

Joana Matos Dias

Localização Dinâmica

Modelos e Técnicas

(Página deixada propositadamente em branco)



I N V E S T I G A Ç Ã O



EDIÇÃO

Imprensa da Universidade de Coimbra
Email: imprensauc@ci.uc.pt
URL: http://www.uc.pt/imprensa_uc
Vendas online: <http://livrariadaimpresa.com>

CONCEPÇÃO GRÁFICA

António Barros

EXECUÇÃO GRÁFICA

Tipografia Lousanense

ISBN

978-989-26-0026-0

ISBN DIGITAL

978-989-26-0193-9

DOI

<http://dx.doi.org/10.14195/978-989-26-0193-9>

DEPÓSITO LEGAL

307209/10

OBRA PUBLICADA COM O APOIO DE:

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR Portugal



Joana Matos Dias

Localização Dinâmica

Modelos e Técnicas



• COIMBRA 2010

(Página deixada propositadamente em branco)

ÍNDICE

CAPÍTULO 1 - INTRODUÇÃO	13
CAPÍTULO 2 - MODELOS DE PROBLEMAS DE LOCALIZAÇÃO DINÂMICA.....	19
2.1. INTRODUÇÃO	19
2.2. PROBLEMAS DE LOCALIZAÇÃO DINÂMICA COM UM NÍVEL.....	21
2.2.1. INTRODUÇÃO	21
2.2.2. PROBLEMAS SEM RESTRIÇÕES DE CAPACIDADE	24
2.2.3. PROBLEMAS COM RESTRIÇÕES DE CAPACIDADES	28
2.3. PROBLEMAS DE LOCALIZAÇÃO MULTI-NÍVEL.....	37
2.3.1. INTRODUÇÃO	37
2.3.2. DESCRIÇÃO DO PROBLEMA	42
2.3.3. VARIÁVEIS DE CONTROLO DE FLUXO.....	43
2.3.4. VARIÁVEIS DE AFECTAÇÃO A CAMINHOS.....	45
2.4. MODELOS MULTIOBJECTIVO	51
CAPÍTULO 3 - HEURÍSTICAS PRIMAIS-DUAIS.....	55
3.1. MOTIVAÇÃO.....	55
3.2. PROBLEMAS DE LOCALIZAÇÃO COM UM NÍVEL	58
3.2.1. PROBLEMAS SEM RESTRIÇÕES DE CAPACIDADES	58
3.2.2. PROBLEMAS COM RESTRIÇÕES DE CAPACIDADE.....	83
3.3. PROBLEMAS DE LOCALIZAÇÃO MULTI-NÍVEL.....	149
3.3.1. PROBLEMAS SEM RESTRIÇÕES DE CAPACIDADES	150
3.3.2. INTRODUÇÃO DE RESTRIÇÕES DE CAPACIDADE MÁXIMA.....	155
3.3.3. INTRODUÇÃO DE RESTRIÇÕES DE CAPACIDADE MÍNIMA.....	160
3.3.4. SERVIÇOS SEM CONSERVAÇÃO DE FLUXO	162
3.3.5. RESULTADOS COMPUTACIONAIS.....	167
CAPÍTULO 4 - METAHEURÍSTICAS.....	182
4.1. MOTIVAÇÃO.....	182
4.2. O PROBLEMA DE LOCALIZAÇÃO.....	187
4.3. O ALGORITMO MEMÉTICO	188
4.3.1. REPRESENTAÇÃO DE SOLUÇÕES	189
4.3.2. FUNÇÃO DE AVALIAÇÃO DE INDIVÍDUOS.....	193
4.3.3. OPERADORES GENÉTICOS	194
4.3.4. PESQUISA LOCAL.....	199
4.3.5. POPULAÇÃO	204
4.3.6. VALORES PARA OS PARÂMETROS	206
4.3.7. PERSPECTIVA GLOBAL	207
4.3.8. RESTRIÇÕES ADICIONAIS	210
4.3.9. RESULTADOS COMPUTACIONAIS.....	214
4.4. CONCLUSÕES	220
4.5. PROBLEMAS MULTI-OBJECTIVO	222
4.5.1. DETALHES DO ALGORITMO.....	229
4.5.2. RESULTADOS COMPUTACIONAIS.....	237
CAPÍTULO 5 - SISTEMA DE APOIO À DECISÃO	251
5.1. INTRODUÇÃO	251
5.2. SISTEMA DE APOIO À DECISÃO PARA A RESOLUÇÃO DE PROBLEMAS DE LOCALIZAÇÃO	255
5.2.1. BASE DE DADOS E BASE DE MODELOS.....	255
5.2.2. BASE DE ALGORITMOS	257
5.2.3. BASE DE ORGANIZAÇÃO	260
CAPÍTULO 6 - CONCLUSÕES.....	263
REFERÊNCIAS	267

(Página deixada propositadamente em branco)

PREFÁCIO

This book is a most welcome addition to the facility location literature and a “must read” for students and researchers in the field as it addresses some of the most difficult and practical problems in facility location analysis. It is the product of doctoral research conducted by Joana Dias. As a member of the jury, I had the great pleasure to participate in Dr. Dias’ defense of her exceptional PhD thesis. Sitting in the 14th century “Sala of student torture” at the University of Coimbra (Portugal) is one of the academia’s greatest honors. It is quite humbling to occupy a seat on the jury balcony that has been occupied by outstanding scholars for over 500 years at a university whose rich traditions and academic excellence date back over 700 years.

The primary focus of this text is on the mathematical modeling of dynamic facility location problems. Facility location analysis has been an important human activity since our earliest ancestors identified the “best” caves for shelter. Its importance continues today as is evidenced by how many times we have heard that “location, location, location” are the three most important components of retail success. Mathematical modeling of location problems dates back at least 100 years to Weber (1909). However, it has been during the past 40 years that most of this research has occurred. The continuing importance of, and widespread interest in, location modeling is demonstrated by the number of articles and presentations addressing the topic by scholars in numerous disciplines including economics, engineering, geography, management science/operations research, production and operations management, and regional science among others.

There are several reasons for this on-going interest. First, facility location decisions are ubiquitous as they are made at all levels of human activity from individuals to corporations and governments. Second, they are strategic in nature as they often involve large capital outlays and the facilities function for long periods of time. The former justifies extensive quantitative analysis and the latter implies that they are often dynamic in nature as parameters such as demand and costs vary over time. Third, they frequently have important social, economic and environmental impacts that may affect various stakeholders differently. As a result, such decisions are often driven by multiple objectives. Fourth, they are “problem specific” in that their constraints and objectives vary greatly depending on the problem setting. Finally, they are generally difficult (or impossible) to solve optimally due to their computational complexity. As a consequence, heuristic solution procedures must be developed to solve most real-world problems. In order to facilitate the solution process, many facility location models simplify the underlying problem by considering only a single objective, and/or by ignoring its dynamic aspects and facility capacity constraints.

The research presented in this book tackles some of the most difficult facility location problems head-on. Specifically, it develops innovative solution approaches to dynamic facility location problems. By definition, dynamic models consider changes over time. Facility location models may be characterized as

“implicitly” dynamic or as “explicitly” dynamic (John Current, Ratick & ReVelle, 1998, John Current, Daskin & Schilling, 2001). Implicitly dynamic models are “static” in that all facilities are opened (or possibly closed) at the beginning of the planning period and remain that way throughout the planning horizon. These models are dynamic in that they account for the effects of future parameters changes in the initial set of locations. Explicitly dynamic models account for future parameter changes by locating facilities and scheduling their opening (and possibly closing) over the planning horizon.

This book presents innovative solution procedures for “explicitly dynamic” location problems which are generally more difficult to solve than are the implicitly dynamic ones. These procedures utilize primal-dual techniques and memetic algorithms. The text also addresses location decisions that involve capacitated facilities and multiple objectives. Historically, these are among the least modeled location problems. For example, a 1985 survey of location planning papers identified over 1,500 papers (W. Domschke & Drexel, 1985) and a 1990 survey of articles addressing multi-objective location problems identified only 46 articles (John Current, Schilling & Min, 1990). The latter article noted the need for additional research on dynamic and capacitated models as only 4 of the 46 articles were dynamic in nature and only 8 considered capacitated facilities. John Current, Daskin & Schilling, 2001, called for the development of new solution heuristics for multi-objective location models. This book makes major contributions regarding these three important areas of facility location research.

I have been conducting research on multi-objective location problems for over 30 years from J. L. Cohon, ReVelle, Eagles & Current, 1980, to Luís Alçada-Almeida, Coutinho-Rodrigues & Current, 2009. It is exciting to know that the future of the field is in good hands as brilliant young scholars like Joana Dias are taking on the challenging problems that still remain. Professors M. Eugénia Captivo and João Clímaco deserve recognition for appreciating Joana’s talent, mentoring her, and supporting her in taking on such challenging research.

John R. Current
Professor, Dept. of Management Science
The Fisher College of Business
The Ohio State University
Columbus, Ohio

«Há um único ponto em que me sinto superior ao comum dos homens: sou ao mesmo tempo mais livre e mais submisso do que eles se atrevem a ser. (...) Inventei um sistema de vida em que a mais pesada tarefa poderia ser cumprida perfeitamente sem nos absorver por completo; na verdade, ousei por vezes propor-me eliminar até a noção física da fadiga. Noutros momentos exercitava-me na prática de uma liberdade de alternativas: as emoções, as ideias, os trabalhos, deviam poder ser interrompidos a cada instante e depois continuados; e a certeza de poder repeli-los ou chamá-los, como aos escravos, tirava-lhes toda a possibilidade de tirania, e a mim todo o sentimento de servidão. Fiz melhor: ordenei um dia inteiro em volta de uma ideia preferida, que eu nunca abandonava; tudo o que poderia ter-me desencorajado ou distraído dessa ideia, os projectos ou os trabalhos de outra ordem, as palavras sem alcance, os mil incidentes do dia apoiavam-se nela como pântanos num fuste de coluna. Outras vezes, pelo contrário, dividia ilimitadamente: cada pensamento, cada facto era cortado por mim, seccionado num grande número de pensamentos ou de factos mais pequenos, mais fáceis de dominar. As resoluções difíceis de tomar esmigalhavam-se numa poeira de decisões minúsculas, perfilhadas uma a uma, conduzindo uma a outra, e tomadas assim inevitáveis e fáceis. (...) Os trabalhos enfadonhos eram executados sem dificuldade por pouco que me agradasse dedicar-me a eles. Logo que um objecto me repugnava, fazia dele objecto de estudo. (...) Mesmo no meio do meu maior desastre vi o momento em que o esgotamento lhe tirava uma parte do seu horror, em que eu o tornava meu, aceitando aceitá-lo. (...) E foi assim, com uma mistura de reserva e de audácia, de submissão e revolta cuidadosamente concertadas, de extrema exigência e prudentes concessões, que eu, finalmente, me aceitei a mim próprio.»

In Memórias de Adriano
Marguerite Yourcenar

(Página deixada propositadamente em branco)

AGRADECIMENTOS

Quero expressar os meus agradecimentos à Professora Doutora Maria Eugénia Captivo e ao Professor Doutor João Clímaco, pela sua orientação durante o desenvolvimento do trabalho de doutoramento que originou este texto, pela motivação para que apresentasse os resultados que se foram obtendo em várias conferências, e pela ajuda na publicação de artigos em revistas nacionais e internacionais.

Agradeço ao Conselho Directivo e ao Centro de Informática da FEUC a disponibilização de um computador onde me foi possível executar o vasto conjunto de testes computacionais realizados. Ao Centro de Investigação Operacional agradeço a possibilidade de utilização do software Cplex. Ao Inesc-Coimbra e ao Conselho Directivo da FEUC agradeço o apoio que me permitiu apresentar o trabalho desenvolvido em várias conferências nacionais e internacionais.

Ao Eng. Cordeiro Santo (ERSUC) e ao Dr. António Manuel Tenreiro da Cruz (Associação de Municípios da Região do Planalto Beirão), agradeço o tempo que me concederam, pois das conversas havidas surgiu a semente do trabalho de cujo o fruto se dá, aqui e agora, conta. Ao Professor Doutor António Antunes (Departamento de Engenharia Civil da FCTUC) o meu muito obrigada por todos os esclarecimentos técnicos e disponibilidade para responder às mensagens de correio electrónico que fomos trocando.

A todos os meus colegas do Núcleo de Métodos Científicos de Gestão da FEUC, pelo óptimo ambiente de trabalho, e em especial aos “rapazes” com quem partilho o Gabinete 500: ao Óscar, ao Paulo, ao Pedro e ao Vítor o meu muito obrigada pela vossa amizade, por todo o vosso apoio e estímulo.

Aos meus pais, pelo seu apoio incondicional.

Ao Pedro, ao Afonso e à Carolina, por todos os nossos dias.

(Página deixada propositadamente em branco)

Capítulo 1

INTRODUÇÃO

Onde localizar? é a questão que hoje, mais do que nunca, está na ordem do dia. Quer estejam em causa serviços públicos como hospitais ou escolas, quer se pense em empresas, lojas ou habitações, a questão de *onde localizar* obriga a reflexões de vária ordem pois a tomada de decisão neste campo acarreta consequências importantes a médio e longo prazos. A questão de *onde localizar* tem muitas vezes a si associada uma outra: a questão *quando localizar*. Não basta, com efeito, saber onde localizar um serviço¹. É igualmente importante saber quando e durante quanto tempo é que estará esse serviço em funcionamento.

Sendo os problemas de localização uma muito interessante área de estudo, agregam um vasto conjunto de disciplinas diferentes. Thisse & Zoller, 1983, consideram que os problemas de localização de serviços públicos, em particular, têm as suas raízes na economia, através das teorias do bem público e da diferenciação de produtos, na geografia e na investigação operacional, e consideram que seria positivo desenvolver a interacção entre estas disciplinas, criando um corpo unificado de conhecimento. Nos dias de hoje é indispensável incluir mais algumas disciplinas nesta lista: a matemática, a física, a geologia, a geografia, as engenharias, a psicologia, a sociologia, a ciência política, para referir apenas aquelas em que se pensa de imediato. Por este motivo, é necessário definir com rigor o âmbito do trabalho que aqui se apresenta, e que consiste na definição de problemas de localização dando enfoque ao desenvolvimento de modelos matemáticos e de métodos de resolução desses modelos que possam ser utilizados no apoio à tomada de decisão.

Quando se equaciona o problema de tomada de decisão e se pensa no desenvolvimento de modelos matemáticos há que ter em conta os dois lados de “uma mesma moeda”: modelos matemáticos mais complexos podem ilustrar de forma mais próxima da realidade os problemas que se pretendem resolver, mas obrigando a algoritmos de resolução mais complexos, a maiores tempos de desenvolvimento, a um maior número de dados acerca do problema, e a um maior esforço por parte do decisor para compreender, aceitar e validar o modelo; modelos matemáticos mais simples são mais fáceis de resolver e interpretar, mas

¹ O termo *serviço* irá ser utilizado neste trabalho para designar todo e qualquer objecto a ser localizado.

obrigando muitas vezes a simplificações e à aceitação de pressupostos demasiadamente fortes que colocam também em risco a aplicação dos resultados obtidos. O equilíbrio entre estes dois lados da mesma moeda não é fácil de atingir, até porque está fortemente dependente do problema que se pretende solucionar bem como dos protagonistas do processo de tomada de decisão. Segundo Chenoweth, Dowling & Louis, 2004, o compromisso entre o esforço dispendido e a precisão dos modelos utilizados favorece sempre os modelos que exigem o menor esforço, até porque, na maioria dos casos, é muito difícil aferir os benefícios de se utilizarem modelos mais complexos.

A localização de um serviço traz, como já foi referido, consequências a curto, médio e longo prazos. Neste trabalho a opção vai no sentido de dar relevância a este aspecto dinâmico da localização que obriga a ter em conta vários períodos temporais, permitindo que os modelos sejam aplicados a um maior número de classes distintas de problemas.

A consideração de vários períodos temporais obriga a que sejam disponibilizados um maior número de dados acerca do problema. Esta necessidade será, à partida, uma desvantagem destes modelos? O que se defende é que nenhum modelo deva ser mais exigente em relação à informação que é necessária conhecer do que seria o agente de decisão, se não estivesse a fazer uso do modelo. Assim, se o agente de decisão sentir um peso demasiado grande e a incapacidade de fornecer os dados necessários, tal poderá significar que o modelo que está a ser utilizado não é o modelo apropriado para o binómio agente de decisão – problema. O oposto pode também acontecer, se e só quando confrontado com as necessidades exigidas pelo modelo, o agente de decisão se der conta da importância de alguns dados do problema que ele poderia, eventualmente, estar a ignorar. Assim, os modelos devem basear-se na informação existente e manipulá-la segundo determinadas regras que devem ser conhecidas e aceites por todos os intervenientes no processo. Deve-se ter sempre em atenção o compromisso (difícil de alcançar) entre a simplicidade do modelo – que permite que o modelo seja compreendido e utilizado – e o realismo com que esse modelo representa a realidade. Tüshaus & Wittmann, 1998, são da opinião que se devem sempre utilizar os modelos, o mais simples possível, e só escolher modelos mais complexos se a compreensão adicional que possibilitam acerca do problema justificar o aumento da complexidade. Cite-se Mitchell, 1996:

Of course the more realistic the model, the more computationally expensive it becomes and the harder it is to analyze the results. At some point the realism of the model can override its usefulness, since studying it would be no more enlightening than studying the actual system in nature. It is the art of effective modelling to strike the proper balance between simplicity (which makes understanding possible) and generality (which ensures that the results are meaningful).

Os modelos desenvolvidos são generalizações de modelos muito estudados na literatura, e que permitem a introdução de características presentes em alguns problemas de localização, de forma a aumentar a sua gama de aplicabilidade. Não são, no entanto, de tal forma complexos que impeçam a sua manipulação e

compreensão. Além disso, todos os modelos desenvolvidos, tais como os algoritmos que os manipulam, são sempre vistos como ferramentas de ajuda à tomada de decisão. Não se considera razoável que a solução óptima obtida através da otimização de um determinado modelo de um problema de localização, por mais completo e complexo que o modelo seja, deva ser escolhida e aplicada sem uma reflexão crítica. As soluções obtidas para estes modelos devem servir, em primeiro lugar, para dar a conhecer ao decisor as alternativas e os pontos de vista acerca do problema, que ele teria dificuldade em encontrar sem a utilização destas ferramentas. Como defende Liebman, 1976, o papel da otimização e dos modelos é mais o de encontrar alternativas do que a selecção de uma delas. É impossível catalogar uma decisão como sendo boa ou má tendo apenas como base um modelo matemático (Cvetkovic & Parmee, 2001). E citando Geoffrion, 1976:

The ostensible purpose of a mathematical programming model is to optimize a stipulated objective function subject to stipulated constraints. But its true purpose, at least in strategic applications as every experienced practitioner should know, is to help develop insights into system behaviour which in turn can be used to guide the development of effective plans and decisions.

Uma das classes de problemas de localização mais aliciantes engloba a localização de serviços não desejáveis, conhecidos por sofrerem do síndrome NYMBY *Not In My Backyard*. Estes serviços possuem características que fazem com que as populações não queiram estar próximas deles, apesar de serem serviços necessários a toda a comunidade. Incluem-se nestes serviços aqueles que constituem os sistemas de recolha e tratamento de resíduos sólidos urbanos. A tomada de decisão acerca da localização destes serviços é muito complexa, implicando sempre vários protagonistas com interesses divergentes e obrigando a processos de negociação mais ou menos complexos. Assim, considerou-se que a análise deste tipo de problemas poderia constituir um bom ponto de partida para o trabalho que tinha sido proposto fazer. A falibilidade das previsões, que obrigatoriamente têm de ser feitas quando se opta por trabalhar com problemas dinâmicos, mostra que a janela temporal considerada nestes modelos não pode ser estática, mas ela, em si própria, tem de ser dinâmica. Isto significa que os modelos dinâmicos não devem ser aplicados uma vez aquando da tomada de decisão, e depois postos de lado. Estes modelos devem ser usados, igualmente, para estudar os desvios em relação ao que tinha sido previsto e devem ser interpretados como ajuda na tomada de decisão de acções que corrijam e acompanhem alterações ao inicialmente previsto.

Naturalmente que algumas simplificações tiveram de ser consideradas nos modelos desenvolvidos. Nas palavras de Menestrel & Wassenhove, 2004, *all models are incomplete* e muitas considerações ficarão sempre fora de qualquer modelo. O mais importante é que se tenha consciência das limitações de cada modelo e dos pressupostos considerados no seu desenvolvimento e na sua aplicação. Não foram consideradas, por exemplo, as questões que se prendem com os erros introduzidos nos modelos devido ao problema de agregação dos

clientes (Francis, Lowe & Tamir, 2000). Também não foi considerado o problema de rotas óptimas, que na maior parte das vezes se encontra fortemente relacionado com os problemas de localização. Esta opção tem a ver não só com a complexidade acrescida que todos os modelos iriam apresentar, mas também por se achar que, na maior parte dos casos, os problemas de localização e de rotas óptimas se encontram em níveis de decisão diferentes: enquanto que os primeiros dão origem a decisões estruturais, com consequências no médio e longo prazos (a que Wirasinghe & Waters, 1983, chama macro-problemas), os últimos poderão ser encarados como problemas de curto-prazo, que poderão ter de ser resolvidos diversas vezes durante o período de vida de um determinado conjunto de serviços. Nos modelos apresentados pressupõe-se, também, que a procura de um determinado cliente é independente das localizações dos serviços. Este pressuposto é, nalgumas situações, demasiado forte mas considerado sem grandes reflexões pela maior parte dos investigadores da área, e toma ainda maior importância quando se consideram modelos dinâmicos. Na realidade, a procura de um determinado serviço por um cliente pode ser fortemente dependente da localização, por exemplo, do serviço mais próximo. Outro pressuposto assumido prende-se com a existência de dados determinísticos que possam ser introduzidos nos modelos: não se considera, de nenhuma forma, a existência de incerteza.

Após o desenvolvimento de modelos matemáticos é preciso saber como manipulá-los para que permitam conhecer uma ou um conjunto de soluções admissíveis para o problema. Neste trabalho, os modelos são manipulados através da utilização de heurísticas primais-duais e de metaheurísticas, como descrito nos capítulos 4 e 5. Estas técnicas de optimização permitem trabalhar com os modelos e encontrar soluções de boa qualidade para problemas cuja complexidade inviabiliza, na maior parte dos casos, a utilização de algoritmos exactos para o cálculo da solução óptima. Aliás, tendo-se a consciência de que nenhum modelo consegue incluir todos os aspectos do problema a resolver e muito menos todas as preferências, pré-conceitos e interesses do decisor, a procura da solução óptima é sempre discutível. Neste trabalho optou-se por ter como objectivo a procura de soluções de boa qualidade, dando especial importância ao processo de procura dessas soluções, como sendo um processo de aprendizagem e de descoberta. Este processo ganha ainda mais ênfase quando se trabalha com problemas multi-objectivo e quando se consideram vários decisores em oposição a apenas um agente de decisão único. No capítulo 5 descreve-se um possível sistema de apoio à decisão, no qual se integram os modelos e metodologias desenvolvidas nos capítulos anteriores, e em que se tenta realçar de novo o carácter multidisciplinar desta área de estudo. Na concepção do sistema de apoio à decisão parte-se do pressuposto que as preferências, os pontos de vista e os conhecimentos dos decisores podem ser de tal modo diferentes que inviabilizem, pelo menos numa fase inicial, a utilização de um modelo matemático comum a todos. Torna-se prioritário discutir o que é realmente característico do problema e aceite consensualmente por todos, mas considerando-se possível que cada decisor deseje escolher as suas próprias funções objectivo e introduzir restrições ao problema. É também neste capítulo

que se demonstra de que forma se podem integrar num mesmo sistema de apoio à decisão as técnicas de optimização que, à partida, poderão parecer incompatíveis.

No início de cada capítulo apresenta-se um pequeno resumo daquilo que nele vai ser apresentado. Como grande parte deste trabalho obrigou ao desenvolvimento de inúmeros procedimentos que são aqui mostrados, aplicou-se especial atenção à sua descrição, sendo que todas as definições utilizadas num determinado procedimento são aí repetidas, para que não se obrigue o leitor a recuar e a perder a linha de raciocínio na interpretação dos procedimentos.

Optou-se por não incluir neste trabalho um capítulo introdutório, em que se faria o levantamento do trabalho realizado na área (*state-of-the-art*) e se introduziriam os conceitos básicos que irão ser utilizados ao longo do texto e que podem ajudar o leitor menos conhecedor a entender e a usufruir daquilo que está escrito. Esta opção deve-se fundamentalmente ao conteúdo fortemente técnico deste trabalho, que faz com que um capítulo introdutório não seja de todo suficiente para suportar tudo aquilo que posteriormente se apresenta, e também ao facto das contribuições feitas não se limitarem apenas a uma parte dos capítulos apresentados. Com efeito, todos os capítulos considerados são motivados pelo trabalho original que foi levado a cabo. Em cada capítulo referem-se os trabalhos mais relevantes (de que o autor tem conhecimento) em relação às matérias que aí são tratadas. Os destinatários deste texto serão leitores com interesse na área da tomada de decisão, em particular no que diz respeito a problemas de localização, familiarizados com a modelação de problemas como modelos de programação linear inteira mista.

(Página deixada propositadamente em branco)

Capítulo 2

MODELOS DE PROBLEMAS DE LOCALIZAÇÃO DINÂMICA

Começa por se fazer uma curta introdução aos problemas de localização que irão ser tratados. Define-se o modelo básico que descreve um problema de localização dinâmica sem capacidades e com um só nível de serviços, em que é possível configurar uma localização mais do que uma vez durante o horizonte temporal considerado. Isto significa que um dado serviço pode ser aberto, fechado e reaberto mais do que uma vez numa determinada localização. Incluem-se restrições de capacidades ao modelo básico, e considera-se uma variante em que é possível abrir vários equipamentos de iguais ou diferentes dimensões em cada localidade. Generaliza-se o modelo básico, considerando-se a situação em que os serviços estão organizados numa estrutura multi-nível. Para cada problema formulado, constrói-se também o problema dual, pois a relação primal-dual é a pedra fundamental dos algoritmos tratados no capítulo seguinte.

What does it mean for a model to be sufficiently accurate? Accuracy, like beauty, is in the eyes of the beholder.

D. E. Goldberg, 2002

2.1. Introdução

Considere-se uma rede com um conjunto \mathcal{N} de nodos e um conjunto \mathcal{A} de arcos. O conjunto \mathcal{N} pode ser dividido em dois subconjuntos disjuntos $\mathcal{N}1$ e $\mathcal{N}2$, tais que os nodos em $\mathcal{N}1$ são caracterizados por terem uma dada procura e os nodos em $\mathcal{N}2$ por poderem vir a satisfazer essa procura. Os nodos em $\mathcal{N}2$ podem ou não ter um limite máximo à quantidade total de procura que conseguem satisfazer. Existem custos fixos associados a estabelecer um determinado nodo em $\mathcal{N}2$ como sendo capaz de satisfazer a procura de clientes, e também custos associados à satisfação da procura por parte dos cliente em $\mathcal{N}1$. O objectivo é decidir quais serão os nodos de $\mathcal{N}2$ que irão satisfazer a procura dos clientes de $\mathcal{N}1$, e ainda quais os arcos de \mathcal{A} que serão utilizados para satisfazer essa procura, de modo a que o custo total seja minimizado. Este é o problema que consiste na escolha de localizações onde instalar serviços (armazéns, fábricas, escolas, hospitais, etc.), de tal forma que seja satisfeita a procura de um determinado bem por parte de um conjunto de clientes,

minimizando o custo total envolvido (geralmente custos fixos de instalação e custos de afectação). Quando não se consideram capacidades associadas aos serviços (ou se considera que a capacidade de cada serviço é suficiente para satisfazer a totalidade da procura dos clientes), este problema é normalmente designado por Problema de Localização Simples (PLS). Cada cliente é servido por apenas um serviço: aquele a que corresponde o menor custo de afectação entre todos os que estão abertos.

O Problema de Localização Dinâmica (PLD) é uma generalização do PLS, em que se considera que a localização dos serviços pode ser feita ao longo de um horizonte temporal determinado.

Segundo Erlenkotter, 1981, são duas as características principais que obrigam à utilização de modelos de localização dinâmica: alterações significativas ao longo do tempo nos custos de afectação de clientes a serviços e custos significativos de fecho e abertura de serviços em novas localizações. Nas situações em que a primeira característica está ausente significa que se poderia utilizar um único problema estático. Na ausência da segunda, significa que se poderão considerar vários problemas estáticos e não relacionados entre si.

Neste capítulo serão apresentados alguns modelos de problemas de localização dinâmica considerando serviços com e sem capacidades. Começar-se-á por apresentar modelos que consideram a existência de um único nível de serviços, passando-se em seguida para os problemas em que se considera a hipótese de existir mais do que um nível de serviços, e estando estes organizados entre si.

Todos os modelos apresentados têm uma característica comum, que os torna generalizações de alguns modelos de localização dinâmica descritos na literatura. Na maior parte dos problemas de localização definidos na literatura e que consideram possível a abertura e o fecho de serviços durante o horizonte temporal definido, a configuração de uma determinada localização pode apenas ser alterada uma vez durante todo o horizonte temporal. Isto significa que, se um serviço é aberto no período temporal t , então manter-se-á aberto até ao fim do horizonte temporal. De modo semelhante, se um determinado serviço é fechado em t manter-se-á fechado até ao fim do horizonte temporal. Existem algumas excepções a esta regra, como é o caso do trabalho de G. Wesolowsky & Truscott, 1975, mas os autores consideram custos fixos de abertura que não têm em conta se numa dada localização já esteve ou não em funcionamento um serviço. Além disso, não são considerados custos fixos de funcionamento dos serviços durante os períodos temporais em que se encontram abertos, nem custos de encerramento de serviços. Também Canel, Khumawala, Law & Loh, 2001, e Pierre Chardaire, Sutter & Costa, 1996, consideram a possibilidade de um serviço ser aberto, fechado e reaberto. No entanto, os autores não diferenciam custos de abertura e reabertura de serviços e apresentam uma função objectivo não linear.

Em todos os modelos desenvolvidos se considera a possibilidade de abertura, fecho e reabertura de serviços durante todo o horizonte temporal. É possível considerar custos fixos de abertura e reabertura diferentes, pois existem diversas situações em que a reabertura de um serviço, numa determinada localização,

tem associado um custo fixo diferente do custo fixo de abertura. Pense-se, por exemplo, em situações em que se tenham adquirido anteriormente instalações, construído infra-estruturas, feito estudos de impacto ambiental. Todos os modelos permitem também ter em conta custos de funcionamento e encerramento de serviços que, na maior parte das situações, não podem ser ignorados. É também, em todos os casos, possível considerar a existência de serviços abertos antes do início do horizonte temporal.

Todos os problemas que a seguir se descrevem são generalizações do PLS, pelo que se pode concluir de imediato serem problemas *NP*-difíceis (Krarup & Pruzan, 1983).

2.2. Problemas de Localização Dinâmica com Um Nível

2.2.1. Introdução

Em C. S. ReVelle & Eiselt, 2005, os autores consideram que o problema de localização dinâmica começou a ser estudado a partir da década de sessenta, tendo-se notado um retrocesso nos desenvolvimentos nesta área durante os anos oitenta. As razões apontadas prendem-se com o facto de serem problemas de difícil resolução e também por obrigarem à previsão de valores, tarefa que acresce à complexidade do problema. Um dos primeiros artigos a referir um problema de localização dinâmica deve-se a Ballou, 1968, em que o autor trata o problema dinâmico de localização e realocação de um único serviço, que é resolvido utilizando programação dinâmica. G. O. Wesolowsky, 1973, generaliza o problema de Weber, considerando a existência de vários períodos temporais. G. Wesolowsky & Truscott, 1975, descrevem um problema de localização discreto em que se fixa, para cada período temporal, o número de serviços abertos. Em cada período temporal, cada serviço pode ser aberto ou removido de uma determinada localização. Van Roy & Erlenkotter, 1982, descrevem o problema de localização dinâmica simples, em que se considera a hipótese de se abrirem novos serviços no início de cada período temporal ou se fecharem serviços (já existentes no início do primeiro período temporal) no fim de cada período temporal. Haug, 1985, apresenta um modelo dinâmico de localização cujo objectivo é a maximização do lucro, através da selecção das localizações óptimas de equipamentos produtivos no estrangeiro. O modelo tem em conta factores como o tempo, risco de expropriações, incentivos dos diversos países considerados, características da mão-de-obra local, etc.. Laporte & Dejax, 1989, descrevem um problema dinâmico em que se considera simultaneamente a localização de serviços e o problema de rotas óptimas que lhe está associado. Jacobsen, 1990, introduz alguns modelos de localização dinâmica, sem capacidades e que consideram a expansão de capacidades. Drezner & Wesolowski, 1991, estudam o problema de localizar um serviço no plano, sendo que a procura por parte dos clientes é uma função contínua do tempo, e a localização do serviço pode ser alterada k vezes durante o horizonte temporal considerado. O objectivo é, então, o de encontrar os k instantes no tempo em que uma alteração na localização do serviço ocorre, e as localizações óptimas

associadas. Roberto Diéguez Galvão & Santibañez-Gonzalez, 1992, consideram a generalização do problema da p -mediana a vários períodos temporais. Neste modelo, os clientes deverão ser afectos a um conjunto K de serviços no período k , sendo o objectivo a minimização dos custos de instalação e transporte. Emanuel Melachrinoudis, Min, & Wu, 1995, propõem um modelo dinâmico, multi-objectivo, para a localização de aterros sanitários com capacidades. Drezner, 1995, descreve o problema da p -mediana progressivo, em que os serviços são localizados, um de cada vez, e em períodos de tempo conhecidos. Current, Ratick, & ReVelle, 19978, estudam um problema dinâmico semelhante ao da p -mediana, mas em que se considera de forma explícita a incerteza na determinação do número exacto de serviços a localizar. Lim & Kim, 1999, consideram um problema dinâmico de localização que trata, de forma explícita, dois tipos diferentes de serviços: serviços *flexíveis* (equipamentos multifuncionais com capacidade para produzir mais do que um tipo de produto ou operação) e serviços *dedicados* (só podem produzir um tipo de produto ou operação). Os autores consideram a possibilidade de existirem serviços dedicados (mas não flexíveis) abertos antes do primeiro período temporal, e todos os serviços que são abertos assim se mantêm até ao fim do horizonte temporal. A capacidade total disponível em cada período temporal depende, assim, do número e do tipo de serviços abertos. Köksalan & Süral, 1999, desenvolvem um modelo dinâmico com capacidades, utilizado para localizar novas instalações produtivas de malte para uma companhia de cerveja turca. Antunes & Peeters, 2001, apresentam um problema de localização dinâmico baseado num problema real, cujas decisões se prendem com a abertura de novos serviços e o fecho, a redução ou expansão de serviços já existentes. Canel *et al.*, 2001, estudam o problema de localização dinâmica multi-produto com uma função objectivo não linear. Saldanha-da-Gama, 2002; Saldanha-da-Gama e Captivo, 2002, estudam o problema de localização dinâmica com capacidades (PLDC), considerando que um serviço aberto antes do início do horizonte de planeamento pode ser fechado (mantendo-se fechado até ao fim do horizonte temporal) e que serviços abertos durante a janela temporal considerada mantêm-se abertos até ao fim do horizonte temporal. As capacidades dos serviços mantêm-se inalteradas durante todos os períodos temporais. Albareda-Sambola, Fernández, Hinojosa & Puerto, 2009, desenvolvem um modelo dinâmico em que se pretende localizar serviços durante um horizonte temporal, sendo que um serviço que seja aberto se manterá aberto até ao fim do horizonte temporal considerado, e estabelecendo um número mínimo de clientes que têm de estar afectos a serviços abertos em cada período.

Um outro conjunto interessante de problemas diz respeito à consideração explícita da possibilidade de expansão de capacidades de serviços. Na maioria das referências encontradas considera-se que esta expansão pode ser feita de forma contínua e não discreta. Hinomoto, 1965, estuda o problema de expansão de capacidades através da adição de novos serviços em períodos discretizados, mas considerando que a dimensão do serviço pode ser tratada como uma variável contínua. Erlenkotter, 1975, desenvolve duas aproximações para tratar o problema de planeamento de capacidades para sistemas de multi-localização de

grandes dimensões: uma é baseada na medida de custo equivalente e outra é baseada em programação dinâmica. Fong & Srinivasan, 1981a, formulam o problema de expansão contínua de capacidade como um problema de localização dinâmica. No artigo seguinte (Fong & Srinivasan, 1981b) generalizam o problema, considerando um custo fixo associado ao aumento de capacidade numa dada localização e ainda um custo variável e proporcional ao aumento considerado. Freindenfelds, 1981b, considera um problema de expansão de capacidades em que existem dois tipos distintos de procura e dois tipos de serviços. O autor considera que a capacidade pode ser aumentada em valores contínuos, a um custo que não depende nem do tempo em que é feito nem dos aumentos que se tenham feito anteriormente. No seu livro (Freindenfelds, 1981a) introduz vários modelos analíticos de problemas de expansão de capacidades, apresentando também várias aplicações e dando importância às decisões de investimento de capital que estão envolvidas sempre que se pensa em localizar novas capacidades produtivas. Smith, 1981, generaliza o trabalho de Manne, 1961, e apresenta um algoritmo eficiente para resolver o problema de expansão de capacidades determinístico, considerando um horizonte temporal finito. É importante referir que, no contexto deste artigo, o conceito de *horizonte temporal* refere o período de tempo durante o qual a capacidade adicional pode ser utilizada, podendo, deste modo, ser interpretado como o *tempo de vida* do equipamento. Esta interpretação é completamente diferente daquela que é dada no contexto do presente trabalho, em que *horizonte temporal* corresponde à janela de tempo dentro da qual é possível tomar decisões de localização ou deslocalização de equipamentos. Em Luss, 1982, pode encontrar-se um *survey* acerca de problemas de expansão de capacidades encontrados na literatura. O autor chama a atenção dos leitores para a escassez de trabalhos que lidam com a expansão dinâmica de capacidades. S.-B. Lee & Luss, 1987, desenvolvem um modelo que considera a expansão de capacidades (com valores contínuos), e também a possibilidade de converter capacidade de um serviço em capacidade de outro serviço, sem que haja a possibilidade de instalar serviços em novas localizações, e considerando que se conhecem à partida os incrementos na procura para cada serviço e período temporal (não se considera nenhum problema de afectação associado). Min, 1988, considera o problema dinâmico de expansão e realocação de serviços públicos com capacidades, considerando vários objectivos. O autor formula o problema como um problema inteiro misto de programação por metas, num ambiente de decisão difuso. O problema é ilustrado através de um caso real de expansão e realocação de bibliotecas públicas dentro do sistema de bibliotecas públicas do Município de Columbus. Shulman, 1991, formula o problema de localizar e expandir a capacidade de serviços dinamicamente. O autor considera um pequeno conjunto discreto de possíveis aumentos das capacidades (a expansão da capacidade é conseguida através da abertura de mais do que um equipamento numa dada localização) e considera duas situações: um problema mais geral que possibilita que equipamentos de diferentes capacidades possam ser abertos na mesma localização e um outro em que, numa dada localização, podem ser localizados vários equipamentos mas todos de igual dimensão. Outros modelos que estão

directamente relacionados com a problemática da escolha das capacidades dos serviços a abrir, ou com a possibilidade de se abrir mais do que um serviço na mesma localização, podem encontrar-se em C. Lee, 1991, que generaliza o problema clássico de localização com capacidades e considera um problema multi-produto em que a capacidade de cada serviço deve ser escolhida dentro de um pequeno conjunto de capacidades possíveis; Sridharan, 1991, e Mazzola & Neebe, 1999, que estudam o problema de localização e escolha do tamanho de um serviço através da resolução de um problema de localização com capacidades, incluindo restrições adicionais (garantindo que, no máximo, um serviço é aberto em cada localização); Ghiani & Musmanno, 2002, que estudam o problema de localizar serviços com capacidades, permitindo que sejam abertos vários serviços de idêntica capacidade na mesma localização (este problema baseia-se na localização de estações de voto num Município italiano); Wu, Zhang, & Zhang, 2006, que apresentam um modelo que permite que mais do que um serviço seja localizado na mesma localização, distinguindo a existência de custos fixos que não dependem do número de serviços instalados, e custos dependentes do tipo de serviço. Estes trabalhos, apesar de se debruçarem sobre problemas estáticos, apresentam algumas características interessantes e a ter em conta nos modelos dinâmicos de localização que se irão desenvolver.

Passa-se de imediato à descrição dos modelos estudados.

2.2.2. Problemas sem Restrições de Capacidade

Vai descrever-se o modelo matemático que formula um problema de localização dinâmico sem capacidades, com possibilidade de alterar a configuração de uma localização mais do que uma vez durante o horizonte temporal considerado (Joana Dias, Captivo & Clímaco, 2005a; Joana Dias, Captivo & Clímaco, 2007c). Vai também formular-se o problema dual.

2.2.2.1. Formulação do Problema

Defina-se a seguinte notação:

$J = \{1, \dots, j, \dots, n\}$ conjunto de índices que correspondem às localizações dos clientes;

$I = \{1, \dots, i, \dots, m\}$ conjunto de índices que correspondem às possíveis localizações dos serviços;

$T =$ número de períodos temporais considerados no horizonte temporal ($1 \leq t \leq T$)

$c'_{ij} =$ custo de afectar um cliente j ao serviço i no período t ;

$FA_{it}^{\xi} =$ custo fixo de abrir um serviço em i no início do período t , e de o fechar no fim do período (o serviço estará em funcionamento desde o início de t até ao fim de t);

$FR_{it}^{\xi} =$ custo fixo de reabrir um serviço em i no início do período t , e de o fechar no fim do período (o serviço estará em funcionamento desde o início de t até ao fim de t);

e definam-se as seguintes variáveis de decisão:

$$a_{it}^{\xi} = \begin{cases} 1 & \text{se o serviço } i \text{ é aberto no início do período } t \\ & \text{e se mantém aberto até ao fim do período } \xi \\ & , i \in I, t=1, \dots, T, \xi=t, \dots, T; \\ 0 & \text{em caso contrário} \end{cases}$$

$$r_{it}^{\xi} = \begin{cases} 1 & \text{se o serviço } i \text{ é reaberto no início do período } t \\ & \text{e se mantém aberto até ao fim do período } \xi \\ & , i \in I, t=2, \dots, T, \xi=t, \dots, T; \\ 0 & \text{em caso contrário} \end{cases}$$

$$x_{ij}^t = \begin{cases} 1 & \text{se o cliente } j \text{ é afecto ao serviço localizado} \\ & \text{em } i \text{ durante o período } t \\ & , i \in I, j \in J, t=1, \dots, T. \\ 0 & \text{em caso contrário} \end{cases}$$

As variáveis r_{it}^{ξ} são apenas definidas para períodos de tempo t maiores do que um. Os custos fixos de abertura ou reabertura de um serviço i do período t até devem considerar os custos de (re) abertura no período t , custos fixos de funcionamento de t até e também custos de encerramento em .

O problema de localização dinâmica com abertura, fecho e reabertura de serviços pode ser formulado como se apresenta seguidamente.

PLDAFR:

$$\text{Min} \sum_{i=1}^T \sum_{r=1}^m \sum_{j=1}^n c_{ij}^t x_{ij}^t + \sum_{i=1}^m \sum_{t=1}^T \sum_{\xi=t}^T FA_{it}^{\xi} a_{it}^{\xi} + \sum_{i=1}^m \sum_{t=2}^T \sum_{\xi=t}^T FR_{it}^{\xi} r_{it}^{\xi} \quad (2-1)$$

sujeito a:

$$\sum_{i=1}^m x_{ij}^t = 1, \quad \forall j \in J, t=1, \dots, T \quad (2-2)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - x_{ij}^t \geq 0, \quad \forall i \in I, j \in J, t=1, \dots, T \quad (2-3)$$

$$\sum_{\tau=1}^{t-1} \sum_{\xi=\tau}^{t-1} a_{i\tau}^{\xi} - \sum_{\xi=t}^T r_{it}^{\xi} \geq 0, \quad \forall i \in I, t=1, \dots, T \quad (2-4)$$

$$\sum_{t=1}^T \sum_{\xi=t}^T a_{it}^{\xi} \leq 1, \quad \forall i \in I \quad (2-5)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \leq 1, \quad \forall i \in I, t=2, \dots, T \quad (2-6)$$

$$a_{it}^{\xi}, x_{ij}^t \in \{0, 1\},$$

$$r_{it}^{\xi} \in \{0, 1\},$$

$$\forall i \in I, j \in J, t=1, \dots, T, \xi \geq t \quad (2-7)$$

$$\forall i \in I, t=2, \dots, T, \xi \geq t$$

As restrições: (2-2) garantem que, em cada período temporal, cada cliente é afecto a exactamente um serviço; (2-3) garantem que, em cada período temporal, um cliente apenas pode ser afecto a serviços que estejam em funcionamento nesse período; (2-4) estabelecem que um serviço só pode ser reaberto no início do período t se já foi aberto num qualquer período temporal anterior e não está em funcionamento no início do período t ; (2-5) impõem que um dado serviço só pode ser aberto uma vez durante todo o horizonte temporal; (2-6) determinam que em cada localização apenas um serviço pode estar em funcionamento em cada período temporal.

As restrições (2-5) e (2-6) necessitam de serem consideradas de forma explícita apenas quando em presença de custos fixos negativos. Se todos os custos fixos forem maiores ou iguais a zero, então poderão ser substituídas por:

$$\sum_{\tau=2}^t \sum_{\xi=\tau}^T r_{i\tau}^{\xi} \leq 1, \quad \forall i \in I, t = 2, \dots, T \quad (2-8)$$

Esta formulação é também válida se se pretender considerar a existência de serviços em funcionamento antes do início do horizonte temporal. Considere-se um conjunto $I_c \subset I$ tal que se $i \in I_c$, então i está aberto antes do início do horizonte temporal. Os serviços que pertencem a I_c poderão continuar abertos durante o primeiro período temporal ou, então, serem imediatamente fechados mesmo antes do início do primeiro período temporal. Considere-se:

$$a_{i1}^{\xi} = \begin{cases} 1 & \text{se o serviço } i \text{ está aberto antes de } t=1 \text{ e continua aberto} \\ & \text{até ao fim do período } \xi \\ 0 & \text{em caso contrário} \end{cases} ;$$

$$a_{it}^{\xi} = \begin{cases} 1 & \text{se o serviço é fechado antes de } t=1 \text{ e é reaberto pela primeira vez} \\ & \text{no início do período } t \text{ e continua aberto até ao fim do período } \xi \\ 0 & \text{em caso contrário} \end{cases} .$$

As variáveis a_{i1}^{ξ} são definidas para $i \in I_c$ e as variáveis a_{it}^{ξ} são definidas para $i \in I_c, t > 1$. Os custos fixos associados às variáveis $a_{i1}^{\xi}, i \in I_c$, correspondem aos custos de funcionamento durante os períodos em que o serviço está aberto, mais os custos de encerrar o serviço no fim do período . Os custos fixos associados às variáveis $a_{it}^{\xi}, i \in I_c$ e $t > 1$, correspondem aos custos que se incorrem por fechar o serviço antes do início do período temporal 1, mais os custos de reabrir o serviço no início do período t , mais os custos de funcionamento durante os períodos em que o serviço está aberto, mais os custos de encerrar o serviço no fim do período. Se se fizer a reinterpretação descrita das variáveis $a_{it}^{\xi}, i \in I_c$, torna-se possível integrar no modelo a pré-existência de serviços em funcionamento sem aumentar o número de variáveis nem restrições.

Note-se que não é obrigatório considerar todas as variáveis de localização apresentadas. O decisor é livre de considerar apenas variáveis $a_{i\tau}^{\xi}$ e $r_{i\tau}^{\xi}$ que façam sentido em relação aos equipamentos com que tem de lidar. Pode, por

exemplo, optar por considerar apenas variáveis tais que $\xi - \tau$ é maior do que um intervalo mínimo de tempo, obrigando, deste modo, a que um serviço que seja aberto se mantenha em funcionamento durante um número mínimo de períodos temporais.

2.2.2.2. Formulação do Problema Dual

Se multiplicarmos as restrições (2-5) e (2-6) por -1 e associarmos variáveis duais v_j^t às restrições (2-2), w_{ij}^t às restrições (2-3), u_i^t às restrições (2-4), ρ_i às restrições (2-5) e π_i^t às restrições (2-6), então o problema dual da relaxação linear do PLDAFR pode ser formulado como D-PLDAFR.

D-PLDAFR:

$$\text{Max} \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$v_j^t - w_{ij}^t \leq c_{ij}^t, \quad \forall i \in I, j \in J, t = 1, \dots, T \quad (2-10)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} w_{ij}^{\tau} + \sum_{\tau=\xi+1}^T u_i^{\tau} - \rho_i - \sum_{\tau=t}^{\xi} \pi_i^{\tau} \leq FA_{it}^{\xi}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-11)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} w_{ij}^{\tau} - u_i^t - \sum_{\tau=t}^{\xi} \pi_i^{\tau} \leq FR_{it}^{\xi}, \quad \forall i \in I, t = 2, \dots, T, \xi = t, \dots, T \quad (2-12)$$

$$w_{ij}^t, u_i^t, \rho_i, \pi_i^t \geq 0, \quad \forall i \in I, j \in J, t = 1, \dots, T \quad (2-13)$$

Uma formulação “condensada” pode ser obtida se se considerar $w_{ij}^t = \max\{0, v_j^t - c_{ij}^t\}$.

CD-PLDAFR:

$$\text{Max} \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-14)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau}\} \leq FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-15)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau}\} \leq FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-16)$$

$$u_i^t, \rho_i, \pi_i^t \geq 0, \quad \forall i \in I, j \in J, t = 1, \dots, T \quad (2-17)$$

2.2.2.3. Condições de Complementaridade

Defina-se:

$$SA_{it}^{\xi} = FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau}\}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-18)$$

$$SR_{it}^{\xi} = FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau}\}, \quad \forall i \in I, t = 2, \dots, T, \xi = t, \dots, T \quad (2-19)$$

$$SA_{it}^{\xi} = \min\{SA_{it}^{\xi}, SR_{it}^{\xi}\}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-20)$$

Na presença de um par de soluções primal e dual óptimas, as condições de complementaridade (2-21) a (2-26) serão satisfeitas (se não existir *gap* de dualidade).

$$\left(\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - x_{ij}^t \right) w_{ij}^t = 0, \quad \forall i \in I, j \in J, t = 1, \dots, T \quad (2-21)$$

$$\left(\sum_{\tau=1}^{t-1} \sum_{\xi=\tau}^{t-1} a_{i\tau}^{\xi} - \sum_{\xi=t}^T r_{it}^{\xi} \right) u_i^t = 0, \quad \forall i \in I, t = 1, \dots, T \quad (2-22)$$

$$\left(\sum_{\tau=1}^T \sum_{\xi=\tau}^T a_{it}^{\xi} - 1 \right) \rho_i = 0, \quad \forall i \in I \quad (2-23)$$

$$\left(\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - 1 \right) \pi_i^t = 0, \quad \forall i \in I, t = 2, \dots, T \quad (2-24)$$

$$SA_{it}^{\xi} a_{it}^{\xi} = 0, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-25)$$

$$SR_{it}^{\xi} r_{it}^{\xi} = 0, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-26)$$

2.2.3. Problemas com Restrições de Capacidades

No modelo desenvolvido anteriormente considera-se que a capacidade de cada serviço é sempre maior do que a totalidade da procura dos clientes. Este pressuposto é, na maioria das situações, demasiado forte. Na realidade, a maioria dos serviços apresenta capacidades máximas, ou seja limites superiores à quantidade de procura que conseguem satisfazer, e podem também apresentar capacidades mínimas no caso de ser necessário um nível mínimo de serviço que justifique o seu funcionamento.

Note-se que, em qualquer dos modelos com capacidades que vai ser apresentado, é possível considerar a situação em que os serviços têm capacidades máximas mas não é possível satisfazer a procura de todos os clientes, tendo de ser atribuída uma penalização (um custo na função objectivo) por essa não satisfação de procura, que pode depender de cliente para cliente. Basta, para isso, considerar uma localização fictícia, com custos fixos nulos e com custos de afectação que traduzem o custo da não satisfação da procura. Esta localização fictícia terá uma capacidade infinita ou igual ao valor máximo das procuras que podem não ser satisfeitas.

Esta secção baseia-se em modelos apresentados em Joana Dias, Captivo & Clímaco, 2004a.

2.2.3.1. Restrições de Capacidade Máxima e/ou Mínima

Defina-se a seguinte notação, que complementa a apresentada na página 24, secção 2.2.2.1:

d_j^t = procura do cliente j no período temporal t ;

Q_i = capacidade máxima de um serviço localizado em i ;

Q'_i = capacidade mínima de um serviço localizado em i .

Redefinam-se as seguintes variáveis de localização:

x_{ij}^t = fracção da procura do cliente j que é satisfeita pelo serviço localizado em i no período t .

No modelo que se apresenta de seguida considera-se que as capacidades máximas e mínimas se mantêm inalteradas durante todo o horizonte temporal. No entanto, é possível considerar que estas capacidades possam sofrer alterações. Neste caso, dever-se-iam considerar capacidades Q_i^t e $Q'_i{}^t$. Esta generalização pode ser facilmente tida em conta, tanto no modelo desenvolvido, como nos métodos de resolução do problema que serão apresentados mais adiante.

PLDAFRFC:

$$\text{Min} \sum_{t=1}^T \sum_{i=1}^m \sum_{j=1}^n c_{ij}^t x_{ij}^t + \sum_{i=1}^m \sum_{t=1}^T \sum_{\xi=t}^T FA_{it}^{\xi} a_{it}^{\xi} + \sum_{i=1}^m \sum_{t=2}^T \sum_{\xi=t}^T FR_{it}^{\xi} r_{it}^{\xi} \quad (2-1)$$

sujeito a:

$$\sum_{i=1}^m x_{ij}^t = 1, \quad \forall j \in J, t=1, \dots, T \quad (2-2)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - x_{ij}^t \geq 0, \quad \forall i \in I, j \in J, t=1, \dots, T \quad (2-3)$$

$$\sum_{\tau=1}^{t-1} \sum_{\xi=\tau}^{t-1} a_{i\tau}^{\xi} - \sum_{\xi=t}^T r_{it}^{\xi} \geq 0, \quad \forall i \in I, t=1, \dots, T \quad (2-4)$$

$$\sum_{i=1}^m \sum_{\xi=t}^T a_{it}^{\xi} \leq 1, \quad \forall i \in I \quad (2-5)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \leq 1, \quad \forall i \in I, t=2, \dots, T \quad (2-6)$$

$$Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{j=1}^n d_j^t x_{ij}^t \geq 0, \quad \forall i \in I, t=1, \dots, T \quad (2-27)$$

$$\sum_{j=1}^n d_j^t x_{ij}^t - Q'_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \geq 0, \quad \forall i \in I, t=1, \dots, T \quad (2-28)$$

$$a_{it}^{\xi} \in \{0, 1\}, \forall i, t, \xi \geq t \quad (2-29)$$

$$r_{it}^{\xi} \in \{0, 1\}, \forall i, t > 1, \xi \geq t$$

$$x_{ij}^t \geq 0, \forall i, j, t$$

As restrições (2-27) impõem limites máximos à procura total que cada serviço

pode satisfazer em cada período temporal. As restrições (2-28) garantem que, se existe um serviço localizado em i durante o período t , então esse serviço terá de satisfazer uma procura igual ou superior a Q'_i . As restrições que consideram capacidades mínimas permitem ter em conta situações em que um determinado serviço só pode estar aberto acima de um determinado limiar de funcionamento que garanta, por exemplo, a sua viabilidade económica.

À semelhança do que foi feito para o problema sem capacidades, vai formular-se o problema dual da relaxação linear do PLDAFRC. Para além das variáveis duais já consideradas no caso do modelo sem capacidades, considerem-se variáveis duais λ_i^t associadas às restrições (2-27) e variáveis duais β_i^t às restrições (2-28). Neste caso vamos ter $w_{ij}^t = \max\{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t + d_j^t \beta_i^t\}$, e o problema dual condensado pode ser formulado como DC-PLDAFRC.

DC-PLDAFRC:

$$\text{Max} \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau} - d_j^{\tau} \lambda_i^{\tau} + d_j^{\tau} \beta_i^{\tau}\} \leq \quad \forall i, t, \xi = t, \dots, T \quad (2-30)$$

$$FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau},$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau} - d_j^{\tau} \lambda_i^{\tau} + d_j^{\tau} \beta_i^{\tau}\} \leq FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau}, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-31)$$

$$u_i^t, \rho_i, \pi_i^t, \lambda_i^t, \beta_i^t \geq 0, \quad \forall i, j, t \quad (2-32)$$

Defina-se:

$$SA_{it}^{\xi} = FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau} - d_j^{\tau} \lambda_i^{\tau} + d_j^{\tau} \beta_i^{\tau}\} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau}, \quad \forall i, t, \xi = t, \dots, T \quad (2-33)$$

$$SR_{it}^{\xi} = FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ij}^{\tau} - d_j^{\tau} \lambda_i^{\tau} + d_j^{\tau} \beta_i^{\tau}\} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau}, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-34)$$

Considerando os problemas primal e dual, as condições de complementaridade (2-21) a (2-26) e (2-35) a (2-36) serão satisfeitas se em presença de um par de soluções primal e dual óptimas e na ausência de *gap* de dualidade.

$$\left(Q_i \sum_{\tau=1}^i \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{j=1}^n d_j^i x_{ij}^i \right) \lambda_i^i = 0, \quad \forall i, t=1, \dots, T \quad (2-35)$$

$$\left(\sum_{j=1}^n d_j^i x_{ij}^i - Q_i \sum_{\tau=1}^i \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \right) \beta_i^i = 0, \quad \forall i, t=1, \dots, T \quad (2-36)$$

2.2.3.2. Restrições de Capacidade Máxima Decrescente

Considere-se a situação em que um dado serviço é aberto com uma determinada capacidade máxima (Joana Dias, Captivo & Clímaco, 2008a). Em vez desta capacidade se considerar constante durante todos os períodos temporais, ou de se considerarem diferentes capacidades máximas para cada período temporal mas definidas independentemente da procura que o serviço satisfaz na realidade, vai agora considerar-se a situação em que a capacidade máxima do serviço diminui à medida que satisfaz a procura dos clientes que lhe são afectos. Não são conhecidos modelos descritos na literatura que incluam este tipo de restrições.

Serviços que apresentam este tipo de comportamento podem ser encontrados, por exemplo, em sistemas de recolha e tratamento de resíduos sólidos urbanos: os aterros sanitários têm uma capacidade máxima aquando da sua abertura, mas esta capacidade vai diminuindo à medida que aí se vai realizando o depósito dos resíduos.

Para se ter em conta esta situação considerando o modelo apresentado, as restrições de capacidades (2-27) e (2-28)² deverão ser substituídas pela restrição (2-37):

$$Q_i \sum_{\tau=1}^i \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{j=1}^n d_j^i x_{ij}^i \geq 0, \quad \forall i, t=1, \dots, T \quad (2-37)$$

O modelo apresentado considera admissível a situação em que um serviço é fechado, apesar da sua capacidade máxima existente ser ainda maior do que zero. As restrições de capacidade apresentadas consideram ainda que, aquando da reabertura de um serviço, a sua capacidade máxima inicial será igual a Q_i mais a capacidade ainda existente aquando do seu fecho. Pode argumentar-se que tal comportamento não é admissível para alguns tipos de equipamentos. Pense-se, por exemplo, num aterro sanitário: será fácil imaginar que se este equipamento for encerrado no período t e reaberto no período $t+1$, então talvez se possa utilizar a capacidade restante no período t (isto no caso de não se terem ainda iniciado os procedimentos de fecho do aterro). No entanto, se o mesmo aterro for reaberto vários períodos temporais mais tarde, toda a sua capacidade restante, no momento em que foi fechado, se terá perdido, devido

² Note-se que a consideração das restrições de capacidade máxima decrescente não invalida a consideração simultânea das restrições de capacidade mínima. Na realidade, também nestes tipos de serviços pode fazer todo o sentido considerar um limiar mínimo de funcionamento em cada período temporal, pelo menos enquanto esse limiar for inferior à capacidade máxima ainda existente. Nos desenvolvimentos que se irão fazer com este modelo, as restrições de capacidade mínima não foram consideradas. No entanto, a sua introdução no modelo levaria a apenas algumas modificações menores, quer nos modelos, quer nos métodos de resolução que se irão apresentar.

aos trabalhos de fecho e manutenção que é necessário realizar. No entanto, neste tipo de equipamentos, os custos fixos de (re)abertura são geralmente de ordem de grandeza várias vezes superior aos custos de transporte. Por este motivo, não será de esperar que um equipamento que ainda tenha disponível capacidade útil seja fechado, a não ser que esta capacidade restante seja insignificante quando comparada com a sua capacidade inicial. Para além disso, como já referido no capítulo 2.2.2.1, neste modelo o decisor é livre de considerar apenas variáveis d_{ir}^{ξ} e r_{ir}^{ξ} que façam sentido em relação ao problema que tem de resolver. Pode, por exemplo, optar por considerar apenas variáveis tais que $\xi - \tau$ é maior do que um intervalo mínimo de tempo. Por estes motivos, pensa-se que o modelo apresentado tem um comportamento aceitável e pode ser considerado útil em várias situações.

Se se associarem as variáveis duais λ_i^t às restrições primais (2-37), considerando $w_{ij}^t = \max \left\{ 0, v_j^t - c_{ij}^t - \sum_{\psi=t}^T d_j^t \lambda_i^\psi \right\}$, então o dual condensado da relaxação linear do problema de localização dinâmica com capacidades máximas decrescentes (PLDCD) pode ser formulado como DC-PLDCD:

DC-PLDCD

$$\text{Max} \sum_{i=1}^T \sum_{j=1}^n v_j^i - \sum_{i=1}^m \rho_i - \sum_{i=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\psi=\tau}^T d_j^\tau \lambda_i^\psi \right\} \leq FA_{ii}^\xi - \sum_{\tau=\xi+1}^T u_i^\tau + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^\tau - Q_i \sum_{\tau=t}^T \lambda_i^\tau, \quad \forall i, t, \xi = t, \dots, T \quad (2-38)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\psi=\tau}^T d_j^\tau \lambda_i^\psi \right\} \leq FR_{ii}^\xi + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^\tau - Q_i \sum_{\tau=t}^T \lambda_i^\tau, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-39)$$

$$u_i^t, \rho_i, \pi_i^t, \lambda_i^t \geq 0, \quad \forall i, j, t \quad (2-40)$$

Defina-se:

$$SA_{ii}^\xi = FA_{ii}^\xi - \sum_{\tau=\xi+1}^T u_i^\tau + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^\tau - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\psi=\tau}^T d_j^\tau \lambda_i^\psi \right\} - Q_i \sum_{\tau=t}^T \lambda_i^\tau, \quad \forall i, t, \xi = t, \dots, T \quad (2-41)$$

$$SR_{ii}^\xi = FR_{ii}^\xi + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^\tau - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\psi=\tau}^T d_j^\tau \lambda_i^\psi \right\} - Q_i \sum_{\tau=t}^T \lambda_i^\tau, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-42)$$

Na presença de um par de soluções dual e primal admissíveis e óptimas, as condições de complementaridade (2-21) a (2-26) serão satisfeitas, mais a seguinte condição (no caso de não existir *gap* de dualidade):

$$\left(Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{\tau=1}^t \sum_{j=1}^n d_j^{\tau} x_{ij}^{\tau} \right) \lambda_i^t = 0, \quad \forall i, t \quad (2-43)$$

2.2.3.3. Serviços Constituídos por Vários Elementos

Nesta secção desenvolve-se um modelo que considera a possibilidade de expandir ou reduzir a capacidade máxima disponível numa determinada localização durante o horizonte temporal considerado (Joana Dias, Captivo & Clímaco, 2004b; Joana Dias, Captivo & Clímaco, 2007a). A expansão ou redução de capacidade disponível é conseguida através da abertura ou do fecho, respectivamente, de um ou mais equipamentos de iguais ou diferentes capacidades.

É interessante notar que a maioria dos problemas de localização dinâmicos com capacidades é, na sua essência, um problema de expansão de capacidades: os serviços são abertos num determinado período temporal, aumentando a capacidade total disponível, de modo a que possa ser satisfeita a procura (geralmente crescente) dos clientes. No modelo desenvolvido a capacidade pode ser aumentada não só através da abertura de serviços em novas localizações, mas também através da abertura de novos equipamentos em localizações onde já existiam serviços abertos (à semelhança do que descreve Shulman, 1991).

Um dado serviço é, assim, constituído por um ou mais elementos de iguais ou diferentes capacidades. A capacidade máxima de um determinado equipamento deverá ser escolhida de um conjunto discreto de capacidades possíveis.

A diferença principal entre o modelo que a seguir se apresenta e os modelos existentes na literatura prende-se com a existência das possibilidades de: reduzir a capacidade em qualquer período temporal; poder localizar vários equipamentos de iguais ou diferentes capacidades em cada localização; poder abrir, fechar e reabrir cada equipamento mais do que uma vez durante o horizonte temporal. Pode ainda considerar-se um limite máximo ao número total de equipamentos que estão simultaneamente em funcionamento em cada localização.

Segundo Luss, 1982, as decisões mais importantes nos problemas de expansão de capacidades são as que definem o *quanto*, o *onde* e o *quando* aumentar as capacidades. No problema que agora se considera o mais importante é decidir o *quanto*, o *onde* e o *quando* aumentar ou reduzir as capacidades.

O problema irá ser agora formulado como um problema de programação linear inteira mista.

2.2.3.3.1. Formulação do Problema

Considere-se a notação definida em 2.2.2.1, página 24. Defina-se ainda:

$S = \{1, \dots, q\}$ conjunto de índices que correspondem às capacidades máximas possíveis dos equipamentos a localizar, em ordem crescente de capacidades;

c_{js}^t = custo de afectar na totalidade o cliente j a um equipamento de

dimensão s localizado em i durante o período t ;

FA_{ist}^{ξ} = custo fixo de abrir um equipamento de dimensão s em i no início do período t , e fechar esse equipamento no fim do período ξ (o equipamento estará em funcionamento desde o início de t até ao fim de ξ), sabendo que é o primeiro equipamento a ser localizado em i ;

FR_{ist}^{ξ} = custo unitário de abertura de um equipamento de dimensão s em i no início do período t , sendo fechado no fim do período ξ (o equipamento estará em funcionamento desde o início de t até ao fim de ξ), sabendo que este equipamento não é o primeiro a ser localizado em i ;

Q_s = capacidade máxima de um equipamento de dimensão s ;

$Nmax$ = número máximo de equipamentos que podem estar em funcionamento em cada localização e em cada período temporal.

Definam-se agora as variáveis de decisão:

$$a_{ist}^{\xi} = \begin{cases} 1 & \text{se o equipamento de dimensão } s \text{ localizado em } i \\ & \text{é aberto no início do período } t \text{ e se mantém em} \\ & \text{funcionamento até ao fim do período } \xi, \text{ sendo} \\ & \text{que este é o primeiro equipamento aberto em } i \\ & ; \\ 0 & \text{em caso contrário} \end{cases}$$

r_{ist}^{ξ} = número de equipamentos de dimensão s abertos em i no início do período t e que se mantêm em funcionamento até ao fim do período ξ , sabendo que nenhum deles é o primeiro equipamento a ser localizado em i ;

x_{ijs}^t = fracção da procura do cliente j que é satisfeita por equipamentos de dimensão s abertos em i durante o período t .

Em relação aos custos fixos de abertura e reabertura, é importante fazer a seguinte observação: o custo fixo FA_{ist}^{ξ} deverá ser igual a FR_{ist}^{ξ} , mas considerando adicionalmente os custos de localizar pela primeira vez um equipamento em i . Estes custos adicionais podem representar custos de aquisição de terrenos, desenvolvimento de infra-estruturas, etc. Se se definir este custo total adicional como f_i^t , então $FA_{ist}^{\xi} = FR_{ist}^{\xi} + f_i^t, \forall i, s, t, \xi \geq t$.

Ao primeiro equipamento a localizar em i será dada a designação de *i-first*, e a todos os outros equipamentos aí localizados a designação de *i-follow*.

Utilizando a notação já definida, o problema dinâmico de expansão e redução de capacidades em que é possível reconfigurar cada localização mais do que uma vez durante o horizonte temporal, pode ser formulado como PLDER.

PLDER

$$Min \sum_{t=1}^T \sum_{i=1}^m \sum_{j=1}^n \sum_{s=1}^q c_{ijs}^t x_{ijs}^t + \sum_{t=1}^T \sum_{i=1}^m \sum_{s=1}^q \sum_{\xi=t}^T FA_{ist}^{\xi} a_{ist}^{\xi} + \sum_{t=1}^T \sum_{i=1}^m \sum_{s=1}^q \sum_{\xi=t}^T FR_{ist}^{\xi} r_{ist}^{\xi} \quad (2-44)$$

sujeito a:

$$\sum_{i=1}^m \sum_{s=1}^q x_{ijs}^t = 1, \quad \forall j, t \quad (2-45)$$

$$\sum_{\tau=1}^t \sum_{\xi=t}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) - x_{ijs}^t \geq 0, \quad \forall i, j, s, t \quad (2-46)$$

$$Nmax \sum_{s=1}^q \sum_{\tau=1}^t \sum_{\psi=\tau}^T a_{is'\tau}^{\psi} - r_{ist}^{\xi} \geq 0, \quad \forall i, s, t, \xi \geq t \quad (2-47)$$

$$\sum_{s=1}^q \sum_{t=1}^T \sum_{\xi=t}^T a_{ist}^{\xi} \leq 1, \quad \forall i \quad (2-48)$$

$$\sum_{s=1}^q \sum_{\tau=1}^t \sum_{\xi=t}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) \leq Nmax, \quad \forall i, t \quad (2-49)$$

$$Q_s \sum_{\tau=1}^t \sum_{\xi=t}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) - \sum_{j=1}^n d_j^t x_{ijs}^t \geq 0, \quad \forall i, s, t \quad (2-50)$$

$$a_{ist}^{\xi} \in \{0, 1\}, r_{ist}^{\xi} \geq 0 \text{ e inteiro}, x_{ijs}^t \geq 0, \quad \forall i, j, s, t, \xi \geq t \quad (2-51)$$

Estas restrições garantem que: a procura de cada cliente será satisfeita em cada período temporal (2-45); cada cliente será apenas afecto a equipamentos que estejam em funcionamento em cada período temporal (2-46); um equipamento de dimensão s que seja aberto em i no início do período t pode ser considerado um equipamento *i-follow* se e só se existir um equipamento *i-first* que tenha sido aberto no início de qualquer período temporal t' , tal que $t' - t$ (2-47)³; em cada localização i poderá apenas existir um equipamento *i-first* durante todo o horizonte temporal considerado (2-48); existe um limite máximo ao número total de equipamentos que podem estar simultaneamente em funcionamento em cada localização i (2-49); as capacidades máximas dos equipamentos em funcionamento em cada período temporal não podem ser ultrapassadas (2-50). A função objectivo (2-44) minimiza os custos totais de localização e de afectação.

Deste modo, o modelo proposto permite que, em cada período temporal, exista um conjunto de equipamentos de iguais ou diferentes dimensões em cada localização, conjunto esse que pode ser alterado em cada período temporal aumentando ou diminuindo a capacidade disponível.

2.2.3.3.2. O Problema Dual e as Condições de Complementaridade

Considerem-se as restrições (2-48') e (2-49') obtidas através da multiplicação por -1 das restrições (2-48) e (2-49).

$$\sum_{s=1}^q \sum_{t=1}^T \sum_{\xi=t}^T -a_{ist}^{\xi} \geq -1, \quad \forall i \quad (2-48')$$

³ Note-se que um equipamento *i-first* e um equipamento *i-follow* podem ser localizados simultaneamente.

$$\sum_{s=1}^q \sum_{\tau=1}^t \sum_{\xi=\tau}^T -\left(a_{is\tau}^\xi + r_{is\tau}^\xi\right) \geq -Nmax, \quad \forall i, t \quad (2-49')$$

Associando variáveis duais v_j^t às restrições (2-45), variáveis duais w_{ijs}^t às restrições (2-46), variáveis duais u_{ist}^ξ às restrições (2-47), variáveis ρ_i às restrições (2-48'), variáveis π_i^t às restrições (2-49') e variáveis duais λ_{is}^t às restrições (2-50), então o problema dual pode ser formulado como D-PLDER.

D-PLDER

$$Max \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - Nmax \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-52)$$

sujeito a:

$$v_j^t - w_{ijs}^t - d_j^t \lambda_{is}^t \leq c_{ijs}^t, \quad \forall i, j, s, t, \quad (2-53)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} w_{ijs}^\tau + Nmax \sum_{s=1}^q \sum_{\tau=t}^T \sum_{\psi=\tau}^T u_{is'\tau}^\psi - \rho_i - \sum_{\tau=t}^{\xi} \pi_i^\tau + Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^\tau \leq FA_{ist}^\xi, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-54)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} w_{ijs}^\tau - u_{ist}^\xi - \sum_{\tau=t}^{\xi} \pi_i^\tau + Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^\tau \leq FR_{ist}^\xi, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-55)$$

$$w_{ijs}^t, u_{ist}^\xi, \rho_i, \pi_i^t, \lambda_{is}^t \geq 0, \quad \forall i, j, s, t, \xi = t, \dots, T \quad (2-56)$$

À semelhança do que foi feito anteriormente (página 27, secção 2.2.2.2) pode formular-se de forma “condensada” este problema dual, considerando $w_{ijs}^t = \max\{0, v_j^t - c_{ijs}^t - d_j^t \lambda_{is}^t\}$.

DC-PLDER

$$Max \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - Nmax \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-52)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^\tau - c_{ijs}^\tau - d_j^\tau \lambda_{is}^\tau\} \leq FA_{ist}^\xi - Nmax \sum_{s=1}^q \sum_{\tau=t}^T \sum_{\psi=\tau}^T u_{is'\tau}^\psi + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^\tau - Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^\tau, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-57)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^\tau - c_{ijs}^\tau - d_j^\tau \lambda_{is}^\tau\} \leq FR_{ist}^\xi + u_{ist}^\xi + \sum_{\tau=t}^{\xi} \pi_i^\tau - Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^\tau, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-58)$$

$$u_{ist}^\xi, \rho_i, \pi_i^t, \lambda_{is}^t \geq 0, \quad \forall i, j, s, t, \xi = t, \dots, T \quad (2-59)$$

Defina-se:

$$SA_{ist}^\xi = FA_{ist}^\xi - Nmax \sum_{s=1}^q \sum_{\tau=t}^T \sum_{\psi=\tau}^T u_{is'\tau}^\psi + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^\tau - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^\tau - c_{ijs}^\tau - d_j^\tau \lambda_{is}^\tau\} - Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^\tau, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-60)$$

$$SR_{ist}^{\xi} = FR_{ist}^{\xi} + u_{ist}^{\xi} + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \max\{0, v_j^{\tau} - c_{ijs}^{\tau} - d_j^{\tau} \lambda_{is}^{\tau}\} - Q_s \sum_{\tau=t}^{\xi} \lambda_{is}^{\tau},$$

$$\forall i, s, t, \xi = t, \dots, T \quad (2-61)$$

$$SA_{ist}^{\xi} = \min\{SA_{ist}^{\xi}, SR_{ist}^{\xi}\},$$

$$\forall i, s, t, \xi = t, \dots, T \quad (2-62)$$

Podem agora estabelecer-se as condições de complementaridade que serão satisfeitas por um par de soluções primal e dual admissíveis e ótimas, no caso de não existência de *gap* de dualidade.

$$\left(\sum_{\tau=1}^t \sum_{\xi=t}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) - x_{ijs}^t \right) w_{ijs}^t = 0, \quad \forall i, j, t \quad (2-63)$$

$$\left(Nmax \sum_{s'=1}^q \sum_{\tau=1}^t \sum_{\psi=\tau}^T a_{is'\tau}^{\psi} - r_{ist}^{\xi} \right) u_{ist}^{\xi} = 0, \quad \forall i, s, t, \xi \geq t \quad (2-64)$$

$$\left(\sum_{s=1}^q \sum_{t=1}^T \sum_{\xi=t}^T a_{ist}^{\xi} - 1 \right) \rho_i = 0, \quad \forall i \quad (2-65)$$

$$\left(\sum_{s=1}^q \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) - Nmax \right) \pi_i^t = 0, \quad \forall i, t \quad (2-66)$$

$$SA_{ist}^{\xi} a_{ist}^{\xi} = 0, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-67)$$

$$SR_{ist}^{\xi} r_{ist}^{\xi} = 0, \quad \forall i, s, t, \xi = t, \dots, T \quad (2-68)$$

$$\left(Q_s \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{is\tau}^{\xi} + r_{is\tau}^{\xi}) - \sum_{j=1}^n d_j^t x_{ijs}^t \right) \lambda_{is}^t = 0, \quad \forall i, s, t \quad (2-69)$$

2.3. Problemas de Localização Multi-Nível

2.3.1. Introdução

Os problemas de localização multi-nível têm sido largamente estudados na literatura, sob designações diferentes de diferentes autores: hierárquicos, multi-estágios, *multi-echelon*, etc. Geralmente, a designação de um determinado problema indica o número máximo de níveis considerados: *k*-hierárquico ou *k*-nível, por exemplo, referem-se a problemas com *k* níveis de serviço.

Neste trabalho, este tipo de problemas são designados por multi-nível porque é o entendimento da autora que esta é a designação mais geral: um problema de localização multi-nível engloba problemas hierárquicos, sem que obrigue a que exista uma estrutura hierárquica definida. Muitos são os exemplos do dia-a-dia que ilustram a importância de se considerarem este tipo de problemas de localização: o sistema de saúde, constituído por centros de saúde, hospitais distritais e hospitais centrais; o sistema escolar, constituído pelas escolas primárias, secundárias e estabelecimentos de ensino superior, e também organizações como correios ou instituições bancárias. Todas estas estruturas de

serviços têm algumas características em comum e outras que as diferenciam. Pode dizer-se que um modelo de localização multi-nível é necessário se os serviços a localizar podem ser agrupados em conjuntos (níveis) tais que todos os serviços de um dado conjunto oferecem o mesmo tipo de bem que por sua vez é distinto de conjunto para conjunto. Os serviços que pertencem a conjuntos diferentes interagem uns com os outros, pelo que não é de todo possível localizar serviços de um conjunto independentemente dos outros. O tipo de ligação existente entre serviços de conjuntos diferentes pode ser implícita (na forma de uma restrição que limite, por exemplo, o montante máximo total a gastar na localização de todos os serviços), ou pode ser explícita determinando de que forma se processa o fluxo de clientes entre os diferentes tipos de serviços (Daskin, 1995).

Narula, 1984, tenta classificar os problemas de localização-afecção hierárquicos. O autor diferencia os problemas através do tipo de hierarquia (*sucessivamente inclusiva* – um equipamento de nível m oferece todos os serviços oferecidos por equipamentos em níveis de 1 a $m-1$ e também serviços que são específicos do nível m ; *sucessivamente exclusiva* – um equipamento de nível m oferece serviços que não podem ser encontrados em qualquer outro nível); disciplinas de arcos e fluxos (o fluxo diz-se *integrado* se ocorre desde um qualquer nível inferior para outro qualquer nível superior, sendo *discriminado* se tiver de ocorrer entre um nível m e o nível $m+1$; uma rede diz-se de *caminho único* se a diferença entre o número de arcos incidentes no nodo e o número de arcos divergentes do nodo é menor ou igual a 1 e diz-se de *caminhos múltiplos* se existe pelo menos um nodo tal que esta diferença é maior do que um). Para que se consiga caracterizar por completo um dado problema é necessário conhecer o número de níveis, o tipo de hierarquia entre serviços e a disciplina de arcos e fluxos. No artigo referenciado, o próprio autor reconhece que a classificação apresentada não é exaustiva. Em Daskin, 1995 (pp 317-333) podem encontrar-se os traços básicos que definem um problema multi-nível, dando o autor alguns exemplos baseados nos problemas da p -mediana e de cobertura.

A maioria dos problemas descritos na literatura são estáticos. Alguns autores consideram serviços sem capacidades, enquanto outros se debruçam no estudo das versões com capacidades. Sahin & Süral, 2007, apresenta uma revisão de alguns dos modelos utilizados nesta área.

Moore & ReVelle, 1982, descrevem um problema de localização com hierarquia sucessivamente inclusiva, baseado num caso real. O objectivo é a minimização da população que não tem acesso a um ou mais componentes dos bens disponibilizados pelos serviços. Uma população é abrangida (*coberta*) por um dado nível de serviço se algum equipamento que seja capaz de fornecer esse serviço esteja localizado a uma distância inferior a um limiar máximo. Tien, El-Tell, & Simons, 1983, desenvolvem e resolvem dois problemas com aplicações ao sistema hierárquico de saúde. Narula & Ogbu, 1985, estudam um problema sem capacidades e com dois níveis de serviços (sucessivamente inclusivo) tal que devem ser localizados p_1 serviços no nível 1 e p_2 serviços no nível 2. Apenas uma parte de toda a procura satisfeita pelos serviços de nível 1 precisa ainda de

ser satisfeita pelos serviços de nível 2. Laporte & Dejax, 1989, definem um problema de localização e rotas óptimas com dois níveis de serviços. Consideram que os serviços no primeiro nível são fixos e conhecidos, e que a localização dos serviços de nível 2 tem de ser determinada. Vernekar, Anandalingam & Dorny, 1990, formulam o problema de localizar recursos (como bases de dados, controladores, etc.) numa rede de computadores organizada de forma hierárquica, por forma a tornar esses recursos disponíveis a todos os utilizadores a um custo mínimo. Eitan, Narula & Tien, 1991, apresentam um modelo de programação linear inteira mista que permite que várias relações de hierarquia estejam simultaneamente presentes, que sejam impostas restrições de capacidade, tanto por tipo de serviço, como por grupos de serviços, considerando tanto custos fixos, como custos variáveis na função objectivo. Este modelo é aplicado a vários problemas descritos na literatura, bem como a um novo problema de grandes dimensões e que é utilizado para ilustrar todas as características do modelo apresentado. Aardal, Labbé, Leung & Queyranne, 1996, investigam as propriedades estruturais dos problemas de dois níveis sem capacidades. São estudadas duas formulações distintas a que os autores chamam *single* e *multi commodity flow*, e que diferem nas variáveis de decisão utilizadas. Bloemhof-Ruwaard, Salomon & Wassenhove, 1996, descrevem um problema que é uma generalização do problema de localização com capacidades, com dois níveis: o problema de distribuição e deposição de resíduos. Caprara & González, 1994, apresentam um modelo que é uma generalização do PLS e que considera que cada cliente pode ser servido por um conjunto de serviços (configuração) em vez de por um único serviço. O modelo apresentado pode ser facilmente interpretado como um problema multi-nível estático (com cada configuração de serviços a corresponder a um caminho de serviços), apesar de tal possibilidade não ser referida pelos autores. Andreas Klose, 1995, 1998, 2000, considera um problema de afectação total em que os clientes são servidos por armazéns que por sua vez são servidos por equipamentos produtivos. No entanto, as variáveis de decisão representam apenas a localização dos armazéns, pois o autor considera que os restantes serviços já foram previamente localizados. Bloemhof-Ruwaard *et al.*, 1996, apresentam a versão com capacidades do problema de localização com dois níveis. Consideram duas possíveis formulações e comparam os resultados obtidos em termos da qualidade da solução primal óptima de ambas as relaxações lineares e de uma relaxação lagrangeana. Tragantalerngsak, Holt & Rönnqvist, 1997, desenvolvem e comparam seis relaxações lagrangeanas diferentes para o problema de localização com capacidades e dois níveis e com afectação total. Marin & Pelegrin, 1997, estudam um caso especial de um problema de localização com dois níveis: o problema de entrepostos, que não estão ainda instalados. Min, Jayaraman & Srivastava, 1998, consideram os problemas conjuntos de localização e rotas óptimas, em que referem o problema de localização com dois níveis. Pirkul & Jayaraman, 1998, desenvolvem um modelo de dois níveis, com capacidades e multi-produto. Tüshaus & Wittmann, 1998, consideram um problema de distribuição com duas configurações possíveis: fábrica, armazém e cliente, ou fábrica, armazém, entreposto e cliente. Formulam ambas as alternativas como PLS, porque apenas

consideram a possibilidade de existência de uma fábrica. Deste modo a primeira alternativa, em que existem dois níveis, resume-se a um problema com apenas um nível e na segunda alternativa o problema é decomposto em dois sub-problemas. Pirkul & Jayaraman, 1998, consideram a versão multi-produto do problema de localização com dois níveis: os clientes são afectos a armazéns abertos que por sua vez recebem produtos de instalações produtivas. P. Chardaire, 1999, lida com problemas de dois níveis sem e com capacidades no contexto do planeamento de redes de telecomunicações. Edwards, 2001 e Ageev, 2002 estudam algumas propriedades e descrevem algoritmos aproximativos para o problema multi-nível. Marianov & Serra, 2001, estudam um problema com uma estrutura hierárquica de dois níveis de serviços, considerando de forma explícita (através de uma descrição probabilística) a congestão no sistema. A congestão ocorre sempre que um determinado serviço não é capaz de simultaneamente servir todos os pedidos de satisfação de procura, sendo que as usuais restrições que impõem capacidades máximas não são capazes de reflectir este problema. Drezner & Wesolowsky, 2001, estudam um problema que pode ser considerado como multi-nível: deseja localizar-se um único serviço (numa linha ou num plano) que serve um conjunto de clientes. A satisfação da procura de um dado cliente compreende percorrer o caminho desde o serviço a localizar até ao cliente, depois o caminho entre o cliente e um depósito e de volta até ao serviço. Bumb, 2001, e Zhang & Ye, 2002, estudam algoritmos aproximativos para o problema com dois níveis, mas com uma função objectivo de maximização. Branas & Reville, 2001, desenvolvem um modelo hierárquico com dois níveis, que maximiza a cobertura de acesso a serviços de apoio e tratamento em situações de acidente. A cobertura é medida pelo número de indivíduos feridos que têm acesso a um ou ambos os recursos médicos: hospitais e helicópteros. Roberto D. Galvão, Espejo, & Boffey, 2002; R. D. Galvão, Espejo & Boffey, 2006, aplicam um modelo de localização com 3 níveis, sucessivamente inclusivo, em que existe um limite máximo ao número de serviços a localizar em cada nível, ao estudo de um caso de localização de serviços pré-natais. Os autores não consideram a existência de custos fixos de abertura. Em Roberto D. Galvão, Espejo, Boffey & Yates, 2006 apresentam um modelo com restrições de capacidade. Syam, 2002, considera um problema de localização com três níveis (clientes, depósitos e serviços de produção) em que introduz algumas componentes de logística. Espejo, Galvão & Boffey, 2003, estudam o problema de cobertura máxima com dois níveis. Jayaraman, Gupta & Pirkul, 2003, estudam um problema de localização hierárquico em que os clientes têm dois tipos distintos de procura: procura por serviços básicos e especializados. O modelo apresentado pretende maximizar a cobertura, afectando tipos de serviços a equipamentos abertos e considerando ainda restrições de capacidade. Landete & Marín, 2003, apresentam uma nova formulação para o problema de localização com dois níveis, sem capacidades. Espejo & Galvão, 2004, trabalham com um problema hierárquico de cobertura máxima, em que as coberturas para os dois níveis considerados são maximizadas de forma independente. Os autores propõem-se utilizar soluções geradas durante o processo de optimização de um problema dual lagrangeano no cálculo de uma aproximação da fronteira

eficiente. Ignacio, Ferreira Filho & Galvão, 2008, propõem a utilização de um modelo hierárquico de dois níveis para localização de equipamentos numa rede de computadores. M. J. Hodgson & Jacobsen, 2009, apresentam um modelo *p*-mediana hierárquico, mas considerando a hipótese de cada cliente não querer ser afecto ao serviço aberto mais próximo, utilizando modelos de interacção espacial.

O número de referências que consideram problemas de localização dinâmica multi-nível é bastante menor. Emanuel Melachrinoudis & Min, 2000, apresentam um caso real em que é necessário relocar um serviço existente que faz parte de uma estrutura hierárquica de dois níveis. A transição entre a localização actual e a localização futura tem de ser feita de forma gradual. Os autores consideram restrições de capacidade e restrições que impõem um limite máximo correspondente ao orçamento disponível para efectuar a realocação. O modelo apresentado é multi-objectivo. Hinojosa, Y., Puerto & Fernández, 2000, lidam com um problema de localização dinâmica, considerando a procura de mais do que um produto por parte dos clientes e o transporte dos vários produtos desde as localizações onde são produzidas até armazéns, e destes até aos clientes. O modelo considera a possibilidade de abrir um serviço no início de um período temporal (mantendo-se este serviço aberto até ao fim do horizonte temporal), e fechar um serviço que esteja em funcionamento antes do primeiro período temporal (sendo que estes serviços, após serem fechados, se mantêm fechados até ao fim do horizonte temporal). Canel *et al.*, 2001, desenvolvem um modelo para o mesmo tipo de problema, mas consideram a possibilidade de um serviço estar aberto em vários períodos temporais, não consecutivos. Os autores consideram custos fixos de abertura e fecho, e apresentam uma função objectivo não linear. Melo *et al.*, 2003, 2006, estudam um problema de localização complexo integrado numa cadeia de aprovisionamento em rede. No modelo apresentado consideram quer a abertura de novos serviços, quer a expansão, redução e realocação de serviços já existentes. Gue, 2003, desenvolve um modelo dinâmico de localização e de determinação de fluxos de material utilizado no planeamento de uma batalha. É necessário providenciar o suporte necessário às forças combatentes, tendo em conta restrições de limitação de recursos. Thanh, Bostel & Péton, 2008, desenvolvem um modelo de localização dinâmica, integrado no desenho de uma cadeia de aprovisionamento, incluindo aspectos relacionados com a gestão de existências.

Alguns aspectos interessantes na área dos problemas de localização multi-nível têm sido igualmente tratados por alguns autores. M. John Hodgson, 1981, estuda o problema de localização de serviços públicos “a caminho do trabalho”. Considera-se que os clientes têm que fazer um percurso diário de casa para o trabalho. Assim, o objectivo do problema é localizar serviços públicos de forma a minimizar o tempo extra necessário para visitar esses serviços, enquanto se faz o percurso diário. Madsen, 1983, estuda o problema de localização e determinação de rotas óptimas num sistema composto por clientes, depósitos e, no máximo, uma fábrica. Serra, Marianov & ReVelle, 1992, estudam o problema de localizar serviços com uma estrutura hierárquica quando existe competição

entre serviços. O modelo desenvolvido permite quer a localização de novos serviços, quer a realocação de serviços existentes. Marianov & Serra, 2001, estudam o problema de localização hierárquico em que as questões de congestionamento são tratadas de forma explícita: há situações em que os clientes têm de esperar numa fila de espera antes de verem a sua procura satisfeita.

No presente capítulo, vários problemas de localização multi-nível serão considerados (Joana Dias, Captivo, & Clímaco, 2004c; Joana Dias, Captivo, & Clímaco, 2007b): sem capacidades, com capacidades, com capacidades e sem conservação de fluxo em alguns nodos intermédios da rede.

Em Joana Dias, Captivo, & Clímaco, 2005c, os modelos aqui apresentados são utilizados no desenvolvimento de um modelo global representando o planeamento de um sistema de tratamento de resíduos sólidos urbanos.

2.3.2. Descrição do Problema

Existe um conjunto de clientes cuja procura necessita de ser satisfeita. A satisfação da procura dos clientes obriga à utilização de bens fornecidos por um ou mais serviços que se encontram organizados numa estrutura de níveis. Os serviços podem ter, ou não, restrições de capacidades máximas, existindo custos fixos de localização e custos de afectação de clientes a serviços ou conjuntos de serviços.

A Figura 2.1 ilustra um possível problema de localização com dois níveis de serviços.

Para se considerar a afectação de clientes a conjuntos de serviços, duas alternativas foram consideradas. Na primeira alternativa consideram-se variáveis de afectação do tipo x_{ij}^t , que representam o fluxo entre o nodo i e o nodo j durante o período temporal t , sendo que o nodo i pode corresponder a um cliente ou a um serviço que esteja localizado num nodo intermédio da rede. Na segunda alternativa consideram-se variáveis do tipo x_{pj}^t , que representam a afectação de um cliente j a um caminho p constituído por um ou vários serviços, no período temporal t .

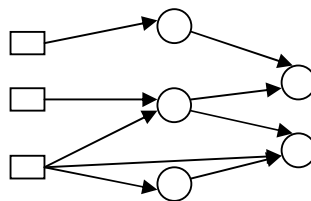


Figura 2.1 – Exemplo de um problema de localização dinâmica com 2 níveis de serviços
 ○ - Possíveis localizações para os serviços; □ - Localizações dos clientes.

Todos os modelos desenvolvidos têm em comum as características seguintes, que os distinguem dos modelos geralmente encontrados na literatura:

- 1) Os problemas são dinâmicos e consideram que um serviço pode ser aberto, fechado e reaberto mais do que uma vez durante o horizonte temporal. É possível considerar custos de abertura, fecho e reabertura, e a existência inicial de serviços abertos.
- 2) Os modelos podem ser usados considerando hierarquias sucessivamente inclusivas ou exclusivas. São também válidos para todas as disciplinas de arcos e fluxos, tais como definidos em Narula, 1984.

Em relação aos modelos apresentados na secção 2.3.4, página 45, que consideram variáveis do tipo x_{pj}^t , pode ainda dizer-se que:

- 3) Os modelos conseguem lidar com caminhos contendo diferentes números de serviços. Num problema de k níveis, então os caminhos poderão ter entre 1 e k serviços. Este aspecto é bastante interessante e com aplicabilidade prática: imagine-se, por exemplo, um sistema de cuidados de saúde. Os pacientes são atendidos no seu centro de saúde, e daí podem ser encaminhados para um hospital. No entanto, se o hospital ficar perto da sua residência, muitos pacientes tenderão a deslocar-se em primeiro lugar ao hospital.
- 4) É possível considerar mais do que um caminho constituído exactamente pelos mesmos serviços. Esta característica apresenta-se como bastante útil se se pretender trabalhar com mais do que uma função objectivo. Considere-se, por exemplo, que se pretende minimizar o custo total e ainda o tempo total de deslocações. Considere-se agora o percurso entre dois serviços que estão ligados entre si através de uma estrada nacional e também de uma auto-estrada. A primeira opção é menos dispendiosa mas mais demorada, passando-se o inverso com a segunda opção. Ambas as opções podem ser explicitamente consideradas nos modelos desenvolvidos.

2.3.3. Variáveis de Controlo de Fluxo

Como se referiu na secção anterior, se se considerarem variáveis de afectação do tipo x_{ij}^t , que representem o fluxo entre o nodo i e o nodo j num determinado período temporal t , então o modelo que se irá obter é indissociável da rede que representa as interacções entre clientes e serviços e entre serviços de níveis diferentes.

Defina-se o conjunto I_k , $0 \leq k \leq K$ como sendo o conjunto contido em I tal que $i \in I_k$ se i está no nível k da rede que define a interacção ente os serviços. Considera-se que $I_0 = \emptyset$. Seja:

K = número máximo de níveis na estrutura multi-nível de serviços (representa simultaneamente o número máximo de serviços que podem pertencer a um caminho, ou seja, o tamanho máximo dos caminhos);

c_{ji}^t = custo unitário de transporte entre o cliente $j \in J$ e o serviço $i \in I$ durante o período t ;

$c_{i'i'}^t$ = custo unitário de transporte entre o serviço $i \in I_k$ e $i' \in I_{k+1}$, $k < K$.

Definam-se as seguintes variáveis de decisão:

x_{ji}^t = quantidade de fluxo entre o cliente $j \in J$ e o serviço $i \in I$ durante o período t ;

$x_{i'i'}^t$ = quantidade de fluxo entre o serviço $i \in I_k$ e $i' \in I_{k+1}$, $k < K$.

2.3.3.1. Conservação de Fluxo em Todos os Nodos da Rede

O problema de localização dinâmica multi-nível sem capacidades pode ser formulado como PLDM1.

PLDM1

$$\text{Min} \sum_{t=1}^T \sum_{j=1}^n \sum_{i=1}^m c_{ij}^t x_{ji}^t + \sum_{k=1}^{K-1} \sum_{t=1}^T \sum_{i \in I_k} \sum_{i' \in I_{k+1}} c_{ii'}^t x_{ii'}^t + \sum_{t=1}^T \sum_{i=1}^m \sum_{\xi=t}^T FA_{ii}^\xi a_{ii}^\xi + \sum_{t=2}^T \sum_{i=1}^m \sum_{\xi=t}^T FR_{ii}^\xi r_{ii}^\xi \quad (2-70)$$

sujeito a:

$$\sum_{i \in I} x_{ji}^t = d_j^t, \quad \forall j \in J, t = 1, \dots, T \quad (2-71)$$

$$\sum_{i \in I_{k+1}} x_{ii'}^t - \left(\sum_{i \in I_{k-1}} x_{i'i}^t + \sum_{j \in J} x_{ji}^t \right) = 0, \quad \forall i \in I_k, k = 1, \dots, K-1, t = 1, \dots, T \quad (2-72)$$

$$Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^\xi + r_{i\tau}^\xi) \leq \sum_{j \in J} x_{ji}^t + \sum_{i' \in I_{k-1}} x_{i'i}^t \leq Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^\xi + r_{i\tau}^\xi), \quad \forall i \in I_k, k = 1, \dots, K-1, t = 1, \dots, T \quad (2-73)$$

$$\sum_{\tau=1}^{t-1} \sum_{\xi=\tau}^{t-1} a_{i\tau}^\xi - \sum_{\xi=t}^T r_{i\tau}^\xi \geq 0, \quad \forall i \in I, t = 1, \dots, T \quad (2-4)$$

$$\sum_{i=1}^T \sum_{\xi=i}^T a_{ii}^\xi \leq 1, \quad \forall i \in I \quad (2-5)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^\xi + r_{i\tau}^\xi) \leq 1, \quad \forall i \in I, t = 2, \dots, T \quad (2-6)$$

$$a_{ii}^\xi, r_{ii}^\xi \in \{0, 1\}, \quad \forall t, \xi \geq t \quad (2-74)$$

$$x_{ji}^t, x_{i'i}^t \geq 0 \quad \forall i, i', t \quad (2-75)$$

A função objectivo minimiza os custos totais de localização e de afectação. As restrições (2-71) garantem que a procura total de cada cliente é satisfeita. As restrições (2-72) são restrições de conservação de fluxo em nodos intermédios da rede, e as restrições (2-73) estabelecem capacidades mínimas e máximas para cada serviço i , garantindo simultaneamente que só pode existir fluxo em serviços que estejam em funcionamento, em cada período temporal. Note-se que, nesta formulação, devem apenas ser consideradas variáveis x_{ji}^t e $x_{i'i}^t$ tais que existe o respectivo arco na rede.

2.3.3.2. Não Conservação de Fluxo em Um ou Mais Nodos da Rede

No modelo apresentado anteriormente considera-se que existe conservação de fluxo em todos os nodos intermédios da rede. Isto significa que, se um nodo recebe um fluxo d_i , então encaminha a mesma quantidade de fluxo para os nodos que lhe são subsequentes.

Considere-se, agora, que existe um parâmetro θ_i associado ao serviço i , tal que i pertence a um nível entre 1 e $K-1$ e $\theta_i \in]0, 1]$. Se um fluxo de d_i unidades chega ao nodo i , então este serviço passará aos serviços subsequentes um fluxo igual a $d_i \theta_i$, tal como é ilustrado na Figura 2.2.

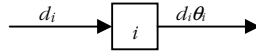


Figura 2.2 – Não conservação de fluxo num nodo intermediário da rede

Existem vários tipos de equipamentos que apresentam este tipo de comportamento. Em sistemas de recolha e tratamento de resíduos, as incineradoras recebem uma dada quantidade de resíduos para incinerar e uma pequena parte deste resíduos tem ainda de ser depositada em aterros sanitários (na forma de escória e cinzas). Num sistema de cuidados de saúde, apenas uma parte dos pacientes que se dirigem a um centro de saúde terão de posteriormente ser encaminhados para um hospital.

A única diferença existente entre o modelo que considera os parâmetros θ_i e o modelo anterior reside nas restrições de conservação de fluxo (2-72) que são substituídas pelas restrições (2-76).

$$\sum_{i' \in I_{k+1}} x_{ii'}^t - \theta_i \left(\sum_{i' \in I_{k-1}} x_{i'i}^t + \sum_{j \in J} x_{ji}^t \right) = 0 \quad \forall i \in I_k, k = 1, \dots, K-1, t = 1, \dots, T \quad (2-76)$$

2.3.4. Variáveis de Afectação a Caminhos

A definição de variáveis do tipo x_{pj}^t , que representam a afectação de um cliente j a um caminho de serviços p no período temporal t , permite que seja desenvolvido um modelo de localização dinâmica multi-nível válido para um conjunto vasto de diferentes configurações da rede.

Defina-se:

$P = \{1, \dots, p, \dots, Pmax\}$ conjunto de todos os possíveis caminhos que contêm um ou mais serviços;

$P(i) = \{p \in P : i \text{ pertence ao caminho } p\}$;

K = número máximo de níveis na estrutura multi-nível de serviços (representa simultaneamente o número máximo de serviços que podem pertencer a um caminho, ou seja, o tamanho máximo dos caminhos);

c_{pj}^t = custo de afectar a totalidade da procura do cliente j ao caminho p ;

e considerem-se as variáveis de decisão:

$$x_{pj}^t = \begin{cases} 1 & \text{se o cliente } j \text{ é afecto ao caminho } p \text{ durante o período } t \\ 0 & \text{em caso contrário} \end{cases}$$

Esta definição das variáveis de afectação é motivada pelo trabalho de Tcha & Lee, 1984. A principal diferença reside no facto de cada caminho p poder ter um qualquer número de arcos, enquanto que no artigo referido todos os caminhos têm de ser exactamente do mesmo tamanho.

Cada caminho pode ser constituído por um ou mais serviços (no máximo um serviço pertencente a cada nível) e cada serviço pode pertencer a um ou mais caminhos.

Um caminho p pode ser representado por um conjunto ordenado (i_1, i_2, \dots, i_k) , com $k \leq K$. Se um cliente j é afecto ao caminho p , isto significa que o cliente j é primeiro servido pelo serviço i_1 , imediatamente a seguir pelo serviço i_2 , e assim

sucessivamente até chegar ao serviço i_b .

Definição 2.1: Diz-se que o caminho p está aberto durante o período t se e só se todos os serviços que pertencem a p estão abertos em t .

Definição 2.2: Diz-se que o caminho p está parcialmente aberto durante o período t se e só se p não está aberto e pelo menos um serviço i pertencente a p está aberto durante o período t .

Definição 2.3: Diz-se que o caminho p está fechado durante o período t se e só se todos os serviços pertencentes ao caminho p estão fechados durante o período t .

2.3.4.1. Problemas Primais

2.3.4.1.1. Conservação de Fluxo em Todos os Nodos da Rede

O problema de localização dinâmica multi-nível sem capacidades pode ser formulado como PLDM2.

PLDM2

$$\text{Min} \sum_{t=1}^T \sum_{p=1}^{P_{\max}} \sum_{j=1}^m C_{pj}^t x_{pj}^t + \sum_{t=1}^T \sum_{i=1}^m \sum_{\xi=t}^T FA_{ii}^{\xi} a_{ii}^{\xi} + \sum_{t=2}^T \sum_{i=1}^m \sum_{\xi=t}^T FR_{ii}^{\xi} r_{ii}^{\xi} \quad (2-77)$$

sujeito a:

$$\sum_{p=1}^{P_{\max}} x_{pj}^t = 1, \quad \forall j, t=1, \dots, T \quad (2-78)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - x_{pj}^t \geq 0, \quad \forall i, j, t, p \in P(i) \quad (2-79)$$

$$\sum_{\tau=1}^{t-1} \sum_{\xi=\tau}^{t-1} a_{i\tau}^{\xi} - \sum_{\xi=t}^T r_{ii}^{\xi} \geq 0, \quad \forall i \in I, t=1, \dots, T \quad (2-4)$$

$$\sum_{i=1}^T \sum_{\xi=i}^T a_{ii}^{\xi} \leq 1, \quad \forall i \in I \quad (2-5)$$

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \leq 1, \quad \forall i \in I, t=2, \dots, T \quad (2-6)$$

$$a_{ii}^{\xi}, x_{pj}^t \in \{0, 1\}, \quad \forall p, j, t = 1, \dots, T, \xi \geq t \quad (2-80)$$

$$r_{ii}^{\xi} \in \{0, 1\}, \quad \forall i, t = 2, \dots, T, \xi \geq t \quad (2-81)$$

A função objectivo (2-77) minimiza os custos totais. As restrições (2-78) garantem que, em cada período temporal, cada cliente seja afecto a um caminho. Um cliente pode apenas ser afecto a caminhos abertos (2-79). As restrições (2-79) poderiam ser substituídas pelas restrições agregadas (2-79').

$$\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{p \in P(i)} x_{pj}^t \geq 0 \quad \forall i, j, t \quad (2-79')$$

No modelo apresentado podem considerar-se restrições adicionais que ilustrem a capacidade máxima de cada serviço (2-82). Neste caso, as variáveis de

afecção x_{pj}^t não serão binárias, mas sim contínuas, dentro do intervalo $[0,1]$ e representam a fracção da procura do cliente j que é satisfeita pelos serviços pertencentes ao caminho p . Poderiam também considerar-se restrições de capacidades mínimas.

$$Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^\xi + r_{i\tau}^\xi) - \sum_{j=1}^n \sum_{p \in P(i)} d_j^t x_{pj}^t \geq 0, \quad \forall i, t \quad (2-82)$$

2.3.4.1.2. Não Conservação de Fluxo em Um ou Mais Nodos da Rede

No modelo apresentado anteriormente considera-se que existe conservação de fluxo em todos os nodos intermédios de todos os caminhos admissíveis. Isto significa que o fluxo que circula num dado caminho p se mantém sempre constante, ou seja, se um nodo recebe um fluxo d então encaminha a mesma quantidade de fluxo para os nodos que lhe são subsequentes.

Considere-se, tal como em 2.3.3.2, que existe um parâmetro θ_i associado ao serviço i , tal que i pertence a um nível entre 1 e $K-1$ e $\theta_i \in]0,1]$. Se um fluxo de d_i unidades chega ao nodo i , então este serviço passará aos serviços subsequentes um fluxo igual a $d_i \theta_i$. A única diferença existente entre o modelo que considera os parâmetros θ_i e o modelo anterior reside apenas nas restrições de capacidade (2-82) que são substituídas pelas restrições (2-83). Este modelo pode ser considerado como uma generalização do anterior em que todos os parâmetros θ_i são iguais a um.

Definição 2.4: Considerem-se dois serviços i e i' e um caminho p tal que $i \in p$ e $i' \in p$. Então $i <_p i'$ se e só se o serviço i se encontra antes de i' no conjunto ordenado p .

$$Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^\xi + r_{i\tau}^\xi) - \sum_{j=1}^n \sum_{p \in P(i)} d_j^t x_{pj}^t \prod_{\substack{i' <_p i \\ p}} \theta_{i'} \geq 0, \quad \forall i, t \quad (2-83)$$

Se se pretender, pode também considerar-se a introdução de restrições de capacidade mínima (2-84).

$$Q_{i'} \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i'\tau}^\xi + r_{i'\tau}^\xi) - \sum_{j=1}^n \sum_{p \in P(i')} d_j^t x_{pj}^t \prod_{\substack{i \in p \\ i' <_p i}} \theta_{i'} \leq 0, \quad \forall i, t \quad (2-84)$$

2.3.4.2. Problemas Duais e Condições de Complementaridade

Considerando as variáveis duais já definidas anteriormente, e associando as variáveis duais w_{ijp}^t às restrições (2-79), o problema dual da relaxação linear do problema PLDM2 pode ser formulado como D-PLDM2.

D-PLDM2

$$Max \sum_{i=1}^T \sum_{j=1}^n v_j^i - \sum_{i=1}^m \rho_i - \sum_{i=1}^T \sum_{i=1}^m \pi_i^i \quad (2-9)$$

sujeito a:

$$v_j^i - \sum_{i \in P} w_{ijp}^i \leq c_{pj}^i, \quad \forall p, j, t \quad (2-85)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} w_{ijp}^{\tau} + \sum_{\tau=\xi+1}^T u_i^{\tau} - \rho_i - \sum_{\tau=t}^{\xi} \pi_i^{\tau} \leq FA_{it}^{\xi}, \quad \forall i, t, \xi = t, \dots, T \quad (2-86)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} w_{ijp}^{\tau} - u_i^t - \sum_{\tau=t}^{\xi} \pi_i^{\tau} \leq FR_{it}^{\xi}, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-87)$$

$$w_{ijp}^i, u_i^i, \rho_i, \pi_i^i \geq 0, \quad \forall i, j, t, p \in P(i) \quad (2-88)$$

Considerando $u_{ijp}^i = \eta_{ijp}^i \max\{0, v_j^i - c_{pj}^i\}$, tal que $\sum_{i \in P} \eta_{ijp}^i = 1$ e $\eta_{ijp}^i \geq 0$,

$\forall i, j, p \in P(i)$, então pode obter-se uma formulação equivalente e “condensada” DC-PLDM2.

DC-PLDM2

$$Max \sum_{i=1}^T \sum_{j=1}^n v_j^i - \sum_{i=1}^m \rho_i - \sum_{i=1}^T \sum_{i=1}^m \pi_i^i \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max\{0, v_j^{\tau} - c_{pj}^{\tau}\} \leq FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau}, \quad \forall i, t, \xi = t, \dots, T \quad (2-89)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max\{0, v_j^{\tau} - c_{pj}^{\tau}\} \leq FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau}, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-90)$$

$$u_i^i, \rho_i, \pi_i^i \geq 0, \quad \forall i, j, t \quad (2-91)$$

Defina-se:

$$SA_{it}^{\xi} = FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max\{0, v_j^{\tau} - c_{pj}^{\tau}\}, \quad \forall i, t, \xi = t, \dots, T \quad (2-92)$$

$$SR_{it}^{\xi} = FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - \sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max\{0, v_j^{\tau} - c_{pj}^{\tau}\}, \quad \forall i, t > 1, \xi = t, \dots, T \quad (2-93)$$

$$S_{it}^{\xi} = \min\{SA_{it}^{\xi}, SR_{it}^{\xi}\}, \quad \forall i \in I, t = 1, \dots, T, \xi = t, \dots, T \quad (2-20)$$

Se estivermos na presença de um par de soluções primal e dual admissíveis e ótimas e na ausência de *gap* de dualidade, então as condições de complementaridade (2-22) a (2-26) e ainda (2-94) serão satisfeitas.

$$\left(\sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - x_{pj}^t \right) w_{ijp}^t = 0, \quad \forall i, j, t, p \in P(i) \quad (2-94)$$

Se se estiverem a considerar as restrições de capacidade máxima então é necessário associar variáveis duais λ_i^t às restrições (2-82) e o dual condensado vem alterado para a seguinte forma:

DC-PLDM2C:

$$Max \sum_{i=1}^T \sum_{j=1}^n v_j^i - \sum_{i=1}^m \rho_i - \sum_{i=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=1}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \sum_{i_i \in p} \lambda_{i_i}^{\tau} \right\} \leq FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau}, \quad \forall i, t, \xi = t, \dots, T \quad (2-95)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \sum_{i_i \in p} \lambda_{i_i}^{\tau} \right\} \leq FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau}, \quad \forall i, t, \xi = t, \dots, T \quad (2-96)$$

$$u_i^t, \rho_i, \pi_i^t, \lambda_i^t \geq 0, \quad \forall i, j, t \quad (2-40)$$

É necessário considerar condições de complementaridade adicionais:

$$\left(Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) - \sum_{j=1}^n \sum_{p \in P(i)} d_j^{\tau} x_{pj}^t \right) \lambda_i^t = 0, \quad \forall i, t \quad (2-97)$$

Se se pretender introduzir restrições de capacidade mínima, as respectivas variáveis duais apresentam um comportamento simétrico em relação às variáveis duais λ_i^t (semelhante ao ilustrado na secção 2.2.3.1).

Quando se considera a existência de parâmetros $\theta_i \neq 1$, para algum serviço i , então o dual condensado pode ser formulado como DC-PLDM2'.

DC-PLDM2':

$$Max \sum_{i=1}^T \sum_{j=1}^n v_j^i - \sum_{i=1}^m \rho_i - \sum_{i=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=1}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \prod_{\substack{i' < i \\ p}} \theta_{i'} \sum_{i_i \in p} \lambda_{i_i}^{\tau} \right\} \leq FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau}, \quad \forall i, t, \xi = t, \dots, T \quad (2-98)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \prod_{\substack{i' < i \\ p}} \theta_{i'} \sum_{i_i \in p} \lambda_{i_i}^{\tau} \right\} \leq FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau},$$

$$\forall i, t, \xi = t, \dots, T \quad (2-99)$$

$$u_i^t, \rho_i, \pi_i^t, \lambda_i^t \geq 0, \quad \forall i, j, t \quad (2-40)$$

Se se considerar a existência, em simultâneo, de restrições de capacidade máxima e mínima, o dual condensado é formulado como DC-PLDM2”.

DC-PLDM2”:

$$Max \sum_{t=1}^T \sum_{j=1}^n v_j^t - \sum_{i=1}^m \rho_i - \sum_{t=1}^T \sum_{i=1}^m \pi_i^t \quad (2-9)$$

sujeito a:

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \prod_{i' < i} \theta_{i'} \left(\sum_{i_i \in p} \lambda_{i_i}^{\tau} - \sum_{i_i \in p} \beta_{i_i}^{\tau} \right) \right\} \leq \\ FA_{it}^{\xi} - \sum_{\tau=\xi+1}^T u_i^{\tau} + \rho_i + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau} \\ \forall i, t, \xi = t, \dots, T \quad (2-100)$$

$$\sum_{j=1}^n \sum_{\tau=t}^{\xi} \sum_{p \in P(i)} \eta_{ijp}^{\tau} \max \left\{ 0, v_j^{\tau} - c_{pj}^{\tau} - d_j^{\tau} \prod_{i' < i} \theta_{i'} \left(\sum_{i_i \in p} \lambda_{i_i}^{\tau} - \sum_{i_i \in p} \beta_{i_i}^{\tau} \right) \right\} \leq \\ FR_{it}^{\xi} + u_i^t + \sum_{\tau=t}^{\xi} \pi_i^{\tau} - Q_i \sum_{\tau=t}^{\xi} \lambda_i^{\tau} + Q'_i \sum_{\tau=t}^{\xi} \beta_i^{\tau} \\ \forall i, t, \xi = t, \dots, T \quad (2-101)$$

$$u_i^t, \rho_i, \pi_i^t, \lambda_i^t, \beta_i^t \geq 0, \quad \forall i, j, t \quad (2-32)$$

As condições de complementaridade que têm de ser satisfeitas pelo par de soluções primal e dual óptimas deverão ser adaptadas por forma a que reflectam estas alterações. Como estas modificações derivam directamente das alterações introduzidas no modelo, que podem ser rapidamente identificadas por comparação com o caso anterior, as condições de complementaridade são omitidas.

2.4. Modelos Multiobjectivo

Todos os modelos descritos até ao momento consideram a existência de apenas uma função objectivo. No entanto, a complexidade da generalidade dos problemas de localização em geral faz com que, na maioria das situações, se possa considerar útil a consideração simultânea de mais do que um objectivo.

Não se encontram muitas referências a trabalhos que tratem explicitamente de problemas de localização com mais do que um objectivo. S. M. Lee, Green, & Kim, 1981, propõem um modelo que descreve a situação de uma empresa que pretende avaliar seis potenciais locais para instalar um equipamento produtivo, servindo quatro centros de distribuição. Os autores referem um conjunto de objectivos interessantes para problemas de localização multi-objectivo em geral, como sejam minimização de custos de produção, maximização da aceitação

pública, minimização de impactos ambientais. Pelegrin & Fernandez, 1988, estudam a determinação de pontos eficientes num problema de localização multi-objectivo em que se pretende localizar apenas um serviço de forma a que esteja o mais próximo possível dos clientes. Batthacharya, Rao, & Tiwari, 1993, estudam um problema de localização no plano de mais do que um serviço, considerando dois critérios, e formulado como um problema de programação por metas em contexto difuso. Pastor, 1994, apresenta o problema de planeamento de uma rede de lojas de retalho, em que se tentam harmonizar dois objectivos opostos: maximizar a acessibilidade total e minimizar os desvios dos investimentos em relação ao planeado. H. W. Hamacher & Nickel, 1996, estudam problemas de localização multi-objectivo no plano, discutindo as inter-relações existentes entre diferentes abordagens para lidar com o problema, dando particular importância à abordagem que se baseia no cálculo de mínimos lexicográficos. Em H. Hamacher, Labbé, & Nickel, 1999, este trabalho é continuado com o estudo de problemas de localização em redes. Mandell & Becker, 1996, aplicam um modelo com dois objectivos ao problema de localizar equipamentos de apoio à vida (desfibrilhadores). Ogryczak, 1997, 1998, 1999, interpreta os problemas de localização como sendo modelos com vários objectivos, em que cada cliente tem uma função objectivo individual definida, que mede a qualidade da solução de localização com respeito à sua satisfação individual (geralmente cada cliente pretende minimizar a sua própria distância ao serviço aberto mais próximo). No primeiro destes artigos, o autor estuda a abordagem *minmax* e melhora-a através da introdução dos conceitos de mínimos lexicográficos. Em Ogryczak, 1999, adapta um método interactivo baseado em pontos de referência. Brimberg & ReVelle, 1998, apresentam um modelo de localização com dois objectivos: a minimização dos custos totais de produção e distribuição e a maximização da procura total satisfeita (não se considera a obrigação da satisfação da procura de todos os clientes). O cálculo de soluções eficientes é feito utilizando a ponderação das funções objectivo. Shimizu, 1999, formula um problema multi-objectivo de localização de serviços para depósito de resíduos perigosos, e desenvolve um algoritmo híbrido que integra algoritmos genéticos e programação matemática. Kolli & Evans, 1999, estuda o problema de localizar novos serviços franchisados numa rede de serviços franchisados já existentes, considerando múltiplos objectivos, que tentam ilustrar o conflito existente entre os franchisados e o franchisador. Tzeng & Chen, 1999, estudam o problema de localização de estações de bombeiros num aeroporto, considerando vários objectivos. Rodríguez-Chía & Puerto, 2002, trabalham com problemas de localização de um único serviço semi-obnócio no plano, maximizando a distância ao habitante mais próximo e minimizando a soma das distâncias a todos os clientes. Fernández & Puerto, 2003, estudam o PLS considerando vários cenários possíveis. Os cenários correspondem a diferentes vectores de custos fixos e de afectação. O problema é decomposto em dois subproblemas, sendo que um diz respeito à localização de serviços e o outro à afectação de clientes a serviços. Para encontrarem soluções não dominadas resolvem problemas de caminho mais curto numa rede constituída por vários estados possíveis, sendo calculado todo o conjunto de

soluções não dominadas (ou uma aproximação deste). Anders J.V. Skriver & Andersen, 2003, consideram o problema de localizar apenas um serviço, em que um objectivo traduz o carácter obnoxious e outro o carácter desejado do serviço. Estudam o problema de localização na rede e no plano. O método utilizado é aplicado à localização de um aeroporto. Emanuel Melachrinoudis & Xanthopoulos, 2003, estudam o problema de localização no plano de um único serviço semi-obnoxious, tendo como objectivos simultâneos a minimização dos efeitos desagradáveis introduzidos pelo serviço e a maximização da distância euclideana mínima com respeito a todos os clientes. Anders J. V. Skriver, Andersen, & Holmberg, 2004, apresentam uma generalização do problema da mediana bicritério. Desenvolvem um método de duas fases que calcula todas as soluções eficientes para o problema (na primeira fase calculam-se as soluções eficientes suportadas e na segunda fase as soluções eficientes não suportadas). Rakas, Teodorovi, & Kim, 2004, apresentam um modelo com dois objectivos aplicado à localização de serviços indesejáveis. No entanto, o método de resolução aparenta não ser capaz de calcular soluções não-dominadas não-suportadas. Chamayou, Moisan, & Scala, 2005, apresentam uma aplicação real, na área das telecomunicações, de um problema de localização com dois objectivos, utilizando a minimização da distância de Tchebycheff, no cálculo de soluções eficientes. Alçada-Almeida, Coutinho-Rodrigues, & Current, 2009, utilizam um modelo multiobjectivo aplicado à localização de incineradoras. Doerner, Gutjahr, & Nolz, 2009, apresentam um modelo multiobjectivo não linear para localização de serviços em áreas costeiras com risco de tsunamis.

Todos os modelos apresentados no capítulo 2 podem ser adaptados para que considerem explicitamente a existência de mais do que uma função objectivo. Tendo sido apresentados como problemas de programação linear inteira mista, condiciona-se todas as funções objectivo a serem lineares. Na literatura relacionada com problemas de localização existem vários exemplos de objectivos que podem ser representados deste modo (Ross & Soland, 1980; Hultz, Klingman, Ross, & Soland, 1981; C. Revelle & Laporte, 1996). A consideração simultânea de mais do que uma função objectivo permite ter em conta, num mesmo modelo, objectivos económicos, sociais, ambientais, etc. Nos modelos descritos na literatura, para além dos objectivos tradicionais que se prendem com a minimização de custos, maximização de lucros (conceitos não equivalentes, nomeadamente se se considerar possível a não satisfação da totalidade da procura dos clientes - Brimberg & ReVelle, 1998) e/ou minimização/maximização de distâncias entre clientes e serviços, podem encontrar-se objectivos que traduzem preocupações no âmbito da minimização do impacto de trânsito, barulho e poluição atmosférica (Chang & Wang, 1997); maximização da equidade entre indivíduos e minimização de riscos de localização e de transporte (Berman & Kaplan, 1990; Current & Ratick, 1995; Erkut & Neuman, 1992; Fishburn & Sarin, 1991; Gopalan, Kolluri, Batta, & Karwan, 1990; Marsh & Schilling, 1994; Ogryczak, 2000), equilíbrio entre serviços abertos relativamente à quantidade total de procura satisfeita por cada serviço (Chamayou *et al.*, 2005). As situações em que os autores expressam maior necessidade de considerarem objectivos que vão para além de questões apenas

relacionadas com a minimização de custos são aquelas em que está em causa a localização de um ou mais serviços obnóxios ou semi-obnóxios, pela dualidade existente entre a necessidade da sua localização, da qual vão beneficiar um vasto conjunto de indivíduos, e as consequências menos agradáveis desta localização, que afectam de forma desigual esses indivíduos (Church & Garfinkel, 1978; Drezner & Wesolowsky, 1989; Horst W. Hamacher, Labbé, Nickel, & Skriver, 2002; E. Melachrinoudis, 1999; Emanuel Melachrinoudis & Xanthopoulos, 2003; Ohsawa & Tamura, 2003; Romero-Morales, carrizosa, & Conde, 1997).

Os modelos apresentados são versáteis em relação às funções objectivo que podem ser consideradas, especialmente devido à maneira como foram definidas as variáveis de decisão. Desta forma, aos coeficientes das variáveis de decisão podem ser dadas novas interpretações, construindo-se assim diferentes objectivos.

(Página deixada propositadamente em branco)

Capítulo 3

HEURÍSTICAS PRIMAIIS-DUAIS

Para cada um dos modelos apresentados no capítulo anterior, desenvolvem-se heurísticas primais-duais capazes de encontrar soluções admissíveis para os problemas. Estas heurísticas baseiam-se na construção de soluções duais admissíveis que servem de base ao cálculo de soluções primais admissíveis. Todos os procedimentos descritos foram programados, permitindo a execução de testes computacionais. Através dos resultados obtidos é possível observar o comportamento das heurísticas que é, de uma forma geral, bom, quer em relação à qualidade da solução primal construída, quer em termos dos tempos computacionais gastos. As excepções a esta regra são identificadas e justificadas.

The term "heuristic" derives from the Greek "heuriskein" meaning to find or discover. If a heuristic is a finding method, we are entitled to ask what is it finding, if not a global optimum. What in Operational Research is now almost universally called a heuristic would be better described as a "seeking method", as it cannot guarantee to find anything. If we wanted to stay with the Greek, we could have used a term like "zetetic", meaning seeking. However it is about 30 years too late for inventing a more consistent vocabulary.

Davis, 1991

3.1. Motivação

Todos os problemas formulados no capítulo 2 podem ser reduzidos ao PLS, pelo que se pode concluir serem todos *NP*-difíceis (Gerard Cornuejols, Nemhauser, & Wolsey, 1990). Sendo assim, não poderão ser resolvidos de forma exacta, utilizando algoritmos de ordem polinomial (Avella *et al.*, 1998).

Quando em presença de modelos com as características dos apresentados, deve sempre pensar-se em alternativas à resolução exacta dos problemas. A utilização de *general solvers* pode não ser exequível, pelo menos se se pensar em problemas de média e grande dimensão que vão ter dificuldades relacionadas com o elevado tempo de execução e com o enorme esforço computacional necessário, não se podendo garantir o cálculo de uma solução óptima (quantas vezes nem sequer admissível) no fim de um limite razoável no tempo de execução do método exacto. As técnicas de enumeração, muitas vezes aplicadas a problemas de natureza combinatória, também são de excluir de

imediatamente porque, na maior parte destes problemas, é inviável pensar em enumerar todas as alternativas possíveis.

Ao lidar com problemas complexos, a dificuldade não está apenas em encontrar a solução ótima. A dificuldade primeira é encontrar soluções admissíveis de boa qualidade. Assim, após o desenvolvimento de modelos complexos, há que desenvolver também métodos de resolução que os permitam tornar capazes de ter alguma aplicabilidade. As técnicas heurísticas, sendo algoritmos menos exigentes, quer em termos de tempos de execução, quer em termos de necessidades a nível computacional, são uma alternativa aliciante. Segundo C. R. Reeves, 1993a:

A heuristic is a technique which seeks good (i.e. near optimal) solutions at reasonable computational cost without being able to guarantee either feasibility or optimality, or even, in many cases, to state how close to optimality a particular feasible solution is.

Vários são os exemplos que podem ser dados de algoritmos exactos e heurísticos que têm sido desenvolvidos ao longo de várias décadas, dedicados à resolução de problemas de localização. L-Lian Gao & Robinson, 1994, atribuem a Balinski o primeiro método de optimização dedicado ao PLS, descrito num artigo em 1964.⁴ A maior parte dos métodos encontrados na bibliografia baseiam-se no uso de: relaxações lagrangeanas com possível optimização pelo método do subgradiente (por exemplo Bitran, Chandru, Sempolinski & Shapiro, 1981; Barceló & Casanovas, 1984; J.E. Beasley, 1993; Christofides & Beasley, 1983; Darby-Dowman & Lewis, 1988; Roberto Diéguez Galvão & Santibañez-Gonzalez, 1992; Pirkul & Schilling, 1991; B. Baker, 1986; Klinecicz & Luss, 1986; J. E. Beasley, 1988; Sridharan, 1991, 1993; Roberto Diéguez Galvão, 1993; Andreas Klose & Stähly, 1996; Holmberg & Ling, 1997; Marín & Pelegrín, 1998; Holmberg, Ronnqvist & Yuan, 1999; Mazzola & Neebe, 1999); métodos de pesquisa em árvore (por exemplo Efrøysom & Ray, 1966; Pinkus, Gross & Soland, 1973; Nauss, 1978; Neebe & Khumawala, 1981; Crainic, Delorme & Dejax, 1993; Ro & Tcha, 1984; Gendron & Crainic, 1997; Goldengorin, Tijssen, Ghosh & Sierksma, 2003; Ceselli & Righini, 2005); técnicas de decomposição (Swain, 1974; Magnanti & Wong, 1990; Chhajed & Lowe, 1998; Francisco Saldanha-da-Gama & Silva, 2005); métodos de optimização através da formulação do problema dual (Erlenkotter, 1978; Guignard & Spielberg, 1979; Kolen, 1983; Klinecicz & Luss, 1987; Tcha, Ro & Yoo, 1988; Körkel, 1989; Li-Lian Gao & Robinson, 1992).

Não é de todo possível apresentar aqui todas as referências conhecidas a trabalhos que aplicam ou apresentam métodos de resolução de problemas de localização. Por este motivo, vão apenas referir-se alguns trabalhos que consideram problemas de localização com capacidades. Na realidade, o aumento da complexidade do problema estudado faz, também, aumentar o número de abordagens diferentes para lidar com o problema. Por isso, o número de métodos que lidam com problemas de localização com capacidades é mais diversificado do que os que lidam, por exemplo, com o PLS.

Para problemas de localização com capacidades, G. Cornuejols, Sridharan &

⁴ "On Finding Integer Solutions to Linear Programs", Mathematica, Princeton, NJ.

Thizy, 1991, apresentam um estudo sistemático acerca da qualidade relativa das heurísticas e relaxações mais utilizadas. Guignard & Spielberg, 1979, descrevem um algoritmo baseado no problema dual da relaxação linear de um problema de localização com capacidades. O método descrito permite calcular limites inferiores de boa qualidade para o valor óptimo da função objectivo e a construção de soluções admissíveis. B. M. Baker, 1982, apresenta restrições válidas para o problema de localização com capacidades que melhoram a relaxação linear do problema. Christofides & Beasley, 1983, utilizam relaxação lagrangeana e optimização pelo subgradiente para encontrarem limites inferiores, que são depois utilizados num algoritmo de *branch-and-bound* que permite o cálculo da solução óptima para problemas com até 50 possíveis localizações para serviços e 150 clientes. Jacobsen, 1983, adapta algumas heurísticas conhecidas para o PLS para que resolvam problemas capacitados. Van Roy, 1986, descreve o método de *cross decomposition* na resolução de um problema de localização com capacidades. O autor apresenta testes computacionais para problemas com até 200 clientes e 100 possíveis localizações para os serviços. B. Baker, 1986, descreve um algoritmo de *branch-and-bound* que faz uso do dual da relaxação linear de uma formulação apertada, permitindo o uso de algoritmos eficientes para problemas de transporte. J. E. Beasley, 1988, descreve um algoritmo de procura em árvore, baseado em relaxação lagrangeana, que consegue resolver problemas com 1000 clientes e 500 possíveis localizações para serviços. Barceló & Casanovas, 1984; Klinecicz & Luss, 1986; Pirkul, 1987, utilizam métodos baseados na relaxação lagrangeana para resolverem o problema em que cada cliente é afecto a exactamente um serviço. Holmberg *et al.*, 1999, estudam o mesmo problema e desenvolvem um método de *branch-and-bound* e Cortinhal & Captivo, 2003a, 2003b, utilizam relaxação lagrangeana, pesquisa tabu e algoritmos genéticos. Barcelo, Fernandez & Jörnsten, 1991, apresentam resultados computacionais obtidos com uma relaxação lagrangeana para o problema de localização com capacidades, em que se considera a introdução de uma variável auxiliar. C. Lee, 1991, utiliza decomposição de Benders para lidar com uma generalização do problema de localização com capacidades em que cada serviço pode fornecer diferentes produtos, incorrendo-se num custo fixo por cada tipo de produto que um serviço pode fornecer. O mesmo autor desenvolve também um algoritmo de *cross decomposition* para problemas de localização com capacidades (C. Y. Lee, 1993). Mazzola & Neebe, 1999, estudam o mesmo problema e utilizam heurísticas lagrangeanas e um algoritmo de *branch-and-bound*. Caprara & González, 1994, apresentam um algoritmo de *branch-and-cut* para resolverem um problema que é uma generalização do PLS. Agar & Salhi, 1998, descrevem várias heurísticas lagrangeanas capazes de lidarem com problemas de grandes dimensões (até 100 possíveis localizações para os serviços e 1000 clientes). Os autores consideram 3 problemas diferentes: com afectação parcial, com afectação total e o problema que considera a escolha da capacidade do serviço a localizar. Aardal, 1998a, estuda a aplicação de planos de corte ao problema de localização com capacidades e estuda também a introdução de restrições redundantes mas computacionalmente eficientes para problemas com um ou

dois níveis de serviços (Aardal, 1998b). Baldaci, Hadjiconstantinou, Maniezzo & Mingozzi, 2002, apresentam um novo método para resolver o problema da p -mediana com capacidades, recorrendo a heurísticas que trabalham com as relações existentes entre soluções duais e primais. Rönnqvist, Tragantalerngsak & Holt, 1999, descrevem uma nova abordagem ao problema de localização com capacidades e afectação total, que se baseia na resolução sistemática de problemas de *matching*. Melkote & Daskin, 2001, trabalham com a versão com capacidades do problema de localização/*network design*, apresentando várias classes de desigualdades válidas que fortalecem as relaxações lineares dos problemas. Saldanha-da-Gama, 2002; Saldanha-da-Gama e Captivo, 2002, descrevem um procedimento de *branch-and-bound* para o PLDC. Correia & Captivo, 2003, utilizam relaxação lagrangeana na resolução do problema de localização com capacidades em que a capacidade de cada serviço a localizar deve ser escolhida de um conjunto finito de capacidades possíveis. L. A. N. Lorena & Senne, 2004, estudam o problema da p -mediana com capacidades, trabalhando com as semelhanças existentes entre a formulação do problema como *set partitioning* e a relaxação lagrangeana obtida quando se relaxam as restrições de satisfação de procura. Desenvolvem um algoritmo de geração de colunas. Zhang, Chen & Ye, 2005, apresentam um algoritmo de pesquisa local que realiza trocas de vários serviços fechados e abertos simultaneamente.

Existem ainda os trabalhos sobre metaheurísticas aplicadas a problemas de localização. A elas se voltará no capítulo 4.

As heurísticas primais duais são, desde há muito, utilizadas na resolução de problemas de localização, apresentando bons compromissos entre a exigência de recursos e a qualidade das soluções obtidas. Em 1999, Mazzola e Neebe consideravam o algoritmo *Dualoc* (Erlenkotter, 1978) como continuando a ser o algoritmo mais eficiente na resolução do PLS. Vários autores se basearam neste algoritmo para desenvolverem algoritmos primais-duais eficientes na resolução de problemas de complexidade crescente. A utilização do problema dual como ponto de partida para a resolução de problemas pode ser observada nos trabalhos de Espejo *et al.*, 2003; Li-Lian Gao & Robinson, 1992; Hanjoul & Peeters, 1985; Herrmann, Ioannou, Minis & Proth, 1996; Klinecwickz & Luss, 1987; Tcha *et al.*, 1988; Van Roy & Erlenkotter, 1982.

Neste capítulo apresentam-se algumas heurísticas primais-duais capazes de resolver os problemas formulados no capítulo 2. Para os problemas com capacidades, aplica-se também relaxação lagrangeana com método do subgradiente. Em relação ao problema mais simples, desenvolve-se um método de pesquisa em árvore (*branch and bound*) baseado na heurística primal-dual.

3.2. Problemas de Localização com um Nível

3.2.1. Problemas sem Restrições de Capacidades

A heurística primal-dual que foi desenvolvida para o problema PLDAFR constrói soluções primais e duais admissíveis, tentando forçar a satisfação de todas as condições de complementaridade (2-21) a (2-26). A solução primal é

sempre construída tendo como base a solução dual. Se a heurística conseguir encontrar um par de soluções primal e dual admissíveis e complementares, então encontrou a solução óptima. Caso tal par de soluções não seja encontrado, a melhor solução dual admissível calculada fornece um limite inferior ao valor óptimo da função objectivo primal, permitindo sempre aferir a qualidade da solução primal construída.

Esquematiza-se agora o algoritmo de funcionamento da heurística, explicitando-se, seguidamente e de forma pormenorizada, cada um dos passos.

Algoritmo 3-1

HEURÍSTICA PRIMAL-DUAL PARA O PLDAFR

1. Inicialização das variáveis duais.
 2. Procedimento Dual Ascendente para as variáveis duais v_j' .
 3. Procedimento Primal.
 4. Procedimento de Ajuste Dual para as variáveis duais ρ_i . Se a solução dual for alterada, então ir para 2; se não ir para 5.
 5. Enquanto a função objectivo do problema dual melhorar, executar o Procedimento de Ajuste Dual para as variáveis duais v_j' .
 6. Procedimento de Ajuste Dual para as variáveis duais ρ_i . Se a solução dual sofrer alterações, então ir para 2; se não ir para 7.
 7. Procedimento Dual Ascendente para as variáveis duais u_i' . Se a solução dual sofrer alterações, ir para 2; se não ir para 8.
 8. Procedimento Dual Descendente para as variáveis duais u_i' . Se a solução dual sofrer alterações, ir para 2; se não ir para 9.
 9. Procedimento de Ajuste Dual para as variáveis duais π_i' . Se a solução dual sofrer alterações; ir para 2; se não parar.
-

A ordem pela qual se executam cada um dos procedimentos referidos pode ser alterada, dando lugar a um elevado número de heurísticas semelhantes. A sequência escolhida foi a que apresentou os melhores resultados nos testes computacionais preliminares realizados, e também aquela que, pela descrição que vai ser feita de cada um dos procedimentos, parece conduzir a uma melhor solução dual. Nestas heurísticas primais-duais não interessa apenas encontrar o melhor valor para a função objectivo dual. Interessa, acima de tudo, que a solução dual tenha uma estrutura facilitadora da construção de uma solução primal de qualidade, com vista à satisfação das condições de complementaridade.

3.2.1.1. Inicialização das Variáveis Duais

As variáveis duais são inicializadas utilizando o Algoritmo 3-2.

Algoritmo 3-2

INICIALIZAÇÃO DAS VARIÁVEIS DUAIS

c_{ij}^t - custo de afectar o cliente j ao serviço i no período t ; FA_{it}^ξ, FR_{it}^ξ - custos fixos de (re)abertura de i ; I - conjunto de possíveis localizações para os serviços; J - conjunto de clientes; T - número de períodos temporais.

1. $v_j^t = \min_i \{ c_{ij}^t \}, \forall j \in J, t = 1, \dots, T; \quad \pi_i^t = 0, \forall i \in I, t = 1, \dots, T;$
 2. $u_i^t = \begin{cases} \max \left\{ 0, -\min_{\xi \geq t} FR_{it}^\xi \right\}, & \text{se } \exists FR_{it}^\xi < 0 \\ 0, & \text{em caso contrário} \end{cases}, \forall i \in I, t = 1, \dots, T;$
 3. $\rho_i^t = \max \left\{ 0, -\min_{\xi \geq t} \left(FA_{it}^\xi - \sum_{\tau=\xi+1}^T u_i^\tau \right) \right\}, \forall i \in I, t = 1, \dots, T$
-
-

3.2.1.2. Procedimento Dual Ascendente para as Variáveis v_j^t

Este procedimento tenta aumentar as variáveis duais $v_j^t, j \in J^+, J^+ \subset J$. Quando este procedimento é executado no passo 2 da heurística (Algoritmo 3-1), então o conjunto J^+ é igual ao conjunto J . Sempre que este procedimento é executado a partir de outros procedimentos, como se poderá verificar nas secções seguintes, então o conjunto J^+ será sempre definido antes do Procedimento de Ajuste Dual ser chamado.

O procedimento que se desenvolveu é uma adaptação directa daquele que é descrito em Van Roy & Erlenkotter, 1982. A única diferença prende-se com a actualização dos valores das *slacks* SA_{it}^ξ e SR_{it}^ξ : cada vez que o valor da variável v_j^t é aumentado, as *slacks* SA_{it}^ξ e $SR_{it}^\xi, \tau \leq t \leq \xi$, têm de ser actualizadas.

O valor das variáveis duais v_j^t é, sempre que possível, aumentado para o menor custo c_{ij}^t maior que v_j^t . Sempre que tal não seja possível, devido ao valor de uma *slack* S_{it}^ξ , então aumenta-se o mais possível por forma a que a solução dual obtida continue admissível. O procedimento termina quando deixa de ser possível aumentar o valor de qualquer variável v_j^t .

Na descrição do algoritmo consideram-se todos os custos c_{ij}^t reindexados à semelhança do que está descrito em Van Roy & Erlenkotter, 1982: para cada cliente j e para cada período temporal t ordenam-se todos os custos c_{ij}^t por ordem crescente, sendo que o primeiro custo é designado por c_j^{t1} , o segundo por c_j^{t2} , o k -ésimo por c_j^{tk} .

Algoritmo 3-3

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS DUAIS v_j^t

Para cada cliente j e período temporal t ordenam-se todos os custos c_{ij}^t por ordem crescente, sendo que o primeiro custo é designado por c_j^{t1} , o segundo por c_j^{t2} , o k -ésimo por c_j^{tk} . As variáveis v_j^t estão organizadas (sem nenhuma ordem especial) num conjunto formado por pares (j, t) , tal que $(j, t)_k$ representa o k -ésimo elemento desse conjunto.

1. Considere-se uma qualquer solução admissível $\{v_j^t\}$ tal que $v_j^t \geq c_j^{t1}, \forall (j, t), S_{it}^\xi \geq 0, \forall i, t, \xi \geq t$.
Para cada par (j, t) , defina-se $k(j, t) = \min\{k: v_j^t \leq c_j^{tk}\}$. Se $v_j^t = c_j^{tk(j,t)}$, então $k(j, t) \leftarrow k(j, t) + 1$.
 2. $(j, t) \leftarrow (j, t)_i; q \leftarrow i - 1; \delta \leftarrow 0$.
 3. Se $(j, t) \notin J^+$, então ir para 7. Se não, ir para 4.
 4. $\Delta_j^t \leftarrow \min\{S_{it}^\xi: v_j^t \geq c_{ij}^t \wedge \tau \leq t \leq \xi\}$.
 5. Se $\Delta_j^t > c_j^{tk(j,t)} - v_j^t$ então $\Delta_j^t \leftarrow c_j^{tk(j,t)} - v_j^t; \delta \leftarrow 1; k(j, t) \leftarrow k(j, t) + 1$.
 6. $v_j^t \leftarrow v_j^t + \Delta_j^t$. Para cada serviço i , se $v_j^t - c_{ij}^t \geq 0$, então $S_{it}^\xi \leftarrow S_{it}^\xi - \Delta_j^t, \tau \leq t \leq \xi$.
 7. Se $q \neq |J^+| \times T$, então $(j, t) \leftarrow (j, t)_{q+1}; q \leftarrow q + 1$. Ir para 3.
 8. Se $\delta = 1$, então ir para 2; se não parar.
-

3.2.1.3. Construção de Soluções Primais Admissíveis

Definam-se os seguintes conjuntos:

$$I^* = \{(i, \tau, \xi) : S_{it}^\xi = 0\}; I_i^* = \{i : (i, \tau, \xi) \in I^* \wedge \tau \leq t \leq \xi\};$$

$$I_i^+ = \{i : i \text{ está aberto durante } t\}; I_A^+ = \{(i, \tau, \xi) : a_{it}^\xi = 1\}; I_R^+ = \{(i, \tau, \xi) : r_{it}^\xi = 1\}.$$

O conjunto I^* corresponde a ternos (i, τ, ξ) tais que SA_{it}^ξ e/ou SR_{it}^ξ são iguais a zero. É a partir deste conjunto que se irão definir quais as variáveis de localização a colocar nos conjuntos I_A^+ e I_R^+ . Os conjuntos I_A^+ e I_R^+ correspondem, no fim do procedimento primal, à solução primal construída. Do conjunto I_i^* fazem parte todos os serviços i que poderão vir a estar abertos durante o período temporal t . Um serviço i irá pertencer ao conjunto I_i^+ se efectivamente estiver aberto durante o período t . Os conjuntos I_i^* e I_i^+ não são, forçosamente, iguais pois o procedimento tenta sempre abrir o menor número possível de serviços, garantindo a afectação de todos os clientes a um serviço aberto.

Os conjuntos I_A^+ e I_R^+ são construídos durante o procedimento primal, e determinam quais os serviços que são abertos e reabertos, quando e por quanto tempo é que estão em funcionamento. O procedimento primal começa por incluir no conjunto I_i^+ aqueles serviços que pertencem a I_i^* e que são considerados essenciais no período t (Definição 3.1). O procedimento tenta então afectar cada cliente j , em cada período temporal t , a um serviço essencial.

Definição 3.1: Um serviço $i, i \in I_i^+$, é considerado *essencial* no período temporal t se existe pelo menos um cliente j que tem de ser afecto ao serviço i durante o período t . Isto acontece se e só se $\exists j \in J : v_j' \geq c_{ij}' \wedge v_j' < c_{ij}', \forall i' \in I, i' \neq i$.

Se num dado período t houver, pelo menos, um cliente j tal que j não possa ser afecto a serviços essenciais, então o procedimento inclui em I_t^+ o serviço $i \in I_i^*$ a que corresponde o menor custo de afectação c_{ij}' . Para tentar evitar que se abram mais serviços do que os necessários e porque, se existirem serviços não essenciais no conjunto I_t^+ , este conjunto depende sempre da ordem pela qual foram analisados os clientes j , voltam a repetir-se os mesmos passos de construção do conjunto I_t^+ , mas agora considerando I_t^* igual a I_t^+ : para os serviços que foram considerados abertos durante o período t detectam-se aqueles que são essenciais. Estes são considerados abertos e outros só o serão se estritamente necessário.

No procedimento primal apresentado, os passos 4 e 5 requerem especial atenção. De facto, a construção dos conjuntos I_A^+ e I_R^+ é bastante mais complexa do que a construção do conjunto I^+ no algoritmo *Dynaloc* (Van Roy & Erlenkotter, 1982, tendo em conta os comentários de Francisco Saldanha-da-Gama & Captivo, 1996), porque poderão ter de se considerar períodos de abertura e reabertura para um mesmo serviço que garantam a admissibilidade da solução primal. Se, para um dado serviço $i \in I_i^+$, se tem $S_{ir}^{\xi} = 0, \tau \leq t \leq \xi$, para mais do que um par (τ, ξ) , a escolha de qual variável incluir no conjunto I_A^+ ou I_R^+ não é trivial e pode tornar-se um problema de natureza combinatória por si só.

Algoritmo 3-4

PROCEDIMENTO PRIMAL

$I^* = \{(i, \tau, \xi) : S_{ir}^{\xi} = 0\}$; $I_i^* = \{i : (i, \tau, \xi) \in I^* \wedge \tau \leq t \leq \xi\}$; $I_t^+ = \{i : i \text{ está aberto durante } t\}$;
 $I_A^+ = \{(i, \tau, \xi) : a_{ir}^{\xi} = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{ir}^{\xi} = 1\}$; c_{ij}' - custo de afectar o cliente j ao serviço i durante t ;
 T - número de períodos temporais

1. $I_A^+ \leftarrow I_R^+ \leftarrow \emptyset$; $I_t^+ \leftarrow \emptyset, \forall t$. Construir os conjuntos I_i^* e I_t^+ . $Num \leftarrow 0$.
 2. Para t de 1 até T , incluir em I_t^+ todos os serviços $i \in I_i^*$ tais que $\exists j : v_j' \geq c_{ij}'$ e $v_j' < c_{ij}', \forall i' \neq i$.
 3. Para cada cliente j tal que $v_j' < c_{ij}', \forall i \in I_t^+$, incluir no conjunto I_t^+ o serviço i tal que $c_{ij}' = \min_{i' \in I_t^+} c_{i'j}'$. $Num \leftarrow Num + 1$. Se $Num = 1$ então $I_A^+ \leftarrow I_R^+ \leftarrow \emptyset$; $I_t^+ \leftarrow I_t^+$; $I_t^+ \leftarrow \emptyset, \forall t$ e ir para 2. Se não, ir para 4.
 4. Construir os conjuntos I_A^+ e I_R^+ . Actualizar o conjunto I_t^+ . Para t de 1 até T , afectar cada cliente j ao serviço $i \in I_t^+$, tal que $c_{ij}' = \min_{i \in I_t^+} \{c_{ij}'\}$.
 5. Verificar a satisfação das condições de complementaridade (2-21) a (2-26).
-

Defina-se: $t_1 = \min\{\tau : i \in I_{\tau}^+\}$ e $t_2 = \max\{\tau : i \in I_{\tau}^+\}$.

Se o terno (i, t_1, t_2) pertencer ao conjunto I^* , então inclui-se a variável $a_{it_1}^{t_2}$

em I_A^+ . Para as situações em que tal não se verifica, foram desenvolvidos dois procedimentos auxiliares utilizados no passo 4 do Algoritmo 3-4 para a construção dos conjuntos I_A^+ e I_R^+ . Para cada serviço i , ambos os procedimentos incluem em I_A^+ ou I_R^+ variáveis que garantem que i estará aberto pelo menos durante os períodos temporais t , tais que $i \in I_t^+$, e simultaneamente garantem a satisfação das restrições (2-4), (2-5) e (2-6). Estes dois procedimentos são muito semelhantes, residindo a diferença entre ambos no facto de, num deles, se começar a construir os conjuntos I_A^+ e I_R^+ , com respeito a cada serviço i , considerando valores crescentes de t de 1 a T , enquanto que o outro considera valores decrescentes de t desde T até 1. Ambos os procedimentos são executados, e escolhe-se depois a solução que corresponde ao menor custo fixo total, relativo às variáveis de localização.

Os procedimentos auxiliares de construção da solução primal tentam, para cada serviço $i \in I_i^+$, incluir em I_A^+ ou I_R^+ o terno $(i, \tau, \xi) \in I^*$ e $\tau \leq t \leq \xi$, que abranja o maior número possível de períodos temporais. Um dos procedimentos (chame-se-lhe Procedimento Auxiliar 1) começa a construir a solução considerando t igual a t_1 e vai tentar inserir em I_A^+ o terno (i, τ, ξ) , com $\tau = 1$ e $\xi = T$. Se tal não for possível, porque $(i, \tau, \xi) \notin I^*$, então diminui o valor de ξ em uma unidade temporal e até que $(i, \tau, \xi) \in I^*$ ou $\xi < t_1$. Se $(i, \tau, \xi) \in I^*$, então insere-se (i, τ, ξ) em I_A^+ , avança-se τ para $\xi + 1$ (porque não é possível haver sobreposições entre os períodos temporais das variáveis de abertura e reabertura devido às restrições (2-6)) e avança-se t para o próximo período temporal tal que $i \in I_t^+$, repetindo-se de novo o processo (sendo que, agora, todos os ternos $(i, \tau, \xi) \in I^*$ que forem considerados corresponderão a variáveis de reabertura – conjunto I_R^+). Se $\xi < t_1$, então não foi possível encontrar nenhum terno $(i, \tau, \xi) \in I^*$ tal que $\tau \leq t \leq \xi$, pelo que se incrementa o valor de τ em uma unidade temporal, até que seja encontrado $(i, \tau, \xi) \in I^*$ ou então $\tau > t_1$. No primeiro caso, procede-se como descrito anteriormente, inserindo-se mais um elemento em I_A^+ ou I_R^+ . No segundo caso, significa que não irá ser possível construir uma solução admissível recorrendo apenas a elementos do conjunto I^* . Vai ter de se considerar igual a um uma variável de localização, tal que a correspondente *slack* $S_{it}^{\xi} \neq 0$. Isto provoca, de imediato, a violação das condições de complementaridade (2-25) e (2-26). O Procedimento Auxiliar 2 constrói a solução primal de forma semelhante à descrita, mas começando por considerar t igual a t_2 , incrementando primeiro o valor de τ e diminuindo depois o valor de ξ . Em ambos os procedimentos tenta experimentar-se, para cada período t , várias possibilidades de ternos (i, τ, ξ) tal que $\tau \leq t \leq \xi$, tendo em conta os elementos já colocados nos conjuntos I_A^+ e I_R^+ e respeitando as restrições (2-6). A execução destes procedimentos garante, assim, a construção de uma solução primal admissível para o PLDAFR, mas pode, nalguns casos,

originar violações das condições de complementaridade (2-25) e (2-26). Nestes casos, optou-se por minimizar o número de violações, garantindo-se que, para cada serviço i , haverá no máximo uma violação de (2-25) ou (2-26).

Algoritmo 3-5

PASSO 4 DO ALGORITMO 3-4

m - número de possíveis localizações para os serviços; $I_i^+ = \{i : i \text{ está aberto durante } t\}$;
 $I_A^+ = \{(i, \tau, \xi) : a_{i\tau}^\xi = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{i\tau}^\xi = 1\}$.

1. $i \leftarrow 1$.
 2. Se $\exists t : i \in I_t^+$, então ir para 3; se não ir para 8.
 3. $t_1 \leftarrow \min\{\tau : i \in I_\tau^+\}$; $t_2 \leftarrow \max\{\tau : i \in I_\tau^+\}$.
 4. $I_{1A}^+ \leftarrow I_A^+$; $I_{1R}^+ \leftarrow I_R^+$. Executar o Procedimento Auxiliar 1 (Algoritmo 3-6).
 5. $I_{2A}^+ \leftarrow I_A^+$; $I_{2R}^+ \leftarrow I_R^+$. Executar o Procedimento Auxiliar 2 (Algoritmo 3-7).
 6. $sum1 \leftarrow \sum_{(i,\tau,\xi) \in I_{1A}^+} FA_{i\tau}^\xi + \sum_{(i,\tau,\xi) \in I_{1R}^+} FR_{i\tau}^\xi$; $sum2 \leftarrow \sum_{(i,\tau,\xi) \in I_{2A}^+} FA_{i\tau}^\xi + \sum_{(i,\tau,\xi) \in I_{2R}^+} FR_{i\tau}^\xi$;
 7. Se $sum1 < sum2$, então $I_A^+ \leftarrow I_{1A}^+$, $I_R^+ \leftarrow I_{1R}^+$. Se não $I_A^+ \leftarrow I_{2A}^+$, $I_R^+ \leftarrow I_{2R}^+$.
 8. $i \leftarrow i + 1$. Se $i > m$, então parar. Se não, ir para 2.
-
-

Algoritmo 3-6

PROCEDIMENTO AUXILIAR 1 DO PASSO 4 (ALGORITMO 3-4)

$t_1 = \min\{\tau : i \in I_\tau^+\}$, $t_2 = \max\{\tau : i \in I_\tau^+\}$; $I_i^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{i\tau}^\xi = 1\}$;
 $I_R^+ = \{(i, \tau, \xi) : r_{i\tau}^\xi = 1\}$; T - número de períodos temporais.

1. início $\leftarrow 1$; tempo $\leftarrow t_1$; stop \leftarrow false; abertura \leftarrow true.
 2. Se tempo $> t_2$, então parar; se não ir para 3.
 3. Se $i \notin I_{tempo}^+$, então tempo \leftarrow tempo + 1 e ir para 2. Se não, ir para 4.
 4. $\tau \leftarrow$ início; $\xi \leftarrow T$; $t \leftarrow$ tempo;
 5. Se $\tau > t$ ou stop = true, então ir para 6. Se não ir para 9.
 6. Se stop = true então tempo \leftarrow tempo + 1 e ir para 2. Se não, ir para 7.
 7. Se abertura = true então abertura \leftarrow false e $I_A^+ \leftarrow I_A^+ \cup \{(i, inicio, t_2)\}$. Se não,
 $I_R^+ \leftarrow I_R^+ \cup \{(i, inicio, t_2)\}$.
 8. tempo \leftarrow tempo + 1. Ir para 2.
 9. Se $\xi < t$ ou stop = true então $\tau \leftarrow \tau + 1$, $\xi \leftarrow T$ e ir para 5. Se não ir para 10.
 10. Se $\exists (i, \tau, \xi) \in I_i^+$, então ir para 11. Se não, $\xi \leftarrow \xi - 1$ e ir para 9.
 11. Se abertura = true, então abertura \leftarrow false e $I_A^+ \leftarrow I_A^+ \cup \{(i, \tau, \xi)\}$, se não, $I_R^+ \leftarrow I_R^+ \cup \{(i, \tau, \xi)\}$.
 12. tempo \leftarrow ξ ; início \leftarrow tempo; stop = true. Ir para 9.
-
-

Algoritmo 3-7

PROCEDIMENTO AUXILIAR 2 DO PASSO 4 (ALGORITMO 3-4)

$t_1 = \min\{\tau : i \in I_\tau^+\}$, $t_2 = \max\{\tau : i \in I_\tau\}$; $I_i^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{it}^\xi = 1\}$;
 $I_R^+ = \{(i, \tau, \xi) : r_{it}^\xi = 1\}$; T – número de períodos temporais.

1. fim $\leftarrow T$; tempo $\leftarrow t_2$; stop \leftarrow false.
 2. Se tempo $< t_1$, então parar; se não ir para 3.
 3. Se $i \notin I_{tempo}^+$, então tempo \leftarrow tempo -1 e ir para 2. Se não, ir para 4.
 4. $\tau \leftarrow -1$; $\xi \leftarrow$ fim; $t \leftarrow$ tempo;
 5. Se $\xi < t$ ou stop=true, então ir para 6. Se não, ir para 9.
 6. Se stop=true, então tempo \leftarrow tempo -1 e ir para 2. Se não, ir para 7.
 7. $I_A^+ \leftarrow I_A^+ \cup \{(i, t_1, fim)\}$.
 8. tempo $\leftarrow t_1 - 1$. Ir para 2.
 9. Se $\tau > t$ ou stop=true, então $\xi \leftarrow \xi - 1$, $\tau \leftarrow -1$ e ir para 5. Se não, ir para 10.
 10. Se $\exists (i, \tau, \xi) \in I^+$, então ir para 11. Se não, $\tau \leftarrow \tau + 1$ e ir para 9.
 11. Se $\tau \leq t_1$, então $I_A^+ \leftarrow I_A^+ \cup \{(i, \tau, \xi)\}$; se não $I_R^+ \leftarrow I_R^+ \cup \{(i, \tau, \xi)\}$.
 12. tempo $\leftarrow \tau$; fim \leftarrow tempo; stop \leftarrow true. Ir para 9.
-

Deve ainda notar-se que a construção dos conjuntos I_A^+ e I_R^+ não tem em conta outras variáveis duais para além das variáveis v_j^i . Isto faz com que a solução primal construída possa violar algumas condições de complementaridade para além das condições (2-25) ou (2-26), nomeadamente as condições (2-22), (2-23) e (2-24). É por este motivo que aparece, no passo 5 do Algoritmo 3-4, a verificação da satisfação das condições de complementaridade. Neste passo vai verificar-se se existe ou não violação de alguma das condições de complementaridade referidas e, em caso afirmativo, vai tentar alterar-se a solução primal, forçando a satisfação daquelas condições. As condições de complementaridade (2-22), (2-23) e (2-24) implicam que:

- ⊕ Se $\rho_i \neq 0$, então terá de se abrir um serviço i durante um ou mais períodos temporais.
- ⊕ Se $\pi_i' \neq 0$, então o serviço i tem de estar em funcionamento durante o período temporal t .
- ⊕ Se $u_i^t \neq 0$, então, se tiver sido aberto um serviço na localidade i antes de t , tal que i esteja fechado no período t , então i terá de ser reaberto em t .

Proposição 3-1: Se, para um dado i e período temporal t , ρ_i , π_i' , e u_i^t são todas maiores do que zero então, se as condições de complementaridade (2-22) são satisfeitas, as condições (2-23) e (2-24) são também satisfeitas.

Proposição 3-2: Se, para um dado i e um dado período temporal t , ρ_i e π_i' são simultaneamente maiores do que zero e as condições de complementaridade (2-24) são satisfeitas, então a condição de complementaridade (2-23) também é satisfeita.

Estas proposições são motivadas directamente pela definição das condições de complementaridade, e são a base do procedimento auxiliar que é utilizado no passo 5 do Algoritmo 3-4. Este procedimento, descrito formalmente como Algoritmo 3-8, começa por verificar a existência de violações das condições de complementaridade (2-22). Se existir uma violação destas condições, para um

dados serviço i e período temporal t , então tenta reabrir o serviço i no período t , mas respeitando sempre as restrições (2-6) (verificando se existe alguma variável $r_{it}^\xi, \tau > t$, em I_R^+ o que, em caso afirmativo, faz com que só seja possível considerar variáveis de reabertura r_{it}^ζ tal que $\zeta < \tau$). Algo de semelhante se processa quando existem violações das condições (2-24). Neste caso tenta (re)abrir-se o serviço i de modo a que este esteja em funcionamento durante o período t , mas considerando intervalos de tempo que não incluam períodos já abrangidos por elementos de I_R^+ e I_A^+ . Quanto às condições de complementaridade (2-23), se estas não estiverem a ser satisfeitas para algum serviço i , significa que i não está em funcionamento em nenhum período temporal. Neste caso tenta encontrar-se uma qualquer *slack* S_{it}^ξ que seja igual a zero e considerar a variável de abertura correspondente igual a um. Note-se que o procedimento de verificação das condições de complementaridade não garante a eliminação de todas as violações.

Proposição 3-3: A execução do procedimento de construção de soluções primais (Algoritmo 3-4) é sempre capaz de gerar uma solução primal admissível para o PLDAFR.

Prova:

Para cada período temporal t existe sempre uma *slack* $S_{it}^\xi, \tau \leq \xi$, tal que $S_{it}^\xi = 0$ (caso contrário, seria possível aumentar o valor das variáveis duais v_j^i pelo que ainda não estariam reunidas as condições de executar o Algoritmo 3-4). Como se considera que a capacidade de cada serviço é suficiente para satisfazer a totalidade da procura dos clientes, então é suficiente ter pelo menos um serviço aberto em cada período t . A forma de construir a solução, através dos procedimentos 1 e 2, garante que existirá sempre uma variável a_{it}^ξ ou r_{it}^ξ igual a 1 tal que $\tau \leq \xi$, para cada t .

c.q.d.

Algoritmo 3-8

PROCEDIMENTO DE VERIFICAÇÃO DAS CONDIÇÕES DE COMPLEMENTARIDADE (PASSO 5 DO ALGORITMO 3-4)

$I^+ = \{(i, \tau, \xi) : S_{it}^\xi = 0\}; I_i^+ = \{i : i \text{ está aberto durante } t\}; I_A^+ = \{(i, \tau, \xi) : a_{it}^\xi = 1\}; I_R^+ = \{(i, \tau, \xi) : r_{it}^\xi = 1\};$;
 m – número de possíveis localizações para os serviços, T – número de períodos temporais.

1. $i \leftarrow 1$.
 2. Se $i > m$, então parar. Se não, $\rho\text{satisfeito} \leftarrow \text{false}; \pi\text{satisfeito} \leftarrow \text{false}; t \leftarrow 1$ e ir para 3.
 3. Se $t > T$, então ir para 9. Se não, ir para 4.
 4. Se $u_i^j \neq 0$ e $\exists (i, \tau, \xi) \in I_A^+, \xi < t$, então ir para 5; se não, ir para 6.
 5. Se $\exists (i, t, \xi) \in I_R^+$, então $\rho\text{satisfeito} \leftarrow \text{true}, \pi\text{satisfeito} \leftarrow \text{true}$ e ir para 6. Se não, $tmax = \min\{T, \min\{\tau : (i, \tau, \gamma) \in I_R^+ \wedge \tau > t\}\}$. Se $\exists (i, t, \xi) \in I^+, \xi < tmax$, então $I_R^+ \leftarrow I_R^+ \cup \{(i, t, \xi)\}$, $\rho\text{satisfeito} \leftarrow \text{true}, \pi\text{satisfeito} \leftarrow \text{true}$ e ir para 6.
 6. Se $\pi_i^j = 0$ e $\pi\text{satisfeito} = \text{false}$, então ir para 7. Se não, $t \leftarrow t + 1$ e ir para 3.
 7. Se $i \in I_i^+$, então $\rho\text{satisfeito} \leftarrow \text{true}$, $t \leftarrow t + 1$ e ir para 3. Se $\exists (i, \tau, \xi) \in I_A^+$, então $tmin = \max\{\xi : (i, \tau, \xi) \in I_A^+ \cup I_R^+, \xi < t\}$ e $tmax = \min\{\tau : (i, \tau, \xi) \in I_A^+ \cup I_R^+, \tau > t\}$. Caso contrário, $tmin \leftarrow 1$ e $tmax \leftarrow T$.
 8. Se $\exists (i, \tau, \xi) \in I^+ : \tau \geq tmin \wedge \xi \leq tmax$, então $I_A^+ \leftarrow I_A^+ \cup \{(i, \tau, \xi)\}$ ou $I_R^+ \leftarrow I_R^+ \cup \{(i, \tau, \xi)\}$, $\rho\text{satisfeito} \leftarrow \text{true}; t \leftarrow t + 1$ e ir para 3.
 9. Se $\rho\text{satisfeito} = \text{false}$ e $\rho_i \neq 0$, então escolher $(i, \tau, \xi) \in I^+$ e $I_A^+ \leftarrow I_A^+ \cup \{(i, \tau, \xi)\}; i \leftarrow i + 1$ e ir para 2.
-

3.2.1.4. Procedimento de Ajuste Dual para as Variáveis Duais ρ_i

Se for possível diminuir o valor de uma variável ρ_i que seja diferente de zero, mantendo-se a admissibilidade da solução dual, então está-se a melhorar de imediato o valor da função objectivo do problema DC-PLDAFR. O valor de uma variável ρ_i poderá ser diminuído se $SA_{it}^\xi \neq 0, \forall 1 \leq t \leq \xi$. Ao aumentar o valor de uma variável é necessário aumentar todas as *slacks* SA_{it}^ξ . O aumento destas *slacks* poderá permitir aumentar o valor de variáveis v_j^t que estivessem impedidas de crescer devido ao valor zero de alguma *slack* SA_{it}^ξ . No entanto, como as variáveis ρ_i apresentam um coeficiente igual a -1 na função objectivo dual, o seu valor só deve ser incrementado se o aumento do valor das variáveis v_j^t compensar a correspondente perda na função objectivo. Note-se, também, que só valerá a pena tentar aumentar uma *slack* SA_{it}^ξ se a correspondente *slack* SR_{it}^ξ for maior do que zero. Caso contrário, não se conseguiria alterar o valor de S_{it}^ξ , pelo que não seria possível aumentar o valor de variáveis v_j^t . Esta situação poderá verificar-se quando $(i, \tau, \xi) \in I_R^+$ mas $SR_{it}^\xi \neq 0$ e $SA_{it}^\xi = 0$.

Sempre que se aumenta o valor de uma variável ρ_i , constrói-se o conjunto J^+ que corresponde a todas as variáveis v_j^t cujo valor poderá ser aumentado. É então executado o procedimento dual ascendente para as variáveis v_j^t . No caso do aumento da variável ρ_i não ter sido totalmente aproveitado pelo aumento das variáveis v_j^t , então diminui-se novamente ρ_i , garantindo-se, deste modo, que a função objectivo dual nunca será piorada.

Defina-se $I_j^{*+} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^+ \text{ e } v_j^t \geq c_{ij}^t \}$. O procedimento de ajuste dual para variáveis ρ_i é formalmente descrito como Algoritmo 3-9.

Proposição 3-4: O procedimento de ajuste dual para as variáveis duais ρ_i nunca piora o valor da função objectivo do problema DC-PLDAFR.

Prova:

Diminuir o valor de uma variável dual ρ_i , mantendo a solução dual admissível, melhora de imediato o valor da função objectivo do problema dual, visto a variável ρ_i ter um coeficiente igual a -1 na função objectivo.

Aumentar o valor de ρ_i aumenta também o valor de algumas *slacks*. Considere-se que Δv_j^t é a alteração ocorrida na variável v_j^t , após a execução do procedimento dual ascendente nos passos 5 e 6 do algoritmo. Seja $\Delta_1 \rho_i$ o aumento calculado no passo 4, $\Delta_2 \rho_i$ a diminuição calculada no passo 8 e $\Delta \rho_i = \Delta_1 \rho_i - \Delta_2 \rho_i$. Considere-se que SA_{it}^{ξ} representa a *slack* obtida após a execução do procedimento de ajuste dual. Note-se que $\Delta_1 \rho_i \geq \Delta_2 \rho_i$. Para verificar que esta desigualdade se verifica sempre, basta pensar na *slack* SA_{it}^{ξ} que era originalmente igual a zero e foi aumentada em $\Delta_1 \rho_i$. Durante a execução do procedimento dual ascendente nos passos 5 e 6 do algoritmo, o valor da *slack* só poderá diminuir, logo tem-se que $\Delta_2 \rho_i \leq SA_{it}^{\xi} \leq \Delta_1 \rho_i$.

Após a execução do procedimento, duas são as situações possíveis:

Situação 1. Se existir alguma *slack* SA_{ir}^{ξ} que tenha retomado o seu valor original, então isto significa que o aumento da variável ρ_i foi totalmente aproveitado pelo procedimento dual ascendente das variáveis v_j' , e por isso $\sum_j \sum_t \Delta v_j' \geq \Delta_1 \rho_i$: $SA_{ir}^{\xi} = SA_{ir}^{\xi} + \Delta \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j'$, logo se $SA_{ir}^{\xi} = SA_{ir}^{\xi}$ então $\Delta \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j' = 0$.

Situação 2. Se $SA_{ir}^{\xi} > SA_{ir}^{\xi}$ para todas as *slacks*, isto significa que todas serão maiores do que zero e o procedimento tentará, então, diminuir o valor de ρ_i . Considere-se SA_{ir}^{ξ} , com $v_j' \geq c_{ij}^t$, que após a execução dos passos 5 e 6 foi alterada para $SA_{ir}^{\xi} = SA_{ir}^{\xi} + \Delta_1 \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j'$. Como $SA_{ir}^{\xi} > SA_{ir}^{\xi}$, isto significa que $\Delta_1 \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j' > 0 \Leftrightarrow \Delta_1 \rho_i > \sum_j \sum_{t=\tau}^{\xi} \Delta v_j'$. Se o valor de ρ_i se mantivesse poderia haver uma deterioração do valor da função objectivo dual. No entanto, o valor desta variável irá ser diminuído em $\Delta_2 \rho_i$, calculado no passo 8 do algoritmo. Isto significa que pelo menos uma *slack* ficará igual a zero, ou seja, todo o aumento real da variável ρ_i foi compensado pelo aumento das variáveis duais v_j' :

$$SA_{ir}^{\xi} = SA_{ir}^{\xi} + \Delta_1 \rho_i - \Delta_2 \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j' = 0. \text{ Como } SA_{ir}^{\xi} \geq 0 \text{ então:}$$

$$\Delta_1 \rho_i - \Delta_2 \rho_i - \sum_j \sum_{t=\tau}^{\xi} \Delta v_j' \leq 0 \Leftrightarrow \Delta_1 \rho_i - \Delta_2 \rho_i \leq \sum_j \sum_{t=\tau}^{\xi} \Delta v_j'.$$

c.q.d.

Algoritmo 3-9

PROCEDIMENTO DE AJUSTE DUAL PARA AS VARIÁVEIS ρ_i

$I_j^{t*} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^* \text{ e } v_j' \geq c_{ij}^t \}$, m - número de possíveis localizações para os serviços

1. $i \leftarrow 1$;
 2. $\Delta \rho_i \leftarrow \min_{\tau \leq \xi} \{ SA_{ir}^{\xi} \}$. Se $\Delta \rho_i = 0$, então ir para 3. Se não ir para 8.
 3. $\Delta \rho_i = \max \{ SR_{ir}^{\xi} : \exists (i, \tau, \xi) \in I_R^* \text{ com } SA_{ir}^{\xi} = 0 \text{ e } SR_{ir}^{\xi} \neq 0 \}$.
 4. Se $\Delta \rho_i \neq 0$, então $\rho_i \leftarrow \rho_i + \Delta \rho_i$; $SA_{ir}^{\xi} \leftarrow SA_{ir}^{\xi} + \Delta \rho_i, \forall \tau, \xi \geq \tau$. Se não, ir para 9.
 5. $J^* = \{ (j, t) : I_j^{t*} = \{i\}, \forall t \}$. Executar o procedimento dual ascendente para as variáveis duais v_j' (Algoritmo 3-3).
 6. $J^* = J \times T$. Executar o procedimento dual ascendente para as variáveis duais v_j' (Algoritmo 3-3).
 7. $\Delta \rho_i = \min_{\substack{\tau \\ \xi \geq \tau}} \{ SA_{ir}^{\xi} \}$.
 8. $\Delta \rho_i = \min \{ \Delta \rho_i, \rho_i \}$. Se $\Delta \rho_i \neq 0$, então $SA_{ir}^{\xi} \leftarrow SA_{ir}^{\xi} - \Delta \rho_i, \forall \tau, \xi \geq \tau$; $\rho_i \leftarrow \rho_i - \Delta \rho_i$.
 9. Se $i = m$, então parar; se não, $i \leftarrow i + 1$ e ir para 2.
-

Temos assim que, no pior cenário, o valor da diminuição no passo 8 do

algoritmo iguala o aumento do passo 4, não havendo qualquer alteração no valor da função objectivo dual.

3.2.1.5. Procedimento de Ajuste Primal-Dual para as Variáveis v_j^t

O procedimento de ajuste primal-dual para variáveis v_j^t detecta violações das condições de complementaridade (2-21) e diminui o valor de algumas variáveis v_j^t , permitindo deste modo que outras variáveis v_j^t possam aumentar. Este procedimento tenta diminuir o número de violações das condições (2-21) e, ao mesmo tempo, melhorar o valor da função objectivo dual. O procedimento desenvolvido baseia-se nos trabalhos de Erlenkotter, 1978, Van Roy & Erlenkotter, 1982 e Francisco Saldanha-da-Gama & Captivo, 1996, com as necessárias adaptações, que têm em conta as características específicas do PLDAFR.

Definam-se os seguintes conjuntos:

$$I_j^{t*} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^* \text{ e } v_j^t \geq c_{ij}^t \}$$

$$I_j^{t+} = \{ i : i \in I_i^+ \text{ e } v_j^t > c_{ij}^t \}$$

$$J_j^{t+} = \{ (j, \tau) : I_j^{t*} = \{i\} \text{ e } (i, \gamma, \xi) \notin I^*, \gamma \leq \xi \leq \tau < t \text{ ou } t < \tau \leq \gamma \leq \xi \}$$

$$c_j^{t-} = \max_i \{ c_{ij}^t : v_j^t > c_{ij}^t \}$$

O conjunto I_j^{t+} indica, para cada cliente j , todos os serviços abertos durante o período t , tais que a variável v_j^t é maior do que o custo de afectação c_{ij}^t . A violação de uma condição de complementaridade (2-21) é detectada pela existência de, pelo menos, um par (j, t) tal que o número de elementos em I_j^{t+} é maior do que um. Ao diminuir o valor de uma variável v_j^t tal que o número de elementos em I_j^{t+} é maior do que um, isto significa que se vão aumentar pelo menos as *slacks* $S_{i\tau}^\xi, \tau \leq t \leq \xi$, para dois serviços distintos. Tal poderá levar a um aumento no valor da função objectivo dual.

O conjunto J_j^{t+} representa todas as variáveis v_j^t cujo valor pode ser aumentado pelo aumento de uma *slack* $S_{i\tau}^\xi, \tau \leq t \leq \xi$. É a partir deste conjunto que se constrói J^+ , sendo depois executado o procedimento dual ascendente para as variáveis v_j^t .

O procedimento é formalmente descrito como Algoritmo 3-10.

Algoritmo 3-10

PROCEDIMENTO DE AJUSTE PRIMAL-DUAL PARA AS VARIÁVEIS v_j^t

$I_j^{t*} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^* \text{ e } v_j^t \geq c_{ij}^t \}; \quad I_j^{t+} = \{ i : i \in I_i^+ \text{ e } v_j^t > c_{ij}^t \};$
 $c_j^{t-} = \max_i \{ c_{ij}^t : v_j^t > c_{ij}^t \}; \quad J_j^{t+} = \{ (j, \tau) : I_j^{t*} = \{i\} \text{ e } (i, \gamma, \xi) \notin I^*, \gamma \leq \xi \leq \tau < t \text{ ou } t < \tau \leq \gamma \leq \xi \}.$ As variáveis v_j^t estão organizadas (sem nenhuma ordem especial) num conjunto formado por pares

(j, t) , tal que $(j, t)_k$ representa o k -ésimo elemento desse conjunto; δ_{\max} - número máximo de tentativas; J - conjunto de clientes; T - número de períodos temporais.

1. $(j, t) \leftarrow (j, t)_1$; $q \leftarrow 1$; $\delta \leftarrow 0$.
2. Se $\#I_j^{t+} \leq 1$, então ir para 11; se não, ir para 3.
3. Se $\bigcup_{\substack{i \\ v'_j \geq c'_{ij}}} I_i^{t+} = \emptyset$, então ir para 11; se não, ir para 4.
4. Para cada (i, τ, ξ) , $\tau \leq t \leq \xi$ e $v'_j > c'_{ij}$, $S_{i\tau}^{\xi} \leftarrow S_{i\tau}^{\xi} + v'_j - c'_{ij}$, $v'_j \leftarrow c'_{ij}$.
5. $J^* \leftarrow \bigcup_{\substack{i \\ v'_j \geq c'_{ij}}} I_i^{t+}$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
6. $J^* \leftarrow J^* \cup \{(j, t)\}$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
7. $J^* \leftarrow J \times T$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
8. Se tiver havido alguma alteração na variável v'_j , então ir para 2; se não, ir para 9.
9. Executar o procedimento primal (Algoritmo 3-4).
10. Se tiver havido melhorias no valor das funções objectivo primal ou dual, então $\delta \leftarrow 1$; se não, $\delta \leftarrow \delta + 1$.
11. Se o valor actual da função objectivo primal igualar o valor da função objectivo dual, ou $\delta = \delta_{\max}$, ou $q = \#J \times T$, então parar. Caso contrário, $q \leftarrow q + 1$, $(j, t) \leftarrow (j, t)_q$ e ir para 2.

3.2.1.6. Procedimento Dual Ascendente para as Variáveis u'_i

Ao aumentar o valor de uma variável u'_i , aumentam-se as *slacks* SR_{it}^{ξ} , $\xi \geq t$, mas, ao mesmo tempo, diminuem-se as *slacks* $SA_{i\tau}^{\xi}$, $\tau \leq \xi < t$. Se for possível aumentar *slacks* $S_{i\tau}^{\xi}$ que estejam a bloquear variáveis v'_j , diminuindo *slacks* $S_{i\tau}^{\xi}$ que não estejam a bloquear nenhuma variável v'_j , então será possível melhorar o valor da função objectivo dual.

Só interessará aumentar o valor de uma variável u'_i se existir $SR_{it}^{\xi} = 0$ e $SA_{it}^{\xi} \neq 0$. Esta situação ocorre, por exemplo, quando $(i, t, \xi) \in I_A^+$ mas $SA_{it}^{\xi} \neq 0$ e $SR_{it}^{\xi} = 0$. Neste caso, interessa aumentar SR_{it}^{ξ} mas não mais do que $SA_{it}^{\xi} - SR_{it}^{\xi}$, porque qualquer aumento para além deste valor não irá alterar o valor de S_{it}^{ξ} . Por outro lado, a variável u'_i não poderá aumentar mais do que o valor mínimo de $SA_{i\tau}^{\xi}$, $\forall \tau \leq \xi < t$, para que a solução dual se mantenha admissível. O aumento do valor das variáveis u'_i pode também diminuir o número de violações das condições de complementaridade (2-25).

O procedimento é formalizado como Algoritmo 3-11.

Algoritmo 3-11

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS u'_i

$I_j^{t+} = \left\{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^* \text{ e } v'_j \geq c'_{ij} \right\}$. As variáveis u'_i estão organizadas numa sequência de pares (i, t) , tal que $(i, t)_k$ representa o k -ésimo elemento dessa sequência; J -

conjunto de clientes; I – conjunto de possíveis localizações para os serviços; T – número de períodos temporais.

-
1. $(i, t) \leftarrow (i, t)_1, q \leftarrow 1$.
 2. $\xi \leftarrow t; \Delta u'_i \leftarrow 0; \delta \leftarrow 0$.
 3. Se $SR_{ii}^\xi = 0$ e $SA_{ii}^\xi \neq 0$, então $\Delta u'_i \leftarrow \max\{\Delta u'_i, SA_{ii}^\xi\}$ e $\delta \leftarrow 1$.
 4. Se $\xi = T$, então ir para 5; se não, $\xi \leftarrow \xi + 1$ e ir para 3.
 5. Se $\delta = 0$, então ir para 8. Caso contrário, $\Delta u'_i \leftarrow \min\left\{\Delta u'_i, \min_{\tau \leq \xi < t} SA_{i\tau}^\xi\right\}$, $SR_{ii}^\xi \leftarrow SR_{ii}^\xi + \Delta u'_i, \forall \xi \geq t$;
 $SA_{i\tau}^\xi \leftarrow SA_{i\tau}^\xi - \Delta u'_i, \forall \tau \leq \xi < t$ e $u'_i \leftarrow u'_i + \Delta u'_i$.
 6. $J^* = \{(j, t) : I_j^* = \{i\}, \forall t\}$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
 7. $J^* = J \times T$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
 8. Se $q \neq I \times T$, então parar; caso contrário, $q \leftarrow q + 1; (i, t) \leftarrow (i, t)_q$ e ir para 2.
-

3.2.1.7. Procedimento Dual Descendente para as Variáveis u'_i

Ao diminuir o valor de uma variável u'_i diminuem-se as *slacks* $SR_{ii}^\xi, \xi \geq t$, e aumentam-se as *slacks* $SA_{i\tau}^\xi, \tau \leq \xi < t$. Para que seja garantida a admissibilidade da solução dual, só se poderá diminuir u'_i se $SR_{ii}^\xi > 0, \forall \xi \geq t$. Se for possível aumentar *slacks* $SA_{i\tau}^\xi$ que estejam a bloquear variáveis v'_j , diminuindo *slacks* que não estejam a limitar o crescimento de nenhuma variável v'_j , então será possível melhorar o valor da função objectivo dual.

Algoritmo 3-12

PROCEDIMENTO DUAL DESCENDENTE PARA AS VARIÁVEIS u'_i

$I_j^* = \{i : \exists(\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^* \text{ e } v'_j \geq c_{ij}^t\}$. As variáveis u'_i estão organizadas numa sequência de pares (i, t) , tal que $(i, t)_k$ representa o k -ésimo elemento dessa sequência; J – conjunto de clientes; I – conjunto de possíveis localizações para os serviços; T – número de períodos temporais.

-
1. $(i, t) \leftarrow (i, t)_1, q \leftarrow 1$.
 2. Se $u'_i = 0$, então ir para 7. Caso contrário, $\Delta u'_i \leftarrow 0; \delta \leftarrow 0$ e ir para 3.
 3. Se $SR_{ii}^\xi > 0, \forall \xi \geq t$, então $\Delta u'_i \leftarrow \min_{\xi \geq t} \{SR_{ii}^\xi\}$ e $\delta \leftarrow 1$.
 4. Se $\delta = 0$, então ir para 7. Caso contrário, $\Delta u'_i \leftarrow \min\{\Delta u'_i, u'_i\}; SR_{ii}^\xi \leftarrow SR_{ii}^\xi - \Delta u'_i, \forall \xi \geq t$.
 $SA_{i\tau}^\xi \leftarrow SA_{i\tau}^\xi + \Delta u'_i, \forall \tau \leq \xi < t$ and $u'_i \leftarrow u'_i - \Delta u'_i$.
 5. $J^* \leftarrow \{(j, t) : I_j^* = \{i\}, \forall t\}$. Executar o procedimento dual ascendente para as variáveis v'_j (Algoritmo 3-3).
-

3.2.1.8. Procedimento de Ajuste Dual para as Variáveis π'_i

Ao aumentar o valor das variáveis π'_i aumenta-se o valor das *slacks* $SA_{i\tau}^\xi, \tau \leq \xi$. O aumento destas *slacks* poderá permitir que variáveis v'_j aumentem o seu valor.

Aliás, só compensará aumentar π_i^t se o que se perde na função objectivo dual (porque esta variável contribui com o coeficiente -1) for compensado pelo que se ganha com o aumento de variáveis v_j^t . Se se diminuir o valor das variáveis π_i^t , mantendo-se a admissibilidade da solução dual, melhora-se automaticamente o valor da função objectivo do problema dual.

O procedimento, que se formaliza como Algoritmo 3-13, tenta diminuir ou aumentar o valor das variáveis π_i^t , mantendo-se a admissibilidade da solução dual e garantindo que o valor da função objectivo dual não piora. Se todas as *slacks* $S_{ir}^{\xi}, \tau \leq t \leq \xi$, forem maiores do que zero, significa que não vale a pena tentar aumentar o valor das variáveis π_i^t . Se existir alguma variável π_i^t maior do que zero, poderá diminuir-se o seu valor, actualizando-se as *slacks* $S_{ir}^{\xi}, \tau \leq t \leq \xi$. Se existir alguma *slack* igual a zero, então aumenta-se o valor da variável π_i^t em M (um valor positivo maior do que o maior custo considerado no problema), e verifica-se quais as variáveis v_j^t que poderão ser aumentadas. Se o aumento de π_i^t não for totalmente aproveitado pelo aumento das variáveis v_j^t , significa que todas as *slacks* $S_{ir}^{\xi}, \tau \leq t \leq \xi$, estarão maiores do que zero, pelo que se procede à diminuição de π_i^t .

Proposição 3-5: O procedimento de ajuste dual para as variáveis duais π_i^t nunca piora o valor da função objectivo do problema DC-PLDAFR.

A prova desta proposição segue de perto a da Proposição 3-4, pelo que se omite.

Algoritmo 3-13

PROCEDIMENTO DE AJUSTE DUAL PARA AS VARIÁVEIS π_i^t

$I_j^{*} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \mid (i, \tau, \xi) \in I^{*} \text{ e } v_j^t \geq c_{ij}^t \}$. As variáveis π_i^t estão organizadas numa sequência de pares (i, t) , tal que $(i, t)_q$ representa o q -ésimo elemento dessa sequência. M – número positivo e maior do que qualquer custo presente; J – conjunto de clientes; I – conjunto de possíveis localizações para os serviços; T – número de períodos temporais.

1. $(i, t) \leftarrow (i, t)_1, q \leftarrow 1$.
 2. $\Delta \pi_i^t \leftarrow \min_{\tau \leq t \leq \xi} S_{ir}^{\xi}$. Se $\Delta \pi_i^t \neq 0$, então ir para 7; caso contrário, $\Delta \pi_i^t \leftarrow M$.
 3. $S_{ir}^{\xi} \leftarrow S_{ir}^{\xi} + \Delta \pi_i^t, \forall \tau \leq t \leq \xi, \pi_i^t \leftarrow \pi_i^t + \Delta \pi_i^t$.
 4. $J^+ \leftarrow \{(j, t) : I_j^{*} = \{i\}, \forall t\}$. Executar o procedimento dual ascendente para as variáveis v_j^t (Algoritmo 3-3).
 5. $J^+ \leftarrow J$. Executar o procedimento dual ascendente para as variáveis v_j^t (Algoritmo 3-3).
 6. $\Delta \pi_i^t \leftarrow \min_{\tau \leq t \leq \xi} S_{ir}^{\xi}$.
 7. $\Delta \pi_i^t \leftarrow \min \{ \Delta \pi_i^t, \pi_i^t \}$. Se $\Delta \pi_i^t \neq 0$, então $S_{ir}^{\xi} \leftarrow S_{ir}^{\xi} - \Delta \pi_i^t, \forall \tau \leq t \leq \xi$ e $\pi_i^t \leftarrow \pi_i^t - \Delta \pi_i^t$.
 8. Se $q = \#I \times T$, então parar; caso contrário, $q \leftarrow q+1; (i, t) \leftarrow (i, t)_q$ e ir para 2.
-

3.2.1.9. Pesquisa Local

Após a execução da heurística primal-dual, e nas situações em que não foi encontrada a solução óptima, é sempre possível tentar melhorar a qualidade da solução através da execução de uma pesquisa local.

Definição 3.2: Uma solução admissível S' para o PLDAFR está na k -vizinhança da solução admissível S se e só se S' difere de S pela introdução ou remoção de, no máximo, k períodos de tempo consecutivos ao serviço i .

Considere-se a seguinte notação:

SOL_{kS} = Conjunto de todas as soluções que pertencem à k -vizinhança da solução S ;

Z_S = Função objectivo primal considerando a solução S .

O algoritmo de pesquisa local pode ser formalmente descrito como Algoritmo 3-14. A pesquisa local é executada após a heurística primal-dual, e tendo como base a melhor solução primal conhecida.

Algoritmo 3-14

ALGORITMO DE PESQUISA LOCAL

S – solução admissível inicial; T – número de períodos temporais; SOL_{kS} - Conjunto de todas as soluções que pertencem à k -vizinhança da solução S ; Z_S - Função objectivo primal considerando a solução S .

1. $k \leftarrow 1$.
 2. Calcular $S^* \in SOL_{kS}$, tal que $Z_{S^*} = \min_{S' \in SOL_{kS}} \{Z_{S'}\}$.
 3. Se $Z_{S^*} < Z_S$, então $S \leftarrow S^*$ e ir para 2. Se não, ir para 4.
 4. $k \leftarrow k+1$. Se $k > T$, então parar. Se não, ir para 2.
-
-

3.2.1.10. Cálculo da Solução Óptima através de Pesquisa em Árvore

Foi desenvolvido um método de pesquisa em árvore (*branch-and-bound*) que permite encontrar a solução óptima do PLDAFR, sempre que tal não é possível através da execução da heurística. Este processo de pesquisa em árvore é inicializado tendo em conta a melhor solução encontrada pela heurística, fixando variáveis de localização escolhidas, tendo como base as condições de complementaridade que estão, em cada momento, a ser violadas (nunca são fixadas as variáveis de afectação).

No algoritmo desenvolvido, as variáveis são sempre fixadas primeiro a 0 e depois a 1, utilizando-se para o efeito mecanismos, muito idênticos aos descritos em Erlenkotter, 1978 (com as obrigatórias adaptações ao PLDAFR). A árvore é pesquisada usando uma estratégia de pesquisa em profundidade.

Em cada nodo da árvore, a heurística primal-dual é executada. A variável a fixar é escolhida tendo em conta o par de soluções primal e dual correntes, respeitando a seguinte ordem:

1. Se $a_{ir}^k = 1$ e $SA_{ir}^k \neq 0$, então escolher a variável r_{ir}^k .

2. Se $r_{it}^{\xi} = 1$ e $SR_{it}^{\xi} \neq 0$, então escolher a variável a_{it}^{ξ} .
3. Se $u_i^t \neq 0$ e $r_{it}^{\xi} = 0, \forall \xi \geq t$, então escolher uma variável r_{it}^{ξ} tal que $SR_{it}^{\xi} = 0$.
4. Se $\pi_i^t \neq 0$ e $a_{it}^{\xi} = r_{it}^{\xi} = 0, \forall \tau \leq t \leq \xi$, então escolher uma variável a_{it}^{ξ} ou r_{it}^{ξ} tal que $S_{it}^{\xi} = 0$.
5. Se $v_j^t > c_{ij}^t$, para mais do que um serviço $i \in I^+$, então escolher a variável a_{it}^{ξ} tal que $(i, \tau, \xi) \in I_A^+$ ou r_{it}^{ξ} tal que $(i, \tau, \xi) \in I_R^+, \tau \leq t \leq \xi$.

A ordem apresentada foi aquela que deu melhores resultados nos testes preliminares que foram levados a cabo.

Para fixar uma variável de localização a 0, o seu custo deve ser alterado para $+\infty$. Isto não irá pôr em risco a admissibilidade da solução dual. Para se fixar a variável de localização a 1, o seu custo fixo é igualado a zero, o que implica proceder a algumas alterações na solução dual que, em geral, se torna não admissível. Se a variável a_{it}^{ξ} ou r_{it}^{ξ} é fixada a 1, então todas as variáveis duais v_j^t tais que $v_j^t > c_{ij}^t, \tau \leq t \leq \xi$, terão de ser alteradas de forma a que $v_j^t = c_{ij}^t$. Consequentemente, as *slacks* irão aumentar e a função objectivo do problema dual irá piorar. Esta deterioração do valor da função objectivo dual é compensada somando-lhe o valor do custo fixo da variável que foi fixada a 1.

A utilização de uma heurística para calcular a solução primal em cada nodo da árvore de pesquisa apresenta uma desvantagem: a solução primal calculada pode não ser óptima. Por este motivo, um nodo só será abandonado quando uma das seguintes condições se verifica:

1. O problema corrente é impossível;
2. Os valores das funções objectivo primal e dual igualam-se (a solução óptima do nodo corrente foi calculada pela heurística);
3. O valor actual para a função objectivo dual é pior do que o melhor valor encontrado até ao momento para a função objectivo primal.

3.2.1.11. Resultados Computacionais

3.2.1.11.1. Descrição das Experiências Computacionais

O comportamento da heurística primal-dual e do algoritmo de pesquisa em árvore foi testado, utilizando um conjunto de problemas de teste gerados aleatoriamente. Não foram utilizados nenhuns problemas de teste retirados da literatura por não se conhecerem formulações de tal forma semelhantes à apresentada que permitissem uma fácil adaptação desses problemas de teste.

Foram considerados para m (número de possíveis localizações para os serviços), n (número de clientes) e T (número de períodos temporais) os valores indicados na Tabela 3.1. Para cada combinação desses valores foram gerados cinco problemas, num total de 360.

n	25	50	100	200	500	1000
m	5	10	50	100		
T	5	20	50			

Tabela 3.1 – Parâmetros dos Testes Computacionais para o PLDAFR

Os problemas teste foram criados tendo por base o seguinte procedimento:

1. Geração aleatória de coordenadas (x,y) , de acordo com uma distribuição uniforme e considerando uma matriz de 500×500 . Estas coordenadas irão corresponder aos $m+n$ nodos da rede.
2. Criação aleatória de arcos entre os nodos da rede, havendo uma probabilidade de 75% de criar um dado arco.
3. Criação de arcos (não criados no passo anterior) entre nodos, tais que a distância euclidiana entre eles é inferior a 50, com uma probabilidade de 80%.
4. Geração de custos associados aos arcos: para o primeiro período temporal os custos são aleatoriamente gerados de acordo com uma distribuição uniforme no intervalo [100,1100]. Para os restantes períodos temporais, o custo associado a um arco no período $t, t > 1$, é igual ao custo em $t-1$ acrescido de um factor de alteração que é gerado aleatoriamente e que corresponde a variações entre -10% e +10%.
5. Para cada período temporal, cálculo do caminho mais curto entre cada cliente e cada possível localização para um serviço, utilizando o algoritmo de Floyd-Warshall (Ahuja, Magnanti, & Orlin, 1993).
6. Para cada serviço i e período t , considera-se $tend=t, \dots, T$. Para $tend=t$, geram-se aleatoriamente os custos fixos para as variáveis a_{ii}^{tend} e r_{ii}^{tend} de acordo com uma distribuição uniforme no intervalo [500,3500]. Para $tend > t$, gera-se aleatoriamente um factor entre 0% e 10% e que representa um aumento no custo fixo em relação a $tend-1$.

Todos os testes computacionais foram executados num computador Pentium 4, 1.80 Ghz, com o sistema operativo Windows 2000, um valor máximo de 2000 MB de memória virtual e 260Mb de RAM.

Tanto a heurística como o algoritmo de pesquisa em árvore foram programados utilizando a linguagem de programação C e o compilador Borland-C++ (versão 5.0). O comportamento destes algoritmos foi comparado com o do Cplex, versão 7.0.

O algoritmo de pesquisa em árvore considera dois critérios de paragem adicionais aos indicados em 3.2.1.10, página 73: o algoritmo termina sempre que se atingir o nível 25 000 na estrutura de árvore ou quando o tempo de execução ultrapassar os 175 000 segundos. O Cplex termina em qualquer uma das situações seguintes: quando encontra a solução óptima; quando atinge os 2100000000 nodos na árvore de pesquisa; quando as iterações do simplex ultrapassam as 2100000000; quando apresenta problemas de falta de memória.

3.2.1.11.2. Resultados

Para cada conjunto (n, m, T) , cinco problemas teste foram gerados. A Tabela 3.5 mostra os resultados obtidos com a heurística primal-dual, com e sem a execução da pesquisa local, em relação à qualidade da solução primal. A tabela mostra o pior, o melhor e a média dos resultados obtidos em cada um dos cinco problemas gerados. A qualidade da solução primal é sempre calculada em relação ao melhor limite inferior conhecido para o valor óptimo da função

objectivo primal. Este limite inferior é igual ao valor óptimo da função objectivo, se se conhecer este valor (no caso do Cplex e/ou o algoritmo de pesquisa em árvore conseguirem calcular a solução óptima). Quando o valor óptimo não se conhece, então o limite inferior é dado pelo máximo valor entre o valor da função objectivo dual calculado pela heurística e calculado pelo Cplex. Os valores apresentados são calculados em percentagem como $(Z - Z_{LB})/Z_{LB}$, em que Z é o valor da função objectivo do problema primal e Z_{LB} é o melhor limite inferior conhecido. Na Tabela 3.2 mostra-se um resumo dos valores obtidos relativos à qualidade da solução primal: o menor de todos os valores da coluna “Melhor”, o valor médio de todos os valores da coluna “Média” e o maior valor de todos os valores da coluna “Pior”.

A Tabela 3.6 mostra, em percentagem, a qualidade do limite inferior calculado pela heurística primal-dual (e que corresponde à melhor solução dual encontrada). A qualidade deste limite é calculada como $(Z^* - Z_{LB})/Z^*$, em que Z^* representa o melhor valor conhecido para a função objectivo primal. A Tabela 3.3 resume os valores apresentados. A Tabela 3.7 compara os tempos computacionais, em segundos, da heurística primal-dual com e sem a execução da pesquisa local. A Tabela 3.8 compara os tempos computacionais obtidos pelo Cplex e pelo algoritmo de pesquisa em árvore. A heurística primal-dual é capaz de encontrar soluções primais admissíveis para todos os 360 testes gerados. Os problemas que nem o Cplex nem o algoritmo de pesquisa em árvore foram capazes de resolver não são considerados na Tabela 3.6 e na Tabela 3.8. A simbologia ‘---’ é utilizada quando o procedimento correspondente não foi capaz de resolver nenhum dos cinco problemas gerados. Para instâncias com $(T, m, n) = (5, 100, 500)$ e $(20, 100, 50)$, o Cplex não foi capaz de resolver um de entre os cinco problemas. Para instâncias com $(T, m, n) = (20, 50, 500)$, o algoritmo de pesquisa em árvore não foi capaz de resolver um de entre os cinco problemas. Nestes casos os valores apresentados têm como base os quatro valores obtidos. A Tabela 3.4 apresenta o resumo dos tempos computacionais obtidos com cada uma das técnicas.

Heurística Primal-Dual			Heurística + Pesquisa Local		
Melhor	Média	Pior	Melhor	Média	Pior
0.00	2.59	40.14	0.00	0.86	8.06

Tabela 3.2 – Resumo dos valores relativos à qualidade da solução primal (em percentagem)

Heurística Primal-Dual		
Melhor	Média	Pior
0.00	0.43	4.62

Tabela 3.3 – Resumo dos valores relativos à qualidade do limite inferior (em percentagem)

Heurística Primal-Dual			Heurística + Pesquisa Local		
Melhor	Média	Pior	Melhor	Média	Pior
0.00	550.24	16768.72	0.00	608.94	18580.30
Cplex			Pesquisa em Árvore		
Melhor	Média	Pior	Melhor	Média	Pior
0.03	6311.03	366072.84	0.00	2931.63	173287.38

Tabela 3.4 – Resumo dos valores relativos aos tempos computacionais (em segundos)

3.2.1.11.3. Conclusões

Os resultados computacionais obtidos permitem que sejam retiradas algumas conclusões. Assim:

1. A heurística primal-dual é um procedimento eficiente capaz de calcular soluções admissíveis de qualidade para o PLDAFR. A heurística é capaz de calcular soluções de muito boa qualidade, necessitando de dispendir tempos computacionais razoáveis mesmo para instâncias de grande dimensão. Para os 360 problemas gerados, o desvio médio das soluções calculadas pela heurística em relação ao melhor limite inferior conhecido é de 2.59%.
2. Pode considerar-se vantajosa a execução da pesquisa local levada a cabo após a heurística primal-dual. O tempo computacional adicional que é necessário pode converter-se em melhorias significativas da qualidade da solução final obtida. Quando se executa a pesquisa local, o desvio médio em relação ao melhor limite inferior conhecido desce para 0.86 %.
3. Os limites inferiores calculados pela heurística primal-dual são, no geral, de boa qualidade. O desvio médio em relação à melhor solução primal conhecida é de 0.43%, sendo o pior resultado igual a 4.62%.
4. O Cplex não consegue resolver as instâncias de maior dimensão.
5. A relaxação linear de 86% dos problemas tem uma solução ótima inteira e, portanto, admissível e ótima para o PLDAFR.
6. O método de pesquisa em árvore desenvolvido é, em média, mais eficiente do que o Cplex. No entanto, tem necessidade de construir árvores de pesquisa com bastante mais nodos do que o Cplex (veja-se o ponto 5). O algoritmo de pesquisa em árvore apresenta mais dificuldades com valores para n grandes. Tal comportamento pode ser explicado pelo facto de elevados valores para n levarem a um aumento no número de violações das condições de complementaridade (2-21). Mesmo assim, o algoritmo de pesquisa em árvore é capaz de resolver instâncias de maior dimensão do que o Cplex.

Se se observarem seguidamente as características das soluções encontradas, verifica-se que podem ser muito diferentes, mesmo para instâncias do PLDAFR da mesma dimensão. Tanto quanto possível, vão agora fazer-se algumas observações acerca das características das soluções encontradas.

1. É em problemas com um elevado número de períodos temporais ($T \geq 20$) e poucas localizações possíveis para serviços ($m \leq 10$) que os serviços tendem a ser abertos, fechados e reabertos com mais frequência. À medida que o número de localizações possíveis para os serviços aumenta, diminui o número de serviços que altera a sua configuração mais do que uma vez.
2. Tendo em conta o conjunto $I^* = \{ i \in I : \exists (i, \tau, \xi) \in I_A^* \cup I_R^* \}$, é possível considerar dois subconjuntos de I^* : um subconjunto contém serviços que se encontram em funcionamento durante a maior parte dos períodos temporais considerados; o outro subconjunto contém serviços que são abertos esporadicamente. Com o aumento no número de possíveis localizações para os serviços, aumentam também o número de elementos pertencentes ao segundo subconjunto.
3. A afectação de clientes a serviços mantém-se, basicamente, estável durante o horizonte temporal, sendo esta afectação feita fundamentalmente recorrendo a serviços que pertencem ao primeiro subconjunto referido no ponto anterior. A excepção a esta regra ocorre nos períodos temporais em que estão abertos os serviços que pertencem ao segundo subconjunto.

<i>T</i>	<i>m</i>	<i>N</i>	Heurística Primal-Dual			Heurística + Pesquisa Local			<i>T</i>	<i>m</i>	<i>n</i>	Heurística Primal-Dual			Heurística + Pesquisa Local			
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>				<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	
5	5	25	0.00	0.71	3.57	0.00	0.00	0.00	0.00	20	50	25	0.00	1.52	4.61	0.00	0.00	0.00
5	5	50	0.00	0.10	0.48	0.00	0.00	0.00	0.53	20	50	50	0.53	1.91	3.66	0.00	1.10	3.20
5	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.48	20	50	100	0.48	1.93	4.06	0.00	0.32	1.20
5	5	200	0.00	0.01	0.03	0.00	0.00	0.00	0.59	20	50	200	0.59	2.52	5.24	0.00	1.86	3.45
5	5	500	0.00	0.01	0.03	0.00	0.00	0.00	0.22	20	50	500	0.22	0.90	2.39	0.00	0.39	1.34
5	5	1000	0.00	0.01	0.04	0.00	0.00	0.00	1.21	20	50	1000	1.21	1.74	2.26	0.67	1.17	1.98
5	10	25	0.00	0.22	1.10	0.00	0.00	0.00	0.39	20	100	25	0.39	2.09	5.40	0.00	0.23	0.93
5	10	50	0.00	0.03	0.15	0.00	0.00	0.00	0.90	20	100	50	0.90	1.99	3.57	0.00	0.72	2.48
5	10	100	0.00	0.00	0.02	0.00	0.00	0.00	0.42	20	100	100	0.42	2.33	4.51	0.00	0.32	0.54
5	10	200	0.00	0.05	0.20	0.00	0.00	0.00	0.15	20	100	200	0.15	1.51	2.79	0.00	0.22	0.53
5	10	500	0.00	0.02	0.07	0.00	0.00	0.00	2.04	20	100	500	2.04	2.58	3.49	0.80	1.50	2.24
5	10	1000	0.00	0.03	0.06	0.00	0.00	0.00	2.38	20	100	1000	2.38	2.82	3.05	2.08	2.62	2.96
5	50	25	0.00	1.38	5.16	0.00	0.11	0.11	0.00	20	100	25	0.00	6.34	15.89	0.00	2.70	8.06
5	50	50	0.00	0.51	1.86	0.00	0.37	0.37	1.98	20	50	50	1.98	5.74	10.75	0.13	1.53	5.19
5	50	100	0.00	0.57	1.17	0.00	0.15	0.15	2.33	20	50	100	2.33	4.40	7.02	0.00	1.12	4.37
5	50	200	0.00	0.18	0.65	0.00	0.01	0.01	3.26	20	50	200	3.26	9.83	9.83	0.00	0.39	0.90
5	50	500	0.00	0.48	1.00	0.00	0.11	0.11	0.33	20	50	500	0.33	1.79	3.34	0.00	0.57	1.79
5	50	1000	0.10	0.46	1.08	0.00	0.07	0.07	0.39	20	50	1000	0.39	1.55	3.62	0.02	0.49	1.53
5	100	25	0.00	0.76	2.08	0.00	0.37	0.37	0.74	20	50	25	0.74	9.59	27.41	0.00	0.70	1.79
5	100	50	0.00	1.53	3.86	0.00	0.55	0.55	6.71	20	50	50	6.71	9.52	11.93	0.65	1.66	2.81
5	100	100	0.55	1.42	3.87	0.00	0.42	0.42	0.64	20	50	100	0.64	3.27	4.92	0.22	1.08	2.81
5	100	200	0.10	0.78	1.18	0.00	0.31	0.31	4.03	20	50	200	4.03	8.03	19.7	0.60	2.16	5.07
5	100	500	0.80	1.75	4.88	0.05	0.98	0.98	0.53	20	50	500	0.53	2.40	4.61	0.03	0.58	2.10
5	100	1000	2.13	3.06	3.63	1.89	2.20	2.20	0.24	20	50	1000	0.24	2.12	4.38	0.06	0.90	3.02
20	5	25	0.00	0.01	0.04	0.00	0.00	0.00	8.32	20	50	25	8.32	12.47	17.01	2.33	4.64	6.09
20	5	50	0.00	0.49	1.99	0.00	0.27	0.27	5.01	20	50	50	5.01	8.15	12.81	0.40	3.04	7.66
20	5	100	0.00	1.05	3.78	0.00	0.22	0.22	1.30	20	50	100	1.30	5.70	12.46	0.88	2.53	3.52
20	5	200	0.00	0.36	0.89	0.00	0.02	0.02	3.41	20	50	200	3.41	5.30	7.26	1.16	2.21	4.72
20	5	500	0.00	0.54	0.88	0.00	0.00	0.00	2.36	20	50	500	2.36	4.19	6.99	0.17	1.17	1.81
20	5	1000	0.00	0.19	0.38	0.00	0.01	0.01	1.40	20	50	1000	1.40	2.40	2.85	0.39	1.07	1.80
20	10	25	0.00	0.54	2.70	0.00	0.16	0.16	4.81	20	100	25	4.81	14.32	40.14	2.27	3.51	6.73
20	10	50	0.00	0.44	1.44	0.00	0.16	0.16	4.91	20	100	50	4.91	9.88	22.98	1.13	2.44	4.20
20	10	100	0.00	1.26	3.30	0.00	0.00	0.00	1.65	20	100	100	1.65	5.78	9.93	0.85	2.73	7.99
20	10	200	0.00	0.77	1.81	0.00	0.05	0.05	3.33	20	100	200	3.33	7.19	8.87	0.73	3.04	7.75
20	10	500	0.00	0.08	0.27	0.00	0.02	0.02	2.50	20	100	500	2.50	4.33	5.83	1.17	2.06	2.50
20	10	1000	0.00	0.20	0.69	0.00	0.02	0.02	3.44	20	100	1000	3.44	4.83	7.06	2.03	2.53	3.16

Tabela 3.5 – Qualidade da solução primal (em percentagem)

<i>T</i>	<i>m</i>	<i>n</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>T</i>	<i>m</i>	<i>n</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	5	25	0.00	0.23	1.17	20	5	200	0.00	0.00	0.00
5	5	50	0.00	0.24	1.21	20	5	500	0.00	0.00	0.00
5	5	100	0.00	0.00	0.00	20	5	1000	0.00	0.00	0.00
5	5	200	0.00	0.07	0.36	20	10	25	0.00	0.01	0.06
5	5	500	0.00	0.00	0.00	20	10	50	0.00	0.13	0.63
5	5	1000	0.00	0.00	0.00	20	10	100	0.00	0.03	0.10
5	10	25	0.00	0.07	0.35	20	10	200	0.00	0.07	0.28
5	10	50	0.00	0.16	0.50	20	10	500	0.00	0.01	0.04
5	10	100	0.00	0.15	0.44	20	10	1000	0.00	0.00	0.00
5	10	200	0.00	0.10	0.26	20	50	25	0.00	0.28	1.06
5	10	500	0.00	0.04	0.09	20	50	50	0.41	1.05	1.59
5	10	1000	0.00	0.00	0.00	20	50	100	0.14	0.45	0.81
5	50	25	0.00	0.73	1.98	20	50	200	0.46	0.79	1.58
5	50	50	0.00	0.41	0.59	20	50	500	0.48	0.56	0.61
5	50	100	0.15	0.99	1.78	20	100	25	0.20	1.20	3.06
5	50	200	0.10	0.53	0.75	20	100	50	1.05	1.35	2.16
5	50	500	0.33	0.73	1.28	20	100	100	0.49	0.98	1.53
5	50	1000	0.53	0.69	1.00	20	100	200	0.52	1.03	1.84
5	100	25	0.15	1.30	4.62	50	5	25	0.00	0.11	0.24
5	100	50	0.12	0.68	1.78	50	5	50	0.04	0.21	0.29
5	100	100	0.24	0.95	1.32	50	5	100	0.00	0.37	0.92
5	100	200	0.49	1.11	1.79	50	10	25	0.00	0.15	0.48
5	100	500	1.18	1.63	2.03	50	10	50	0.00	0.72	2.03
20	5	25	0.00	0.08	0.39	50	10	100	0.20	0.84	2.11
20	5	50	0.00	0.01	0.03	50	10	200	0.01	0.21	0.49
20	5	100	0.00	0.41	1.24						

Tabela 3.6 – Qualidade do limite inferior (em porcentagem)

Heurística Primal-Dual						Heurística + Pesquisa Local						Heurística Primal-Dual						Heurística + Pesquisa Local						
T	m	n	Melhor	Média	Pior	Melhor	Média	Pior	T	M	n	Melhor	Média	Pior	Melhor	Média	Pior	T	M	n	Melhor	Média	Pior	
5	5	25	0.00	0.00	0.02	0.00	0.00	0.02	20	50	25	0.56	0.74	1.00	0.72	0.87	1.17	5	5	25	16691.70	17532.73	18177.23	18580.30
5	5	50	0.00	0.00	0.02	0.00	0.01	0.02	20	50	50	3.72	4.70	5.17	4.09	5.04	5.52	5	5	50	3302.00	3465.17	3600.64	3668.97
5	5	100	0.00	0.00	0.00	0.00	0.00	0.00	20	50	100	13.69	13.91	14.31	14.97	15.46	16.13	5	5	100	8298.72	8751.92	9026.67	9392.36
5	5	200	0.00	0.00	0.02	0.00	0.00	0.02	20	50	200	30.52	30.98	31.59	34.59	37.22	40.42	5	5	200	3323.39	3465.17	3600.64	3668.97
5	5	500	0.02	0.02	0.03	0.02	0.02	0.03	20	50	500	78.81	80.70	81.97	92.80	101.58	108.61	5	5	500	8329.88	8751.92	9026.67	9392.36
5	5	1000	0.02	0.04	0.08	0.02	0.04	0.08	20	50	1000	158.58	160.75	163.03	186.12	201.54	215.64	5	5	1000	8329.88	8751.92	9026.67	9392.36
5	10	25	0.00	0.00	0.00	0.00	0.00	0.00	20	100	25	2.78	3.17	3.45	3.39	3.59	3.80	5	10	25	16691.70	17532.73	18177.23	18580.30
5	10	50	0.00	0.00	0.00	0.00	0.00	0.02	20	100	50	17.44	21.03	25.50	18.19	21.87	26.25	5	10	50	3302.00	3465.17	3600.64	3668.97
5	10	100	0.00	0.00	0.02	0.00	0.00	0.02	20	100	100	56.77	57.84	59.89	59.38	61.51	63.73	5	10	100	8298.72	8751.92	9026.67	9392.36
5	10	200	0.00	0.02	0.03	0.00	0.02	0.03	20	100	200	132.03	136.47	140.67	149.20	153.09	157.37	5	10	200	3323.39	3465.17	3600.64	3668.97
5	10	500	0.05	0.06	0.08	0.05	0.07	0.09	20	100	500	334.80	337.49	339.73	371.55	397.94	415.45	5	10	500	8329.88	8751.92	9026.67	9392.36
5	10	1000	0.03	0.08	0.13	0.03	0.09	0.14	20	100	1000	680.94	705.74	721.00	730.00	774.13	848.05	5	10	1000	16691.70	17532.73	18177.23	18580.30
5	50	25	0.00	0.01	0.02	0.00	0.01	0.02	50	5	25	0.25	0.34	0.50	0.25	0.49	0.67	5	50	25	16691.70	17532.73	18177.23	18580.30
5	50	50	0.00	0.01	0.02	0.00	0.01	0.02	50	5	50	0.86	0.98	1.11	1.38	1.44	1.48	5	50	50	3302.00	3465.17	3600.64	3668.97
5	50	100	0.02	0.03	0.03	0.03	0.04	0.05	50	5	100	1.73	1.83	1.92	2.47	2.59	2.66	5	50	100	8298.72	8751.92	9026.67	9392.36
5	50	200	0.13	0.17	0.23	0.03	0.21	0.30	50	5	200	3.83	3.97	4.17	4.39	5.77	6.89	5	50	200	16691.70	17532.73	18177.23	18580.30
5	50	500	0.66	0.94	1.08	0.81	1.17	1.41	50	5	500	8.86	10.17	11.77	9.55	14.92	20.28	5	50	500	3302.00	3465.17	3600.64	3668.97
5	50	1000	2.64	2.98	3.36	3.03	3.76	4.39	50	5	1000	18.63	21.25	22.84	31.52	33.61	37.56	5	50	1000	16691.70	17532.73	18177.23	18580.30
5	100	25	0.00	0.01	0.02	0.00	0.01	0.02	50	10	25	1.31	1.56	1.77	1.98	2.22	2.49	5	100	25	16691.70	17532.73	18177.23	18580.30
5	100	50	0.00	0.02	0.03	0.02	0.03	0.05	50	10	50	3.69	4.00	4.50	3.88	5.07	5.61	5	100	50	3302.00	3465.17	3600.64	3668.97
5	100	100	0.05	0.06	0.08	0.08	0.10	0.14	50	10	100	7.78	8.41	9.63	9.98	10.65	12.09	5	100	100	8298.72	8751.92	9026.67	9392.36
5	100	200	0.63	0.71	0.81	0.80	0.88	0.88	50	10	200	15.64	16.16	16.95	23.59	24.83	25.50	5	100	200	16691.70	17532.73	18177.23	18580.30
5	100	500	2.13	3.55	4.22	2.86	4.27	5.00	50	10	500	40.33	42.37	44.81	58.67	62.34	65.98	5	100	500	3302.00	3465.17	3600.64	3668.97
5	100	1000	6.98	9.16	10.44	10.11	11.70	14.77	50	10	1000	80.45	81.38	83.20	111.48	126.60	138.47	5	100	1000	16691.70	17532.73	18177.23	18580.30
20	5	25	0.00	0.00	0.02	0.00	0.01	0.03	50	50	25	56.97	69.11	72.94	60.64	73.43	77.48	20	5	25	16691.70	17532.73	18177.23	18580.30
20	5	50	0.02	0.05	0.06	0.02	0.07	0.09	50	50	50	168.17	172.48	178.14	183.80	187.94	199.36	20	5	50	3302.00	3465.17	3600.64	3668.97
20	5	100	0.06	0.12	0.14	0.06	0.16	0.23	50	50	100	359.78	367.83	375.63	401.36	419.13	449.36	20	5	100	8298.72	8751.92	9026.67	9392.36
20	5	200	0.23	0.28	0.31	0.23	0.33	0.38	50	50	200	736.67	744.20	750.41	814.38	871.40	896.09	20	5	200	16691.70	17532.73	18177.23	18580.30
20	5	500	0.61	0.85	1.30	0.61	0.99	1.61	50	50	500	1852.78	1864.77	1876.05	2122.88	2241.62	2387.52	20	5	500	3302.00	3465.17	3600.64	3668.97
20	5	1000	1.19	1.87	2.38	1.19	2.12	2.92	50	50	1000	2746.17	3535.07	3756.53	4276.05	4752.17	5447.20	20	5	1000	16691.70	17532.73	18177.23	18580.30
20	10	25	0.00	0.03	0.06	0.00	0.04	0.11	50	100	25	283.94	307.47	329.75	298.63	320.75	340.08	20	10	25	16691.70	17532.73	18177.23	18580.30
20	10	50	0.14	0.17	0.22	0.14	0.18	0.25	50	100	50	755.41	773.04	792.61	780.92	805.16	837.28	20	10	50	3302.00	3465.17	3600.64	3668.97
20	10	100	0.39	0.49	0.55	0.45	0.58	0.75	50	100	100	1611.16	1630.04	1638.49	1695.56	1739.48	1790.31	20	10	100	8298.72	8751.92	9026.67	9392.36
20	10	200	1.02	1.09	1.19	1.02	1.35	1.47	50	100	200	3302.00	3313.96	3323.39	3465.17	3600.64	3668.97	20	10	200	16691.70	17532.73	18177.23	18580.30
20	10	500	2.59	3.04	3.59	2.59	3.44	4.33	50	100	500	8298.72	8325.88	8355.55	8751.92	9026.67	9392.36	20	10	500	16691.70	17532.73	18177.23	18580.30
20	10	1000	5.31	6.04	6.70	5.31	8.01	10.61	50	100	1000	16691.70	16735.65	16768.72	17532.73	18177.23	18580.30	20	10	1000	16691.70	17532.73	18177.23	18580.30

Tabela 3.7 – Tempos Computacionais da Heurística Primal-Dual (em segundos)

<i>T</i>	<i>m</i>	<i>n</i>	Cplex			Pesquisa em Árvore		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	5	25	0.03	0.03	0.05	0.00	0.00	0.02
5	5	50	0.06	0.20	0.72	0.00	0.00	0.02
5	5	100	0.11	0.13	0.14	0.00	0.00	0.00
5	5	200	0.30	4.49	21.14	0.00	0.02	0.08
5	5	500	1.09	1.15	1.20	0.00	0.03	0.06
5	5	1000	2.81	2.89	2.97	0.02	0.11	0.28
5	10	25	0.06	0.07	0.09	0.00	0.00	0.02
5	10	50	0.11	0.37	1.33	0.00	0.00	0.00
5	10	100	0.25	1.05	4.19	0.00	0.02	0.05
5	10	200	0.67	2.48	9.19	0.02	0.04	0.08
5	10	500	2.75	28.36	70.03	0.03	0.19	0.44
5	10	1000	6.61	7.37	8.13	0.03	0.74	2.02
5	50	25	0.34	0.36	0.39	0.00	0.02	0.03
5	50	50	0.70	0.72	0.73	0.00	0.05	0.13
5	50	100	1.74	110.53	442.84	0.08	3.88	13.80
5	50	200	3.84	146.13	568.73	0.27	8.55	29.29
5	50	500	37.33	2463.08	5283.55	182.30	4852.24	14854.80
5	50	1000	207.06	37328.21	150023.69	---	---	---
5	100	25	0.81	2.86	10.61	0.02	0.10	0.22
5	100	50	1.51	6.72	27.30	0.05	0.18	0.42
5	100	100	4.05	103.50	197.42	0.44	12.11	41.89
5	100	200	24.42	1128.55	3414.66	97.78	1990.17	7938.69
5	100	500	1379.13	170075.45	366072.84	---	---	---
20	5	25	1.11	1.17	1.36	0.00	0.34	1.67
20	5	50	2.23	2.30	2.34	0.00	1.75	8.47
20	5	100	4.80	28.70	123.00	0.06	2.45	9.03
20	5	200	13.03	15.70	22.64	0.28	3.26	11.78
20	5	500	475.97	563.61	790.44	1.45	37.00	85.24
20	5	1000	7531.26	7821.59	8478.95	1.42	26.07	95.20
20	10	25	2.58	2.88	3.91	0.02	0.03	0.05
20	10	50	5.81	5.97	6.14	0.16	2.30	6.66
20	10	100	13.53	14.21	14.56	0.56	2.48	6.44
20	10	200	174.77	271.28	541.56	1.25	8.85	17.42
20	10	500	9019.70	9379.23	9690.03	3.27	21.42	84.86
20	10	1000	---	---	---	6.61	41.83	128.67
20	50	25	48.45	118.22	345.88	2.72	22.42	98.88
20	50	50	619.64	1881.24	3163.72	6.05	79.23	240.58
20	50	100	9160.84	9657.41	9891.80	26.02	154.06	386.70
20	50	200	---	---	---	174.41	10218.88	38633.63
20	50	500	---	---	---	1335.17	24601.85	81552.56
20	100	25	618.36	1574.34	3349.52	12.95	44.52	81.11
20	100	50	9325.13	14092.71	20264.30	64.16	5202.91	20157.00
20	100	100	---	---	---	164.64	597.01	1254.75
20	100	200	---	---	---	9323.72	52290.97	173287.38
50	5	25	115.61	152.88	186.00	0.31	30.79	113.61
50	5	50	3560.16	3994.17	4898.30	1.69	2211.72	8148.28
50	5	100	---	---	---	579.05	1168.56	1819.83
50	10	25	3913.23	4070.83	4242.41	32.08	2343.90	4377.22
50	10	50	---	---	---	432.34	9585.01	21667.70
50	10	100	---	---	---	142.61	6248.54	13398.52
50	10	200	---	---	---	2391.20	21833.07	46346.81

Tabela 3.8 – Tempos computacionais do Cplex e do Algoritmo de Pesquisa em Árvore (em segundos)

3.2.2. Problemas com Restrições de Capacidade

Nesta secção descrevem-se as heurísticas que foram desenvolvidas para se resolverem os problemas de localização dinâmicos com capacidades, descritos e formulados em 2.2.3.1, 2.2.3.2 e 2.2.3.3. Estas heurísticas são baseadas na heurística descrita no capítulo anterior, com as alterações necessárias à acomodação de mais variáveis duais, e com procedimentos de construção da solução primal necessariamente diferentes.

Trataremos separadamente a heurística para os problemas descritos em 2.2.3.3. Para os restantes problemas, o esquema de funcionamento das heurísticas é muito semelhante ao apresentado no Algoritmo 3-1, página 59.

Algoritmo 3-15

HEURÍSTICA PRIMAL-DUAL PARA O PLDAFR COM CAPACIDADES

1. Inicialização das variáveis duais.
 2. Procedimento Dual Ascendente para as variáveis duais v_j^l .
 3. Procedimento Primal.
 4. Procedimento de Ajuste Dual para as variáveis duais ρ_i . Se a solução dual for alterada, então ir para 2; se não, ir para 5.
 5. Enquanto a função objectivo do problema dual melhorar, executar o Procedimento de Ajuste Dual para as variáveis duais v_j^l .
 6. Procedimento de Ajuste Dual para as variáveis duais ρ_i . Se a solução dual sofrer alterações, então ir para 2; se não, ir para 7.
 7. Procedimento Dual Ascendente para as variáveis duais u_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 8.
 8. Procedimento Dual Descendente para as variáveis duais u_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 9.
 9. Procedimento Dual Ascendente para as variáveis duais λ_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 10.
 10. Procedimento Dual Descendente para as variáveis duais λ_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 11.
 11. Procedimento Dual Ascendente para as variáveis duais β_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 12.
 12. Procedimento Dual Descendente para as variáveis duais β_i^l . Se a solução dual sofrer alterações ir para 2; se não, ir para 13.
 13. Procedimento de Ajuste Dual para as variáveis π_i^l . Se a solução dual sofrer alterações ir para 2; caso contrário, parar.
-

Os passos 1, 4-8 e 13 utilizam os procedimentos descritos no capítulo anterior, sem quaisquer alterações. No passo 1, a inicialização das variáveis duais é feita utilizando o Algoritmo 3-2, página 60, sendo que as variáveis duais, que não estão presentes nesse algoritmo, são inicializadas a zero (a menos que explicitado de outro modo).

Quanto ao passo 2 da heurística, em que é executado o Procedimento Dual Ascendente para as variáveis duais v_j^l , este procedimento é igual ao referido anteriormente (Algoritmo 3-3, página 59) mas com as seguintes alterações: se estivermos a considerar restrições de capacidade máxima (2-27), então os custos

de afectação c_{ij}^t deverão ser substituídos por $c_{ij}^t + d_j^t \lambda_i^t$; se estivermos a considerar restrições de capacidade máxima (2-27) e mínima (2-28), então os custos de afectação a considerar são $c_{ij}^t + d_j^t (\lambda_i^t - \beta_i^t)$; se estivermos a considerar restrições de capacidade máxima decrescente (2-37), então deve considerar-se $c_{ij}^t + d_j^t \sum_{\tau=1}^T \lambda_i^\tau$.

Nas secções seguintes irão ser descritos os procedimentos executados nos passos 3 e 9 a 12. Na secção III.2.2.6 tratar-se-á dos problemas descritos em II.2.2.3.

3.2.2.1. Problemas com Restrições de Capacidade Máxima

As expressões (2-33) e (2-34) explicitam a relação existente entre as *slacks* SA_{it}^ξ e SR_{it}^ξ e as variáveis duais λ_i^t . Uma alteração na variável dual λ_i^t vai influenciar todas as *slacks* SA_{it}^ξ e SR_{it}^ξ , tais que $\tau \leq t \leq \xi$. Cada uma destas variáveis duais contribui para cada *slack* com um valor dado por (3-1). Pode, assim, concluir-se que o valor das *slacks* pode aumentar com a variação do valor da variável dual: o aumento do valor da variável λ_i^t aumentará a segunda parte da expressão (3-1) e diminuirá a primeira parte, e a diminuição de λ_i^t terá o efeito oposto.

$$\sum_j \max \{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} + Q_i \lambda_i^t \quad (3-1)$$

3.2.2.1.1. Procedimento Dual Ascendente para as Variáveis λ_i^t

Considerem-se as seguintes definições:

$$J1(\delta) = \left\{ j \in J : \frac{1}{d_j^t} (v_j^t - c_{ij}^t - d_j^t \lambda_i^t) \leq \delta \right\} \quad (3-2)$$

$$\Delta = \max_{j \in J} \left\{ \frac{1}{d_j^t} \max \{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} \right\} \quad (3-3)$$

Proposição 3-6: Considere-se $\delta \in]0, \Delta]$, com Δ definido como em (3-3). Considere-se ainda que a variável dual λ_i^t é aumentada em δ . Se $Q_i \delta < \sum_{j \in J1(\delta)} \max \{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} + \sum_{j \in J1(\delta)} d_j^t \delta$ então todas as *slacks* SA_{it}^ξ e SR_{it}^ξ com $\tau \leq t \leq \xi$ irão ser aumentadas em:

$$\Omega(\delta) = \sum_{j \in J1(\delta)} \max \{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} + \sum_{j \in J1(\delta)} d_j^t \delta - Q_i \delta \quad (3-4)$$

Este resultado segue de perto outros que podem ser encontrados em Guignard & Spielberg, 1979, e Saldanha-da-Gama, 2002, pelo que a respectiva prova é omitida.

Proposição 3-7: Para cada $\delta' > \Delta$ é sempre possível encontrar $\delta \leq \Delta$ tal que $\Omega(\delta) > \Omega(\delta')$. Por este motivo, o máximo valor que δ deve tomar é dado por Δ .

A prova da Proposição 3-7 é trivial e obtém-se directamente através da Proposição 3-6 e de (3-3), pelo que é omitida.

O procedimento dual ascendente para variáveis duais λ_i^t é baseado na Proposição 3-6 e segue o trabalho desenvolvido por Saldanha-da-Gama, 2002. Este procedimento é formalizado como Algoritmo 3-16.

Algoritmo 3-16

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS λ_i^t

d_j^t - procura do cliente j durante o período t ; c_{ij}^t - custo de afectar o cliente j ao serviço i ;

$\Omega(\delta) = \sum_{j \in J1(\delta)} \max\{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} + \sum_{j \in J1(\delta)} d_j^t \delta - Q_t \delta$; $J1(\delta) = \left\{ j \in J : \frac{1}{d_j^t} (v_j^t - c_{ij}^t - d_j^t \lambda_i^t) \leq \delta \right\}$; T - número de períodos temporais; m - número de possíveis localizações para os serviços.

1. $t \leftarrow 1$;
2. $i \leftarrow 1$;
3. $\delta \leftarrow 0$; $\delta' \leftarrow +\infty$; *changed* \leftarrow false;
4. $\delta \leftarrow \max_{\substack{j \in J \\ \frac{1}{d_j^t} \max\{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} < \delta}} \left\{ \frac{1}{d_j^t} \max\{0, v_j^t - c_{ij}^t - d_j^t \lambda_i^t\} \right\}$; se $\delta = 0$, então ir para 9.
5. Calcular $J1(\delta)$.
6. Se $\Omega(\delta) > 0$; então ir para 7; se não, ir para 8.
7. $SA_{ir}^{\xi} \leftarrow SA_{ir}^{\xi} + \Omega(\delta)$, $SR_{ir}^{\xi} \leftarrow SR_{ir}^{\xi} + \Omega(\delta)$, $\forall \tau, \xi$ tal que $\tau \leq \xi$, $\lambda_i^t \leftarrow \lambda_i^t + \delta$; *changed* \leftarrow true; ir para 9.
8. $\delta' \leftarrow \delta$;
9. Se $\delta \neq 0$ e *changed* = false, então ir para 3; caso contrário, $i \leftarrow i + 1$;
10. Se $i > m$, então $t \leftarrow t + 1$; se não, ir para 3.
11. Se $t > T$, então parar. Caso contrário, ir para 2.

3.2.2.1.2. Procedimento Dual Descendente para as Variáveis λ_i^t

Todos os valores $v_j^t - c_{ij}^t - d_j^t \lambda_i^t$ serão aumentados com a diminuição da variável dual λ_i^t . Para valores $v_j^t - c_{ij}^t - d_j^t \lambda_i^t \geq 0$, este aumento pode ser calculado como δd_j^t , sendo δ a variação no valor da variável dual. Se δ é tal que todos os valores $v_j^t - c_{ij}^t - d_j^t \lambda_i^t < 0$ se mantêm menores ou iguais a zero, então as alterações nas *slacks* SA_{ir}^{ξ} e SR_{ir}^{ξ} , com $\tau \leq \xi$, são dadas por (3-5). O procedimento dual descendente para as variáveis λ_i^t (Algoritmo 3-17) é baseado nesta observação e segue o trabalho desenvolvido por Saldanha-da-Gama, 2002.

$$\delta \left(Q_i - \sum_{\substack{j \in J \\ v_j^t - c_{ij}^t - d_j^t \lambda_i^t \geq 0}} d_j^t \right) \quad (3-5)$$

Algoritmo 3-17

PROCEDIMENTO DUAL DESCENDENTE PARA AS VARIÁVEIS λ_i^t

m – número de possíveis localizações para serviços; T – número de períodos temporais considerados ; d_j^t - procura do cliente j durante o período t ; c_{ij}^t - custo de afectar o cliente j ao serviço i ; Q_i - capacidade máxima do serviço i .

1. $t \leftarrow 1$;
 2. $i \leftarrow 1$;
 3. Se $\left(Q_i - \sum_{\substack{j \in J \\ v_j^t - c_{ij}^t - d_j^t \lambda_i^t \geq 0}} d_j^t \right) < 0$, então ir para 6.
 4. $\delta \leftarrow \min \left\{ \lambda_i^t, \min_{\substack{j \in J \\ v_j^t - c_{ij}^t - d_j^t \lambda_i^t < 0}} \left\{ -\frac{v_j^t - c_{ij}^t - d_j^t \lambda_i^t}{d_j^t} \right\} \right\}$; se $\delta = 0$, então ir para 6.
 5. $SA_{ir}^\xi \leftarrow SA_{ir}^\xi + \delta \left(Q_i - \sum_{\substack{j \in J \\ v_j^t - c_{ij}^t - d_j^t \lambda_i^t \geq 0}} d_j^t \right)$, $SR_{ir}^\xi \leftarrow SR_{ir}^\xi + \delta \left(Q_i - \sum_{\substack{j \in J \\ v_j^t - c_{ij}^t - d_j^t \lambda_i^t \geq 0}} d_j^t \right)$, para τ, ξ tais que $\tau \leq t \leq \xi$;
 $\lambda_i^t \leftarrow \lambda_i^t - \delta$.
 6. $i \leftarrow i + 1$;
 7. Se $i > m$, então $t \leftarrow t + 1$, se não ir para 3.
 8. Se $t > T$, então parar, caso contrário ir para 2.
-
-

3.2.2.1.3. Procedimento Primal

O procedimento primal que foi desenvolvido é muito semelhante ao descrito em 3.2.1.3, página 61, para o PLDAFR. Com efeito, começa por construir-se uma solução primal utilizando o Algoritmo 3-4, testando depois a admissibilidade da solução construída (pode não ser admissível devido às restrições de capacidade). Se em algum período temporal os serviços que estão em funcionamento não são suficientes para assegurarem a satisfação da totalidade da procura dos clientes, então mais serviços terão de ser abertos.

Na descrição deste algoritmo vão utilizar-se alguns conjuntos já definidos anteriormente e também:

F_i^t = custo mais baixo em que se incorre por considerar um serviço $i \notin I_t^+$ em funcionamento durante o período temporal t .

O procedimento primal é descrito formalmente como Algoritmo 3-18.

Algoritmo 3-18

PROCEDIMENTO PRIMAL PARA PLDAFR COM CAPACIDADES MÁXIMAS

$I_t^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{it}^\xi = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{it}^\xi = 1\}$; $F_i^t =$ custo mais baixo em que se incorre por considerar um serviço $i \notin I_t^+$ em funcionamento durante o período temporal t ; T – número de períodos temporais considerados; J – conjunto de todos os clientes; d_j^t – Procura do cliente j no período t ; Q_i – capacidade máxima do serviço i ; c_{ij}^t – custo de afectar o cliente j ao serviço i n o período t .

1. Executar o Algoritmo 3-4, construindo os conjuntos I_t^+ , I_A^+ e I_R^+ .
 2. $t \leftarrow 1$;
 3. $D \leftarrow \sum_j d_j^t$; $C \leftarrow \sum_{i \in I_t^+} Q_i$. Se $D \leq C$, então ir para 8.
 4. Calcular $F_i^t, \forall i \notin I_t^+$.
 5. Calcular $F_i^t \leftarrow \frac{F_i^t}{Q_i} \left[\frac{\phi}{Q_i} \right], \forall i \notin I_t^+$, com $\phi = \begin{cases} D - C, & \text{se } C + Q_i < D \\ Q_i, & \text{em caso contrário} \end{cases}$.
 6. Considerar i^t , tal que $F_{i^t}^t \leftarrow \min_{i \in I_t^+} \{F_i^t\}$;
 7. $I_t^+ \leftarrow I_t^+ \cup \{i^t\}$; Actualizar os conjuntos I_A^+ , I_R^+ e $I_t^+, \forall t$; $C \leftarrow C + Q_{i^t}$; Se $D \leq C$ então ir para 8. Caso contrário, ir para 5.
 8. $t \leftarrow t + 1$. Se $t \leq T$, então ir para 3; se não, ir para 9.
 9. $t \leftarrow 1$;
 10. Resolver um problema de transportes considerando o conjunto J de clientes como origens (com ofertas d_j^t), os serviços pertencentes ao conjunto I_t^+ como destinos (com procuras Q_i), e custos unitários de transporte dados por c_{ij}^t/d_j^t .
 11. $t \leftarrow t + 1$; Se $t \leq T$, ir para 10. Caso contrário, parar.
-

O passo 5 do algoritmo serve para penalizar todos os serviços cuja capacidade não é suficiente para cobrir a diferença entre a totalidade da procura por parte dos clientes e a capacidade total dos serviços em funcionamento. Esta penalização é justificada porque, no caso de se optar por abrir algum destes serviços, pelo menos um outro também terá de ser aberto.

O cálculo de $F_i^t, \forall i \notin I_t^+$, não é trivial, especialmente se o serviço i pertencer a um ou mais conjuntos $I_{t'}^+, t' \neq t$. Todas as hipóteses necessitam de ser testadas, considerando a introdução de uma nova variável em I_A^+ ou I_R^+ e a possível remoção de uma ou mais variáveis destes conjuntos.

Considere-se que o serviço i não se encontra em funcionamento durante o período t mas está em funcionamento em períodos temporais anteriores e posteriores a t , como exemplificado na Figura 3.1.

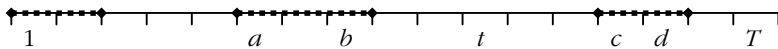


Figura 3.1– Períodos de funcionamento de um serviço i (O traçado $\blacklozenge \bullet \bullet \bullet \blacklozenge$ representa os períodos de funcionamento de i)

Os períodos temporais a, b, c, d podem ser definidos matematicamente como:
 $b = \max\left\{0, \max_{t' < t} \{t' : i \in I_t^+\}\right\}$; $a = t'$ tal que $\exists(i, t', b) \in I_A^+ \cup I_R^+$;
 $c = \min\left\{T + 1, \min_{t' > t} \{t' : i \in I_t^+\}\right\}$; $d = t'$ tal que $\exists(i, c, t') \in I_A^+ \cup I_R^+$. O período temporal b representa o período anterior a t e mais próximo deste, tal que o serviço i está aberto. O período temporal c representa o período posterior e mais próximo de t , tal que i está em funcionamento. O Algoritmo 3-19 permite calcular o valor de F_i^t , para um dado i .

Algoritmo 3-19

PROCEDIMENTO PARA CÁLCULO DE F_i^t , PARA $i \notin I_t^+$

i – serviço que se pondera abrir em t ; t – período temporal considerado; T – número de períodos temporais; $b = \max\left\{0, \max_{t' < t} \{t' : i \in I_t^+\}\right\}$; $a = t'$ tal que $\exists(i, t', b) \in I_A^+ \cup I_R^+$;

$c = \min\left\{T + 1, \min_{t' > t} \{t' : i \in I_t^+\}\right\}$; $d = t'$ tal que $\exists(i, c, t') \in I_A^+ \cup I_R^+$.

1. Se $b = 0$ e $c = T + 1$, então $F_i^t \leftarrow \min\{FA_{\tau}^{\xi} : 1 \leq \tau \leq t \leq \xi \leq T\}$; Parar.

2. Se $b = 0$ e $c \leq T$ então

$$F_i^t \leftarrow \min\left\{\min\{FA_{\tau}^{\xi} - FA_{ic}^d + FR_{ic}^d : 1 \leq \tau \leq t \leq \xi < c\}, \min\{FA_{\tau}^d - FA_{ic}^d : 1 \leq \tau \leq t\}\right\}, \text{ parar.}$$

3. Se $b > 0$ e $c = T + 1$, então ir para 4, se não ir para 7.

4. Se $(i, a, b) \in I_A^+$ ir para 5. Caso contrário, ir para 6.

5. $F_i^t \leftarrow \min\left\{\min\{FA_{ia}^{\xi} - FA_{ia}^b : t \leq \xi \leq T\}, \min\{FR_{\tau}^{\xi} : b < \tau \leq t \leq \xi \leq T\}\right\}$. Parar.

6. $F_i^t \leftarrow \min\left\{\min\{FR_{ia}^{\xi} - FR_{ia}^b : t \leq \xi \leq T\}, \min\{FR_{\tau}^{\xi} : b < \tau \leq t \leq \xi \leq T\}\right\}$. Parar.

7. Se $(i, a, b) \in I_A^+$ ir para 8; caso contrário, ir para 9.

8. $F_i^t \leftarrow \min\left\{FA_{ia}^d - FA_{ia}^b - FR_{ic}^d, \min\{FR_{\tau}^{\xi}, b < \tau \leq t \leq \xi < c\}, \right.$
 $\left. \min\{FA_{ia}^{\xi} - FA_{ia}^b, t \leq \xi < c\}, \min\{FR_{\tau}^d - FR_c^d, b < \tau \leq t\}\right\}$. Parar.

9. $F_i^t \leftarrow \min\left\{FR_{ia}^d - FR_{ia}^b - FR_{ic}^d, \min\{FR_{\tau}^{\xi}, b < \tau \leq t \leq \xi < c\}, \right.$
 $\left. \min\{FR_{ia}^{\xi} - FR_{ia}^b, t \leq \xi < c\}, \min\{FR_{\tau}^d - FR_c^d, b < \tau \leq t\}\right\}$; Parar.

Explica-se agora o funcionamento deste algoritmo:

- A. O passo 1 considera a situação em que não existe nenhum $(i, \tau, \xi) \in I_A^+ \cup I_R^+$. Neste caso o procedimento procura uma variável a_{τ}^{ξ} , $\tau \leq t \leq \xi$, com o menor custo fixo possível.
- B. O passo 2 considera a situação em que não existe $(i, \tau, \xi) \in I_A^+ \cup I_R^+$ com $\xi < t$, mas existe $(i, c, d) \in I_A^+$. Neste caso, o menor custo de incluir i em I_t^+ é dado pelo menor custo entre:
- i. Considerar o menor custo fixo entre as variáveis a_{τ}^{ξ} , $\tau \leq t \leq \xi < c$, mais o custo de alterar a variável a_{ic}^d de um para zero e a variável r_{ic}^d de zero para um;
 - ii. Considerar o menor custo fixo entre as variáveis a_{τ}^d , $\tau \leq t$, menos o custo fixo de a_{ic}^d , que será alterada de um para zero.
- C. Os passos 4 a 6 consideram a situação em que $(i, a, b) \in I_A^+ \cup I_R^+$ mas não existe (i, τ, ξ) , $t < \tau \leq \xi$.

Se $(i, a, b) \in I_A^+$, então F_i^t será calculado como o menor de entre:

- i. O menor custo fixo entre as variáveis a_{ia}^{ξ} , $\xi \geq t$, menos o custo fixo da variável a_{ia}^b que é alterada de um para zero.
- ii. O custo fixo mais pequeno entre variáveis r_{it}^{ξ} , $b < \tau < t \leq \xi$.

Se $(i, a, b) \in I_R^+$, então F_i^t será calculado como o menor de entre:

- i. O custo fixo mais pequeno entre as variáveis r_{ia}^{ξ} , $\xi \geq t$, menos o custo fixo da variável r_{ia}^b , que é alterada de um para zero;
- ii. O menor custo fixo entre as variáveis r_{it}^{ξ} , $b < \tau < t \leq \xi$.

D. Os passos 7-9 consideram a situação em que existe $(i, a, b) \in I_A^+ \cup I_R^+$ e também $(i, c, d) \in I_A^+ \cup I_R^+$. Se $(i, a, b) \in I_A^+$ ($(i, a, b) \in I_R^+$), então F_i^t será calculado como o menor de entre:

- i. O custo fixo da variável a_{ia}^d (r_{ia}^d), menos os custos fixos respeitantes às variáveis a_{ia}^b (r_{ia}^b) e r_{ic}^d cujo valor é modificado de um para zero;
- ii. O menor custo fixo de entre as variáveis r_{it}^{ξ} , $b < \tau \leq t \leq \xi < c$.
- iii. O menor custo fixo entre as variáveis a_{ia}^{ξ} (r_{ia}^{ξ}), $t \leq \xi < c$, menos o custo fixo da variável a_{ia}^b (r_{ia}^b) que é modificada de um para zero.
- iv. O menor custo fixo entre variáveis a_{it}^d (r_{it}^d), $b < \tau \leq t$, menos o custo fixo da variável r_{ic}^d cujo valor se modifica de um para zero.

No passo 7 do Algoritmo 3-18, após ter sido escolhido o serviço i que irá ser aberto durante o período temporal t , é necessário actualizar os conjuntos I_A^+ , I_R^+ e I_t^+ , $\forall t$, de acordo com o cálculo, utilizando o Algoritmo 3-19, do valor de F_i^t . Nos passos 9 a 11 do Algoritmo 3-18 são resolvidos T problemas de transporte (um para cada período temporal), que permitem calcular o valor óptimo das variáveis de afectação x_{ij}^t , dados os conjuntos I_t^+ que fixam quais os serviços abertos em cada período t .

3.2.2.2. Introdução de Restrições de Capacidade Mínima

Como se pode concluir da observação de (2-33) e (2-34), as variáveis duais β_i^t apresentam um comportamento simétrico quando comparado com o comportamento das variáveis λ_i^t . Por este motivo, o procedimento dual ascendente para as variáveis λ_i^t (Algoritmo 3-16) torna-se o procedimento dual descendente para as variáveis β_i^t . De igual modo, o Algoritmo 3-17 é agora o algoritmo dual ascendente para β_i^t .

A única diferença significativa entre a heurística desenvolvida para o problema que considera apenas as restrições de capacidade máxima e para o problema que considera também restrições de capacidade mínima reside no procedimento primal.

Dados os conjuntos I_t^+ , $\forall t$, uma solução primal é admissível, em relação às restrições de capacidade, se verifica as condições (3-6) e (3-7).

$$\sum_{i \in I_t^+} Q_i^1 \leq \sum_j d_j^t, \quad \forall t \quad (3-6)$$

$$\sum_j d_j^t \leq \sum_{i \in I_t^+} Q_i, \quad \forall t \quad (3-7)$$

Para um dado período temporal t no máximo uma destas condições pode ser violada. Descreve-se agora o procedimento primal desenvolvido.

3.2.2.2.1. Procedimento Primal

Para que se construa uma solução admissível para o problema PLDAFR com restrições de capacidade máxima e mínima, optou-se por garantir primeiro a satisfação das condições (3-7). Se as condições (3-6) estiverem a ser violadas, o procedimento altera a solução primal, fechando serviços abertos ou trocando serviços abertos por outros com capacidades mínimas menores.

Defina-se:

f_i^t = o menor custo em que se incorre por fechar o serviço $i \in I_t^+$ durante o período t .

g_i^t = o menor custo em que se incorre por fechar o serviço $i \in I_t^+$ durante o período t e abrir um ou mais serviços $i' \notin I_t^+$ tais que a capacidade máxima total dos serviços i' é suficiente para satisfazer a condição (3-7) e a capacidade mínima total é menor do que Q_i^1 . O conjunto de serviços $i' \notin I_t^+$ que são abertos é designado por $K(i,t)$.

Descreve-se agora o procedimento primal como Algoritmo 3-20.

Algoritmo 3-20

PROCEDIMENTO PRIMAL PARA PLDAFR COM CAPACIDADES MÁXIMAS E MÍNIMAS

$I_t^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{it}^{\tau} = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{it}^{\tau} = 1\}$; F_i^t = custo mais baixo em que se incorre por considerar um serviço $i \notin I_t^+$ em funcionamento durante o período temporal t ; T - número de períodos temporais considerados; J - conjunto de todos os clientes; d_j^t - procura do cliente j no período t ; Q_i - capacidade máxima do serviço i ; Q_i^1 - capacidade mínima do serviço i ; c_{ij}^t - custo de afectar o cliente j ao serviço i no período t .

1. Execute-se o Algoritmo 3-4, construindo os conjuntos I_t^+ , I_A^+ e I_R^+ .
2. $t \leftarrow 1$;
3. $D \leftarrow \sum_j d_j^t$; $C_{\max} \leftarrow \sum_{i \in I_t^+} Q_i$; $C_{\min} \leftarrow \sum_{i \in I_t^+} Q_i^1$. Se $C_{\min} \leq D \leq C_{\max}$, então ir para 16. Caso contrário, ir para 4.
4. Calcular $F_i^t, \forall i \notin I_t^+$. $F_i^t \leftarrow F_i^t (1 + Q_i^1 / Q_i), \forall i \notin I_t^+$. Se $D \leq C_{\max}$, então ir para 8; se não, ir para 5.
5. Calcular $F_i^{t'} = \frac{F_i^t}{Q_i} \left[\frac{\phi_i}{Q_i} \right], \forall i \notin I_t^+$, com $\phi_i = \begin{cases} D - C_{\max}, & \text{se } C_{\max} + Q_i < D \\ Q_i, & \text{caso contrário} \end{cases}$.
6. Considerar i' , tal que $F_{i'}^{t'} = \min_{i \in I_t^+} \{F_i^{t'}\}$.
7. $I_t^+ \leftarrow I_t^+ \cup \{i'\}$. Reconstruir os conjuntos I_A^+ , I_R^+ e $I_t^+, \forall t$. $C_{\max} \leftarrow C_{\max} + Q_{i'}$; $C_{\min} \leftarrow C_{\min} + Q_{i'}^1$. Se $D \leq C_{\max}$, então ir para 8; se não, ir para 5.

8. Se $C_{\min} \leq D$, então ir para 16; se não, ir para 9.
9. Calcular $f_i^t, \forall i \in I_i^+$.
10. Se $\{i \in I_i^+ : C_{\max} - Q_i \geq D\} = \emptyset$, então ir para 12. Caso contrário, considerar i' , tal que

$$f_{i'}^t = \min_{\substack{i \in I_i^+ \\ C_{\max} - Q_i \geq D}} \{f_i^t \phi_i\}, \text{ com } \phi_i = \begin{cases} 1 + \frac{C_{\min} - D - Q_i}{C_{\min} - D}, & \text{se } f_i^t > 0 \\ 1, & \text{caso contrário} \end{cases}.$$
11. $I_i^+ \leftarrow I_i^+ \setminus \{i'\}$. Reconstruir I_A^+ e I_R^+ . $C_{\min} \leftarrow C_{\min} - Q_{i'}$; $C_{\max} \leftarrow C_{\max} - Q_{i'}$. Se $C_{\min} \leq D$, então ir para 16; se não, ir para 12.
12. Calcular $g_i^t, \forall i \in I_i^+$. Se $g_i^t = +\infty, \forall i \in I_i^+$, então parar: não é possível encontrar uma solução admissível.
13. Considerar i' , tal que $g_{i'}^t = \min_{i \in I_i^+} \{g_i^t\}$. $I_i^+ \leftarrow I_i^+ \setminus \{i'\}$. $I_i^+ \leftarrow I_i^+ \cup I(i', t)$.
14. Reconstruir I_A^+, I_R^+ e $I_i^+, \forall t$. $C_{\min} \leftarrow C_{\min} - Q_{i'} + \sum_{i \in I(i', t)} Q_i$; $C_{\max} \leftarrow C_{\max} - Q_{i'} + \sum_{i \in I(i', t)} Q_i$.
15. Se $C_{\min} \leq D$, então ir para 16; se não ir para 12.
16. $t \leftarrow t + 1$. Se $t \leq T$, ir para 3; se não ir para 17.
17. $t \leftarrow 1$;
18. Resolver um problema de transportes considerando o conjunto J de clientes como sendo as origens (com ofertas d_j^t), dois destinos (i_1, i_2) correspondendo a cada $i \in I_i^+$ (um com procura igual a Q_i e outra com procura igual a $Q_i - Q_{i'}$), e custos unitários de transporte dados por c_{ij}^t / d_j^t . Se houver necessidade de considerar uma origem fictícia, então os custos de transporte desta origem para todos os destinos i_1 serão considerados iguais a $+\infty$ (de forma a que sejam satisfeitas (2-28)).
19. $t \leftarrow t + 1$. Se $t \leq T$, ir para 19; se não, parar.

Alguns dos passos deste procedimento requerem explicações adicionais. No passo 4 do algoritmo, F_i^t é calculado utilizando o Algoritmo 3-19, mas garantindo que (3-6) é satisfeita para todos os períodos temporais $t' < t$. Este valor é depois alterado para que tenha em conta a relação existente, em cada serviço, entre as capacidades máximas e mínimas. Um serviço cuja capacidade mínima esteja muito próxima da sua capacidade máxima poderá mais facilmente ser responsável pela violação da condição (3-6).

No passo 9, o cálculo de f_i^t é feito de forma simples como mostra o Algoritmo 3-21.

Algoritmo 3-21

PROCEDIMENTO DE CÁLCULO DE f_i^t

t e $i \in I_i^+$ – período temporal e serviço em relação aos quais se pretende calcular f_i^t . Como $i \in I_i^+$, então existe obrigatoriamente $(i, a, b) \in I_A^+ \cup I_R^+$ tal que $a \leq t \leq b$.

1. Se $(i, a, b) \in I_A^+$, então $f_i^t = \begin{cases} -FA_{ia}^b, & \text{se } a = b \\ FA_{ia}^{t-1} + FR_{a+1}^b - FA_{ia}^b, & \text{caso contrário} \end{cases}$.
 2. Se $(i, a, b) \in I_R^+$, então $f_i^t = \begin{cases} -FR_{ia}^b, & \text{se } a = b \\ FR_{ia}^{t-1} + FR_{a+1}^b - FR_{ia}^b, & \text{caso contrário} \end{cases}$.
-

No passo 10, o procedimento tenta fechar serviços cuja capacidade máxima não é necessária para garantir a satisfação das condições (3-7). Todos aqueles serviços tais que o seu fecho não é suficiente para garantir a satisfação das condições (3-6) são penalizados. O procedimento chega ao passo 12 sempre que não é possível satisfazer simultaneamente as condições (3-6) e (3-7) através da introdução ou remoção de serviços do conjunto I_t^* . Se tal acontecer, então será necessário “trocar” serviços abertos por serviços fechados. Sempre que se fecha um serviço num período t , a capacidade total máxima disponível neste período diminui. Como tal, será necessário abrir um ou mais serviços (de outro modo as condições (3-7) seriam violadas). No cálculo de g_i^t será necessário ter em conta o custo fixo do serviço que se irá fechar, mas também os custos fixos dos serviços que se irão abrir, como se mostra no Algoritmo 3-22.

Algoritmo 3-22

PROCEDIMENTO DE CÁLCULO DE g_i^t

t e $i \in I_t^*$ - período temporal e serviço em relação aos quais se pretende calcular g_i^t . Q_i - capacidade máxima do serviço i ; Q_i' - capacidade mínima do serviço i .

1. $Dif \leftarrow D - (C_{\max} - Q_i)$; $g_i^t \leftarrow f_i^t$; $I(i, t) \leftarrow \emptyset$; $Cap \leftarrow 0$.
 2. Calcular $F_i^u = \frac{F_i^t}{Q_i} \left[\frac{\phi_i}{Q_i} \right]$, $\forall i \in I_t^*$, com $\phi_i = \begin{cases} Dif, & \text{se } Q_i < Dif \\ Q_i, & \text{caso contrário} \end{cases}$.
 3. Se $\exists i' \in I_t^* : Cap + Q_{i'}' < Q_{i'}$, então $g_i^t \leftarrow +\infty$ e parar; se não, ir para 4.
 4. Considerar $i' \in I_t^*$, tal que $F_{i'}^u = \min_{\substack{i \in I_t^* \cup I(i, t) \\ Cap + Q_{i'}' < Q_{i'}}} \{F_{i'}^u\}$.
 5. $I(i, t) \leftarrow I(i, t) \cup \{i'\}$; $g_i^t \leftarrow g_i^t + F_{i'}^u$; $Dif \leftarrow Dif - Q_{i'}$; $Cap \leftarrow Cap + Q_{i'}'$.
 6. Se $Dif \leq 0$, então parar; se não, ir para 2.
-

O Algoritmo 3-20 não garante o cálculo de uma solução admissível em todas as situações. De facto, no passo 12 do algoritmo, se $g_i^t = +\infty, \forall i \in I_t^*$, então não será possível construir uma solução admissível. O valor de g_i^t toma o valor $+\infty$ em casos em que não é possível construir um conjunto $I(i, t)$ tal que o somatório das capacidades mínimas dos serviços pertencentes a este conjunto é menor do que $Q_{i'}$. Apesar de tal situação ser possível de ocorrer, não mostrou ser um problema em todos os testes computacionais feitos.

3.2.2.3. Relaxação Lagrangeana

Se se optar por relaxar lagrangeanamente a(s) restrição(ões) de capacidade, vai obter-se o PLDAFR. Os bons resultados com a heurística primal-dual para o PLDAFR motivaram o desenvolvimento de heurísticas que se baseiam no método do subgradiente, mas que resolvem, em cada iteração, o problema relaxado de forma heurística, em vez de óptima.

Considerem-se apenas as restrições de capacidade máxima, e associem-se a estas restrições os multiplicadores λ_i^t . O problema obtido é o PLDAFR com a

função objectivo (3-8).

$$\sum_i \sum_j (c'_{ij} + \lambda'_i d'_j) + \sum_i \sum_{\xi=i}^T \left(FA_{ii}^{\xi} - Q_i \sum_{\tau=i}^{\xi} \lambda_i^{\tau} \right) a_{i\tau}^{\xi} + \sum_i \sum_{\xi=i}^T \left(FR_{ii}^{\xi} - Q_i \sum_{\tau=i}^{\xi} \lambda_i^{\tau} \right) r_{i\tau}^{\xi} \quad (3-8)$$

Em cada iteração do algoritmo, o PLDAFR é resolvido, utilizando a heurística primal-dual. Note-se que o valor da função objectivo dual calculado pela heurística é um limite inferior válido e que pode ser utilizado em vez de se optar por resolver de forma óptima o PLDAFR (com vantagens evidentes em relação aos tempos computacionais necessários, sendo que a qualidade da solução dual é, em geral, muito boa). A solução primal admissível para o PLDAFR pode não ser quando se introduzem as restrições de capacidade. Por esse motivo, e no caso de não admissibilidade, a solução primal é alterada, utilizando os procedimentos primais descritos neste capítulo. Em cada iteração o valor de G_i^t é calculado como em (3-9).

$$G_i^t = \sum_j d'_j x'_{ij} - Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) \quad (3-9)$$

O valor do multiplicador é actualizado como em (3-11), sendo o valor de π calculado como em (3-10), em que Z_{UB} e Z_{LB} representam, respectivamente, os melhores limites superior e inferior conhecidos. O primeiro corresponde à melhor solução primal encontrada até ao momento. O segundo corresponde ao melhor valor da função objectivo dual do PLDAFR calculada até ao momento. O valor de π é inicializado a 2, e o seu valor é actualizado para metade sempre que se executam 5 iterações sem que haja melhorias no valor de Z_{LB} . A

heurística termina quando $\frac{Z_{UB} - Z_{LB}}{Z_{LB}} < 0.001$, ou quando $\pi < 0.005$.

$$\pi = \frac{\pi(Z_{UB} - Z_{LB})}{\sum G_i^2} \quad (3-10)$$

$$\lambda_i^t = \max(0, \lambda_i^t + \pi G_i^t) \quad (3-11)$$

A introdução das restrições de capacidades mínimas obriga a que se considere um outro multiplicador que terá um comportamento simétrico em relação a λ_i^t .

3.2.2.4. Resultados Computacionais para Problemas com Restrições de Capacidades Máxima e Mínima

3.2.2.4.1. Descrição dos Testes

Tal como no caso dos problemas sem restrições de capacidades, as heurísticas primais-duais foram testadas através de um conjunto de problemas de teste gerados aleatoriamente. Foram considerados valores para n , m e T (com $n > m$) indicados na Tabela 3.9. Para cada combinação considerada, cinco problemas foram gerados num total de 270 que consideram apenas restrições de capacidade máxima, e outros 270, que consideram simultaneamente restrições

de capacidades máxima e mínima.

n	25	50	100	200	500
m	5	10	20	50	
T	5	10	20		

Tabela 3.9 – Geração de problemas de teste

Os problemas foram gerados como descrito em 3.2.1.11.1. As capacidades máximas associadas a cada possível localização de um serviço foram geradas como descrito em Saldanha-da-Gama, 2002. As capacidades mínimas são aleatoriamente geradas, e estão compreendidas entre 35% a 85% da capacidade máxima do serviço. Todos os testes computacionais foram executados num Pentium 4, 1.80 Ghz, com sistema operativo Windows 2000, memória virtual de 2000 MB e 260Mb de Ram. As heurísticas foram programadas em linguagem C e utilizando o compilador Microsoft Visual C++. O desempenho das heurísticas foi comparado com o desempenho do Cplex, versão 7.0. O Cplex termina sem conseguir calcular a solução óptima quando se verifica uma das seguintes situações: *i*) ultrapassa os 2 100 000 000 nodos na árvore de *branch-and-bound*; *ii*) o número de iterações do simplex ultrapassa o mesmo número; *iii*) não existe memória suficiente; *iv*) o tempo de execução excede os 200 000 segundos. Tal como foi descrito no caso do problema sem restrições de capacidade, também nestes problemas se optou por executar uma heurística de pesquisa local, partindo da melhor solução primal obtida pela heurística primal-dual.

3.2.2.4.2. Resultados

Tal como no caso do problema sem capacidades, para cada (n, m, T) , cinco problemas foram gerados e resolvidos. As tabelas mostram o melhor, o pior e o valor médio obtido. Os valores percentuais apresentados são calculados como descrito em 3.2.1.11.2. A Tabela 3.10 mostra os resultados obtidos relativos à qualidade da solução calculada pela heurística primal-dual, e após a execução da pesquisa local, quando se consideram apenas restrições de capacidades máximas (no primeiro grupo de problemas teste consideraram-se as capacidades mínimas iguais a zero para todos os serviços). A mesma tabela mostra também os resultados obtidos quando a heurística lagrangeana foi executada e quando se optou por inicializar as variáveis duais através da resolução de um problema de programação linear. Este problema de programação linear resulta do problema dual quando se consideram todas as variáveis duais iguais a zero à excepção das variáveis v'_j e λ'_i (Saldanha-da-Gama, 2002). A Tabela 3.11 mostra a qualidade do limite inferior calculado por todas as heurísticas. A Tabela 3.12 mostra a qualidade das soluções obtidas nos problemas em que se consideraram restrições de capacidades máxima e mínima. A Tabela 3.13 mostra a qualidade dos limites inferiores calculados pelas heurísticas. A Tabela 3.14 e a Tabela 3.15 mostram os tempos computacionais, em segundos, dispendidos pelas heurísticas e pelo Cplex na resolução de problemas com capacidades máximas e com capacidades máximas e mínimas, respectivamente. A linha “Resumo de Valores”,

que se apresenta em cada tabela, mostra o menor valor de todos os valores apresentados na coluna “Melhor”, a média dos valores apresentados na coluna “Média”, e o maior valor de todos os apresentados na coluna “Pior”. É de notar que, para problemas com restrições de capacidades máximas, o Cplex não foi capaz de resolver uma das cinco instâncias com (T, n, m) igual a $(10, 500, 50)$, nenhuma das instâncias com (T, n, m) igual a $(20, 100, 50)$, uma instância com (T, n, m) igual a $(10, 200, 10)$, nenhuma das instâncias com (T, n, m) igual a $(20, 200, 50)$, duas instâncias com (T, n, m) igual a $(20, 500, 10)$, e nenhuma das instâncias com (T, n, m) igual a $(20, 500, 20)$ e $(20, 500, 50)$. Se se considerarem restrições de capacidade máxima e mínima, então o Cplex não consegue encontrar soluções admissíveis para todos os problemas com (T, n, m) igual a $(20, 100, 50)$, $(20, 200, 50)$, $(20, 500, 20)$ e $(20, 500, 50)$. Para problemas com (T, n, m) igual a $(20, 500, 10)$ o Cplex não consegue resolver uma instância. Foram executados mais alguns testes computacionais com o objectivo de comparar o desempenho da heurística primal-dual e o desempenho do Cplex, quando este último é utilizado como um procedimento heurístico. As regras de paragem escolhidas para terminar a execução do Cplex foram as seguintes: tempo máximo igual a 15000 e 35000 segundos, para problemas com restrições de capacidade máxima e restrições de capacidade máxima e mínima, respectivamente (estes valores são superiores ao maior tempo computacional gasto pela heurística), e qualidade da solução primal igual à qualidade média das soluções calculadas para a heurística, em problemas de igual dimensão⁵.

Foi calculada a relação existente entre os tempos computacionais médios obtidos, da seguinte forma:

$$\frac{\text{Tempo computacional médio do Cplex}}{\text{Tempo computacional médio da heurística primal-dual + pesquisa local}}$$

Os resultados são apresentados nas Tabela 3.16 e Tabela 3.17. Em alguns casos, o Cplex não é capaz de encontrar soluções primais com a qualidade pretendida, dentro do limite de tempo estabelecido. Para problemas com restrições de capacidade máxima, isto acontece em problemas com (m,n,T) igual a $(10,500,50)$ e $(20,100,50)$. Para problemas com (m,n,T) igual a $(20,200,20)$, é apenas capaz de calcular uma solução primal para um dos cinco problemas, e com (m,n,T) igual a $(20,500,5)$ e $(20,500,10)$, não consegue resolver um dos cinco problemas. O Cplex necessita, em média, de tempos computacionais dez vezes superiores aos tempos gastos pela heurística primal-dual, para chegar a soluções de qualidade semelhante. Nos problemas em que se consideram simultaneamente restrições de capacidade máxima e mínima, o Cplex não consegue resolver nenhum dos problemas com (m,n,T) igual a $(10,500,50)$ e $(20,100,50)$. Os tempos computacionais gastos pelo Cplex são, em média, mais de cinco vezes superiores aos gastos pela heurística, na situação em que ambos os procedimentos tentam encontrar soluções com qualidade semelhante.

⁵ A fórmula utilizada pelo Cplex para calcular a qualidade da solução primal é ligeiramente diferente da utilizada neste trabalho, mas tais diferenças foram tidas em conta na execução destes testes.

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando LP			Heurística com inicialização usando LP + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	25	5	0.00	0.09	0.28	0.00	0.06	0.28	0.00	2.42	5.21	0.00	1.04	5.21	0.00	0.09	0.28	0.00	0.00	0.00
5	25	10	0.36	3.20	11.51	0.00	1.85	6.11	0.00	1.77	5.41	0.00	1.30	3.57	0.00	0.59	1.92	0.00	0.00	0.00
5	25	20	1.27	2.53	3.43	0.85	1.74	3.05	0.68	1.77	3.17	0.68	1.51	2.07	0.00	0.32	0.85	0.00	0.24	0.64
5	50	5	0.00	1.68	6.43	0.00	0.07	0.33	0.00	3.26	13.94	0.00	1.40	7.01	0.00	0.52	0.26	0.00	0.00	0.00
5	50	10	0.00	1.08	2.78	0.00	0.92	2.66	0.00	2.60	5.85	0.00	0.50	1.39	0.00	0.12	0.62	0.00	0.12	0.62
5	50	20	2.82	3.54	6.08	2.12	3.33	6.08	2.08	2.98	4.12	0.30	0.91	1.97	0.00	0.43	1.53	0.00	0.13	0.38
5	100	5	0.00	1.42	7.09	0.00	0.56	2.79	0.00	3.29	9.52	0.00	1.07	4.56	0.00	0.00	0.00	0.00	0.00	0.00
5	100	10	0.00	1.50	4.74	0.00	1.01	2.84	1.18	3.33	6.25	0.06	1.73	3.86	0.00	0.03	0.03	0.01	0.03	0.03
5	100	20	1.77	2.48	4.00	0.28	1.74	3.85	0.66	1.93	4.64	0.66	1.17	2.76	0.00	0.09	0.22	0.00	0.08	0.22
5	100	50	0.79	2.13	3.16	0.36	1.43	2.91	1.37	3.11	5.81	0.78	1.28	2.11	0.16	0.35	0.56	0.15	0.29	0.48
5	200	5	0.00	0.32	1.30	0.00	0.32	1.30	0.50	1.64	4.64	0.00	0.85	4.24	0.00	0.04	0.22	0.00	0.00	0.00
5	200	10	0.00	1.27	3.33	0.00	0.60	2.25	0.00	3.25	6.38	0.00	0.83	2.07	0.00	0.28	0.83	0.00	0.12	0.41
5	200	20	0.82	2.86	4.71	0.37	1.89	2.97	1.63	4.07	6.65	0.00	2.38	6.65	0.00	0.08	0.16	0.00	0.08	0.16
5	200	50	1.55	2.75	3.77	1.55	2.13	3.21	1.18	2.92	4.05	0.13	1.62	3.76	0.00	0.34	0.57	0.00	0.32	0.48
5	500	5	0.00	1.19	2.85	0.00	0.57	2.85	0.00	1.33	4.45	0.00	0.44	2.22	0.00	0.00	0.00	0.00	0.00	0.00
5	500	10	0.00	1.18	3.22	0.00	1.18	3.22	1.77	4.41	6.89	0.00	2.28	4.74	0.00	0.03	0.15	0.00	0.03	0.15
5	500	20	0.78	1.80	3.71	0.21	0.80	1.16	0.58	2.83	4.82	0.00	0.68	1.47	0.00	0.07	0.18	0.00	0.03	0.17
5	500	50	1.41	4.00	5.85	0.81	3.44	5.85	1.36	2.55	3.49	1.11	1.91	2.95	0.00	0.41	1.54	0.00	0.24	0.77
10	25	10	0.00	2.68	4.64	0.00	2.39	4.64	0.42	4.41	4.10	0.00	0.44	1.38	0.00	0.38	1.22	0.00	0.24	1.22
10	25	20	0.00	1.31	4.97	0.00	0.61	1.74	0.00	2.50	4.10	0.00	1.08	1.98	0.00	0.13	0.57	0.00	0.11	0.48
10	25	50	0.50	3.51	7.30	0.25	2.60	5.46	0.46	1.74	2.95	0.08	1.43	2.81	0.00	0.74	1.57	0.00	0.64	1.39
10	50	5	0.00	3.03	9.71	0.00	1.28	4.61	0.00	0.62	1.74	0.00	0.49	1.74	0.00	0.07	0.32	0.00	0.06	0.32
10	50	10	0.83	2.77	4.49	0.83	1.92	2.86	0.74	0.94	13.27	0.00	0.94	3.12	0.00	0.25	0.74	0.00	0.08	0.42
10	50	20	0.97	1.54	2.10	0.97	1.52	2.10	1.92	2.62	3.91	0.38	1.19	2.02	0.02	0.72	2.42	0.02	0.22	0.50
10	100	5	0.00	2.14	4.73	0.00	1.55	4.72	0.00	2.65	7.75	0.00	0.50	2.51	0.00	0.40	1.14	0.00	0.23	0.65
10	100	10	0.00	4.08	5.21	1.62	3.48	4.98	0.58	1.88	3.62	0.58	0.68	1.99	0.40	0.60	0.78	0.17	0.45	0.73
10	100	20	3.10	3.10	3.10	0.00	0.24	0.98	0.00	2.42	3.62	0.00	1.99	3.62	0.40	0.60	0.78	0.00	0.48	1.60
10	100	50	2.01	2.95	3.95	0.71	1.30	1.87	2.40	3.01	4.30	0.45	1.42	2.51	0.10	0.24	0.39	0.02	0.16	0.27
10	200	5	0.00	1.08	3.03	0.00	0.89	1.51	0.00	1.57	3.59	0.00	1.14	3.47	0.00	0.56	1.95	0.00	0.31	0.85
10	200	10	0.74	1.61	3.22	0.13	1.30	3.22	0.12	1.57	1.40	0.08	0.25	0.78	0.00	0.42	1.27	0.00	0.10	0.17
10	200	20	2.33	2.33	4.23	0.19	1.26	3.00	1.06	2.02	2.74	0.79	1.33	2.61	0.04	0.82	2.25	0.00	0.19	0.49
10	200	50	3.33	4.58	5.10	0.98	3.42	5.10	2.04	4.27	6.33	1.62	2.98	4.53	0.40	0.78	1.30	0.25	0.45	0.66

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Primal-Dual		Heurística Primal-Dual + Pesquisa Local		Heurística com inicialização usando LP		Heurística com inicialização usando LP + Pesquisa Local		Heurística Lagrangeana		Heurística Lagrangeana + Pesquisa Local							
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>			
10	500	5	0.14	0.86	1.41	0.14	0.81	1.41	0.00	1.60	3.90	0.00	0.24	1.21	0.00	0.07	0.33	0.00	0.00	0.00
10	500	10	0.00	1.29	2.34	0.00	0.69	1.90	0.16	1.23	2.48	0.00	0.59	1.89	0.00	0.16	0.58	0.00	0.06	0.16
10	500	20	0.76	2.14	4.93	0.76	1.28	2.12	0.91	2.02	4.29	0.29	0.97	1.68	0.00	0.51	0.88	0.00	0.21	0.49
10	500	50	3.99	5.00	6.11	2.46	3.86	6.11	2.19	4.12	6.01	1.28	2.28	4.39	0.18	0.41	0.75	0.14	0.24	0.51
20	25	5	0.00	1.02	3.24	0.00	0.45	1.52	0.00	1.95	4.09	0.00	1.47	3.88	0.00	0.73	2.44	0.00	0.27	0.84
20	25	10	0.67	2.86	5.54	0.15	1.07	2.37	1.38	3.41	5.81	0.62	1.35	2.69	0.45	1.36	2.17	0.00	0.62	1.38
20	25	20	3.19	4.41	4.86	0.84	2.25	4.56	3.03	3.61	4.92	0.39	1.09	1.76	1.37	3.01	4.42	0.69	1.29	2.29
20	50	5	0.00	0.89	2.17	0.00	0.09	0.43	0.07	1.93	6.55	0.00	0.77	3.04	0.03	0.91	2.10	0.00	0.57	2.03
20	50	10	0.98	5.04	11.48	0.98	3.27	6.68	1.38	2.74	4.29	0.58	1.51	2.02	0.56	1.58	3.79	0.26	0.98	2.24
20	50	20	2.51	4.21	6.91	1.03	2.20	3.31	2.71	6.83	11.07	0.94	3.69	11.07	0.79	1.88	4.15	0.30	0.95	2.14
20	100	5	0.00	2.29	4.47	0.00	0.86	1.77	0.00	2.21	5.30	0.00	1.30	3.85	0.00	0.89	2.34	0.00	0.34	0.87
20	100	10	2.89	5.19	6.67	0.75	2.81	5.88	1.01	2.97	4.36	0.93	1.77	4.36	0.07	1.46	2.78	0.00	0.49	0.77
20	100	20	4.58	4.51	6.11	1.62	2.53	3.56	2.99	3.94	5.62	1.21	1.49	3.58	1.03	2.65	4.60	0.74	1.38	2.21
20	100	50	4.58	6.33	7.81	1.72	5.13	7.81	2.12	4.07	5.78	1.21	1.89	3.62	0.76	3.19	4.25	0.66	2.80	4.25
20	200	5	0.00	2.10	6.88	0.00	0.55	1.34	0.00	2.46	9.05	0.00	1.11	3.98	0.00	0.49	1.21	0.00	0.49	1.21
20	200	10	1.65	3.21	5.78	0.63	1.98	3.35	0.97	2.99	6.27	0.19	0.78	1.55	0.00	1.96	3.92	0.00	0.81	1.72
20	200	20	2.54	3.40	4.87	0.52	2.50	4.87	1.06	2.63	4.03	0.61	2.09	3.07	2.09	2.38	2.89	0.42	1.13	2.18
20	200	50	5.24	6.54	8.64	3.32	4.47	6.96	4.80	5.72	7.05	1.73	2.75	4.80	1.78	3.30	5.95	1.02	2.25	4.09
20	500	5	0.05	2.40	6.98	0.05	1.15	1.77	0.00	0.82	2.19	0.00	0.66	2.19	0.02	1.74	3.39	0.02	1.15	2.76
20	500	10	1.88	3.52	5.41	0.58	2.12	3.77	0.60	3.07	5.14	0.59	1.61	2.17	0.48	1.32	2.35	0.17	1.01	1.80
20	500	20	3.41	6.88	9.48	2.09	3.30	5.03	3.34	4.64	6.41	1.42	2.22	2.86	1.86	2.72	3.70	1.22	1.73	2.47
20	500	50	6.27	7.45	9.17	2.08	3.96	6.63	3.65	4.99	6.49	1.69	2.91	5.92	2.11	3.22	5.12	0.84	1.28	1.96
Resumo dos Valores			0.00	2.80	11.51	0.00	1.75	7.81	0.00	2.80	13.94	0.00	1.36	11.07	0.00	0.85	5.95	0.00	0.47	4.25

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Primal-Dual			Heurística com inicialização usando IP			Heurística Lagrangeana		
			<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	0.09	3.64	7.37	1.62	4.68	7.63	0.09	1.13	3.84
5	25	10	5.81	7.27	8.75	2.66	3.77	5.40	0.31	2.27	6.91
5	25	20	6.43	7.97	9.35	1.97	2.44	2.81	0.47	1.39	3.64
5	50	5	0.00	6.61	15.07	2.14	6.75	9.27	0.00	2.34	5.81
5	50	10	1.94	5.37	9.19	2.14	3.37	5.57	0.20	1.44	2.93
5	50	20	4.86	6.74	9.97	1.59	2.83	5.04	0.29	3.06	11.77
5	100	5	0.34	4.44	12.46	1.39	5.68	8.93	0.10	1.40	2.50
5	100	10	2.14	4.80	7.62	1.22	4.17	6.82	0.13	1.49	2.85
5	100	20	6.54	7.60	8.46	2.35	2.93	3.28	0.69	1.04	1.20
5	100	50	5.78	7.90	9.83	1.35	2.73	5.16	0.57	0.93	1.17
5	200	5	0.00	4.64	9.18	2.96	5.38	9.61	0.00	1.66	4.65
5	200	10	2.24	7.02	9.37	3.04	4.08	4.90	0.59	2.05	3.30
5	200	20	6.18	7.26	8.15	3.35	3.72	4.27	0.55	1.00	1.91
5	200	50	7.99	8.93	9.95	1.69	2.80	3.78	0.40	1.79	3.63
5	500	5	1.12	5.77	12.61	2.33	4.13	5.75	0.27	1.66	2.44
5	500	10	4.01	6.50	8.73	3.36	4.19	5.53	0.57	4.35	12.29
5	500	20	3.98	7.29	9.25	2.38	3.13	4.85	0.39	1.01	1.64
5	500	50	8.48	9.88	12.05	2.30	3.02	3.60	0.67	2.40	3.93
10	25	5	1.35	2.79	5.21	2.48	3.71	4.90	1.10	1.71	2.97
10	25	10	2.28	3.32	4.49	1.31	2.40	3.46	0.72	1.08	1.56
10	25	20	5.53	6.91	9.10	2.43	2.80	3.19	0.95	3.19	7.44
10	50	5	0.31	2.89	7.92	3.53	4.34	5.37	0.22	0.89	1.79
10	50	10	1.38	4.00	5.49	2.28	3.11	4.73	0.13	1.11	1.83
10	50	20	2.41	4.77	6.03	1.46	2.69	3.79	0.61	2.86	10.38
10	100	5	2.74	5.17	8.80	4.22	5.68	6.98	1.36	2.40	3.39
10	100	10	0.57	3.53	6.70	3.29	4.27	5.39	0.57	1.14	1.70
10	100	20	3.04	4.22	5.81	2.50	2.84	3.19	0.76	1.15	1.87
10	100	50	4.28	5.52	6.81	1.78	2.20	2.90	0.40	0.64	0.80
10	200	5	3.01	4.65	6.82	4.52	6.48	8.12	1.43	2.10	2.81
10	200	10	2.67	3.67	5.24	3.92	4.71	5.25	1.24	1.74	2.36
10	200	20	1.44	3.62	5.51	2.51	3.60	4.21	0.66	1.01	1.53
10	200	50	4.76	5.57	6.18	2.14	2.64	3.18	0.43	0.80	1.22
10	500	5	3.44	5.41	7.10	3.42	4.80	6.85	0.85	1.79	3.77
10	500	10	1.17	3.45	6.47	2.40	4.75	5.78	0.33	1.00	1.82
10	500	20	2.85	3.98	5.53	3.67	4.14	4.68	0.64	0.86	1.25
10	500	50	4.75	6.31	7.22	2.73	3.33	3.53	0.47	0.75	1.38
20	25	5	0.00	0.95	2.51	1.48	2.71	4.22	0.00	0.70	1.62
20	25	10	0.88	2.56	3.73	1.94	2.27	2.60	0.63	1.00	1.47
20	25	20	3.82	5.47	7.72	2.28	3.05	4.41	4.14	6.42	8.56
20	50	5	0.98	3.02	5.04	2.92	3.75	4.18	0.67	1.27	2.06
20	50	10	2.19	3.58	5.20	3.02	3.25	3.55	1.26	1.58	2.45
20	50	20	2.59	4.21	4.81	2.76	3.12	3.65	1.59	2.17	2.92
20	100	5	0.29	2.12	4.14	2.47	4.22	5.82	0.20	1.32	3.35
20	100	10	3.35	3.81	4.05	2.58	3.69	4.89	0.99	1.53	2.19
20	100	20	2.18	3.55	5.37	3.66	4.10	4.46	0.71	1.63	3.38
20	100	50	5.06	6.70	8.15	3.31	4.07	4.83	1.88	3.18	5.47
20	200	5	0.05	2.01	4.26	1.66	3.88	5.17	0.05	1.26	1.88
20	200	10	1.03	2.98	5.31	2.84	4.34	5.23	0.51	1.59	2.91
20	200	20	2.50	3.11	3.78	3.56	4.34	4.83	0.93	1.37	1.85
20	200	50	5.01	5.63	6.28	3.74	4.21	4.48	1.01	1.99	3.93
20	500	5	0.46	1.97	4.68	1.64	2.54	4.11	0.08	0.84	1.62
20	500	10	1.80	2.58	2.94	3.99	4.38	4.71	0.95	1.59	2.06
20	500	20	2.64	3.33	3.84	4.58	5.35	6.09	1.20	1.66	2.41
20	500	50	3.43	5.11	6.06	4.55	5.44	6.38	0.83	1.25	1.85
Resumo dos Valores			0.00	4.85	15.07	1.22	3.83	9.61	0.00	1.68	12.29

Tabela 3.11 – Qualidade do limite inferior – restrições de capacidades máximas

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com Inicialização usando LP			Heurística com Inicialização usando LP + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			
			Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	
5	25	5	0,00	4,08	8,64	0,00	1,39	6,96	0,00	9,25	34,16	0,00	7,17	34,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	25	10	0,00	1,70	4,37	0,00	1,28	4,32	0,00	2,31	3,97	0,00	2,09	3,55	0,00	0,14	0,69	0,00	0,14	0,69	0,00
5	125	20	1,50	2,42	3,13	1,25	1,99	2,98	0,09	4,61	11,92	0,09	1,75	6,01	0,01	0,34	1,09	0,01	0,33	1,08	0,00
5	50	5	0,00	1,21	5,05	0,00	0,00	0,00	0,00	0,57	1,86	0,00	0,37	1,86	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	50	10	0,00	2,42	8,10	0,00	1,98	5,94	0,00	8,80	32,71	0,00	3,64	14,00	0,00	0,03	0,13	0,00	0,03	0,13	0,00
5	50	20	0,77	1,29	1,78	0,01	0,98	1,76	1,60	2,41	3,69	0,14	1,17	1,94	0,04	0,98	3,70	0,00	0,69	3,32	0,00
5	100	5	0,00	1,91	5,73	0,00	0,80	5,73	3,23	9,05	17,26	0,00	2,63	6,8	0,00	0,39	1,81	0,00	0,02	1,12	0,00
5	100	10	0,00	2,64	4,70	0,00	0,70	2,98	2,03	4,67	6,85	0,00	2,49	5,26	0,00	0,64	2,15	0,00	0,58	2,14	0,00
5	100	20	0,82	4,09	5,49	0,14	0,14	4,90	2,24	3,27	4,31	0,14	2,09	4,31	0,00	0,17	0,59	0,00	0,02	0,71	0,00
5	100	50	1,92	3,08	4,71	0,80	1,62	2,57	1,35	2,99	15,66	0,52	0,76	15,66	0,08	0,50	0,89	0,03	0,31	0,54	0,00
5	200	5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	4,08	15,66	0,00	3,26	15,66	0,00	0,14	0,71	0,00	0,14	0,71	0,00
5	200	10	0,03	2,50	4,60	0,03	1,28	3,28	0,07	2,44	5,15	0,04	1,5	5,15	0,00	0,39	1,44	0,00	0,09	0,39	0,00
5	200	20	1,13	2,59	5,24	0,32	1,73	4,35	2,10	5,43	9,19	0,60	1,69	4,43	0,00	0,47	1,46	0,00	0,08	0,26	0,00
5	200	50	1,29	2,08	2,59	1,05	1,61	2,59	1,47	2,95	4,35	0,47	2,15	4,24	0,00	0,71	1,29	0,09	0,49	1,04	0,00
5	500	5	0,00	1,45	6,67	0,00	1,45	6,67	0,00	1,17	4,16	0,00	0,94	4,15	0,00	0,34	1,16	0,00	0,11	0,53	0,00
5	500	10	0,00	1,45	5,39	0,00	1,15	3,88	0,51	3,30	10,01	0,01	3,13	10,01	0,00	0,16	0,44	0,00	0,00	0,01	0,00
5	500	20	0,00	1,10	2,17	0,00	0,25	0,66	0,00	6,72	13,6	0,00	3,53	11,82	0,00	0,72	1,81	0,00	0,36	0,80	0,00
5	500	50	0,00	1,45	6,67	0,00	1,45	6,67	0,00	3,48	6,76	1,11	2,09	3,37	0,14	0,53	1,08	0,01	0,30	0,87	0,00
10	25	5	0,20	0,61	1,29	0,00	0,49	1,29	0,28	16,41	46,70	0,00	12,12	46,70	0,20	0,46	1,16	0,00	0,32	1,16	0,00
10	25	10	0,96	3,08	5,31	0,53	2,83	4,83	1,17	3,90	6,22	0,26	1,30	2,64	0,02	0,68	1,57	0,00	0,52	0,97	0,00
10	25	20	0,74	3,21	5,74	0,12	1,89	3,36	2,25	3,15	4,03	0,36	1,46	2,74	0,52	1,27	2,04	0,48	0,85	1,47	0,00
10	50	5	0,00	2,57	5,93	0,00	1,15	2,37	0,00	1,32	3,06	0,00	1,12	3,06	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10	50	10	2,80	3,90	6,76	0,20	2,31	5,04	1,73	8,69	25,64	0,00	9,46	25,64	0,01	1,15	2,42	0,01	0,71	2,24	0,00
10	50	20	1,17	2,51	4,29	0,33	1,54	3,59	1,86	4,75	8,22	0,95	1,94	8,22	0,45	0,88	1,24	0,01	0,87	0,95	0,00
10	100	5	0,00	2,70	8,62	0,00	0,54	2,69	0,00	10,72	41,15	0,00	5,28	22,84	0,00	1,54	4,48	0,00	0,84	2,51	0,00
10	100	10	2,45	9,78	30,23	1,03	4,06	7,79	2,38	20,00	34,95	2,38	11,06	33,90	0,00	1,70	3,40	0,00	0,89	1,68	0,00
10	100	20	1,67	7,09	3,86	0,69	2,51	7,09	3,77	5,30	7,00	0,05	2,31	5,56	0,01	1,94	5,54	0,01	1,34	3,93	0,00
10	100	50	3,99	5,17	6,17	1,85	2,58	2,58	3,09	4,50	7,00	1,24	2,69	7,00	0,75	1,27	2,72	0,35	0,65	1,41	0,00
10	200	5	0,00	0,95	2,74	0,00	0,76	2,74	0,00	0,39	1,89	0,00	0,01	0,03	0,06	0,46	0,79	0,00	0,21	0,79	0,00
10	200	10	0,94	2,83	4,19	0,94	2,15	4,19	2,19	6,09	12,31	0,06	3,32	12,31	0,13	0,83	1,28	0,01	0,66	1,23	0,00

Tabela 3.12 – Qualidade da solução primal (em percentagem) – restrições de capacidades máxima e mínima

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando LP			Heurística com inicialização usando LP + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local		
			Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior
10	200	20	1.42	5.98	10.96	1.15	4.77	10.96	4.38	24.64	40.85	1.34	24.03	40.85	0.50	1.59	2.76	0.05	0.74	1.70
10	200	50	2.69	4.50	6.22	0.91	3.05	6.22	1.60	3.36	5.03	0.65	2.23	3.06	0.84	1.03	1.26	0.39	0.63	0.91
10	500	5	0.00	1.28	4.62	0.00	0.43	1.36	0.00	1.13	3.94	0.00	0.48	1.84	0.00	0.00	0.01	0.00	0.00	0.01
10	500	10	0.44	1.60	3.66	0.44	1.44	3.66	0.14	2.03	7.98	0.09	0.65	1.75	0.01	0.84	2.23	0.00	0.36	1.11
10	500	20	0.13	2.60	4.78	0.13	1.82	4.78	2.67	5.62	11.85	0.19	3.63	11.85	0.44	0.83	1.69	0.07	0.32	0.68
10	500	50	3.46	4.41	5.67	3.46	4.22	4.71	5.42	8.42	12.27	1.69	6.32	12.27	1.05	1.39	1.65	0.48	0.71	0.89
20	25	5	0.00	24.69	64.63	0.00	14.70	40.32	3.19	12.42	25.66	2.35	9.63	21.61	0.33	2.64	8.26	0.33	1.33	3.25
20	25	10	3.56	9.20	16.20	1.85	6.91	16.20	2.74	19.95	55.32	0.18	12.23	33.60	0.24	2.29	1.95	0.11	1.47	4.42
20	25	20	4.46	9.02	12.38	2.25	5.03	7.30	3.76	11.57	23.55	0.79	5.32	14.03	1.45	2.29	3.25	0.68	1.15	1.76
20	50	5	7.85	45.70	81.78	3.60	41.71	72.54	15.59	47.77	77.18	13.17	46.10	77.18	0.00	5.49	11.35	0.00	1.51	3.40
20	50	10	0.40	14.94	42.42	0.24	10.86	42.42	1.36	25.93	54.27	1.36	9.61	23.29	0.84	1.90	4.06	0.32	0.92	1.66
20	50	20	1.23	7.56	21.09	1.20	5.31	18.59	4.48	13.34	24.39	0.77	9.98	24.39	1.55	5.57	18.66	0.41	4.54	18.17
20	100	5	0.03	3.44	13.08	0.00	2.83	10.66	0.01	2.08	4.64	0.01	1.62	3.69	0.03	4.74	10.28	0.00	2.20	5.40
20	100	10	1.01	33.15	67.68	1.01	22.49	67.68	0.39	31.65	63.00	0.35	19.76	43.83	1.02	4.38	7.42	0.18	2.38	4.26
20	100	20	2.87	23.58	83.41	1.66	19.74	83.41	3.41	42.87	109.33	0.90	42.57	109.33	1.75	4.80	11.37	0.79	1.23	2.48
20	100	50	2.41	3.64	4.57	1.56	3.12	4.57	4.23	7.10	12.41	1.11	2.14	3.56	1.75	2.66	3.26	0.83	1.41	1.95
20	200	5	0.00	8.60	34.25	0.00	8.43	34.25	0.00	11.69	34.25	0.00	10.13	34.25	0.00	1.82	3.56	0.00	0.58	1.64
20	200	10	0.68	19.93	44.37	0.11	6.43	12.10	6.13	18.87	43.18	1.60	12.31	43.18	2.05	5.30	9.51	0.81	2.26	4.52
20	200	20	3.33	23.02	46.20	1.01	6.07	12.59	10.01	33.31	55.48	2.59	27.98	55.48	0.77	5.70	10.09	0.50	1.99	4.62
20	200	50	6.46	7.42	8.60	4.58	7.04	8.60	8.58	21.27	48.82	4.94	19.75	48.82	5.02	5.67	6.34	4.11	4.37	5.04
20	500	5	0.01	0.70	1.74	0.01	0.30	1.00	0.00	2.42	9.64	0.00	1.68	6.69	0.01	1.54	5.75	0.00	1.09	4.98
20	500	10	1.79	13.16	34.18	1.17	6.65	13.62	0.32	2.89	4.60	0.30	2.41	4.29	1.40	4.21	6.96	0.36	1.80	4.35
20	500	20	2.32	23.12	34.70	1.43	5.76	15.21	5.73	28.96	60.14	4.78	10.01	21.32	5.72	8.86	11.29	2.80	4.86	6.58
20	500	50	5.98	25.56	42.45	3.54	6.23	17.23	7.80	32.12	48.13	6.20	12.78	19.37	6.25	9.70	12.38	5.50	6.27	7.98
Resumo dos Valores			0.00	7.37	83.41	0.00	4.38	83.41	0.00	10.62	109.33	0.00	7.18	109.33	0.00	1.92	18.66	0.37	1.02	18.17

Tabela 3.12 – Qualidade da solução primal (em porcentagem) – restrições de capacidades máxima e mínima (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Primal-Dual			Heurística com inicialização usando LP			Heurística Lagrangeana		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	1.27	4.91	11.60	1.54	3.46	4.65	1.27	2.81	5.21
5	25	10	2.59	5.05	9.87	0.96	3.18	5.48	2.81	5.01	9.47
5	25	20	4.83	5.96	7.42	1.80	2.60	3.54	5.65	6.54	8.11
5	50	5	0.92	6.33	15.71	1.08	4.94	9.93	0.92	5.54	15.71
5	50	10	2.16	3.21	4.91	2.33	3.72	4.73	2.24	3.49	4.83
5	50	20	5.65	7.19	8.16	2.15	2.52	2.93	5.05	6.16	6.99
5	100	5	2.53	5.45	9.33	4.72	5.60	7.29	1.03	4.69	7.84
5	100	10	2.89	7.10	9.40	2.49	4.50	6.77	1.62	6.37	10.19
5	100	20	5.01	6.33	7.58	1.11	2.05	2.63	5.08	5.84	6.99
5	100	50	6.31	7.16	8.30	1.65	1.86	2.24	4.72	5.56	6.59
5	200	5	1.22	3.92	7.09	3.93	5.24	6.60	1.16	3.87	6.66
5	200	10	5.47	7.24	9.41	3.73	4.71	5.88	4.19	5.44	6.93
5	200	20	5.33	7.68	11.83	1.83	2.90	3.57	5.09	6.85	10.65
5	200	50	7.14	7.87	9.52	1.81	2.18	2.85	5.33	6.70	9.80
5	500	5	0.70	4.10	7.91	2.35	4.31	6.42	0.70	3.92	7.37
5	500	10	2.28	6.20	10.20	3.42	4.21	4.95	2.29	4.97	6.37
5	500	20	3.54	7.72	13.10	2.22	4.88	11.02	3.53	8.08	15.51
5	500	50	7.94	10.18	12.62	2.18	2.80	3.61	4.30	5.46	7.46
10	25	5	2.48	4.96	8.02	3.32	4.61	5.75	0.00	3.63	6.65
10	25	10	2.87	5.13	8.15	2.17	3.59	5.73	4.24	6.19	9.92
10	25	20	2.66	4.44	6.26	2.08	2.70	3.63	3.04	5.40	8.34
10	50	5	0.00	4.07	8.32	1.69	2.85	4.96	1.38	4.56	8.16
10	50	10	1.90	3.54	5.47	3.51	3.83	4.36	1.84	3.60	6.27
10	50	20	2.34	4.28	6.06	1.90	2.51	2.85	2.32	5.05	7.92
10	100	5	2.65	4.59	7.09	2.75	5.96	7.48	2.64	4.59	7.09
10	100	10	2.10	3.84	5.01	4.16	5.44	6.68	1.69	3.81	5.35
10	100	20	1.96	4.18	6.92	2.25	2.98	3.78	1.97	4.15	6.55
10	100	50	4.30	5.47	6.77	1.86	2.31	2.75	5.81	7.51	10.01
10	200	5	2.52	4.74	7.77	3.59	5.32	6.44	2.52	4.74	7.77
10	200	10	2.77	4.83	6.89	3.40	4.74	6.50	2.70	5.03	6.89
10	200	20	3.93	4.83	7.03	2.42	3.68	4.73	4.00	5.01	7.41
10	200	50	4.21	5.70	6.35	1.73	2.21	2.57	4.48	7.22	9.23
10	500	5	2.35	5.16	9.64	2.17	3.31	5.21	2.20	4.39	9.64
10	500	10	1.00	3.05	5.32	2.46	4.22	5.29	1.00	3.19	5.35
10	500	20	2.51	3.89	5.69	3.39	3.66	4.10	2.56	3.91	5.73
10	500	50	6.00	6.54	6.96	2.38	3.22	3.81	6.37	6.76	7.33
20	25	5	2.91	4.19	6.58	3.00	3.92	6.43	3.14	4.10	6.13
20	25	10	2.93	4.70	6.68	3.08	4.53	5.89	2.72	5.04	7.61
20	25	20	5.09	6.10	8.08	2.78	3.76	5.80	6.07	8.30	11.02
20	50	5	5.28	7.40	12.11	7.73	9.73	14.18	5.97	7.89	12.11
20	50	10	2.81	3.26	3.94	3.51	4.72	6.00	3.65	4.08	4.66
20	50	20	0.43	2.80	3.67	0.01	2.40	3.31	0.01	4.77	7.31
20	100	5	2.65	4.13	5.83	4.88	6.18	8.08	2.72	4.81	5.98
20	100	10	2.93	4.10	5.41	4.30	6.05	6.91	3.18	4.83	7.61
20	100	20	3.15	3.86	5.24	3.43	4.54	6.99	3.88	5.46	6.75
20	100	50	3.72	4.97	6.59	2.37	3.53	4.53	6.06	8.23	11.30
20	200	5	0.92	4.53	6.17	1.97	6.44	8.16	1.26	4.94	6.95
20	200	10	2.37	3.42	3.92	4.61	5.91	7.05	3.25	4.50	5.81
20	200	20	1.89	3.75	5.68	3.78	6.04	8.12	3.27	4.89	6.80
20	200	50	4.03	4.39	4.80	3.95	4.26	4.82	5.88	6.79	7.33
20	500	5	1.85	3.15	4.96	1.20	3.54	7.11	1.73	3.83	5.65
20	500	10	1.27	3.05	4.51	3.30	4.87	6.19	1.54	3.44	5.63
20	500	20	5.24	6.15	7.21	5.62	6.39	7.43	2.73	4.61	6.17
20	500	50	6.21	8.39	8.93	6.56	7.43	8.98	5.22	5.89	7.39
Resumo dos Valores			0.00	5.17	15.71	0.01	4.20	14.18	0.00	5.23	15.71

Tabela 3.13 – Qualidade do limite inferior – restrições de capacidades máximas e mínimas

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando LP			Heurística com inicialização usando LP + Pesquisa Local			
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	
5	25	5	0,00	0,01	0,02	0,00	0,02	0,03	0,05	0,06	0,17	0,56	0,11	0,22	0,58
5	25	10	0,02	0,03	0,05	0,05	0,10	0,19	0,06	0,10	0,10	0,12	0,11	0,29	0,50
5	25	20	0,05	0,10	0,16	0,17	0,32	1,00	0,55	0,32	0,32	0,91	0,55	0,79	1,36
5	50	5	0,00	0,02	0,05	0,02	0,08	0,13	0,09	0,19	0,19	0,41	0,11	0,33	0,63
5	50	10	0,05	0,07	0,09	0,14	0,21	0,31	0,14	0,20	0,20	0,31	0,47	0,76	1,16
5	50	20	0,13	0,19	0,36	0,66	0,74	0,94	0,36	0,44	0,44	0,64	1,59	2,79	3,94
5	100	5	0,03	0,06	0,11	0,06	0,18	0,25	0,17	0,17	0,44	0,95	0,20	0,83	1,36
5	100	10	0,11	0,16	0,31	0,20	0,53	0,81	0,23	0,65	0,65	0,95	1,69	2,94	2,94
5	100	20	0,42	0,61	0,75	1,22	3,63	8,53	0,84	1,32	1,32	1,69	4,58	6,07	7,97
5	100	50	2,03	3,02	3,72	5,98	25,99	53,56	2,08	2,50	2,50	3,31	20,02	60,38	76,73
5	200	5	0,05	0,18	0,52	0,05	0,32	0,81	0,30	1,13	1,13	3,14	0,73	1,75	3,36
5	200	10	0,52	0,92	1,83	0,73	1,81	3,84	1,84	2,94	2,94	4,77	5,03	6,81	10,09
5	200	20	0,86	2,43	5,22	1,97	11,44	16,28	3,63	5,18	5,18	7,11	14,16	23,59	36,34
5	200	50	8,58	13,73	18,47	23,84	124,77	328,17	10,59	13,93	13,93	20,66	73,69	177,98	390,97
5	500	5	0,36	0,81	1,77	0,77	1,94	3,11	1,00	2,96	2,96	5,25	3,41	5,46	6,98
5	500	10	0,89	2,89	4,11	1,78	5,06	8,69	2,19	10,54	10,54	18,33	11,91	28,35	39,70
5	500	20	9,56	16,35	33,80	34,52	70,51	127,47	8,67	32,77	32,77	58,63	55,47	141,44	223,30
5	500	50	30,22	48,80	82,38	70,89	200,67	504,52	90,06	130,96	130,96	166,78	382,12	730,91	1564,87
10	25	5	0,02	0,06	0,16	0,05	0,23	0,41	0,08	0,12	0,12	0,16	0,44	0,56	0,64
10	25	10	0,05	0,09	0,16	0,19	0,53	0,92	0,19	0,33	0,33	0,55	1,52	2,14	3,39
10	25	20	0,16	0,28	0,45	0,88	2,22	3,91	0,38	0,48	0,48	0,56	3,79	5,13	7,20
10	50	5	0,03	0,05	0,08	0,17	0,46	0,74	0,14	0,47	0,47	1,19	0,48	1,19	1,89
10	50	10	0,08	0,23	0,58	1,13	2,08	3,11	0,36	0,63	0,63	1,05	3,92	5,99	8,95
10	50	20	0,17	0,33	0,80	1,80	5,74	7,20	0,64	0,96	0,96	1,20	22,70	37,92	49,8
10	100	5	0,08	0,17	0,28	1,53	1,86	2,03	0,31	0,89	0,89	1,59	4,98	4,98	4,98
10	100	10	0,33	0,63	0,97	4,33	5,63	7,92	1,34	2,19	2,19	3,39	12,80	16,58	16,58
10	100	20	0,83	1,54	2,95	14,97	20,43	25,19	2,61	4,61	4,61	5,70	30,14	84,19	84,19
10	100	50	15,75	18,32	21,52	363,14	545,30	772,33	10,08	12,56	12,56	15,06	140,05	913,37	1451,49
10	200	5	0,20	0,41	0,77	2,03	4,79	6,77	2,95	4,10	4,10	6,25	11,45	16,86	16,86
10	200	10	1,08	2,29	3,80	11,97	16,97	20,52	3,81	5,70	5,70	7,22	34,95	37,60	39,56
10	200	20	4,69	6,08	8,33	63,73	96,87	128,47	17,03	24,40	24,40	35,27	116,60	207,80	352,99

Tabela 3.14 – Tempos computacionais (em segundos) – restrições de capacidade máxima

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando LP			Heurística com inicialização usando LP + Pesquisa Local		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
10	200	50	25,47	41,02	67,53	245,38	83,52	1441,47	39,84	57,87	68,78	415,96	1730,01	4297,08
10	500	5	1,28	2,82	5,50	3,50	14,12	22,22	7,45	11,41	11,41	16,30	43,93	66,99
10	500	10	2,23	6,12	15,59	39,59	76,16	125,89	15,53	51,59	88,50	142,46	198,49	246,96
10	500	20	43,50	63,62	81,50	331,09	411,78	684,42	55,97	146,86	248,02	602,22	1092,17	1657,25
10	500	50	60,63	192,12	323,27	1115,25	3695,84	6509,09	388,07	501,95	620,47	3245,52	13266,23	25421,89
20	25	5	0,00	0,08	0,14	0,00	1,77	3,31	0,28	0,89	2,77	3,95	5,48	7,91
20	25	10	0,36	0,44	0,66	0,66	8,43	12,03	0,58	0,98	1,58	10,97	14,58	18,45
20	25	20	0,98	1,19	1,47	1,47	32,03	51,28	1,06	1,80	2,13	50,53	71,44	89,22
20	50	5	0,11	0,19	0,28	3,78	4,86	5,61	0,58	0,74	1,08	9,67	11,25	12,42
20	50	10	0,41	0,71	1,45	5,94	12,08	17,47	1,00	2,51	4,55	26,20	35,94	56,36
20	50	20	2,72	4,73	6,91	41,47	101,29	165,05	3,17	4,36	6,97	68,17	203,80	343,40
20	100	5	0,28	0,44	0,72	10,03	12,69	14,97	1,86	3,11	4,31	17,58	28,04	37,30
20	100	10	1,14	1,83	3,49	31,05	59,40	90,95	3,17	5,61	9,53	63,55	102,44	164,36
20	100	20	4,00	7,49	14,25	141,75	250,02	310,81	14,05	22,93	43,97	237,57	619,19	835,30
20	100	50	74,64	112,69	136,59	644,89	2267,22	5261,22	87,88	105,08	118,72	1149,81	5550,66	10366,14
20	200	5	0,86	1,29	1,89	31,80	46,49	65,52	2,92	6,76	15,99	78,83	100,48	131,74
20	200	10	3,67	8,86	15,13	123,48	187,55	242,14	11,13	16,29	23,41	287,41	410,40	621,28
20	200	20	19,74	44,28	88,30	415,77	724,20	901,94	76,22	95,40	103,13	733,13	1281,45	2688,93
20	200	50	123,14	274,71	370,59	1799,64	7263,48	17872,13	339,65	435,94	505,48	3695,32	26648,22	37110,47
20	500	5	3,13	4,59	7,77	63,20	181,43	351,08	10,23	23,15	29,78	183,08	282,09	385,87
20	500	10	18,14	36,74	60,45	761,52	1065,20	1410,83	51,70	93,31	180,47	1301,27	1887,16	2718,07
20	500	20	60,31	145,98	198,00	4512,47	5866,99	8457,64	216,40	311,71	474,06	7354,92	8643,07	10078,09
20	500	50	455,02	992,43	1385,45	14007,66	73280,84	120092,98	2050,13	3541,76	4611,51	25898,51	124819,51	185722,09
Resumo dos Valores			0,00	38,25	1385,45	0,00	1806,83	120092,98	0,05	105,60	4611,51	0,11	3508,94	185722,09

Tabela 3.14 – Tempos computacionais (em segundos) – restrições de capacidade máxima (cont.)

T	n	m	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex	
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média
5	25	5	0,00	0,15	0,25	0,02	0,17	0,27	0,19	0,34
5	25	10	0,06	0,36	0,72	0,23	0,46	0,80	1,09	2,69
5	25	20	0,69	1,21	1,63	1,28	1,56	1,84	4,06	8,52
5	50	5	0,00	0,19	0,41	0,06	0,26	0,42	0,13	0,13
5	50	10	0,55	0,72	1,16	0,69	0,90	1,34	2,87	9,91
5	50	20	0,41	1,65	3,31	1,53	2,81	4,09	10,90	29,06
5	100	5	0,09	0,50	0,78	0,12	0,63	0,95	2,15	6,75
5	100	10	0,63	1,30	2,16	1,19	1,80	2,80	16,22	31,59
5	100	20	2,19	5,27	11,09	4,37	7,62	13,27	57,44	112,64
5	100	50	8,24	19,54	36,14	27,61	39,98	56,67	145,94	360,00
5	200	5	0,02	0,65	1,13	0,02	0,90	1,55	0,36	15,52
5	200	10	1,95	3,44	5,19	3,03	4,88	6,86	15,86	113,55
5	200	20	7,08	9,42	12,06	13,99	16,79	20,00	92,84	296,22
5	200	50	59,84	76,98	95,70	90,83	128,58	156,97	2059,51	28346,59
5	500	5	2,73	4,81	7,41	3,94	6,03	8,47	4,41	58,44
5	500	10	2,80	13,03	35,28	5,95	18,24	37,48	66,41	142,08
5	500	20	25,50	54,44	89,67	51,00	86,06	125,89	477,58	22034,05
5	500	50	386,28	459,74	519,67	630,97	708,90	851,23	18471,34	59907,95
10	25	5	0,33	0,20	2,22	0,53	1,15	2,39	2,42	5,89
10	25	10	0,97	2,65	5,70	1,59	3,27	6,20	1,33	6,09
10	25	20	1,81	4,59	7,28	6,19	8,02	10,55	17,77	119,08
10	50	5	0,41	0,73	1,42	0,78	1,19	1,88	0,61	2,67
10	50	10	2,22	3,14	4,33	3,75	5,12	6,50	3,75	19,06
10	50	20	2,64	8,07	20,88	10,83	22,03	31,39	24,08	201,02
10	100	5	2,42	3,84	5,52	3,25	5,19	7,30	6,67	12,01
10	100	10	1,39	5,14	9,92	8,39	10,73	15,28	6,33	70,76
10	100	20	8,42	18,37	34,03	36,33	46,85	63,30	103,06	1681,93
10	100	50	153,11	233,67	380,52	317,55	456,50	564,36	427,39	121381,21
10	200	5	3,36	4,77	5,20	6,69	8,75	10,78	23,06	104,48
10	200	10	9,44	11,18	13,91	22,05	28,50	35,88	88,88	277,97
10	200	20	39,00	84,89	142,70	130,36	181,03	211,14	191,48	14902,67
10	200	50	456,11	830,15	1033,59	1467,63	1790,68	2502,47	7772,91	58429,24
10	500	5	16,00	19,20	23,45	24,05	31,38	39,74	49,78	217,97

Tabela 3.14 – Tempos computacionais (em segundos) – restrições de capacidade máxima (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
10	500	10	55,335	100,47	179,81	92,39	179,36	250,28	438,05	2367,11	5635,58
10	500	20	373,08	533,52	635,22	826,50	988,93	1415,34	4977,28	47058,44	122573,23
10	500	50	2536,22	3368,75	3973,06	4292,44	6620,52	8131,98	48753,72	131890,38	206322,89
20	25	5	0,02	1,44	3,14	0,02	2,94	4,75	1,31	3,21	6,92
20	25	10	5,13	7,77	11,34	10,61	15,08	21,92	9,00	36,36	74,88
20	25	50	6,23	7,63	9,50	33,78	50,35	74,67	609,97	3429,81	10656,53
20	50	5	1,45	3,42	6,83	5,84	7,70	11,61	5,86	33,49	119,05
20	50	10	7,49	14,64	19,30	22,55	30,05	36,14	42,09	424,09	1515,24
20	50	20	22,47	35,61	53,31	88,92	119,37	142,52	2018,53	9870,99	37712,34
20	100	5	4,94	20,98	46,14	17,88	33,70	57,66	5,94	30,55	108,75
20	100	10	20,45	30,63	37,84	65,52	81,59	102,30	287,77	468,31	885,69
20	100	20	142,75	186,93	280,63	291,75	432,46	537,17	3103,13	12335,47	20057,30
20	100	50	1843,94	2166,79	2756,95	2622,09	3621,01	5373,38	21878,83	22866,17	23798,11
20	200	5	0,06	12,68	20,67	37,17	47,62	55,42	35,23	284,31	744,72
20	200	10	49,80	72,25	99,44	194,30	315,08	567,55	1184,64	45247,54	200000,00
20	200	20	487,36	700,63	1321,45	1048,91	1592,38	2330,70	47401,41	130972,33	207812,78
20	200	50	4560,16	7401,84	9197,25	11492,45	17164,81	24238,78	---	---	---
20	500	5	27,41	97,58	222,30	200,27	242,89	313,02	856,77	21588,11	99602,95
20	500	10	259,45	421,81	634,48	1014,00	1267,88	1468,92	22321,50	72091,20	215343,67
20	500	20	1835,41	2309,06	2736,77	5270,97	6217,60	6729,39	---	---	---
20	500	50	20062,11	25361,11	36029,19	68028,02	89669,27	125148,49	---	---	---
Resumo dos Valores			0,00	828,52	36029,19	0,02	2450,51	125148,49	0,06	15589,31	215343,67

Tabela 3.14 – Tempos computacionais (em segundos) – restrições de capacidade máxima (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando LP			Heurística com inicialização usando LP + Pesquisa Local		
			<i>Meior</i>	<i>Média</i>	<i>Pior</i>	<i>Meior</i>	<i>Média</i>	<i>Pior</i>	<i>Meior</i>	<i>Média</i>	<i>Pior</i>	<i>Meior</i>	<i>Média</i>	<i>Pior</i>
5	25	5	0.00	0.02	0.05	0.02	0.05	0.09	0.08	0.20	0.63	0.08	0.25	0.64
5	25	5	0.03	0.07	0.14	0.17	0.23	0.30	0.16	0.20	0.25	0.41	0.45	0.56
5	25	20	0.22	0.34	0.48	0.88	2.10	3.16	1.83	4.16	6.13	6.00	8.57	11.09
5	50	5	0.02	0.02	0.03	0.02	0.08	0.13	0.13	0.21	0.33	0.17	0.31	0.38
5	50	20	0.09	0.14	0.25	0.39	0.65	1.14	0.45	0.47	0.66	0.95	1.51	2.47
5	50	50	0.28	0.59	0.99	2.06	3.32	5.25	1.09	2.11	5.03	4.16	6.73	10.20
5	100	5	0.03	0.14	0.36	0.06	0.25	0.52	0.20	0.58	0.99	0.48	0.93	1.52
5	100	10	0.17	0.49	0.80	1.38	1.83	2.75	1.27	1.89	2.36	3.52	5.09	6.58
5	100	20	0.70	1.46	2.13	5.59	11.75	19.47	2.30	3.11	3.97	9.59	16.76	29.50
5	100	50	6.39	11.99	14.84	57.02	224.68	369.30	16.73	29.80	59.00	179.70	336.13	467.74
5	200	5	0.13	0.31	0.73	0.34	0.67	0.97	1.25	1.69	2.66	1.58	2.68	3.81
5	200	10	1.27	1.89	2.77	4.53	7.09	10.33	3.31	4.66	9.38	11.23	13.61	15.98
5	200	20	4.22	6.47	10.14	21.44	29.83	47.63	8.75	12.56	23.97	59.28	105.63	182.33
5	200	50	15.38	28.11	49.33	87.38	268.18	529.50	46.81	67.03	83.78	203.47	484.22	1281.23
5	500	5	0.89	1.17	1.97	2.31	3.82	5.88	2.53	6.67	12.44	4.89	11.03	20.59
5	500	10	6.14	12.84	23.83	16.03	24.68	39.13	14.30	17.79	25.23	39.16	46.69	49.67
5	500	20	5.47	27.61	59.41	5.47	123.88	196.73	15.44	53.04	98.86	15.44	226.19	573.49
5	500	50	75.31	93.01	120.61	422.02	1134.88	2089.31	198.28	245.42	270.33	912.57	2940.95	8936.93
10	25	5	0.03	0.05	0.06	0.06	0.32	0.63	0.11	0.19	0.27	0.20	0.77	1.28
10	25	10	0.06	0.27	0.70	1.34	1.89	2.66	0.31	1.08	3.33	4.89	6.63	8.36
10	25	20	0.45	0.68	1.34	6.67	17.40	44.28	1.55	5.32	11.64	21.17	35.74	42.53
10	50	5	0.00	0.08	0.20	0.00	0.57	1.22	0.33	0.52	0.95	1.19	1.64	2.11
10	50	10	0.23	0.32	0.47	4.97	5.84	6.77	0.95	1.32	1.70	2.49	7.74	16.52
10	50	20	0.38	1.17	3.11	13.58	36.48	59.50	2.80	36.15	165.94	32.61	111.25	263.92
10	100	5	0.06	0.18	0.34	0.83	2.31	3.77	0.52	1.18	2.92	2.28	5.57	8.11
10	100	10	0.30	1.50	3.70	8.64	15.47	22.42	1.36	3.84	5.66	10.39	29.06	44.44
10	100	20	2.34	3.47	4.33	42.98	117.74	192.28	7.52	52.79	187.63	99.63	249.40	507.60
10	100	50	18.31	31.10	44.64	2344.91	2747.64	3131.39	38.80	614.19	2226.53	304.82	2303.61	6456.07

Tabela 3.15 – Tempos computacionais (em segundos) – restrições de capacidade máxima e mínima

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa			Heurística com inicialização usando IP			Heurística com inicialização usando IP + Pesquisa Local		
			Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior
10	200	5	0.23	0.35	0.45	0.45	6.58	15.28	2.41	3.49	4.11	3.59	11.14	18.19
10	200	10	2.03	3.18	4.94	25.78	45.77	93.08	3.52	10.30	15.49	41.53	98.92	159.58
10	200	10	10.64	20.73	32.72	97.30	305.15	540.03	27.09	38.43	58.59	107.55	233.87	610.10
10	200	50	32.39	101.02	190.42	793.50	3249.04	5787.31	194.70	378.95	584.86	1357.79	4605.74	12516.49
10	500	5	0.80	4.76	11.13	5.67	33.79	49.69	6.19	11.71	25.69	25.69	60.43	90.70
10	500	10	11.64	21.45	40.80	158.84	217.04	312.06	34.30	47.80	70.63	231.11	323.12	403.58
10	500	20	30.06	78.27	144.97	482.34	862.72	1429.02	62.53	246.14	405.67	973.16	2132.99	3261.76
10	500	50	368.42	591.33	986.98	4426.80	8251.40	22409.02	943.15	1422.86	2050.47	4751.16	14169.04	46515.57
20	25	5	0.08	0.14	0.17	0.45	2.71	3.72	0.38	0.66	1.39	2.45	4.27	6.75
20	25	10	0.25	0.59	0.80	4.42	13.39	25.47	1.33	1.93	3.06	12.44	39.59	82.47
20	25	20	1.80	2.23	3.05	38.69	129.40	234.72	3.66	22.83	81.36	76.66	317.80	445.42
20	50	5	0.19	0.61	0.97	0.58	2.28	6.33	0.53	2.31	6.39	0.81	6.09	12.80
20	50	10	2.61	3.47	5.28	12.72	46.21	69.61	2.92	4.84	7.19	57.64	105.74	187.07
20	50	20	10.74	22.53	30.03	133.44	262.09	424.41	8.42	40.29	143.24	93.38	741.28	1417.55
20	100	5	1.72	2.08	2.88	12.72	17.08	21.61	2.97	3.96	4.66	5.08	28.65	48.17
20	100	10	5.41	6.59	7.78	23.70	92.43	178.75	9.22	12.28	16.38	60.21	168.48	295.98
20	100	20	23.00	34.90	55.33	70.53	715.11	1215.08	27.50	50.20	79.07	122.35	553.34	1867.09
20	100	50	519.47	896.28	1527.77	1917.09	4411.09	10374.73	296.98	1444.22	3088.26	7676.75	29690.75	57809.68
20	200	5	0.89	4.63	7.28	3.39	28.56	44.19	2.91	7.56	12.89	7.61	61.02	82.19
20	200	10	10.94	21.22	32.92	288.95	413.42	678.52	28.95	55.46	89.99	198.12	596.23	1084.08
20	200	20	80.00	110.78	146.52	2768.63	3850.61	5313.91	91.64	195.26	369.87	643.54	3363.31	7564.26
20	200	50	908.73	1503.27	2275.72	6137.77	11690.45	31282.44	1058.49	2433.15	3778.01	7614.06	51450.66	117633.20
20	500	5	10.61	25.18	57.28	184.58	295.19	295.19	13.50	34.56	63.94	63.94	323.18	533.72
20	500	10	32.22	56.61	104.75	873.47	1755.38	2303.25	79.58	113.77	128.11	1124.94	2117.22	2892.00
20	500	20	443.94	662.33	1206.82	7742.75	29096.84	41065.28	443.94	662.33	1206.82	7742.75	29096.84	41065.28
20	500	50	1234.65	1689.54	2456.98	9065.45	34987.76	54231.98	1134.78	2341.98	3542.80	9865.76	35213.90	48775.56
Resumo dos Valores			0.00	112.77	2456.98	0.00	1952.86	54231.98	0.08	199.17	3778.01	0.08	3379.12	117633.20

Tabela 3.15 – Tempos computacionais (em segundos) – restrições de capacidade máxima e mínima (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	0,03	0,09	0,23	0,05	0,11	0,28	0,23	0,40	0,72
5	25	10	0,14	0,22	0,31	0,25	0,34	0,44	0,33	0,27	41,03
5	25	20	0,25	1,17	2,28	1,66	2,13	2,77	3,09	16,67	37,47
5	50	5	0,02	0,11	0,19	0,03	0,15	0,20	0,16	0,79	2,08
5	50	10	0,20	0,23	0,25	0,53	0,60	0,73	1,33	5,49	11,77
5	50	20	0,44	0,80	1,59	2,58	3,26	3,86	9,42	29,94	58,33
5	100	5	0,27	0,36	0,50	1,39	1,82	2,48	5,94	4,40	8,08
5	100	10	0,63	0,79	1,03	1,39	1,82	2,48	5,94	25,81	72,16
5	100	20	1,17	1,45	2,09	4,69	8,56	11,74	17,38	39,28	83,17
5	100	50	7,55	14,07	23,11	65,50	107,16	165,27	1116,49	2855,03	8183,02
5	200	5	0,52	0,64	0,95	0,72	0,96	1,27	5,31	15,52	35,69
5	200	10	1,80	3,17	6,99	5,20	6,89	9,27	31,50	93,11	205,56
5	200	20	2,69	4,21	7,08	13,49	21,33	34,47	62,91	224,72	332,14
5	200	50	24,94	47,48	73,58	175,86	223,33	268,00	3240,61	9912,31	19233,86
5	500	5	1,84	2,89	4,73	2,69	5,25	7,78	12,59	87,92	225,99
5	500	10	4,53	8,01	13,38	14,52	21,96	35,59	82,88	263,78	655,36
5	500	20	17,64	47,09	94,08	54,11	128,91	214,52	915,42	5765,67	18665,14
5	500	50	193,42	322,09	428,94	870,88	1310,61	2377,64	41373,02	89253,82	200131,45
10	25	5	0,00	0,26	0,38	0,00	0,46	0,91	1,34	4,03	6,97
10	25	10	0,39	0,62	1,19	1,27	2,10	3,02	3,09	70,51	268,39
10	25	20	0,83	0,97	1,11	5,67	9,70	12,63	35,11	196,64	649,66
10	50	5	0,34	0,50	0,77	0,47	0,83	1,02	1,42	2,61	3,80
10	50	10	0,91	1,25	1,72	3,34	5,74	8,94	7,78	21,29	51,28
10	50	20	1,64	2,07	2,80	24,63	38,72	56,89	124,81	272,34	608,09
10	100	5	1,03	1,48	2,45	2,31	3,43	4,91	4,38	285,41	1303,00
10	100	10	1,56	2,86	3,52	13,13	18,86	27,66	26,84	897,92	2216,03
10	100	20	4,44	5,56	6,80	56,27	85,90	132,66	180,22	842,55	1810,92
10	100	50	25,34	39,11	59,81	772,42	983,91	1286,28	15441,70	129797,45	200099,86
10	200	5	2,08	2,97	3,48	3,06	7,17	14,08	6,53	132,55	376,39

Tabela 3.15 – Tempos computacionais (em segundos) – restrições de capacidade máxima e mínima (cont.)

T	n	m	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
10	200	10	5,06	6,91	9,34	26,23	38,10	50,91	441,83	823,49	1881,06
10	200	20	20,41	24,95	33,48	172,28	332,27	464,53	10323,25	29053,02	68171,75
10	200	50	104,39	125,96	149,94	1969,36	2659,32	4306,55	28039,30	155419,43	200670,52
10	500	5	9,36	12,63	15,34	15,34	37,30	60,67	22,33	809,05	2067,52
10	500	10	31,91	38,33	53,45	187,17	230,93	297,33	309,56	2931,09	7926,33
10	500	20	98,83	136,34	178,06	1036,06	1415,53	2172,38	8400,23	51039,10	200179,20
10	500	50	397,84	692,90	1464,92	10326,70	15195,27	21602,55	98082,49	185498,22	209429,28
20	25	5	1,11	1,93	2,30	2,63	4,48	7,70	3,41	24,08	83,67
20	25	10	2,42	2,97	3,42	7,95	14,80	27,44	42,08	406,37	1427,89
20	25	20	5,80	7,94	9,30	62,33	103,09	166,66	181,78	4026,50	11075,95
20	50	5	4,11	4,51	5,61	4,53	7,76	12,48	118,22	6948,91	31764,66
20	50	10	6,16	8,04	10,33	36,09	50,17	72,30	47,06	174,66	305,94
20	50	20	11,28	13,86	15,45	206,11	294,07	329,28	1427,94	14901,44	32001,84
20	100	5	8,03	11,86	19,31	18,95	30,48	47,14	21,88	1441,08	5378,03
20	100	10	14,49	18,09	22,95	98,13	193,05	319,47	1185,22	3513,88	6677,58
20	100	20	42,00	51,39	62,63	839,09	1381,96	2193,64	17093,19	105626,20	200730,03
20	100	50	265,30	432,22	812,22	7250,86	10150,21	13498,53	227,17	1187,06	3751,28
20	200	5	11,86	26,63	41,08	41,73	66,47	103,22	4425,61	29735,83	85064,50
20	200	10	46,55	76,50	118,92	412,77	692,57	1046,63	4425,61	29735,83	85064,50
20	200	20	221,92	273,37	366,86	1631,53	4366,28	7889,44	208507,37	210793,87	212773,47
20	200	50	1004,05	1226,67	1425,09	20767,50	28328,73	39467,14	---	---	---
20	500	5	66,84	105,86	176,83	76,89	241,80	412,61	2162,06	15028,74	59889,05
20	500	10	291,58	339,82	423,19	2188,31	2787,27	3843,78	58041,42	111318,66	213978,91
20	500	20	1062,48	1336,02	1692,22	11796,77	25372,99	39182,78	---	---	---
Resumo dos Valores	500	50	3014,64	4092,24	5406,86	100161,05	280815,02	480442,45	0,16	23425,71	213978,91

Tabela 3.15 – Tempos computacionais (em segundos) – restrições de capacidade máxima e mínima (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>	<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>
5	100	20	33.58	9.26	10	500	10	1113.19	14.62
5	100	50	56.98	2.19	10	500	20	4587.44	11.14
5	200	20	91.69	8.01	10	500	50	---	---
5	200	50	414.07	3.32	20	25	20	85.65	2.67
5	500	20	3640.90	51.64	20	50	20	413.31	4.08
5	500	50	5799.46	28.90	20	100	20	1237.60	4.95
10	100	20	88.60	4.34	20	100	50	---	---
10	100	50	691.53	1.27	20	200	10	908.26	4.84
10	200	10	145.35	8.57	20	200	20	15083.91	20.83
10	200	20	482.65	4.98	20	500	5	646.11	3.56
10	200	50	4740.12	5.55	20	500	10	8772.38	8.24
10	500	5	186.82	13.24					
Valores Médios Globais								2343.79	10.30

Tabela 3.16 – Tempos computacionais quando o Cplex é utilizado como procedimento heurístico – restrições de capacidade máxima

<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>	<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>
5	100	20	20.13	1.71	10	500	10	623.55	2.87
5	100	50	127.69	0.57	10	500	20	4127.56	4.78
5	200	20	64.82	2.17	10	500	50	---	---
5	200	50	389.04	1.45	20	25	20	124.37	0.96
5	500	20	6275.15	50.65	20	50	20	319.56	1.22
5	500	50	6689.22	5.89	20	100	20	2757.08	3.86
10	100	20	167.68	1.42	20	100	50	---	---
10	100	50	1291.88	0.47	20	200	10	1410.51	3.41
10	200	10	96.42	2.11	20	200	20	8288.08	2.15
10	200	20	424.14	1.39	20	500	5	4234.05	22.94
10	200	50	5482.39	1.69	20	500	10	7677.50	4.37
10	500	5	211.76	6.27					
Valores Médios Globais								3284.70	5.83

Tabela 3.17 – Tempos computacionais quando o Cplex é utilizado como procedimento heurístico – restrições de capacidade máxima e mínima

3.2.2.4.3. Conclusões

A análise dos resultados computacionais permite retirar algumas conclusões acerca do desempenho das heurísticas desenvolvidas. Assim:

7. A heurística primal-dual mostra ser eficiente. É capaz de calcular boas soluções em tempos computacionais razoáveis mesmo para instâncias dos problemas de grandes dimensões. Os piores resultados aparecem quando se consideram simultaneamente capacidades máximas e mínimas, nas situações em que a solução primal calculada apenas com base na solução dual não é admissível e é necessário, durante o procedimento primal, executar o passo 12. Será necessário definir um processo diferente de “trocar” serviços fechados e abertos para que se melhore a qualidade da solução primal. Note-se ainda que estas situações ocorrem apenas quando existem serviços, tais que a sua capacidade mínima se encontra muito perto da respectiva capacidade máxima.
8. A execução da pesquisa local resulta numa melhoria significativa da qualidade da solução primal obtida.
9. A inicialização das variáveis duais através da resolução de um problema de programação linear não apresenta vantagens: as soluções primais são, em média, de pior qualidade e os tempos computacionais são também, em média, maiores. No entanto, encontram geralmente melhores soluções duais.
10. Os limites inferiores calculados pelas heurísticas são, em geral, de boa qualidade.
11. A heurística lagrangeana é, em geral, capaz de calcular melhores soluções primais, mas à custa de tempos computacionais mais elevados.
12. O Cplex não consegue resolver as instâncias dos problemas de maiores dimensões.
13. Considerando as relaxações lineares dos problemas com restrições de capacidades verifica-se que cerca de 15% destas relaxações apresentam soluções óptimas com valores inteiros para as variáveis de localização. Este resultado é muito diferente do apresentado para os problemas sem restrições de capacidades.

3.2.2.5. Capacidades Máximas Decrescentes

As restrições de capacidades máximas decrescentes complicam bastante o problema. O comportamento das variáveis duais λ_i^t é diferente daquele que foi apresentado para os problemas anteriores, e é muito mais difícil encontrar soluções admissíveis. Nos problemas tratados anteriormente, após ser tomada a decisão de quais os serviços a abrir em cada período temporal t , o valor óptimo das variáveis de afectação poderia ser facilmente calculado através da resolução de T problemas de transporte. No problema com restrições de capacidades máximas decrescentes, a resolução de T problemas de transporte não garante o cálculo do valor óptimo das variáveis de afectação. Para tal é necessário resolver o problema de programação linear PL1.

$$\begin{array}{l} \text{PL1} \\ \text{Min} \sum_t \sum_i \sum_j c_{ij}^t x_{ij}^t \end{array} \quad (3-12)$$

sujeito a:

$$\sum_{i \in I_t^+} x_{ij}^t = 1, \quad \forall j, t \quad (3-13)$$

$$Q_i \sum_{\tau=1}^t \sum_{\xi=\tau}^T (a_{ir}^{\xi} + r_{ir}^{\xi}) - \sum_{\tau=1}^t \sum_j d_j^{\tau} x_{ij}^{\tau} \geq 0^6, \quad \forall t, i \in I_t^+ \quad (3-14)$$

$$x_{ij}^t \geq 0, \forall j, t, i \in I_t^+ \quad (3-15)$$

⁶ As variáveis a_{ir}^{ξ} e r_{ir}^{ξ} estão fixas a um ou a zero.

3.2.2.5.1. Inicialização das Variáveis Duais

As variáveis duais são inicializadas como se apresenta no Algoritmo 3-23.

Algoritmo 3-23	
INICIALIZAÇÃO DAS VARIÁVEIS DUAIS PARA O PROBLEMA COM RESTRIÇÕES DE CAPACIDADE MÁXIMA	
DECRESCENTE	
c_{ij}^t - custo de afectar o cliente j ao serviço i no período t ; FA_{ii}^ξ, FR_{ii}^ξ - custos fixos de (re)abertura de i ; I - conjunto de possíveis localizações para os serviços; J - conjunto de clientes; T - número de períodos temporais.	
1.	$v_j^t = \min \{ c_{ij}^t \}, \forall j, t; \quad \pi_i^t = 0 \text{ e } \lambda_i^t = 0, \forall i, t;$
2.	$u_i^t = \begin{cases} \max \left\{ 0, -\min_{\xi \geq t} FR_{ii}^\xi \right\}, & \text{se } \exists FR_{ii}^\xi < 0 \\ 0, & \text{caso contrário} \end{cases}, \forall i, t$
3.	$\rho_t = \max \left\{ 0, -\min_{\xi \geq t} \left(FA_{ii}^\xi - \sum_{\tau=\xi+1}^T u_i^\tau \right) \right\}, \forall i, t.$

3.2.2.5.2. Procedimento Dual Ascendente para as Variáveis λ_i^t

A variável dual λ_i^t influencia o valor das *slacks* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi, \tau \leq t$. Considerem-se as definições seguintes:

$$\Delta = \max_{\substack{j \in J \\ \tau \leq t}} \left\{ \frac{1}{d_j^\tau} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\xi=\tau}^T d_j^\xi \lambda_i^\xi \right\} \right\} \quad (3-16)$$

$$J1(t) = \left\{ j \in J : v_j^t - c_{ij}^t - \sum_{\xi=t}^T d_j^\xi \lambda_i^\xi \leq \delta d_j^t \right\} \quad (3-17)$$

Proposição 3-8: Se a variável λ_i^t é aumentada em $\delta \in]0, \Delta]$, então as *slacks* $SA_{i\tau}^\xi$ and $SR_{i\tau}^\xi, \tau \leq t$, serão alteradas em:

$$\Phi(\delta, \tau, \xi) = \sum_{s=\tau}^{\min\{\xi, t\}} \left(\sum_{j \in J1(s)} \max \left\{ 0, v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi \right\} + \sum_{j \notin J1(s)} \delta d_j^s \right) - \delta Q_i \quad (3-18)$$

Prova:

Se λ_i^t é aumentado em $\delta \in]0, \Delta]$, então as somas $\sum_{\psi=s}^T d_j^\psi \lambda_i^\psi$, com $s \leq t$, serão aumentadas em $d_j^s \delta$.

Estes somatórios influenciam os valores de todas as *slacks* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi$, com $\tau \leq t$. Se $t > \xi$, então todos os somatórios com $\tau \leq \xi$ têm de ser tomados em conta. Se $t \leq \xi$, então apenas somatórios com $\tau \leq t$ serão alterados (somatórios com $s > t$ não serão alterados). Para cada $s, \tau \leq s \leq \min\{\xi, t\}$, $v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi$ com $j \in J1(s)$ ficará menor ou igual a zero, e as correspondentes variáveis u_{ij}^s serão iguais a zero. Para todo $j \notin J1(s)$, as variáveis u_{ij}^s irão ser decrementadas em $d_j^s \delta$. A variável dual λ_i^t influencia todas as *slacks* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi, \tau \leq t$, não apenas devido a $\sum_{\psi=s}^T d_j^\psi \lambda_i^\psi$, com $\tau \leq s \leq t$, mas

também devido a $Q_i \sum_{\psi=\tau}^T \lambda_i^\psi$. O valor deste somatório será aumentado em δQ_i . Pode, portanto, concluir-se que a alteração total reflectida nas *slacks* SA_{it}^ξ e SR_{it}^ξ , $\tau \leq t$, devido a uma alteração δ na variável dual λ_i^t é dada por $\Phi(\delta, \tau, \xi)$.

c.q.d.

Como pode ser visto por observação de (3-18), as *slacks* influenciadas pelo aumento do valor da variável dual poderão ter comportamentos distintos: algumas aumentam enquanto outras diminuem.

Proposição 3-9: Considere-se que λ_i^t é aumentado em δ , com $\delta > \Delta$, sendo as *slacks* resultantes deste aumento representadas por SA_{it}^ξ e SR_{it}^ξ , com $\tau \leq t$. É possível encontrar $\delta \in]0, \Delta]$ tal que se λ_i^t é aumentada em δ em vez de δ' , os valores correspondentes das *slacks* SA_{it}^ξ e SR_{it}^ξ , com $\tau \leq t$, são maiores ou iguais a SA_{it}^ξ e SR_{it}^ξ , com $\tau \leq t$.

A prova é trivial e resulta directamente da aplicação da Proposição 3-8 e de (3-16). Esta proposição motiva o procedimento dual ascendente para as variáveis λ_i^t que se formaliza no Algoritmo 3-24.

Algoritmo 3-24

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS λ_i^t

$J1(t) = \left\{ j \in J : v_j^t - c_{ij}^t - \sum_{\xi=t}^T d_j^\xi \lambda_i^\xi \leq \delta \cdot d_j^t \right\}$; m - número de possíveis localizações para serviços; T - número de períodos temporais; c_{ij}^t - custo de afectar o cliente j ao serviço i no período t ; d_j^t - procura do cliente j durante o período t .

1. $t \leftarrow 1$;
 2. $i \leftarrow 1$;
 3. $\delta \leftarrow +\infty$;
 4. $\delta \leftarrow \max_{\substack{j \in J \\ \tau \leq t}} \left\{ \frac{1}{d_j^\tau} \max \left\{ 0, v_j^\tau - c_{ij}^\tau - \sum_{\xi=\tau}^T d_j^\xi \lambda_i^\xi \right\} \right\}$; Se $\delta = 0$, então ir para 9.
 5. Calcular $J1(t)$. Se $\exists SA_{it}^\xi$ ou SR_{it}^ξ , $\tau \leq t$, tal que $SA_{it}^\xi + \Phi(\delta, \tau, \xi) < 0$ ou $SR_{it}^\xi + \Phi(\delta, \tau, \xi) < 0$, então ir para 6; se não, ir para 7.
 6. Se $\delta = 0$, então ir para 9; se não, $\delta \leftarrow \delta$. Ir para 4.
 7. $\lambda_i^t \leftarrow \lambda_i^t + \delta$; $SA_{it}^\xi \leftarrow SA_{it}^\xi + \Phi(\delta, \tau, \xi)$ e $SR_{it}^\xi \leftarrow SR_{it}^\xi + \Phi(\delta, \tau, \xi)$, $\forall \tau \leq t$.
 8. Executar o procedimento dual ascendente para variáveis v_j^t (Algoritmo 3-3) com $J^* = J$.
 9. $i \leftarrow i + 1$; Se $i > m$, então ir para 10; se não, ir para 3.
 10. $t \leftarrow t + 1$; Se $t > T$, então parar; se não, ir para 2.
-
-

3.2.2.5.3. Procedimento Dual Descendente para as Variáveis λ_i^t

Uma diminuição no valor da variável dual λ_i^t irá aumentar o valor de

$$v_j^\tau - c_{ij}^\tau - \sum_{\xi=\tau}^T d_j^\xi \lambda_i^\xi, \forall \tau \leq t.$$

Proposição 3-10: Se λ_i^t diminui em δ com:

$$0 < \delta \leq \min_{\substack{\tau \leq t \\ j \in J \\ v_j^\tau - c_{ij}^\tau - \sum_{\xi=\tau}^T d_j^\xi \lambda_i^\xi < 0}} \left\{ \frac{v_j^\tau - c_{ij}^\tau - \sum_{\xi=\tau}^T d_j^\xi \lambda_i^\xi}{d_j^\tau} \right\}, \quad (3-19)$$

então todas as *slacks* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi$, com $\tau \leq t$, serão alteradas em:

$$\Omega(\delta, \tau, \xi) = \delta \left(Q_i - \sum_{\psi=\tau}^{\min\{\xi, t\}} \sum_{\substack{j \in J \\ v_j^\psi - c_{ij}^\psi - \sum_{\zeta=\psi}^T d_j^\zeta \lambda_i^\zeta \geq 0}} d_j^\psi \right). \quad (3-20)$$

Prova:

Se λ_i^t diminui em δ , todos os valores $v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi$, $s \leq t$, serão aumentados em $d_j^s \delta$. Para que se garanta que $v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi < 0$ continuem menores ou iguais a zero, é necessário que:

$$v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi + \delta d_j^s \leq 0 \Leftrightarrow \delta \leq - \frac{v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi}{d_j^s}. \text{ Portanto, o limite superior definido por}$$

(3-19) garante que, para todo o j e s tal que $v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi < 0$, este valor irá continuar menor do

que zero. Para cada *slack* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi$, com $\tau \leq t$, todos os valores $v_j^s - c_{ij}^s - \sum_{\psi=s}^T d_j^\psi \lambda_i^\psi \geq 0$, com

$s \leq \min\{\xi, t\}$, serão aumentados em $d_j^s \delta$. Cada uma destas *slacks* é também influenciada pela descida

do valor de δ . Q_i no somatório $Q_i \sum_{\psi=\tau}^T \lambda_i^\psi$. Logo, $\Omega(\delta, \tau, \xi)$ representa a alteração que ocorre em $SA_{i\tau}^\xi$

and $SR_{i\tau}^\xi$, com $\tau \leq t$, devido a uma diminuição de δ no valor da variável dual λ_i^t .

c.q.d.

Esta proposição motiva o procedimento dual descendente para as variáveis duais λ_i^t , que é formalizado como Algoritmo 3-25.

Algoritmo 3-25

PROCEDIMENTO DUAL DESCENDENTE PARA AS VARIÁVEIS λ_i^t

m – número de possíveis localizações para serviços; T – número de períodos temporais; c_{ij}^t – custo de afectar o cliente j ao serviço i no período t ; d_j^t – procura do cliente j durante o período t ;

$$\Omega(\delta, \tau, \xi) = \delta \left(Q_i - \sum_{\psi=\tau}^{\min\{\xi, t\}} \sum_{\substack{j \in J \\ v_j^\psi - c_{ij}^\psi - \sum_{\zeta=\psi}^T d_j^\zeta \lambda_i^\zeta \geq 0}} d_j^\psi \right).$$

1. $t \leftarrow 1$;

2. $i \leftarrow 1$;

3. $\delta \leftarrow \min \left\{ \lambda_i^t, \min_{\substack{\tau \leq t \\ v_j^\tau - c_{ij}^\tau - \sum_{\zeta=\tau}^T d_j^\zeta \lambda_i^\zeta < 0}} \left\{ -\frac{v_j^\tau - c_{ij}^\tau - \sum_{\zeta=\tau}^T d_j^\zeta \lambda_i^\zeta}{d_j^\tau} \right\} \right\}$; Se $\delta = 0$, então ir para 8.

4. Se $Cap_i^t \leftarrow Cap_i^{t-1} + Q_i + \Omega(\delta, \tau, \xi) < 0$ ou $SR_{i\tau}^\xi + \Omega(\delta, \tau, \xi) < 0$, para algum $\tau \leq t$, então:

$$\delta \leftarrow \min_{\substack{\tau \leq t \\ \Omega(\delta, \tau, \xi) < 0}} \left\{ -\frac{SA_{i\tau}^\xi}{\Omega(\delta, \tau, \xi)/\delta}, -\frac{SR_{i\tau}^\xi}{\Omega(\delta, \tau, \xi)/\delta} \right\}.$$

5. Se $\delta = 0$, então ir para 8; se não, $SA_{i\tau}^\xi \leftarrow SA_{i\tau}^\xi + \Omega(\delta, \tau, \xi)$ e $SR_{i\tau}^\xi \leftarrow SR_{i\tau}^\xi + \Omega(\delta, \tau, \xi), \forall \tau \leq t$.

6. $\lambda_i^t \leftarrow \lambda_i^t + \delta$.

7. Executar o procedimento dual ascendente para variáveis v_j^t (Algoritmo 3-3) com $J^* = J$.

8. $i \leftarrow i + 1$. Se $i > m$, então ir para 9; se não, ir para 3.

9. $t \leftarrow t + 1$. Se $t > T$, então parar; se não, ir para 2.

3.2.2.5.4. Procedimento Primal

O procedimento primal que aqui se descreve garante o cálculo de uma solução primal admissível (se alguma existir) para o PLDCD. O procedimento começa por calcular uma solução admissível para o PLDAFR. Se for necessário abrir mais serviços durante o período t , o procedimento calcula o custo de abrir serviços que estão fechados em t mas também o custo de abrir ou reabrir serviços antes de t , pois essas (re)aberturas irão influenciar a capacidade máxima total disponível em t : cada vez que um serviço é (re)aberto a sua capacidade máxima disponível aumenta.

Defina-se a seguinte notação:

b_i^t = menor custo em que se incorre por abrir um serviço $i \notin I_t^+$ durante o período t .

p_i^t = menor custo em que se incorre por reabrir um serviço $i \in I_t^+$ no início de um período $t' < t$.

Cap_i^t = capacidade máxima disponível no serviço i , no início do período t .

O cálculo de b_i^t é muito semelhante ao cálculo de F_i^t (Algoritmo 3-19). As únicas alterações referem-se aos passos 5, 6, 8 e 9. A possibilidade de alterar o

valor de variáveis a_{ia}^b ou r_{ia}^b (com a e b definidos como para o Algoritmo 3-19, Figura 3.1) é considerada apenas se o serviço i tem uma capacidade máxima disponível maior do que zero no fim do período temporal b . De outro modo, mesmo que o serviço estivesse em funcionamento em t , ele não iria contribuir para o aumento da capacidade máxima total disponível. No Algoritmo 3-26 descrevem-se os passos que permitem o cálculo de b_i^t .

Algoritmo 3-26

PROCEDIMENTO DE CÁLCULO DE b_i^t PARA $i \in I_t^+$

$$b = \max\left\{0, \max_{t' < t} \{t' : t \in I_{t'}^+\}\right\}; \quad a = t' \text{ tal que } \exists (i, t', b) \in I_A^+ \cup I_R^+; \quad c = \min\left\{T+1, \min_{t' > t} \{t' : i \in I_{t'}^+\}\right\};$$

$$d = t' \text{ tal que } \exists (i, c, t') \in I_A^+ \cup I_R^+$$

1. Se $b = 0$ e $c = T+1$, então $b_i^t \leftarrow \min\{FA_{it}^\xi : 1 \leq \tau \leq t \leq \xi \leq T\}$; ir para 10.
 2. Se $b = 0$ e $c \leq T$, então

$$b_i^t \leftarrow \min\left\{\min\{FA_{it}^\xi - FA_{ic}^d + FR_{ic}^d : 1 \leq \tau \leq t \leq \xi < c\}, \min\{FA_{it}^d - FA_{ic}^d : 1 \leq \tau \leq t\}\right\}$$
; ir para 10.
 3. Se $b > 0$ e $c = T+1$, então ir para 4; se não, ir para 7.
 4. Se $(i, a, b) \in I_A^+$, então ir para 5; se não, ir para 6.
 5. Se $Cap_i^t > 0$, então $b_i^t \leftarrow \min\left\{\min\{FA_{ia}^\xi - FA_{ia}^b : t \leq \xi \leq T\}, \min\{FR_{it}^\xi : b < \tau \leq t \leq \xi \leq T\}\right\}$; se não,

$$b_i^t \leftarrow \min\{FR_{it}^\xi : b < \tau \leq t \leq \xi \leq T\}$$
, ir para 10.
 6. Se $Cap_i^t > 0$, então $b_i^t \leftarrow \min\left\{\min\{FR_{ia}^\xi - FR_{ia}^b : t \leq \xi \leq T\}, \min\{FR_{it}^\xi : b < \tau \leq t \leq \xi \leq T\}\right\}$; se não

$$b_i^t \leftarrow \min\{FR_{it}^\xi : b < \tau \leq t \leq \xi \leq T\}$$
, ir para 10.
 7. Se $(i, a, b) \in I_A^+$; então ir para 8; se não, ir para 9.
 8. Se $Cap_i^t > 0$, então $b_i^t \leftarrow \min\left\{FA_{ia}^d - FA_{ia}^b - FR_{ic}^d, \min\{FR_{it}^\xi, b < \tau \leq t \leq \xi < c\}, \min\{FA_{ia}^\xi - FA_{ia}^b, t \leq \xi < c\}, \min\{FR_{it}^d - FR_{ic}^d, b < \tau \leq t\}\right\}$;

se não, $b_i^t \leftarrow \min\left\{\min\{FR_{it}^\xi, b < \tau \leq t \leq \xi < c\}, \min\{FR_{it}^d - FR_{ic}^d, b < \tau \leq t\}\right\}$. Ir para 10.
 9. Se $Cap_i^t > 0$ então $b_i^t \leftarrow \min\left\{FR_{ia}^d - FR_{ia}^b - FR_{ic}^d, \min\{FR_{it}^\xi, b < \tau \leq t \leq \xi < c\}, \min\{FR_{ia}^\xi - FR_{ia}^b, t \leq \xi < c\}, \min\{FR_{it}^d - FR_{ic}^d, b < \tau \leq t\}\right\}$;

se não, $b_i^t \leftarrow \min\left\{\min\{FR_{it}^\xi, b < \tau \leq t \leq \xi < c\}, \min\{FR_{it}^d - FR_{ic}^d, b < \tau \leq t\}\right\}$. Ir para 10.
 10. Se o valor de b_i^t foi calculado considerando os custos fixos de uma variável a_{it}^ξ ou r_{it}^ξ tal que $b < \tau$, então $C_i^t \leftarrow Q_i + Cap_i^t$; se não, $C_i^t \leftarrow Cap_i^t$.
 11. $b_i^t \leftarrow \frac{b_i^t}{C_i^t} \left[\frac{\phi_i}{C_i^t} \right]$, com $\phi_i = \begin{cases} D - C, & \text{se } C + C_i^t < D \\ C_i^t, & \text{em caso contrário} \end{cases}$.
-
-

Se um serviço já estiver aberto durante o período t , então existe a possibilidade de aumentar a sua capacidade máxima disponível através da sua (re)abertura num período temporal anterior a t . Isto pode ser conseguido através da “divisão” de variáveis a_{it}^ξ ou r_{it}^ξ , com $\tau \leq t$, que são consideradas iguais a um na solução primal actual, ou então considerando novas variáveis a_{it}^ξ ou r_{it}^ξ tais

que $i \notin I_t^+, \tau \leq t' \leq \xi < t$. O cálculo de p_t^i tem de ter em conta estas possibilidades, e é feito conforme descrito no Algoritmo 3-27.

Algoritmo 3-27

PROCEDIMENTO DE CÁLCULO DE $p_t^i, i \in I_t^+$

$I_A^+ = \{(i, \tau, \xi) : a_{i\tau}^\xi = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{i\tau}^\xi = 1\}$; FA_{ii}^ξ, FR_{ii}^ξ - custos fixos de (re)abertura de i ; I - conjunto de possíveis localizações para os serviços; $I_t^+ = \{i : i \text{ está aberto durante } t\}$.

1. $p_t^i \leftarrow \min \left\{ +\infty, \min_{\substack{(i, \tau, \xi) \in I_A^+ \\ \tau \leq t \\ \xi \leq t}} \{FA_{i\tau}^\xi + FR_{i\tau}^\xi - FA_{i\tau}^\xi\}, \min_{\substack{(i, \tau, \xi) \in I_R^+ \\ \tau \leq t \\ \xi \leq t}} \{FR_{i\tau}^\xi + FR_{i\tau}^\xi - FR_{i\tau}^\xi\} \right\}$.
 2. $t1 \leftarrow 1$;
 3. Se $i \in I_{t1}^+$, então ir para 6; se não, ir para 4.
 4. $b = \max \left\{ 0, \max_{t' < t1} \{t' : i \in I_{t'}^+\} \right\}$; $c = \min \left\{ t, \min_{t' > t1} \{t' : i \in I_{t'}^+\} \right\}$.
 5. Se $b=0$, existe $(i, a, d) \in I_A^+$. Então $p_t^i \leftarrow \min \left\{ p_t^i, \min_{b < \tau \leq \xi < c} \{FA_{i\tau}^\xi\} + FR_{ii}^d - FA_{ii}^d \right\}$; se não,
 $p_t^i \leftarrow \min \left\{ p_t^i, \min_{b < \tau \leq \xi < c} \{FR_{i\tau}^\xi\} \right\}$.
 6. $t1 \leftarrow t1 + 1$. Se $t1 = t$, então ir para 7; se não, ir para 3.
 7. $p_t^i \leftarrow \left\lfloor \frac{p_t^i}{Q_i} \right\rfloor \left\lceil \frac{\phi_i}{Q_i} \right\rceil$, com $\phi_i = \begin{cases} D - C, & \text{se } C + Q_i < D \\ Q_i, & \text{em caso contrário} \end{cases}$.
-
-

O procedimento primal descreve-se formalmente como Algoritmo 3-28.

Algoritmo 3-28

PROCEDIMENTO PRIMAL PARA PLDAFR COM CAPACIDADES MÁXIMAS DECRESCENTES

$I_t^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{i\tau}^\xi = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{i\tau}^\xi = 1\}$; d_j^i - procura do cliente j no período t ; Q_i - capacidade máxima do serviço i aquando da sua (re)abertura; c_{ij}^t - custo de afectar o cliente j ao serviço i no período t ; b_i^t = menor custo em que se incorre por abrir um serviço $i \notin I_t^+$ durante o período t ; p_i^t - menor custo em que se incorre por reabrir um serviço $i \in I_t^+$ no início de um período $t' < t$; Cap_i^t - capacidade máxima disponível no serviço i no início do período t .

1. Executar o procedimento primal para o PLDAFR (Algoritmo 3-4), construindo os conjuntos $I_t^+, \forall t, I_A^+$ e I_R^+ .
2. Resolver o problema PL1 utilizando um *general solver*. Se PL1 for um problema impossível então ir para 3, se não parar.
3. $t \leftarrow 1$. $Cap_i^1 \leftarrow Q_i, \forall i \in I_1^+$ e $Cap_i^1 \leftarrow 0, \forall i \notin I_1^+$.

4. $D \leftarrow \sum_j d_j^t$; $C \leftarrow \sum_{i \in I_i^+} Cap_i^t$. Se $D \leq C$, então ir para 9.
 5. Calcular $b_i^t, \forall i \in I_i^+$ e $p_i^t, \forall i \in I_i^+$. $b_i^t \leftarrow +\infty, \forall i \in I_i^+$ e $p_i^t \leftarrow +\infty, \forall i \notin I_i^+$.
 6. Escolher i^t , tal que $\min\{b_i^t, p_i^t\} = \min_{i \in I_i^+}\{b_i^t, p_i^t\}$.
 7. Recalcular os conjuntos $I_i^+, \forall t$, I_A^+ e I_R^+ e recalcular $Cap_i^t, \forall i$, e C de acordo com a escolha feita em 6.
 8. Se $D > C$, então ir para 5; se não, ir para 9.
 9. Resolver um problema de transportes, considerando o conjunto J de clientes como as origens (com ofertas d_j^t) e destinos dados pelo conjunto I_i^+ (com procuras Cap_i^t), e custos unitários de transporte iguais a c_{ij}^t/d_j^t . Designem-se os valores óptimos das variáveis de transporte por x_{ij}^t .
 10. $Cap_i^t \leftarrow Cap_i^t - \sum_{j \in J} x_{ij}^t, \forall i \in I_i^+$.
 11. $t \leftarrow t+1$. Se $t > T$, então ir para 2; se não, ir para 12.
 12. Se existe $(i, t, \xi) \in I_A^+ \cup I_R^+$, então $Cap_i^t \leftarrow Cap_i^{t-1} + Q_i$; se não, $Cap_i^t \leftarrow Cap_i^{t-1}$. Ir para 4.
-

3.2.2.5.5. Resultados Computacionais

Tal como aconteceu no caso dos problemas de capacidade máxima e mínima, a heurística para os problemas com capacidades máximas decrescentes foi testada, utilizando um conjunto de problemas de teste aleatoriamente gerados. Na realidade, foram utilizados os mesmos problemas de teste gerados para o caso do PLDAFR com restrições de capacidades máximas.

O computador utilizado, linguagem de programação, versão do Cplex, e condições de paragem do Cplex são exactamente as mesmas que foram descritas em 3.2.2.4. Também para estes problemas se executa uma pesquisa local, tendo como ponto de partida a melhor solução calculada pela heurística, e se executa uma heurística lagrangeana à semelhança do referido em 3.2.2.3.

A Tabela 3.18 mostra a qualidade das soluções primais calculadas pela heurística primal-dual. Nas colunas 4 a 9 mostram-se os resultados, considerando a inicialização das variáveis duais como descrito no Algoritmo 3-23, sem (colunas 4 a 6) e com (colunas 7 a 9) a pesquisa local. Nas colunas 10 a 15 apresentam-se os resultados obtidos quando se inicializam as variáveis duais através da resolução de um problema de programação linear, à semelhança do que é descrito em Saldanha-da-Gama, 2002, mostrando-se os resultados sem e com a execução da pesquisa local.

Na Tabela 3.19 mostram-se os resultados obtidos pela execução da heurística lagrangeana. Esta heurística foi construída, considerando a relaxação das restrições de capacidade máxima decrescente com multiplicadores λ_i^t (à semelhança do que é descrito em I.1.1.3), sendo que se obtém o PLDAFR com a função objectivo modificada (3-10).

$$\sum_t \sum_i \sum_j \left(c_{ij}^t + d_j^t \sum_{\tau=t}^T \lambda_i^\tau \right) x_{ij}^t + \sum_t \sum_i \sum_{\xi=t}^T \left(FA_{it}^\xi - Q_i \sum_{\tau=t}^T \lambda_i^\tau \right) a_{it}^\xi + \sum_t \sum_i \sum_{\xi=t}^T \left(FR_{it}^\xi - Q_i \sum_{\tau=t}^T \lambda_i^\tau \right) b_{it}^\xi \quad (3-21)$$

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando IP			Heurística com inicialização usando IP + Pesquisa Local		
			Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior	Meior	Média	Pior
5	25	5	0,22	1,17	3,47	0,00	0,12	0,62	0,24	2,21	4,95	0,00	1,26	4,88
5	25	10	1,21	2,01	4,17	0,89	1,59	3,45	0,77	2,28	3,85	0,68	1,94	3,56
5	25	20	1,46	2,72	3,89	1,40	2,12	3,26	1,06	2,53	3,87	0,41	2,02	3,51
5	50	5	0,11	0,96	1,30	0,02	0,69	1,21	0,80	3,05	6,30	0,00	1,56	4,17
5	50	10	1,31	2,93	4,68	1,30	2,43	3,40	1,46	2,22	3,62	0,74	1,51	1,98
5	50	20	1,42	2,72	3,86	0,74	2,21	3,86	1,31	3,20	4,51	0,00	2,43	3,99
5	100	5	0,42	1,75	4,65	0,10	1,38	3,86	0,00	2,38	4,57	0,00	2,10	4,46
5	100	10	1,00	1,46	2,05	0,72	1,16	2,05	0,73	1,52	2,25	0,44	1,09	1,79
5	100	20	1,40	2,44	3,49	1,40	1,99	2,89	0,87	2,03	3,24	0,65	1,09	3,24
5	100	50	1,40	3,23	4,40	1,34	2,86	3,59	1,70	3,18	3,78	1,70	1,59	4,88
5	200	5	0,03	1,45	2,82	0,00	0,77	2,02	0,03	1,38	2,82	0,01	1,11	2,12
5	200	10	1,22	1,91	2,82	0,99	1,56	2,05	1,30	2,13	3,11	0,99	1,84	2,89
5	200	20	1,76	2,61	3,88	1,41	2,21	3,88	0,97	2,16	3,69	0,68	1,71	2,56
5	500	5	0,00	4,07	5,10	0,00	3,54	5,10	2,82	4,07	4,88	2,80	3,82	4,88
5	500	10	0,42	1,13	1,58	0,36	1,02	1,18	0,00	1,00	2,84	0,00	0,72	1,76
5	500	20	0,78	2,32	2,58	0,70	1,79	2,21	0,46	1,82	2,78	0,39	1,58	2,44
5	500	50	2,92	3,64	5,13	0,69	3,43	5,13	3,06	3,97	4,91	2,32	2,90	3,33
10	25	5	0,76	1,32	2,35	0,29	2,05	3,92	0,81	2,26	3,65	0,54	1,75	5,52
10	25	10	1,53	2,62	3,92	0,85	2,47	3,73	2,98	3,97	5,07	0,81	1,64	2,70
10	25	20	2,20	3,52	4,69	1,46	2,47	5,07	2,98	3,97	5,07	1,33	2,49	3,88
10	50	5	0,54	1,82	4,43	0,54	1,53	2,99	0,06	2,05	6,04	0,01	1,98	5,86
10	50	10	1,79	2,02	2,57	1,08	1,72	2,57	0,76	2,12	3,99	0,52	1,39	1,92
10	50	20	2,96	3,24	3,81	1,90	2,53	3,81	2,60	3,12	3,62	2,25	2,78	3,22
10	100	5	0,16	0,70	1,26	0,01	0,42	0,99	0,26	1,27	2,57	0,14	0,68	2,00
10	100	10	1,19	1,90	3,00	0,29	1,35	3,00	1,80	2,12	2,43	1,33	1,68	2,18
10	100	20	1,92	2,45	2,91	1,15	1,95	2,91	1,75	2,39	3,00	1,21	1,99	2,40
10	100	50	3,14	4,11	4,98	2,25	3,42	4,19	3,30	4,25	4,95	2,65	3,42	4,31
10	200	5	0,23	1,52	2,26	0,01	1,11	2,19	0,23	1,31	2,14	0,23	1,13	1,51
10	200	10	1,10	1,94	3,03	0,40	1,53	2,52	1,62	2,83	5,19	1,32	2,24	3,50
10	200	20	2,05	2,94	3,53	1,22	1,73	2,38	1,94	3,07	3,62	1,94	2,57	3,14
10	200	50	3,60	3,70	3,85	2,97	3,48	3,85	3,14	3,49	3,82	2,18	2,57	3,01
10	500	5	0,14	0,75	1,36	0,01	0,58	1,33	0,41	0,88	1,19	0,41	0,76	0,99
10	500	10	0,73	1,61	2,38	0,65	1,31	1,63	1,12	1,77	2,75	1,01	1,45	2,54
10	500	20	1,80	2,26	2,83	1,43	1,66	2,05	1,68	2,38	3,18	1,17	2,07	2,62
10	500	50	3,07	3,78	4,32	2,47	3,08	3,96	3,04	3,73	4,57	2,58	3,31	4,57
20	25	5	1,45	3,09	4,31	1,36	2,13	3,29	1,32	3,64	5,24	1,21	2,37	2,77
20	25	10	2,63	3,15	3,90	2,04	2,67	3,11	2,67	3,40	5,91	1,89	2,42	3,00
20	25	20	3,96	5,13	5,97	2,78	3,94	5,97	4,14	6,02	7,94	2,51	3,47	4,51
20	50	5	1,47	2,55	4,34	0,50	1,77	3,40	0,67	2,32	4,64	0,28	1,49	3,06
20	50	10	1,35	1,98	2,77	0,85	1,48	2,21	1,34	2,69	5,30	1,02	2,11	4,52
20	50	20	4,17	4,77	5,48	3,99	4,62	5,48	3,92	4,48	5,31	2,08	3,07	3,92
20	100	5	0,93	2,34	3,74	0,93	2,06	3,31	0,66	2,45	4,87	0,58	1,84	2,85

Tabela 3.18 – Qualidade da solução primal (em percentagem)

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística com inicialização usando IP			Heurística com inicialização usando IP + Pesquisa Local		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
20	100	20	2,99	3,94	4,50	2,52	3,18	4,40	3,28	3,72	4,07	2,31	2,63	3,28
20	100	50	109,91	128,20	137,60	108,56	127,02	136,43	120,99	130,55	156,65	110,39	131,50	139,25
20	200	5	1,79	2,13	2,49	1,51	1,88	2,40	2,36	2,89	3,25	1,89	2,38	2,95
20	200	10	1,10	1,91	2,70	0,96	1,69	2,21	2,61	2,91	3,10	2,25	2,58	2,76
20	200	20	2,86	3,39	4,26	1,88	2,75	3,55	3,02	4,23	5,02	2,42	3,12	3,77
20	200	50	115,41	132,04	151,36	113,99	130,83	150,07	120,25	136,96	165,30	119,25	128,59	162,20
20	500	5	1,58	2,85	3,95	1,19	2,28	3,56	1,75	3,12	4,03	1,49	2,98	3,63
20	500	10	2,43	3,04	4,50	1,42	2,02	2,28	2,23	4,15	5,11	2,23	3,51	4,25
20	500	20	3,65	4,37	7,29	2,44	3,86	4,24	3,98	5,12	7,43	3,18	4,98	6,69
20	500	50	118,87	143,93	165,74	116,27	141,29	162,08	123,60	152,60	163,97	122,55	151,07	161,87
Resumo dos Valores			0,00	9,88	165,74	0,00	9,30	162,08	0,00	10,42	165,30	0,00	9,67	162,20

Tabela 3.18 – Qualidade da solução primal (em percentagem) (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	0.00	3.77	9.44	0.00	2.38	6.05
5	25	10	0.91	5.59	7.58	0.86	4.35	6.98
5	25	20	3.06	4.97	7.71	2.09	4.08	7.17
5	50	5	0.00	1.58	4.68	0.00	1.13	2.56
5	50	10	1.48	4.67	8.34	0.95	2.80	4.00
5	50	20	3.19	5.49	9.73	1.52	4.51	8.58
5	100	5	0.97	3.42	4.73	0.97	3.23	4.64
5	100	10	2.62	6.05	13.04	1.88	4.51	10.47
5	100	20	3.81	5.68	8.54	3.22	4.60	6.66
5	100	50	3.66	5.47	8.04	3.03	4.65	7.58
5	200	5	2.08	3.09	4.16	0.00	1.93	3.50
5	200	10	3.36	4.64	6.69	2.33	3.82	6.23
5	200	20	4.20	5.73	8.20	3.03	3.98	4.89
5	200	50	5.19	6.28	7.40	4.03	5.11	6.15
5	500	5	0.00	1.54	3.12	0.00	1.02	1.81
5	500	10	0.57	3.85	6.53	0.44	2.66	4.96
5	500	20	2.87	4.23	5.96	1.66	3.49	5.35
5	500	50	4.66	6.03	7.04	3.89	4.56	5.39
10	25	5	4.27	8.40	11.44	4.12	5.16	6.38
10	25	10	4.73	7.89	10.78	3.13	4.95	6.67
10	25	20	4.53	7.77	10.70	2.39	3.32	3.90
10	50	5	4.63	7.81	15.96	3.23	6.35	15.61
10	50	10	3.42	11.54	30.00	0.77	7.67	20.89
10	50	20	5.92	7.84	10.15	3.28	5.40	9.46
10	100	5	4.95	9.25	22.99	2.08	5.57	15.02
10	100	10	3.45	6.51	13.56	2.60	4.63	9.72
10	100	20	5.32	8.89	13.61	2.20	5.82	10.35
10	100	50	5.31	6.61	8.23	3.74	4.52	5.17
10	200	5	4.76	7.17	11.77	4.22	6.18	8.80
10	200	10	3.24	6.12	8.45	2.45	4.39	6.32
10	200	20	5.74	10.21	14.36	2.86	6.78	11.60
10	200	50	5.02	7.15	7.84	4.02	5.05	5.37
10	500	5	0.20	0.95	1.57	0.17	0.89	1.18
10	500	10	1.89	2.35	3.86	1.32	2.15	3.28
10	500	20	2.88	3.14	3.90	2.73	3.10	3.71
10	500	50	5.62	6.04	7.32	4.21	5.12	5.42
20	25	5	2.32	3.86	5.75	1.85	2.89	5.52
20	25	10	3.91	4.52	4.96	3.17	3.24	3.47
20	25	20	5.59	6.25	7.04	4.42	5.07	5.28
20	50	5	1.24	3.57	6.29	1.11	2.48	5.09
20	50	10	2.22	2.89	4.15	1.91	2.75	3.28
20	50	20	6.50	6.75	7.24	5.65	6.21	6.52
20	100	5	1.38	2.84	3.56	1.22	2.45	2.88
20	100	10	4.50	4.81	5.92	3.60	4.21	4.44
20	100	20	4.75	5.94	8.34	3.56	5.45	6.67
20	100	50	135.26	142.50	158.58	128.96	139.54	149.66
20	200	5	2.80	3.68	5.28	2.10	3.12	4.43
20	200	10	1.72	2.98	3.64	1.63	2.46	2.87
20	200	20	5.23	5.98	6.64	4.19	5.34	5.71
20	200	50	125.21	142.35	166.54	122.15	134.77	163.25
20	500	5	2.75	3.39	5.40	2.31	2.96	4.38
20	500	10	3.84	4.21	4.44	3.34	3.68	3.86
20	500	20	7.04	7.56	9.03	4.93	5.38	6.77
20	500	50	125.80	151.80	164.85	123.54	148.99	162.80
Resumo dos Valores			0.00	13.21	166.54	0.00	11.68	163.25

Tabela 3.19 – Qualidade da solução primal (em percentagem) - Heurística Lagrangeana

			Heurística Primal-Dual			Heurística Primal-Dual com Inicialização usando LP			Heurística Lagrangeana		
<i>T</i>	<i>n</i>	<i>m</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	15.37	24.61	34.78	6.42	13.58	21.55	0.66	24.40	42.61
5	25	10	16.85	25.93	30.60	7.66	11.17	14.31	1.42	31.82	43.69
5	25	20	27.06	27.77	28.48	9.27	12.90	17.34	42.85	44.12	45.66
5	50	5	13.81	25.93	42.56	7.66	13.60	27.78	2.54	20.83	47.52
5	50	10	27.48	29.08	30.57	8.92	12.72	17.22	31.77	35.96	39.07
5	50	20	22.80	27.60	32.87	7.68	12.32	19.68	32.86	37.10	41.30
5	100	5	19.64	25.90	34.22	8.32	11.73	16.22	18.01	28.97	40.68
5	100	10	21.12	29.14	39.52	7.31	13.23	22.07	29.15	36.49	46.93
5	100	20	27.21	30.08	32.28	9.03	11.55	13.29	37.44	38.80	40.44
5	100	50	28.81	30.91	32.39	13.56	14.32	15.57	40.22	43.43	48.50
5	200	5	22.77	32.69	37.77	14.90	18.61	21.87	21.08	32.03	40.31
5	200	10	28.37	34.25	42.09	8.85	16.13	28.39	27.84	37.55	48.64
5	200	20	26.51	29.87	36.00	8.33	13.54	19.86	31.80	37.04	41.14
5	200	50	31.12	35.29	37.98	15.36	18.13	21.23	42.39	45.33	46.95
5	500	5	26.08	34.32	42.55	12.59	17.94	22.17	2.13	21.94	35.85
5	500	10	30.13	36.73	46.93	15.34	18.28	24.36	28.29	32.75	36.39
5	500	20	31.60	35.71	42.46	14.89	18.31	22.70	27.89	35.15	43.98
5	500	50	33.76	36.10	39.87	15.11	18.08	21.07	42.01	44.13	47.48
10	25	5	29.32	35.06	39.98	10.01	16.01	20.72	33.14	40.42	44.43
10	25	10	31.63	36.93	46.23	12.34	16.44	22.67	38.27	43.31	52.12
10	25	20	34.91	38.45	41.15	17.27	20.16	22.99	45.37	47.50	49.62
10	50	5	24.70	34.44	49.43	10.09	16.10	25.01	29.34	38.33	52.85
10	50	10	31.51	36.58	43.69	11.09	16.45	21.88	37.35	42.07	48.68
10	50	20	37.27	42.21	44.35	13.34	20.36	24.41	44.01	49.28	51.52
10	100	5	30.74	37.89	45.61	13.74	19.52	24.73	21.96	39.04	48.12
10	100	10	35.83	38.59	41.84	13.74	18.92	22.22	25.89	39.41	47.86
10	100	20	34.74	37.95	41.87	15.46	17.54	19.53	40.81	44.46	48.97
10	100	50	41.19	42.06	43.98	20.76	21.59	23.10	49.17	50.44	52.51
10	200	5	30.11	34.12	38.48	13.77	16.94	22.79	33.26	36.37	39.21
10	200	10	35.48	39.41	44.36	17.69	34.27	91.55	35.72	40.81	44.93
10	200	20	35.92	38.85	41.43	15.44	18.38	20.72	42.75	44.20	46.19
10	200	50	38.66	40.58	42.15	16.31	19.56	22.78	48.87	53.78	64.89
10	500	5	37.95	40.84	44.88	19.52	22.44	29.85	31.48	40.21	42.85
10	500	10	35.29	42.90	51.35	16.46	22.82	32.07	45.96	46.89	47.98
10	500	20	37.18	42.14	48.31	18.10	21.91	25.09	38.15	40.82	41.76
10	500	50	42.55	45.88	48.33	21.60	24.94	27.56	37.28	42.17	44.69
20	25	5	36.26	40.15	45.39	19.09	20.54	22.52	36.07	41.89	44.06
20	25	10	41.03	43.95	46.20	20.46	22.50	24.32	27.60	35.25	41.21
20	25	20	46.98	48.71	50.55	21.36	24.40	26.54	31.43	40.01	42.26
20	50	5	33.73	43.10	50.21	15.99	21.93	28.18	35.25	41.89	44.88
20	50	10	40.97	46.34	52.90	18.49	24.33	29.68	38.97	40.56	43.80
20	50	20	45.08	48.14	50.65	22.16	23.70	26.19	28.65	38.10	42.83
20	100	5	33.05	39.38	42.43	14.23	18.96	21.16	43.89	45.87	46.58
20	100	10	42.60	48.22	53.90	20.62	23.81	29.20	40.69	42.54	44.58
20	100	20	45.05	49.20	52.02	20.36	25.01	28.68	38.74	39.85	43.77
20	100	50	52.05	55.85	57.70	32.54	36.45	42.26	26.52	32.58	46.82
20	200	5	38.53	41.71	46.95	21.52	33.54	38.95	42.62	43.12	44.21
20	200	10	39.87	48.77	57.88	37.59	39.78	48.48	40.90	42.03	43.51
20	200	20	46.73	48.99	50.08	42.48	45.26	48.89	32.11	36.56	42.31
20	200	50	47.80	49.22	51.80	43.54	47.95	49.25	37.02	39.60	46.25
20	500	5	35.69	42.57	49.88	33.98	38.96	45.87	32.47	35.98	43.25
20	500	10	37.55	42.69	51.49	34.58	39.48	48.54	41.50	43.13	44.62
20	500	20	34.57	43.54	56.00	33.48	38.45	54.47	33.35	38.77	44.43
20	500	50	42.69	44.52	58.10	43.02	46.15	53.87	37.46	38.99	39.83
Resumo dos Valores			13.81	38.55	58.10	6.42	22.25	91.55	0.66	39.33	64.89

Tabela 3.20 – Qualidade da solução dual (em porcentagem)

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística Primal-Dual com Inicialização usando LP			Heurística Primal-Dual com Inicialização usando LP + Pesquisa Local		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	25	5	0.20	0.42	0.88	0.33	0.55	1.00	0.03	0.14	0.34	0.13	0.24	0.38
5	25	10	1.34	1.86	3.00	2.00	2.46	3.50	0.20	0.98	1.72	0.53	1.54	2.28
5	25	20	2.80	3.55	4.27	3.42	8.14	11.09	0.50	4.49	13.97	3.56	9.21	18.91
5	50	5	0.45	0.71	1.06	0.64	0.96	1.23	0.33	0.15	0.33	0.23	0.37	0.47
5	50	10	1.89	3.11	6.77	2.52	4.51	7.49	0.80	1.83	4.78	2.31	3.64	7.23
5	50	20	5.50	9.29	15.97	6.94	14.63	24.31	0.53	7.72	19.42	8.41	17.01	31.21
5	100	5	1.44	3.23	4.41	1.80	3.77	4.64	0.38	1.78	4.03	0.81	2.15	4.27
5	100	10	4.24	9.63	14.16	8.27	13.32	20.63	2.52	8.08	13.20	6.03	12.40	21.05
5	100	20	19.36	26.46	32.41	29.41	45.70	69.36	4.84	45.04	102.86	8.19	69.85	140.74
5	100	50	262.63	458.47	796.91	291.85	782.83	1091.03	25.42	349.80	513.02	438.70	733.44	1044.17
5	200	5	6.06	8.51	10.58	7.14	9.68	12.55	0.95	2.62	5.17	2.45	3.64	5.81
5	200	10	11.13	29.67	61.47	21.99	43.25	85.06	3.50	11.57	28.53	15.20	21.72	31.06
5	200	20	117.21	180.56	239.43	217.35	296.89	465.51	5.61	80.62	170.69	92.55	168.56	277.67
5	200	50	720.38	1913.01	3493.57	1905.17	3762.35	6137.44	126.39	1071.32	4112.93	252.78	1885.70	4245.37
5	500	5	28.81	73.23	153.93	34.63	80.17	157.83	4.16	43.20	152.43	8.44	52.00	164.04
5	500	10	163.58	257.08	399.35	194.10	301.51	426.36	9.50	76.67	127.63	75.78	108.19	163.18
5	500	20	380.54	1158.09	2700.24	625.45	1783.19	3673.38	120.27	444.72	875.58	707.55	1104.20	1685.69
5	500	50	13159.91	21361.88	35254.99	19572.96	30073.71	37640.63	1931.89	9657.81	34525.64	10459.74	22350.64	47733.16
10	25	5	0.91	1.48	2.27	1.53	2.63	3.27	0.17	0.81	1.58	0.80	1.92	3.44
10	25	10	2.56	5.00	5.88	6.77	13.04	16.27	0.69	3.17	6.92	5.80	9.90	15.80
10	25	20	10.71	23.97	37.78	67.11	88.02	101.80	1.88	8.71	33.30	61.89	75.57	97.35
10	50	5	2.09	3.19	4.91	3.27	4.63	5.77	0.42	2.29	4.80	1.80	3.66	6.53
10	50	10	6.17	15.03	24.19	9.41	25.71	51.92	0.44	7.94	18.17	8.89	23.45	30.67
10	50	20	25.77	41.04	51.30	62.50	176.65	303.95	32.56	61.58	109.52	80.05	124.32	169.96
10	100	5	5.67	9.96	22.74	9.92	14.21	30.08	1.02	6.10	20.36	5.89	10.79	24.36
10	100	10	12.70	44.33	110.25	43.67	94.60	121.52	4.00	16.41	37.81	40.61	75.78	125.25
10	100	20	162.77	278.30	254.31	195.49	623.43	966.03	23.69	252.62	505.69	290.62	568.79	834.03
10	100	50	1151.59	2450.31	4898.91	3167.88	10803.39	17058.68	183.63	616.71	2110.29	6236.89	7902.28	9580.65
10	200	5	29.88	48.02	75.72	44.28	65.14	110.35	4.92	20.26	43.95	25.80	34.00	50.53
10	200	10	144.10	338.53	713.32	343.09	508.26	891.98	13.74	77.97	277.12	151.13	264.57	312.74
10	200	20	790.11	1710.94	2785.07	2904.90	4500.50	6568.19	122.02	528.48	1143.71	542.86	1785.06	2782.21

Tabela 3.21 – Tempos computacionais (em segundos)

T	n	m	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística Primal-Dual com Inicialização usando IP			Heurística Primal-Dual com Inicialização usando IP + Pesquisa Local		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
10	200	50	2698,40	12783,56	18453,68	15686,25	29510,98	58466,82	259,87	8244,00	19655,89	27446,93	39700,67	47021,97
10	500	5	127,68	355,15	685,68	188,68	426,35	713,16	12,83	212,42	418,70	98,03	285,72	445,98
10	500	10	631,33	1157,69	1916,48	1325,34	2056,20	3180,71	76,82	491,51	1039,58	641,18	1134,25	2214,31
10	500	20	9947,82	16868,06	28665,21	23422,00	32148,47	39152,14	646,60	2535,50	4657,12	5599,65	9902,68	14757,58
10	500	50	55877,62	87158,30	114482,84	121500,18	341679,19	439949,13	17890,18	75416,58	198013,84	43864,48	203211,04	355459,95
20	25	5	2,78	4,38	8,03	8,22	13,12	18,22	0,72	5,73	20,00	9,11	15,20	27,11
20	25	10	9,91	17,39	29,69	22,13	78,27	140,18	4,67	13,49	35,02	81,05	110,63	166,44
20	25	20	82,00	130,74	205,35	132,52	969,14	1402,61	15,16	72,76	160,07	737,85	999,77	1258,79
20	50	5	4,97	9,14	16,20	21,14	34,25	50,45	1,31	7,99	14,16	23,30	27,70	36,53
20	50	10	37,00	65,28	101,27	214,52	247,59	296,79	10,38	28,98	52,35	167,22	202,75	233,18
20	50	20	212,32	305,11	416,69	320,90	606,40	1390,17	41,61	102,97	153,30	242,15	1614,30	2699,95
20	100	5	11,66	4,22	77,72	71,85	89,46	113,54	4,30	19,33	56,05	27,56	79,09	131,49
20	100	10	86,47	278,47	471,61	567,86	893,76	1198,43	31,24	161,60	421,20	438,26	799,69	1214,46
20	100	20	1450,75	2752,20	4235,43	1868,81	9267,28	15822,16	200,12	1246,58	3065,46	1793,96	6798,73	11137,45
20	100	50	13708,00	21059,52	35063,17	15663,27	75838,17	165274,26	10281,00	26545,14	34361,91	10763,78	50408,33	158986,56
20	200	5	105,68	162,71	228,45	297,81	323,16	388,70	84,54	152,25	194,18	254,89	394,24	451,68
20	200	10	603,57	1054,88	1686,79	1137,32	2318,56	4018,93	513,03	1125,29	1653,05	1789,51	3986,43	5785,68
20	200	20	10263,45	18159,94	22563,39	16863,95	40523,63	55927,49	4105,38	12052,29	22789,03	12120,15	28521,85	45122,28
20	200	50	18888,80	26570,56	56299,04	32089,78	42896,57	65498,17	6611,08	32407,13	50106,14	16250,03	82518,15	99210,16
20	500	5	633,27	2145,54	3400,96	996,22	3164,43	4956,48	660,62	2131,80	2550,72	800,46	6485,75	9157,09
20	500	10	3799,64	12084,54	20405,78	5379,59	25832,22	34695,38	2849,73	13262,10	18365,20	23225,29	44010,65	77152,20
20	500	20	24869,55	51304,14	71663,04	50123,08	118529,91	139381,63	10445,21	52285,88	75246,19	158890,22	320097,28	320097,28
20	500	50	110637,68	198468,35	22676,01	218340,32	225739,35	227589,15	54212,46	210932,96	255114,33	169413,95	245868,22	265157,35
Resumo dos Valores			0,20	8948,27	22676,01	0,33	18654,38	439949,13	0,03	8386,40	255114,33	0,13	17099,09	355459,95

Tabela 3.21 – Tempos computacionais (em segundos) (cont.)

T	n	m	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	25	5	1.53	2.49	5.08	3.06	4.97	6.11	0.19	2.48	7.19
5	25	10	2.98	5.29	13.34	20.81	24.24	28.95	3.34	12.76	28.78
5	25	20	6.41	6.64	6.91	127.41	143.22	162.07	46.63	276.56	852.86
5	50	5	2.98	6.38	12.00	5.09	9.29	13.84	5.52	15.71	42.08
5	50	10	6.39	6.76	7.23	35.66	55.04	71.75	29.78	54.62	113.39
5	50	20	12.02	12.99	15.09	140.96	274.35	358.17	81.33	2081.09	7447.09
5	100	5	5.56	6.74	8.34	9.70	12.06	14.00	7.17	11.97	21.75
5	100	10	11.52	12.83	14.25	50.55	85.51	111.89	70.51	209.63	619.99
5	100	20	25.94	27.60	29.81	455.69	623.93	856.45	218.34	1532.67	1532.67
5	100	50	73.69	78.63	88.68	6758.69	7690.21	9363.03	9182.86	31019.28	59177.34
5	200	5	12.81	14.18	15.24	24.95	36.07	44.33	62.45	121.59	214.34
5	200	10	24.35	27.36	30.03	109.64	160.96	227.79	94.84	454.05	951.91
5	200	20	55.39	62.49	69.55	1204.09	1581.21	2116.01	972.22	5333.65	10604.09
5	200	50	190.79	210.14	256.18	16947.59	21096.12	25722.68	5344.91	127499.79	200098.61
5	500	5	11.54	33.38	40.58	53.46	73.78	91.08	162.48	366.43	868.34
5	500	10	88.08	92.97	97.69	244.71	375.52	525.92	847.89	1682.37	3734.39
5	500	20	209.73	228.93	245.77	2634.49	3806.99	4765.82	3912.63	19078.13	65750.22
5	500	50	673.01	698.55	743.24	48027.18	72579.75	85677.87	200031.30	200064.78	200126.44
10	25	5	4.74	5.13	5.81	19.05	32.07	50.75	3.03	12.26	31.75
10	25	10	9.59	10.30	11.22	215.79	317.89	398.45	30.45	118.87	355.52
10	25	20	21.33	22.43	22.97	1896.23	2386.58	2731.78	205.14	9983.48	25885.67
10	50	5	9.19	9.76	10.50	30.24	54.74	69.82	6.23	23.48	52.23
10	50	10	17.88	20.04	22.50	372.72	508.45	686.77	47.92	260.59	747.17
10	50	20	41.41	44.41	48.38	3523.69	4825.56	6280.89	406.52	10888.38	20202.53
10	100	5	19.80	21.65	23.53	86.52	144.65	224.59	36.48	230.03	517.28
10	100	10	40.31	43.33	46.25	695.91	940.62	1222.46	223.14	548.67	845.94
10	100	20	85.43	93.96	99.16	8165.39	9857.16	14093.44	2218.50	50962.52	200029.78
10	100	50	275.45	322.53	341.81	6328.70	38952.52	167314.85	121567.22	172208.70	200202.58
10	200	5	39.99	41.39	46.81	157.58	235.82	279.13	68.84	1112.95	3901.81

Tabela 3.22 – Tempos computacionais (em segundos)

T	n	m	Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
10	200	10	87,08	100,55	105,18	1681,84	2186,68	3062,90	1186,14	5574,84	10370,64
10	200	20	211,21	225,56	241,62	16712,11	23580,19	29049,70	6939,69	103152,79	200120,47
10	200	50	553,89	756,96	896,53	23578,11	56244,52	77055,15	200192,84	200277,92	200428,22
10	500	5	64,44	580,55	2521,21	447,92	18545,54	85219,55	519,70	1866,89	3208,50
10	500	10	502,37	2586,22	3976,91	2929,54	10425,55	12510,82	12906,09	89244,19	200227,27
10	500	20	6828,09	18683,53	26647,39	25584,79	45235,24	83380,34	116841,61	179875,60	200409,99
10	500	50	148916,44	150080,57	152871,61	200416,81	203762,28	208348,72	---	---	---
20	25	5	2,34	33,84	132,98	41,62	120,24	153,17	25,95	98,94	209,48
20	25	10	16,36	187,78	293,14	370,32	526,21	940,40	482,13	1281,91	2556,92
20	25	20	2,86	67,95	86,73	3371,25	5365,87	7112,17	3760,39	62446,97	163523,95
20	50	5	2,86	52,54	115,80	106,45	176,41	206,41	74,34	252,07	531,64
20	50	10	68,33	138,11	168,06	764,05	906,54	1317,47	396,89	8609,62	32815,39
20	50	20	240,11	255,61	261,98	1106,37	8545,42	15254,69	13779,34	87671,28	200092,05
20	100	5	29,91	123,78	469,26	125,94	421,58	742,91	179,94	510,24	951,38
20	100	10	229,71	936,41	3488,07	2002,43	5854,55	6861,70	1066,59	4422,62	10658,67
20	100	20	1018,50	7010,42	8534,66	8196,62	22612,36	62926,60	27390,05	105007,92	200471,97
20	100	50	51824,02	66985,45	99792,16	49179,70	154652,55	198274,06	---	---	---
20	200	5	526,15	848,32	1053,87	1164,59	2175,12	2551,99	1029,44	1219,48	1776,34
20	200	10	3075,75	5862,25	14863,60	8176,27	22786,54	32689,11	9815,08	66700,98	178804,48
20	200	20	8491,28	32542,24	81564,55	55376,97	97286,55	254940,86	210511,41	210511,41	210511,41
20	200	50	56634,58	99215,88	248584,02	74246,41	112503,24	260537,48	---	---	---
20	500	5	579,67	7512,25	16680,60	3657,29	34255,55	51737,57	---	---	---
20	500	10	16336,91	85453,24	147860,42	106116,33	126950,69	135909,29	---	---	---
20	500	20	54678,66	157000,44	182658,54	127608,46	168651,25	199136,33	---	---	---
20	500	50	117410,51	156421,56	166228,48	174052,52	193571,55	219449,75	---	---	---
Resumo dos Valores			1,53	14737,62	248584,02	3,06	27485,40	260537,48	0,19	37533,24	210511,41

Tabela 3.22 – Tempos computacionais (em segundos) (cont.)

<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>	<i>T</i>	<i>n</i>	<i>m</i>	<i>Tempo Computacional Médio</i>	<i>Relação</i>
5	25	5	2.35	4.31	10	100	10	94.39	1.00
5	25	10	3.98	1.62	10	100	20	574.49	0.92
5	25	20	12.55	1.54	10	100	50	2660.71	0.25
5	50	5	9.35	9.75	10	200	5	559.64	8.59
5	50	10	9.96	2.21	10	200	10	329.24	0.65
5	50	20	23.07	1.58	10	200	20	1724.92	0.38
5	100	5	9.07	2.40	10	200	50	16277.85	0.55
5	100	10	54.82	4.12	10	500	5	718.31	1.68
5	100	20	57.14	1.25	10	500	10	2247.82	1.09
5	100	50	392.12	0.50	10	500	20	19215.96	0.60
5	200	5	74.50	7.70	10	500	50	---	---
5	200	10	102.24	2.36	20	25	5	44.75	3.41
5	200	20	304.55	1.03	20	25	10	126.83	1.62
5	200	50	1353.32	0.36	20	25	20	257.95	0.27
5	500	5	272.90	3.40	20	50	5	64.05	1.87
5	500	10	476.68	1.58	20	50	10	186.49	0.75
5	500	20	2518.52	1.41	20	50	20	1323.76	2.18
5	500	50	6779.43	0.23	20	100	5	101.91	1.14
10	25	5	7.60	2.89	20	100	10	591.86	0.66
10	25	10	15.07	1.16	20	100	20	5592.08	0.60
10	25	20	86.51	0.98	20	100	50	---	---
10	50	5	11.45	2.47	20	200	5	1598.74	4.95
10	50	10	53.60	2.09	20	200	10	2887.51	1.25
10	50	20	140.12	0.79	20	200	20	---	---
10	100	5	140.09	9.86					
Valores Médios Globais								1523.70	2.22

Tabela 3.23 – Tempos computacionais quando o Cplex é utilizado como procedimento heurístico – restrições de capacidade máxima decrescente

O PLDAFR é resolvido, utilizando a heurística primal-dual, executando-se de seguida o procedimento primal em tudo semelhante ao Algoritmo 3-28 para que se atinja a admissibilidade da solução.

Na Tabela 3.18 e na Tabela 3.19 alguns dos valores apresentados são maiores do que 100%. Isto acontece em problemas que o Cplex não foi capaz de resolver, e não foi possível sequer encontrar uma solução admissível para a relaxação linear. Nestes casos o melhor limite inferior conhecido é aquele que é calculado pela própria heurística. Como pode ser observado na Tabela 3.20, a qualidade do limite inferior calculado pelas heurísticas é de muito má qualidade. É por este motivo que aparecem os valores maiores a 100%: nestes casos não é possível aferir a real qualidade das soluções primais calculadas.

As Tabela 3.21 e Tabela 3.22 mostram os tempos computacionais gastos pelas heurísticas e pelo Cplex. O Cplex não conseguiu resolver uma das cinco instâncias com (T,n,m) igual a $(20,100,20)$ e foi capaz de resolver apenas uma das cinco instâncias com (T,n,m) igual a $(20,200,20)$. O símbolo '---' é utilizado nos casos em que o Cplex não foi capaz de resolver nenhuma das cinco instâncias.

A Tabela 3.23 apresenta os tempos médios computacionais obtidos quando o Cplex é utilizado como um procedimento heurístico, com o objectivo de encontrar soluções de qualidade semelhante às encontradas pela heurística primal-dual. O limite de tempo considerado foi de 76 000 segundos, cerca de três vezes superior ao valor médio considerando os maiores tempos computacionais gastos pela heurística.

Da análise dos resultados computacionais podem retirar-se algumas conclusões:

1. A heurística primal-dual desenvolvida é capaz de gerar soluções primais de boa qualidade.
2. A inicialização das variáveis duais através da resolução de um problema de programação linear diminui a qualidade da solução primal calculada.
A heurística lagrangeana calcula, em média, soluções de pior qualidade quando comparadas com as geradas pela heurística primal-dual. Este resultado é substancialmente diferente daquilo que foi reportado para os problemas com restrições de capacidade máxima e restrições de capacidades máxima e mínima
3. À semelhança do que acontece nos problemas já estudados, a pesquisa local consegue melhorar de forma significativa a qualidade da solução primal, à custa de um aumento considerável nos tempos computacionais necessários.
4. O tempo computacional de que o Cplex necessita é, em média, dez vezes superior ao tempo computacional gasto pelas heurísticas. É também interessante notar que em muitos problemas o Cplex termina sem conseguir encontrar a solução óptima. Em média, calcula soluções que distam 0.02% do melhor limite inferior conhecido.
5. As heurísticas desenvolvidas não são capazes de fornecer limites inferiores de qualidade.
6. Apenas em 0.03% dos problemas se verifica ser a solução óptima da relaxação linear uma solução admissível (e portanto óptima) para o problema.

Quando o Cplex é utilizado como procedimento heurístico, em alguns casos conseguem obter-se tempos computacionais menores do que os obtidos aquando da execução da heurística primal-dual. No entanto, em média, os tempos computacionais são mais do que duas vezes superiores, e não se consegue encontrar soluções para alguns conjuntos de problemas

3.2.2.6. Serviços Constituídos por Vários Elementos

A heurística primal-dual desenvolvida para os problemas com restrições de capacidades em que se considera que um serviço pode ser constituído por mais do que um elemento de iguais ou diferentes dimensões é muito semelhante às heurísticas anteriormente descritas. O esquema de funcionamento da heurística é descrito no Algoritmo 3-29, sendo que as principais diferenças em relação ao Algoritmo 3-15 se prendem com a designação das variáveis duais que são, em alguns casos, diferentes devido às especificidades deste problema e à necessidade de se considerar um índice adicional que identifica a dimensão de cada elemento.

A heurística pára quando a solução primal óptima é encontrada ou quando não há melhorias no valor das funções objectivo primal e dual.

O procedimento dual ascendente para variáveis v_j^t , o procedimento de ajuste primal-dual para variáveis v_j^t , o procedimento de ajuste dual para variáveis π_i^t e ρ_i são os mesmos descritos nas secções anteriores. A única diferença reside no facto de, em vez de se considerar o conjunto I de possíveis localizações para os serviços, dever considerar-se um conjunto $I \times S$. No procedimento dual ascendente para as variáveis v_j^t e no procedimento de ajuste primal-dual para as mesmas variáveis, os custos de afectação a considerar são dados por $c_{ijs}^t + d_j^t \lambda_{is}^t$. Por este motivo os procedimentos referidos não se repetem neste momento.

Algoritmo 3-29

ESQUEMA DE FUNCIONAMENTO DA HEURÍSTICA PRIMAL-DUAL

1. Inicialização das variáveis duais.
 2. Procedimento Dual Ascendente para as variáveis duais v_j^t .
 3. Procedimento Primal.
 4. Procedimento de Ajuste Dual para variáveis ρ_i . Se a solução dual for alterada, ir para 2.
 5. Repetir o Procedimento de Ajuste Dual para as variáveis v_j^t até que não haja qualquer melhoria no valor da função objectivo dual.
 6. Procedimento de Ajuste Dual para as variáveis ρ_i . Se a solução dual for alterada, ir para 2.
 7. Procedimento Dual Ascendente para variáveis u_{ist}^s . Se a solução dual for alterada, ir para 2.
 8. Procedimento Dual Descendente para variáveis u_{ist}^s . Se a solução dual for alterada, ir para 2.
 9. Se $Nmax=1$, então executar o Procedimento de Ajuste Dual para variáveis π_i^t . Se a solução dual for alterada, ir para 2.
 10. Procedimento Dual Ascendente para variáveis λ_{is}^t . Se a solução dual for alterada, ir para 2.
 11. Procedimento Dual Descendente para variáveis λ_{is}^t . Se a solução dual for alterada, ir para 2.
-

O procedimento de ajuste dual para as variáveis π_i^t só é executado quando $Nmax$ é igual a 1. Os testes computacionais preliminares feitos mostraram que, em qualquer outra situação, a alteração desta variável dual não melhoraria o valor da função objectivo dual, devido ao coeficiente igual a $-Nmax$ com que esta variável contribui para a função objectivo.

Os procedimentos dual ascendente e dual descendente para as variáveis λ_{is}^t são também semelhantes aos já desenvolvidos para o problema com restrições de capacidade máxima. Nestes procedimentos basta considerar serviços (i,s) com capacidades máximas iguais a Q_s , em vez de serviços i com capacidades máximas iguais a Q_i . Também estes procedimentos não se irão repetir agora.

3.2.2.6.1. Inicialização das variáveis duais

As variáveis duais são inicializadas como descrito no Algoritmo 3-30.

Algoritmo 3-30
INICIALIZAÇÃO DAS VARIÁVEIS DUAIS
<p>c_{js}^t = custo de afectar o cliente j a um equipamento de dimensão s localizado em i, durante o período t; FA_{ist}^ξ = custo fixo de abrir pela primeira vez o serviço i localizando um equipamento de dimensão s que estará em funcionamento de t a ξ; FR_{ist}^ξ = custo fixo de localizar um equipamento de dimensão s em i que está em funcionamento de t a ξ; $Nmax$ – número máximo de equipamentos que podem estar simultaneamente em funcionamento em i.</p>
<ol style="list-style-type: none"> 1. $v_j^t \leftarrow \min_{i,s} \{c_{js}^t\}, \forall j, t; \pi_i^t \leftarrow 0, \forall i, t;$ 2. $u_{ist}^\xi \leftarrow \max \{0, -FR_{ist}^\xi\}, \forall i, s, t, \xi = t, \dots, T$ 3. $\rho_i \leftarrow \max \left\{ 0, -\min_{\substack{i,s \\ \xi \geq t}} \left(FA_{ist}^\xi - Nmax \sum_s \sum_{\tau=t}^T \sum_{\psi=\tau}^T u_{is\tau}^\psi \right) \right\}, \forall i$

3.2.2.6.2. Procedimento Primal

Considerem-se as seguintes definições, adaptadas das apresentadas anteriormente:

$$I^* = \{(i, s, \tau, \xi) : S_{is\tau}^\xi = 0\};$$

$$I_t^* = \{(i, s) : (i, s, \tau, \xi) \in I^* \text{ e } \tau \leq t \leq \xi\};$$

$$I_t^+ = \{(i, s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}; I^+ = \{(i, s, \tau, \xi) : a_{is\tau}^\xi = 1 \text{ ou } r_{is\tau}^\xi > 0\};$$

F_{is}^t = menor custo em que se incorre por abrir um equipamento de dimensão s em i durante o período t ;

$Num[i,s,t]$ = número total de equipamentos de dimensão s abertos em i durante o período t ;⁷

$M[i,t]$ = número total de equipamentos em funcionamento em i durante o período t .⁸

Está-se agora em condições de descrever o procedimento que constrói soluções primais admissíveis.

⁷ Representa o número total de elementos $(i,s,\tau,\xi) \in I^*$ tais que $\tau \leq t \leq \xi$.

⁸ Representa o número total de elementos $(i,s,\tau,\xi) \in I^*$ tais que $\tau \leq t \leq \xi, \forall s \in S$.

Algoritmo 3-31

PROCEDIMENTO PRIMAL

I - conjunto de todas as possíveis localizações para serviços; S - conjunto de possíveis dimensões para os equipamentos a localizar; T - número de períodos temporais considerados; $I^* = \{(i, s, \tau, \xi) : S_{i\tau}^\xi = 0\}$; $I_t^* = \{(i, s) : (i, s, \tau, \xi) \in I^* \text{ e } \tau \leq t \leq \xi\}$; $I_t^+ = \{(i, s) : \text{ pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$; $I^+ = \{(i, s, \tau, \xi) : a_{i\tau}^\xi = 1 \text{ ou } r_{i\tau}^\xi > 0\}$; F_{is}^t = menor custo em que se incorre por abrir um equipamento de dimensão s em i durante o período t ; $Num[i, s, t]$ = número total de equipamentos de dimensão s abertos em i durante o período t ; $M[i, t]$ = número total de equipamentos em funcionamento em i durante o período t ; c_{js}^t = custo de afectar o cliente j a um equipamento de dimensão s localizado em i , durante o período t ; Q_s = capacidade máxima de um equipamento de dimensão s , d_j^t = procura do cliente j durante o período t .

1. $I^+ \leftarrow \emptyset$. $I_t^+ \leftarrow \emptyset, \forall t$. Construir os conjuntos I^+ e I_t^+ . $Num[i, s, t] \leftarrow 0, \forall i, s, t$. $Num[i, t] \leftarrow 0, \forall i, t$.
2. Para $t=1$ até T , incluir em I_t^+ todos os pares $(i, s) \in I_t^+$, tais que $\exists j : v_j' \geq c_{js}^t$ e $v_j' < c_{i's'}^t, \forall (i', s') \neq (i, s)$.⁹
3. Para cada cliente j , tal que $v_j' < c_{js}^t, \forall (i, s) \in I_t^+$, incluir em I_t^+ o par (i, s) , tal que $c_{js}^t = \min_{v_j' \geq c_{i'j}^t} c_{i'j}^t$.
4. Construir o conjunto I^+ . Actualizar $I_t^+, \forall t$.
5. $t \leftarrow 1$;
6. $M[i, t] \leftarrow \sum_s Num[i, s, t], \forall i \in I$.
7. $D \leftarrow \sum_j d_j^t$; $C \leftarrow \sum_{(i, s) \in I_t^+} (Q_s Num[i, s, t])$. Se $D \leq C$, então ir para 10.
8. Calcular $F_{is}^t, \forall i \in I, s \in S$.
9. Se $F_{is}^t = +\infty, \forall i \in I, s \in S$, então ir para 13.
10. Calcular $F_{is}^{t'} = \begin{bmatrix} F_{is}^t \\ Q_s \end{bmatrix}, \forall i \in I, s \in S$, com $\phi_s = \begin{cases} D - C, & \text{se } C + Q_s < D \\ Q_s, & \text{em caso contrário} \end{cases}$.
11. Considerar o par (i', s') , tal que $F_{i's'}^{t'} = \min_{i \in I, s \in S} \{F_{is}^{t'}\}$.
12. $I_t^+ \leftarrow I_t^+ \cup \{(i', s')\}$. Reconstruir os conjuntos $I_t^+, \forall t$ e I^+ ; $C \leftarrow C + Q_s$; $Num[i, s, t] \leftarrow Num[i, s, t] + 1$; $M[i, t] \leftarrow M[i, t] + 1$. Se $D \leq C$, então ir para 19; se não, ir para 8.
13. Se $s = q$ para cada $(i, s, \tau, \xi) \in I^+$, com $\tau \leq t \leq \xi$, então parar. O procedimento não consegue encontrar uma solução admissível. Em caso contrário, ir para 14.
14. Se $D > C$, então, para cada $(i, s, \tau, \xi) \in I^+$, com $s < q$ e $\tau \leq t \leq \xi$, calcular $H_{i\tau}^\xi = \left(FR_{(s+1)\tau}^\xi - FR_{i\tau}^\xi \right) \begin{bmatrix} D - C \\ Q_{s+1} - Q_s \end{bmatrix}$.
15. Escolher $(i, s, \tau, \xi) \in I^+$ com $s < q$ e $\tau \leq t \leq \xi$ a que corresponde o menor $H_{i\tau}^\xi$.
16. $I^+ \leftarrow I^+ \setminus (i, s, \tau, \xi)$; $I^+ \leftarrow I^+ \cup \{(i, s+1, \tau, \xi)\}$; $Num[i, s, t] \leftarrow Num[i, s, t] - 1$;
17. $Num[i, s+1, t] \leftarrow Num[i, s+1, t] + 1$; $C \leftarrow C + Q_{s+1} - Q_s$.
18. Se $D \leq C$, então ir para 19; se não, ir para 13.
19. $t \leftarrow t + 1$; Se $t \leq T$, ir para 6; se não, ir para 20.
20. $t \leftarrow 1$;

⁹ Correspondem a serviços essenciais.

21. Resolver um problema de transportes, considerando o conjunto J de clientes como origens (com ofertas d_j^t), e como destinos os pares $(i, s) \in I_t^+$ (com procuras iguais a $Q_s \text{ Num}[i, s, t]$), e custos unitários de transporte dados por $\frac{c_{js}^t}{d_j^t}$.
 22. $t \leftarrow t + 1$; Se $t \leq T$ ir para 21; se não, ir para 23.
 23. Calcular os valores das variáveis primais a_{isr}^ξ e r_{isr}^ξ .
 24. Executar uma heurística de trocas locais.
-

Existem alguns passos deste algoritmo que merecem uma atenção especial.

O passo 4 deste algoritmo é muito semelhante ao passo 4 dos procedimentos primais anteriormente descritos. No entanto, existem algumas particularidades deste problema que deverão ser tidas em conta. Por este motivo, este passo é descrito formalmente como Algoritmo 3-32.

Algoritmo 3-32

PASSO 4 DO PROCEDIMENTO PRIMAL

m - número de possíveis localizações de serviços; q - número máximo de possíveis dimensões para os equipamentos; t - período temporal considerado; $I_t^+ = \{(i, s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$; $I^+ = \{(i, s, \tau, \xi) : a_{isr}^\xi = 1 \text{ ou } r_{isr}^\xi > 0\}$; $\text{Num}[i, s, t]$ = número total de equipamentos de dimensão s abertos em i durante o período t ; $M[i, t]$ = número total de equipamentos em funcionamento em i durante o período t ; FR_{isr}^ξ = custo fixo de localizar um equipamento de dimensão s em i que está em funcionamento de t a ξ .

1. $i \leftarrow 1$.
 2. $s \leftarrow 1$.
 3. Se $\exists t : (i, s) \in I_t^+$, ir para 4; se não, ir para 9.
 4. $t_1 \leftarrow \min\{\tau : (i, s) \in I_\tau^+\}$; $t_2 \leftarrow \max\{\tau : (i, s) \in I_\tau^+\}$.
 5. Calcular $\text{Num}[i, s, t]$ e $M[i, t]$, $\forall t$. $\Pi^+ \leftarrow I^+$. Executar o Procedimento Auxiliar 3 (Algoritmo 3-33).
 6. $I2^+ \leftarrow I^+$. Executar o Procedimento Auxiliar 4 (Algoritmo 3-34).
 7. $\text{sum1} \leftarrow \sum_{(i, s, \tau, \xi) \in I1^+} FR_{isr}^\xi$; $\text{sum2} \leftarrow \sum_{(i, s, \tau, \xi) \in I2^+} FR_{isr}^\xi$.
 8. Se $(\text{sum1} < \text{sum2})$ $I^+ \leftarrow \Pi^+$; se não, $I^+ \leftarrow I2^+$. Calcular $\text{Num}[i, s, t]$, $\forall t$.
 9. $s \leftarrow s + 1$. Se $s > q$, então ir para 10; se não, ir para 3.
 10. $i \leftarrow i + 1$. Se $i > m$, parar; se não, ir para 2.
-

Após a execução do passo 4, o conjunto I^+ tem tantos elementos (i, s, τ, ξ) quantos o número de equipamentos de dimensão s que estão em funcionamento em i desde o início do período temporal τ até ao fim do período temporal ξ . O Algoritmo 3-33 e o Algoritmo 3-34 são baseados no Algoritmo 3-6 e no Algoritmo 3-7, respectivamente. As principais diferenças prendem-se com o facto de ter de se considerar a possibilidade de vários equipamentos estarem simultaneamente em funcionamento, numa dada localização.

No passo 7 do procedimento primal testa-se a admissibilidade da solução primal construída, tendo em conta as restrições de capacidade máxima.

No passo 8, o cálculo de F_{is}^t considera todas as hipóteses de ter um equipamento de dimensão s aberto em i durante o período t : um novo

equipamento é localizado ou, então, são alterados os limites temporais inferiores e/ou superiores de uma variável de localização já considerada igual a um. O cálculo de F_{is}^t tem em conta a melhor escolha em termos dos custos fixos em que se incorre.

Defina-se:

$$F_i^\tau = \begin{cases} FA_{is\tau}^\xi - FR_{is\tau}^\xi, \forall s, \xi, & \text{se } \exists (i, s') \in I_i^+ : t < \tau, \forall s' \in S \\ 0, & \text{em caso contrário} \end{cases}$$

Este valor representa o custo fixo correspondente a localizar o primeiro equipamento em i no início do período temporal τ .

Algoritmo 3-33

PROCEDIMENTO AUXILIAR 3

$Num[i, s, t]$ = número total de equipamentos de dimensão s abertos em i durante o período t ; $M[i, t]$ = número total de equipamentos em funcionamento em i durante o período t ; $I_i^+ = \{(i, s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$; $I^* = \{(i, s, \tau, \xi) : S_{is\tau}^\xi = 0\}$; $Nmax$ – número máximo de equipamentos simultaneamente em funcionamento em cada localização i ; $t_1 \leftarrow \min\{\tau : (i, s) \in I_i^+\}$; $t_2 \leftarrow \max\{\tau : (i, s) \in I_i^+\}$.

-
1. $início \leftarrow 1$; $tempo \leftarrow t_1$.
 2. Se $tempo > t_2$, então parar; se não, ir para 3.
 3. Se $M[i, tempo] = Nmax$, então $tempo \leftarrow tempo + 1$ e ir para 2; se não, ir para 4.
 4. Se $(i, s) \in I_{tempo}^+$, então ir para 5; se não, $tempo \leftarrow tempo + 1$ e ir para 2.
 5. $\tau \leftarrow início$; $\xi \leftarrow T$; $t \leftarrow tempo$; $stop \leftarrow false$.
 6. Se $\tau > t$ ou $stop = true$, então ir para 14.
 7. Se $\xi < t$ ou $stop = true$, então $\tau \leftarrow \tau + 1$, $\xi \leftarrow T$, e ir para 6.
 8. Se $\exists (i, s, \tau, \xi) \in I^*$, então $\Pi^+ \leftarrow \Pi^+ \cup \{(i, s, \tau, \xi)\}$; se não, ir para 13.
 9. $k \leftarrow \tau$.
 10. Se $k > \xi$, então $tempo \leftarrow \xi + 1$ e $stop \leftarrow true$. Ir para 7.
 11. $Num[i, s, k] \leftarrow Num[i, s, k] + 1$; $M[i, k] \leftarrow M[i, k] + 1$. Se $M[i, k] = Nmax$ e $início \leq k$, então $início \leftarrow k + 1$.
 12. $k \leftarrow k + 1$ e ir para 10.
 13. $\xi \leftarrow \xi - 1$ e ir para 7.
 14. Se $stop = true$, então ir para 2; se não $\xi \leftarrow t_2$, $k \leftarrow tempo$ e ir para 15.
 15. Se $k > t_2$, então ir para 17.
 16. Se $M[i, k] = Nmax$, então $\xi \leftarrow k - 1$ e ir para 17; se não, $k \leftarrow k + 1$ e ir para 15.
 17. Se $tempo > \xi$, então $tempo \leftarrow tempo + 1$, $início \leftarrow \xi + 1$ e ir para 2; se não, ir para 18.
 18. $\Pi^+ \leftarrow \Pi^+ \cup \{(i, s, tempo, \xi)\}$, $k \leftarrow tempo$.
 19. $M[i, k] \leftarrow M[i, k] + 1$, $Num[i, s, k] \leftarrow Num[i, s, k] + 1$. Se $M[i, k] = Nmax$ e $início \leq k$, então $início \leftarrow k + 1$.
 20. $k \leftarrow k + 1$. Se $k \leq \xi$, então ir para 19; se não, ir para 21.
 21. $tempo \leftarrow tempo + 1$ e ir para 2.
-
-

Algoritmo 3-34

PROCEDIMENTO AUXILIAR 4

$Num[i,s,t]$ = número total de equipamentos de dimensão s abertos em i durante o período t ; $M[i,t]$ = número total de equipamentos em funcionamento em i durante o período t ; $I_t^+ = \{(i,s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$; $I^* = \{(i,s,\tau,\xi) : S_{i\tau}^\xi = 0\}$; $Nmax$ – número máximo de equipamentos simultaneamente em funcionamento em cada localização i ; $t_1 \leftarrow \min\{\tau : (i,s) \in I_t^+\}$; $t_2 \leftarrow \max\{\tau : (i,s) \in I_t^+\}$.

1. $fim \leftarrow T$; $tempo \leftarrow t_2$.
 2. Se $tempo < t_1$, então parar; se não, ir para 3.
 3. Se $M[i,tempo]=Nmax$, então $tempo \leftarrow tempo-1$ e ir para 2; se não, ir para 4.
 4. Se $(i,s) \in I_{tempo}^+$, então ir para 5; se não, $tempo \leftarrow tempo-1$ e ir para 2.
 5. $\tau \leftarrow 1$; $\xi \leftarrow fim$; $t \leftarrow tempo$; $stop \leftarrow false$.
 6. Se $\xi < t$ ou $stop=true$, ir para 14.
 7. Se $\tau > t$ ou $stop=true$, então $\xi \leftarrow \xi-1$, $\tau \leftarrow \tau-1$, e ir para 6; se não, ir para 8.
 8. Se $\exists (i,s,\tau,\xi) \in I^*$, então $I2^+ \leftarrow I2^+ \cup \{(i,s,\tau,\xi)\}$; se não, ir para 13.
 9. $k \leftarrow \tau$.
 10. Se $k > \xi$, então $tempo \leftarrow \tau-1$ e $stop \leftarrow true$. Ir para 7.
 11. $Num[i,s,k] \leftarrow Num[i,s,k] + 1$; $M[i,k] \leftarrow M[i,k] + 1$. Se $M[i,k]=Nmax$ e $fim \geq k$, então $fim \leftarrow k-1$.
 12. $k \leftarrow k+1$ e ir para 10.
 13. $\tau \leftarrow \tau+1$ e ir para 7.
 14. Se $stop=true$, então ir para 2; se não, $\tau \leftarrow t$, $k \leftarrow tempo$ e ir para 15.
 15. Se $k < \tau$, então ir para 17.
 16. Se $M[i,k]=Nmax$, então $\tau \leftarrow k+1$ e ir para 17; se não, $k \leftarrow k-1$ e ir para 15.
 17. Se $tempo < \tau$, então $tempo \leftarrow tempo-1$, $fim \leftarrow \tau-1$ e ir para 2; se não, ir para 18.
 18. $I2^+ \leftarrow I2^+ \cup \{(i,s,\tau,tempo)\}$, $k \leftarrow \tau$.
 19. $M[i,k] \leftarrow M[i,k]+1$, $Num[i,s,k] \leftarrow Num[i,s,k] + 1$. Se $M[i,k]=Nmax$ e $fim \geq k$, então $fim \leftarrow k-1$.
 20. $k \leftarrow k+1$. Se $k \leq tempo$, então ir para 19; se não, ir para 21.
 21. $tempo \leftarrow tempo-1$ e ir para 2.
-
-

Considere-se que o serviço i não se encontra em funcionamento durante o período t mas está em funcionamento em períodos temporais anteriores e posteriores a t , como exemplificado na Figura 3.1, página 86.

Defina-se ainda F_i^t como representando o custo fixo de localizar pela primeira vez um equipamento em i , considerando os conjuntos I_t^+ :

$$F_i^t = \begin{cases} F_i^\tau & \text{se } \exists (i,s') \in I_\tau^+, s' \in S \wedge \exists (i,s') \in I_t^+ : t < \tau, \forall s' \in S \\ 0, & \text{em caso contrário} \end{cases}$$

O Algoritmo 3-35 descreve o modo como se calcula o valor de F_{is}^t .

Algoritmo 3-35

PROCEDIMENTO DE CÁLCULO DE F_{is}^t

$M[i,t]$ = número total de equipamentos em funcionamento em i durante o período t ; $I_t^+ = \{(i,s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$; $Nmax$ – número máximo de equipamentos simultaneamente em funcionamento em cada localização i ;

$I^+ = \{(i, s, \tau, \xi) : a_{is\tau}^\xi = 1 \text{ ou } r_{is\tau}^\xi > 0\}$; $EA_{is\tau}^\xi =$ custo fixo de abrir pela primeira vez o serviço i , localizando um equipamento de dimensão s que estará em funcionamento de t a ξ , $FR_{is\tau}^\xi =$ custo fixo de localizar um equipamento de dimensão s em i , que está em funcionamento de t a ξ .

1. $F_{is}^t \leftarrow +\infty$. Se $N[i, t] = Nmax$ então parar, se não ir para 2.
2. $F_{is}^t \leftarrow \min \left\{ F_{is}^t, \begin{cases} \min \{ EA_{is\tau}^\xi - F_i^t : \tau \leq t \leq \xi \text{ e } N[i, k] < Nmax, k \in [\tau, \xi] \} , \\ \min \{ FR_{is\tau}^\xi : \tau \leq t \leq \xi \text{ e } N[i, k] < Nmax, k \in [\tau, \xi] \} , \text{ em caso contrário} \end{cases} \right\}^{10}$
3. Calcular $b = \max \left\{ 0, \max_{t' < t} \{ t' : (i, s) \in I_{t'}^+ \} \right\}$; $a = \{ t' : (i, s, t', b) \in I^+ \}$;
 $c = \min \left\{ T + 1, \min_{t' > t} \{ t' : (i, s) \in I_{t'}^+ \} \right\}$; $d = \{ t' : (i, s, c, t') \in I^+ \}$.
4. Se $b = 0$ e $c \leq T$, então
 $F_{is}^t \leftarrow \min \left\{ F_{is}^t, \min_{\tau} \{ FR_{is\tau}^d + F_i^t - FR_{isc}^d - F_i^c : \tau \leq t, N[i, k] < Nmax, k \in [\tau, c] \} \right\}$ e parar.¹¹ Caso contrário, ir para 5.
5. Se $c = T + 1$ e $b > 0$, então
 $F_{is}^t \leftarrow \min \left\{ F_{is}^t, \min_{\xi} \{ FR_{isa}^\xi - FR_{isb}^b : \xi \geq t, N[i, k] < Nmax, k \in]b, \xi] \} \right\}$ e parar. Caso contrário, ir para 6.
6. Se $c \leq T$ e $b > 0$ e $N[i, k] < Nmax, k \in]b, c[$ então $F_{is}^t \leftarrow \min \{ F_{is}^t, FR_{isa}^d - FR_{isc}^b - FR_{isc}^d \}$.

O passo 10 do procedimento primal penaliza os equipamentos cuja dimensão não é suficiente para cobrir a diferença existente entre a procura total e a capacidade máxima disponível (neste caso seria necessário abrir mais do que um equipamento numa ou mais localizações). O conjunto I^+ é atualizado, no passo 12 do procedimento, tendo em conta os limites inferiores e superiores que correspondem ao menor valor de F_{is}^t . É importante notar que, no conjunto I^+ , é possível ter vários elementos (i, s, τ, ξ) idênticos. Tal ocorre na situação em que existem vários equipamentos da mesma dimensão localizados em i que são abertos e fechados nos mesmos períodos de tempo.

Se não for possível encontrar uma solução admissível através da abertura de equipamentos que estejam em funcionamento durante t , os passos 13 a 18 tentam encontrar uma solução admissível através da alteração da dimensão dos equipamentos que estão já localizados. Se todos os equipamentos instalados forem de dimensão q , então não será possível encontrar uma solução admissível através deste procedimento. Caso contrário, para cada $(i, s, \tau, \xi) \in I^+$ com $s < q$ e $\tau \leq t \leq \xi$, o procedimento calcula o custo de alterar a dimensão s para $s+1$ (penalizando aquelas alterações que não forem suficientes para garantir a

¹⁰Se existir algum equipamento em funcionamento em i durante um período de tempo anterior ou igual a t , então o procedimento considera apenas custos de reabertura. Em caso contrário, o procedimento considera o custo de abertura pela primeira vez, mas “desconta” o custo fixo de abrir pela primeira vez um serviço no período $t > \tau$ (se for esse o caso).

¹¹Neste passo, o procedimento tenta calcular as alterações sentidas nos custos fixos em virtude de se fundirem dois intervalos: (τ, ξ) e (c, d) . Se o período c corresponde ao período em que se instalou pela primeira vez um equipamento em i , então $\mathcal{B}(i, s^1) \in I_{t'}^+, t < \tau, \forall s^1$, por isso F_i^t corresponde ao custo fixo de se localizar o primeiro equipamento em i . Se o serviço foi aberto pela primeira vez antes de c , então $F_i^c = 0$. O mesmo se aplica a $F_i^t = 0$.

admissibilidade primal).

Os passos 20 a 22 determinam os valores óptimos das variáveis de afectação, considerando os equipamentos em funcionamento em cada período de tempo, através da resolução de T problemas de transporte.

No passo 23 são calculados os valores das variáveis de localização $a_{is\tau}^{\xi}$ e $r_{is\tau}^{\xi}$. Este cálculo é feito de forma simples, tal como descrito no Algoritmo 3-36. Como os custos $FA_{is\tau}^{\xi}$ são iguais a $FR_{is\tau}^{\xi}$ mais os custos adicionais de se abrir pela primeira vez um equipamento em i , é indiferente qual o equipamento que se considera como i -first.

No passo 24, o procedimento tenta melhorar a solução admissível construída através da execução de uma heurística de trocas locais muito simples. Esta heurística tenta trocar a variável $a_{is\tau}^{\xi}$ (e $r_{is\tau}^{\xi}$) por $a_{is'\tau}^{\xi}$ ($r_{is'\tau}^{\xi}$), $\forall s' \neq s$ e $s' \in S$. A heurística escolhe então a troca que conduz à maior diminuição do valor da função objectivo primal. O processo repete-se, até que não seja possível melhorar a solução primal deste modo.

Algoritmo 3-36

PROCEDIMENTO QUE CALCULA OS VALORES DAS VARIÁVEIS DE LOCALIZAÇÃO

$a_{is\tau}^{\xi}$ e $r_{is\tau}^{\xi}$ - variáveis de localização cujo valor se vai determinar e que são inicialmente iguais a zero;
 $M[i,t]$ = número total de equipamentos em funcionamento em i durante o período t ;
 $I^+ = \{(i,s,\tau,\xi) : a_{is\tau}^{\xi} = 1 \text{ ou } r_{is\tau}^{\xi} > 0\}$; $I_i^+ = \{(i,s) : \text{pelo menos um equipamento de dimensão } s \text{ está aberto em } i \text{ durante } t\}$.

1. Se $M[i,t] = 0, \forall t$, então parar.
 2. Escolher arbitrariamente um elemento $(i,s,\tau,\xi) \in I^+$, tal que $\tau = \min\{t : \exists (i,s) \in I_i^+\}$.
 3. Fixar $a_{is\tau}^{\xi} = 1$. $I^+ \leftarrow I^+ \setminus \{(i,s,\tau,\xi)\}$.
 4. A variável $r_{is\tau}^{\xi}$ é igual ao número de elementos $(i,s,\tau,\xi) \in I^+$.
-

3.2.2.6.3. Procedimento Dual Ascendente para as Variáveis u_{ist}^{ξ}

A observação de (2-60) e (2-61) mostra que o aumento do valor da variável dual u_{ist}^{ξ} pode levar ao aumento das *slacks* SR_{ist}^{ξ} , mas diminui as *slacks* $SA_{is\tau}^{\xi}$, $\tau \leq t, \forall s' \in S$. Só valerá a pena aumentar *slacks* SR_{ist}^{ξ} tais que $SR_{ist}^{\xi} = 0$ e $SA_{is\tau}^{\xi} > 0$; caso contrário, o valor de S_{ist}^{ξ} não seria alterado o que inviabilizaria alterações nos valores das variáveis duais v_j^t . Este procedimento é descrito como Algoritmo 3-37.

Algoritmo 3-37

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS u_{ist}^{ξ}

1. $i \leftarrow 1$.
2. $s \leftarrow 1$.
3. $t \leftarrow 1$.
4. $\xi \leftarrow t$.

5. $\Delta u_{ist}^{\xi} \leftarrow 0$.
 6. Se $SR_{ist}^{\xi} = 0$ e $SA_{ist}^{\xi} > 0$, então $\Delta u_{ist}^{\xi} \leftarrow \frac{SA_{ist}^{\xi}}{2}$; se não, ir para 10.
 7. $\Delta u_{ist}^{\xi} \leftarrow \min \left\{ \Delta u_{ist}^{\xi}, \min_{\substack{s \in S \\ \tau \leq t \\ \psi \geq \tau}} SA_{is'\tau}^{\psi} \right\}$, $\Delta u_{is'\tau}^{\xi} \leftarrow \frac{\Delta u_{is'\tau}^{\xi}}{Nmax}$. Se $\Delta u_{ist}^{\xi} = 0$, então ir para 10; se não, ir para 8.
 8. $SR_{ist}^{\xi} \leftarrow SR_{ist}^{\xi} + \Delta u_{ist}^{\xi}$, $SA_{is'\tau}^{\xi} \leftarrow SA_{is'\tau}^{\xi} - \Delta u_{is'\tau}^{\xi} Nmax$, $\tau \leq t$, $\forall s' \in S$, $u_{ist}^{\xi} \leftarrow u_{ist}^{\xi} + \Delta u_{ist}^{\xi}$.
 9. Executar o procedimento dual ascendente para as variáveis v_j^i (Algoritmo 3-4) com $J^* = j$.
 10. $\xi \leftarrow \xi + 1$. Se $\xi > T$, então $t \leftarrow t + 1$ e ir para 11. Caso contrário, ir para 5.
 11. Se $t > T$, então $s \leftarrow s + 1$ e ir para 12; se não, ir para 4.
 12. Se $s > q$, então $i \leftarrow i + 1$ e ir para 13; se não, ir para 3.
 13. Se $i > m$, então parar; se não, ir para 2.
-

No passo 6 deste procedimento, Δu_{ist}^{ξ} toma o valor $\frac{SA_{ist}^{\xi}}{2}$ porque se tomasse o valor SA_{ist}^{ξ} esta *slack* ficaria igual a zero no passo 8, diminuindo muito as possibilidades de melhorar o valor da função objectivo dual. Se $SA_{ist}^{\xi} = \min \left\{ SA_{is'\tau}^{\psi} \right\}_{\substack{s \in S \\ \tau \leq t \\ \psi \geq \tau}}$, então SA_{ist}^{ξ} e SR_{ist}^{ξ} terão o mesmo valor no fim do passo 8 do procedimento.

3.2.2.6.4. Procedimento Dual Descendente para as Variáveis u_{ist}^{ξ}

Ao diminuir-se o valor da variável dual u_{ist}^{ξ} diminui-se o valor da *slack* SR_{ist}^{ξ} , mas aumenta-se o valor das *slacks* $SA_{is'\tau}^{\xi}$, $\tau \leq t$, $\forall s' \in S$. Se for possível aumentar o valor de uma *slack* que está a bloquear uma ou mais variáveis duais v_j^i , então será possível melhorar o valor da função objectivo dual.

Algoritmo 3-38

PROCEDIMENTO DUAL DESCENDENTE PARA AS VARIÁVEIS u_{ist}^{ξ}

1. $i \leftarrow 1$.
 2. $s \leftarrow 1$.
 3. $t \leftarrow 1$.
 4. $\xi \leftarrow t$.
 5. $\Delta u_{ist}^{\xi} \leftarrow \min \left\{ u_{ist}^{\xi}, \frac{SR_{ist}^{\xi}}{2} \right\}$. Se $\Delta u_{ist}^{\xi} = 0$, então ir para 8; se não, ir para 6.
 6. $SR_{ist}^{\xi} \leftarrow SR_{ist}^{\xi} - \Delta u_{ist}^{\xi}$; $SA_{is'\tau}^{\xi} \leftarrow SA_{is'\tau}^{\xi} + \Delta u_{is'\tau}^{\xi} Nmax$, $\tau \leq t$ e $\psi \geq \tau$, $\forall s' \in S$; $u_{ist}^{\xi} \leftarrow u_{ist}^{\xi} - \Delta u_{ist}^{\xi}$.
 7. Executar o procedimento dual descendente para variáveis v_j^i (Algoritmo 3-4) com $J^* = j$.
 8. $\xi \leftarrow \xi + 1$. Se $\xi > T$, então $t \leftarrow t + 1$ e ir para 11; se não, ir para 5.
 9. Se $t > T$, então $s \leftarrow s + 1$ e ir para 12; se não, ir para 4.
 10. Se $s > q$, então $i \leftarrow i + 1$ e ir para 13; se não, ir para 3.
 11. Se $i > m$, então parar; se não, ir para 2.
-

No passo 5 deste procedimento, Δu_{ist}^e toma o menor valor entre u_{ist}^e e $\frac{SR_{ist}^e}{2}$ pelas mesmas razões apontadas anteriormente para o procedimento dual ascendente.

3.2.2.7. Resultados Computacionais

À semelhança do que foi descrito anteriormente, foram gerados problemas de teste de forma aleatória, tendo como objectivo aferir o desempenho da heurística primal-dual, e compará-la com o desempenho de um *general solver*, o Cplex (v. 9.0). O algoritmo utilizado para gerar estes problemas teste é em tudo semelhante ao desenvolvido para gerar os problemas de localização com restrições de capacidade, com as seguintes especificidades: devem ser introduzidos pelo utilizador os dados referentes ao número de dimensões para os equipamentos, e ao número máximo de equipamentos permitidos em cada localização; o custo fixo associado a um equipamento de uma determinada dimensão é calculado aleatoriamente como sendo entre 20% a 80% mais elevado do que o custo fixo associado à dimensão imediatamente anterior, acontecendo o inverso com os custos unitários de funcionamento: a equipamentos de menor dimensão correspondem custos maiores do que os associados aos equipamentos de maior dimensão. Foram gerados problemas com a seguinte dimensão: $T = 5$ ou 10; $m = 10$ ou 20; $n = 50$ ou 100; $q = 2$ ou 3; $Nmax = 1$ ou 2. Para cada combinação de parâmetros foram gerados cinco problemas de teste, num total de 160 problemas.

A Tabela 3.24 mostra a qualidade da solução primal encontrada pela heurística primal-dual sem e com a execução da pesquisa local e pela heurística lagrangeana (que considera a relaxação das restrições de capacidade associadas aos elementos a localizar), sem e com a execução da pesquisa local; a Tabela 3.25 mostra a qualidade da solução dual obtida pela heurística primal-dual e pela heurística lagrangeana e a Tabela 3.26 mostra os tempos computacionais obtidos quando se executaram: a heurística primal-dual sem e com a execução da pesquisa local; a heurística lagrangeana, sem e com a execução da pesquisa local; o Cplex. A Tabela 3.27 mostra os resultados quando se utiliza o Cplex como procedimento heurístico, à semelhança do que foi feito anteriormente. O limite de tempo utilizado foi de 600 segundos, cerca de três vezes superior ao maior tempo computacional gasto pela heurística. Optou por incluir-se uma coluna que indica, para cada combinação de parâmetros, o número de problemas para os quais não foi encontrada nenhuma solução primal com a qualidade desejada.

Dos resultados apresentados pode concluir-se que a heurística primal-dual é capaz de encontrar boas soluções primais, tendo um desempenho superior à heurística lagrangeana. A heurística é muito mais rápida do que o *general solver*, e mesmo quando este é utilizado como procedimento heurístico demora, em média, mais de vinte vezes a encontrar uma solução de qualidade equivalente à construída pela heurística, não sendo mesmo capaz de o fazer em muitas das instâncias consideradas. As soluções duais encontradas não são, em média, de boa qualidade.

<i>T</i>	<i>m</i>	<i>n</i>	<i>q</i>	<i>Nmax</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local		
					Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	10	50	2	1	0.00	1.29	4.25	0.00	1.29	4.25	0.00	2.10	5.24	0.00	1.09	5.24
5	10	50	2	2	0.31	3.83	8.17	0.31	2.67	4.94	0.31	1.83	4.51	0.31	1.83	4.51
5	10	50	3	1	2.33	3.14	3.68	0.00	2.57	3.68	0.00	3.00	5.92	0.00	2.28	3.91
5	10	50	3	2	2.02	3.62	7.06	1.84	3.55	7.06	0.46	3.79	7.69	0.46	3.79	7.69
5	10	100	2	1	0.00	3.03	7.72	0.00	0.93	2.02	0.00	4.08	10.27	0.00	1.89	5.03
5	10	100	2	2	0.75	2.58	4.20	0.71	2.57	4.20	0.40	3.02	5.15	0.40	3.02	5.15
5	10	100	3	1	0.00	5.85	9.72	0.00	3.93	7.11	0.04	5.85	12.79	0.04	3.78	6.76
5	10	100	3	2	0.68	2.89	5.57	0.48	2.76	5.23	0.96	3.58	5.58	0.96	3.58	5.58
5	20	50	2	1	0.89	2.59	6.15	0.89	2.59	6.15	0.02	2.50	9.08	0.02	2.50	9.08
5	20	50	2	2	0.55	2.85	5.21	0.55	2.61	4.01	2.14	3.96	5.59	2.14	3.96	5.59
5	20	50	3	1	2.43	5.02	7.61	2.07	4.12	5.98	0.44	3.95	8.98	0.44	3.00	4.42
5	20	50	3	2	1.71	3.65	5.84	1.71	3.55	5.84	4.19	6.22	9.36	4.19	6.22	9.36
5	20	100	2	1	2.27	4.23	6.33	0.35	2.79	6.33	3.58	4.47	6.64	1.75	3.17	4.46
5	20	100	2	2	0.66	1.92	2.72	0.66	1.78	2.72	1.76	2.20	3.01	1.76	2.20	3.01
5	20	100	3	1	1.31	4.27	7.89	0.00	3.49	5.60	1.03	3.76	9.56	0.00	2.55	4.83
5	20	100	3	2	2.40	4.59	6.73	1.41	4.24	6.73	3.32	5.85	10.10	3.32	5.85	10.10
10	10	50	2	1	0.01	2.35	4.34	0.01	2.34	4.34	0.01	1.84	3.59	0.01	1.83	3.59
10	10	50	2	2	0.49	1.90	3.70	0.49	1.41	3.42	0.70	4.36	9.20	0.70	4.36	9.20
10	10	50	3	1	2.01	4.37	5.69	1.03	2.10	3.27	0.57	2.84	5.26	0.57	1.91	2.63
10	10	50	3	2	1.15	3.13	5.72	0.99	2.71	4.64	1.79	3.27	7.12	1.79	3.13	6.44
10	10	100	2	1	0.01	1.57	2.69	0.01	1.00	2.20	1.45	2.32	2.76	0.44	1.39	2.76
10	10	100	2	2	0.81	2.09	2.72	0.81	1.79	2.59	2.45	3.87	7.67	2.03	3.66	7.67
10	10	100	3	1	2.69	4.07	5.91	0.49	1.79	3.60	1.74	4.09	5.91	0.49	1.85	2.91
10	10	100	3	2	0.95	3.91	6.97	0.95	3.24	6.97	1.32	6.10	12.75	1.32	6.10	12.75
10	20	50	2	1	1.45	2.60	3.27	0.60	2.19	3.27	1.45	3.35	5.30	0.60	2.70	5.30
10	20	50	2	2	1.42	5.42	15.91	0.53	5.10	15.91	1.14	6.33	18.69	1.14	6.33	18.69
10	20	50	3	1	0.76	4.12	7.57	0.18	3.35	7.21	0.76	5.45	8.36	0.76	5.22	8.36
10	20	50	3	2	0.86	4.37	7.48	0.71	3.39	7.48	1.37	5.39	12.82	1.37	4.76	12.82
10	20	100	2	1	3.04	4.14	5.15	2.11	3.03	4.64	2.75	3.99	6.41	1.25	3.13	6.41
10	20	100	2	2	1.73	3.70	5.48	1.40	2.98	5.48	2.97	5.27	11.11	2.97	5.27	11.11
10	20	100	3	1	2.22	4.84	6.68	0.06	1.54	2.70	1.96	6.87	11.44	0.47	2.89	5.30
10	20	100	3	2	2.18	4.68	7.00	1.04	3.75	7.00	3.57	6.63	12.19	1.14	6.15	12.19
Resumo dos Valores					0.00	3.52	15.91	0.00	2.72	15.91	0.00	4.13	18.69	0.00	3.48	18.69

Tabela 3.24 – Qualidade da solução primal (em porcentagem) – serviços constituídos por vários elementos

<i>T</i>	<i>m</i>	<i>n</i>	<i>q</i>	<i>Nmax</i>	Heurística Primal-Dual			Heurística Lagrangeana		
					<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	10	50	2	1	3.61	10.68	19.37	4.03	9.87	20.88
5	10	50	2	2	11.98	15.91	21.91	2.44	14.29	31.30
5	10	50	3	1	11.48	15.05	26.47	10.10	13.81	16.10
5	10	50	3	2	7.36	16.07	23.48	1.94	14.28	24.24
5	10	100	2	1	5.27	11.70	23.05	7.09	15.46	30.29
5	10	100	2	2	9.28	17.63	27.62	11.24	24.44	38.13
5	10	100	3	1	6.29	13.10	19.41	8.57	14.05	23.29
5	10	100	3	2	8.89	11.85	16.63	7.63	13.48	19.63
5	20	50	2	1	9.57	14.97	20.49	1.48	13.13	26.60
5	20	50	2	2	8.55	14.59	19.47	1.54	18.11	30.05
5	20	50	3	1	8.11	15.35	20.44	11.06	16.80	26.21
5	20	50	3	2	9.76	11.78	14.95	12.50	16.06	23.22
5	20	100	2	1	6.26	11.06	17.86	4.88	10.66	15.84
5	20	100	2	2	8.62	12.17	15.97	11.12	15.80	20.50
5	20	100	3	1	7.62	16.06	21.12	11.53	16.19	24.43
5	20	100	3	2	11.85	15.86	20.25	14.19	21.83	31.02
10	10	50	2	1	5.77	10.94	14.75	6.15	8.17	11.51
10	10	50	2	2	5.43	9.68	19.14	1.95	10.77	24.18
10	10	50	3	1	5.43	9.86	15.57	7.74	10.31	14.38
10	10	50	3	2	4.54	6.84	8.83	1.30	4.77	8.00
10	10	100	2	1	5.84	8.75	15.43	6.63	10.92	18.11
10	10	100	2	2	2.64	13.35	34.46	0.07	6.77	12.54
10	10	100	3	1	6.28	8.28	12.20	7.61	9.53	10.72
10	10	100	3	2	5.82	10.49	19.55	0.60	5.21	7.69
10	20	50	2	1	5.32	10.64	19.08	1.03	7.09	9.76
10	20	50	2	2	6.49	10.92	13.73	3.72	10.13	17.66
10	20	50	3	1	10.52	14.18	18.19	0.28	7.17	10.74
10	20	50	3	2	5.17	7.59	14.47	0.10	7.06	17.64
10	20	100	2	1	5.01	9.82	12.64	7.39	10.40	12.85
10	20	100	2	2	4.44	11.14	25.54	0.12	9.31	31.54
10	20	100	3	1	5.63	8.72	11.32	7.45	10.64	13.81
10	20	100	3	2	7.14	11.11	16.92	4.65	12.22	21.10
Resumo dos Valores					2.64	12.07	34.46	0.07	12.15	38.13

Tabela 3.25 – Qualidade da solução dual (em percentagem) – serviços constituídos por vários elementos

T	m	n	q	Nmax	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Complex		
					Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	10	50	2	1	0.13	0.36	0.84	0.41	0.57	0.97	0.44	1.09	1.73	0.88	1.73	2.25	0.44	46.22	191.45
5	10	50	2	2	0.36	0.67	1.19	0.70	1.05	1.30	0.95	2.62	3.47	1.91	3.47	4.39	21.48	2260.93	10983.00
5	10	50	3	1	0.22	0.49	1.05	0.59	0.87	1.28	2.38	5.25	11.02	3.73	6.51	12.48	9.91	56.02	135.92
5	10	50	3	2	0.16	1.17	2.70	1.00	1.82	3.11	3.31	8.85	24.80	5.56	10.81	26.48	113.06	267.60	693.34
5	10	100	2	1	0.50	0.68	0.99	1.17	1.49	1.91	2.08	3.82	6.30	4.17	5.63	7.14	24.98	60.58	105.72
5	10	100	2	2	1.42	2.65	4.34	2.39	3.26	4.44	3.28	7.17	10.44	6.08	8.83	10.77	30.73	76.09	2767.20
5	10	100	3	1	0.55	1.23	3.55	1.45	2.97	4.16	6.06	7.85	11.08	8.77	12.75	17.25	30.67	55.494	866.27
5	10	100	3	2	0.30	1.60	4.81	1.83	4.13	6.42	8.50	14.83	22.19	15.31	21.00	27.27	2259.44	6472.31	12276.30
5	20	50	2	1	0.28	1.20	2.19	0.98	1.89	2.83	3.41	9.92	25.80	6.30	12.50	28.17	41.25	587.75	2714.19
5	20	50	2	2	0.53	1.93	4.61	1.53	4.38	10.95	5.67	13.08	34.30	9.89	17.24	39.36	204.48	3782.21	10249.77
5	20	50	3	1	0.81	2.08	3.89	3.39	5.37	9.48	4.06	12.25	19.06	12.11	20.34	26.73	58.09	428.30	792.27
5	20	50	3	2	1.27	2.86	4.78	3.78	5.72	7.08	8.00	12.90	18.61	17.41	22.06	29.06	4633.76	15678.92	28898.99
5	20	100	2	1	1.02	1.73	2.36	2.36	5.33	9.41	5.84	13.70	17.77	11.97	22.17	29.06	85.83	104.20	159.76
5	20	100	2	2	1.44	4.21	8.19	4.42	7.89	12.97	12.83	17.64	22.88	23.92	28.28	31.92	2593.66	37562.57	137829.75
5	20	100	3	1	2.50	6.15	10.81	9.13	11.51	13.75	3.97	30.95	65.36	22.61	50.13	76.23	97.13	501.49	1943.77
5	20	100	3	2	4.78	16.17	38.41	10.69	22.12	41.91	25.53	43.03	69.56	45.00	64.46	84.14	2544.06	30310.08	69891.59
10	10	50	2	1	0.24	1.19	4.34	0.75	1.97	4.61	1.81	9.88	18.55	7.13	13.72	20.17	7.67	7810.15	1330.44
10	10	50	2	2	0.16	1.02	2.42	1.44	3.10	6.38	3.34	8.79	13.17	11.92	15.59	19.53	124.61	810.15	18430.33
10	10	50	3	1	0.97	1.68	3.02	4.09	7.08	10.97	7.58	17.13	31.53	25.00	30.50	42.14	62.70	693.62	2195.17
10	10	50	3	2	0.23	2.21	5.75	2.53	6.56	12.99	5.66	25.47	51.92	18.81	45.88	68.67	1563.01	32558.67	98431.45
10	10	100	2	1	0.58	1.82	3.05	1.95	6.87	9.94	7.00	8.80	12.17	19.50	23.65	27.44	374.89	796.45	30588.27
10	10	100	2	2	0.47	3.80	9.42	4.53	9.03	18.84	14.99	24.93	43.89	30.64	45.75	63.55	29.14	132207.80	8431.45
10	10	100	3	1	2.42	5.25	9.38	10.69	16.83	27.58	20.16	34.74	59.44	53.45	71.04	90.50	204.14	837.88	3237.26
10	10	100	3	2	3.03	8.19	13.31	7.84	24.36	57.11	37.81	73.96	122.17	80.74	115.02	158.83	3455.34	125059.55	224544.61
10	20	50	2	1	0.97	3.04	6.22	7.36	11.80	20.81	10.52	31.05	91.94	28.59	56.49	116.31	131.97	974.98	2910.52

Tabela 3.26 – Tempos computacionais (em segundos) – serviços constituidos por vários elementos

<i>T</i>	<i>m</i>	<i>n</i>	<i>q</i>	<i>Nmax</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Heurística Lagrangeana			Heurística Lagrangeana + Pesquisa Local			Cplex		
					<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
10	20	50	2	2	0.22	3.92	7.09	7.14	10.96	14.55	12.56	36.69	92.50	39.72	64.49	121.05	0.00	3445.21	6486.00
10	20	50	3	1	2.42	9.91	21.92	8.31	26.66	60.92	35.38	126.08	315.61	74.63	169.03	357.94	319.56	1284.30	7694.13
10	20	50	3	2	1.39	6.32	12.42	17.63	32.17	65.58	33.69	53.67	78.74	106.81	147.20	257.77	43718.31	87540.41	175002.00
10	20	100	2	1	2.17	7.34	12.66	22.45	39.63	58.19	14.98	43.13	72.67	90.56	126.27	193.39	240.38	53154.97	103354.49
10	20	100	2	1	2.56	12.26	24.94	36.14	63.64	121.58	37.03	92.88	135.97	125.77	173.30	224.84	904.22	43921.56	175005.77
10	20	100	3	1	7.66	27.85	69.72	49.09	131.36	192.66	43.33	82.96	163.58	212.74	310.83	503.59	1343.20	11208.79	39835.94
10	20	100	3	2	6.11	33.07	84.38	62.22	132.34	203.91	104.20	124.02	155.75	262.97	321.72	453.52	15039.22	80031.52	175004.94
Resumo dos Valores					0.13	5.44	84.38	0.41	18.90	203.91	0.44	31.22	315.61	0.88	63.70	503.59	0.00	18096.13	224544.61

Tabela 3.26 – Tempos computacionais (em segundos) – serviços constituídos por vários elementos (cont.)

<i>T</i>	<i>m</i>	<i>n</i>	<i>q</i>	<i>Nmax</i>	<i>Tempo Computacional Médio</i>	<i>n° problemas que não resolve</i>	<i>Relação</i>
5	10	50	2	1	40.50	0	71.57
5	10	50	2	2	118.87	0	113.17
5	10	50	3	1	34.08	0	39.23
5	10	50	3	2	83.92	0	46.22
5	10	100	2	1	36.44	0	24.45
5	10	100	2	2	31.76	1	9.74
5	10	100	3	1	149.49	1	50.25
5	10	100	3	2	---	5	---
5	20	50	2	1	12.89	0	6.81
5	20	50	2	2	127.76	0	29.18
5	20	50	3	1	70.94	0	13.21
5	20	50	3	2	---	5	---
5	20	100	2	1	38.37	0	7.21
5	20	100	2	2	121.12	3	15.34
5	20	100	3	1	124.14	0	10.79
5	20	100	3	2	318.12	2	14.38
10	10	50	2	1	56.13	0	28.56
10	10	50	2	2	126.73	0	40.92
10	10	50	3	1	34.61	0	4.89
10	10	50	3	2	367.44	4	56.02
10	10	100	2	1	234.29	0	34.09
10	10	100	2	2	112.66	2	12.47
10	10	100	3	1	172.40	0	10.24
10	10	100	3	2	378.72	4	15.55
10	20	50	2	1	47.33	0	4.01
10	20	50	2	2	54.56	0	4.98
10	20	50	3	1	200.13	1	7.51
10	20	50	3	2	273.56	3	8.50
10	20	100	2	1	113.11	0	2.85
10	20	100	2	2	284.71	1	4.47
10	20	100	3	1	249.03	4	1.90
10	20	100	3	2	---	5	---
Valores Médios Globais					138.41		23.74

Tabela 3.27 – Tempos Computacionais quando o Cplex é utilizado como procedimento heurístico – serviços com vários elementos

3.3. Problemas de Localização Multi-Nível

Uma das primeiras tentativas de resolver problemas de localização dinâmica multi-nível deve-se a Erlenkotter, 1975, que utiliza programação dinâmica para resolver um problema de planeamento de capacidades para várias localizações. Sweeney & Tatham, 1976, fazem também uso da programação dinâmica para desenvolverem um procedimento dedicado a problemas de localização dinâmica, recorrendo à resolução de problemas estáticos. Tcha & Lee, 1984, apresentam um algoritmo de *branch-and-bound* baseado num procedimento dual ascendente (semelhante ao desenvolvido por Erlenkotter, 1978), para o problema de localização multi-nível sem capacidades. Dez anos mais tarde, um artigo por Barros & Labbé, 1994, prova que este problema não é *sub modular*, pelo que questionam a validade do procedimento de *branch-and-bound* descrito por Tcha e Lee. Ro & Tcha, 1984, descrevem um algoritmo de *branch-and-bound* para a resolução do problema de localização com 2 níveis, sem capacidades, com restrições adicionais que representam as relações próximas de alguns armazéns com determinados equipamentos produtivos (se o equipamento estiver aberto, então existe um conjunto de armazéns que também vão ter de estar abertos). Madsen, 1983, considera simultaneamente o problema de localização e o problema de determinação de rotas óptimas quando existem clientes, fábricas e armazéns, referindo alguns métodos de resolução. Narula & Ogbu, 1985, utilizam relaxação lagrangeana e o método do subgradiente para resolver um problema de localização com dois níveis e sem capacidades. Li-Lian Gao & Robinson, 1992, investigam a utilização de procedimentos duais semelhantes aos desenvolvidos por Erlenkotter, 1978, na resolução de problemas com dois níveis. Andreas Klose, 1995, 1998, 2000, utiliza relaxação lagrangeana aplicada ao problema de localizar depósitos, sendo preciso considerar a afectação dos clientes aos depósitos e os fluxos dos equipamentos produtivos para os depósitos. Bloemhof-Ruwaard et al., 1996, descrevem heurísticas baseadas no arredondamento da solução óptima da relaxação linear de um problema de dois níveis com capacidades. Tragantalerngsak *et al.*, 1997, estudam relaxações lagrangeanas, aplicando o método do subgradiente, e heurísticas baseadas numa formulação de *general assignment* no problema de dois níveis, capacitado e com afectação total. Os mesmos autores apresentam também um algoritmo de *branch-and-bound* para o mesmo problema (Tragantalerngsak, Holt & Rönnqvist, 2000). Alminyana, Borrás & Pastor, 1998, descrevem uma heurística que resolve o problema da *pq*-mediana. P. Chardaire, 1999, utiliza relaxação lagrangeana e *simulated annealing* para resolver problemas de planeamento de redes de telecomunicações com dois níveis sem e com capacidades. A. Klose, 1999, apresenta uma heurística baseada na relaxação linear de um problema de localização com dois níveis e afectação total, mas em que as localizações dos serviços no segundo nível são previamente determinadas. Marín & Pelegrín, 1999, comparam a utilização de relaxações lagrangeanas a dois modelos diferentes de problemas de localização com dois níveis: um modelo utiliza variáveis de decisão com dois índices e, o outro, variáveis com

três índices. Edwards, 2001, trabalha com algoritmos aproximativos para o problema de localização multi-nível. Espejo *et al.*, 2003, utilizam uma relaxação *lagrangean-surrogate* incluída num algoritmo de otimização pelo método de subgradiente para encontrar limites inferiores e superiores ao valor óptimo da função objectivo do problema de máxima cobertura numa rede com dois níveis. Jayaraman *et al.*, 2003, descreve uma relaxação lagrangeana seguida de uma heurística primal, para encontrar soluções admissíveis para um problema de localização hierárquico em que é necessário decidir que tipo de serviço se deve afectar a cada equipamento aberto (serviço básico ou especializado).

As heurísticas primais-duais que foram desenvolvidas para o problema multi-nível têm como base a formulação do problema em que se considera que os clientes são afectos a caminhos. Consideraram-se os casos em que o problema não apresenta restrições de capacidade, apresenta restrições de capacidade máxima ou restrições de capacidade máxima e mínima e ainda o caso de não conservação de fluxo em nodos intermédios da rede.

3.3.1. Problemas sem Restrições de Capacidades

A heurística primal-dual, que se desenvolveu para a resolução dos problemas multi-nível sem restrições de capacidade, apresenta um esquema de funcionamento igual ao Algoritmo III-1. Na realidade, os únicos procedimentos que têm de ser adaptados a este problema são os procedimentos dual ascendente e de ajuste primal-dual para variáveis v_j^t e o procedimento primal que constrói soluções admissíveis. Por este motivo, apenas estes três procedimentos serão agora descritos.

3.3.1.1. Procedimento Dual Ascendente para as Variáveis v_j^t

Este procedimento tenta aumentar o valor das variáveis duais v_j^t , $j \in J^*$, $J^* \subset J$. Sempre que se altera uma variável v_j^t existe a necessidade de actualizar o valor de todas as *slacks* SA_{ir}^{ξ} e SR_{ir}^{ξ} , com $\tau \leq t \leq \xi$, tais que $v_j^t \geq c_{pj}^t$ e $p \in P(i)$. É interessante notar que, se $v_j^t \geq c_{pj}^t$ para mais do que um caminho $p \in P(i)$, então as *slacks* SA_{ir}^{ξ} e SR_{ir}^{ξ} , com $\tau \leq t \leq \xi$, verão o seu valor ser diminuído mais do que uma vez (este comportamento é completamente distinto do comportamento descrito para o caso de um único nível). O procedimento é descrito no Algoritmo 3-39.

Algoritmo 3-39

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS v_j^t - CASO MULTI-NÍVEL

Para cada cliente j e período temporal t ordenam-se todos os custos c_{pj}^t por ordem crescente, sendo que o primeiro custo é designado por c_j^{t1} , o segundo por c_j^{t2} , o k -ésimo por c_j^{tk} . As variáveis v_j^t estão organizadas (sem nenhuma ordem especial) num conjunto formado por pares (j, t) , tal que $(j, t)_k$ representa o k -ésimo elemento desse conjunto.

1. Considere-se uma qualquer solução admissível inicial $\{v_j^t\}$ tal que

- $v_j \geq c_j^t, \forall (j,t), S_i^\xi \geq 0, \forall i,t, \xi \geq t$. Para cada (j,t) , defina-se $k(j,t) = \min\{k : v_j^t \leq c_j^{tk}\}$. Se $v_j^t = c_j^{tk(j,t)}$, então $k(j,t) \leftarrow k(j,t) + 1$.
2. $(j,t) \leftarrow (j,t)_1; q \leftarrow 1; \delta \leftarrow 0$.
 3. Se $(j,t) \notin J^+$, então ir para 7; se não, ir para 4.
 4. $\Delta_j^t \leftarrow \min_i \left\{ \frac{S_{it}^\xi}{\sum_{\substack{p \in P(i) \\ v_j^t \geq c_{pj}^t}} \eta_{ip}^t} : \tau \leq t \leq \xi \right\}$. Se $\Delta_j^t = 0$ ir para 7; se não, ir para 5.
 5. Se $\Delta_j^t > c_j^{tk(j,t)} - v_j^t$, então $\Delta_j^t \leftarrow c_j^{tk(j,t)} - v_j^t; \delta \leftarrow 1; k(j,t) \leftarrow k(j,t) + 1$.
 6. Para cada serviço i , $SR_{it}^\xi \leftarrow SR_{it}^\xi - \Delta_j^t \sum_{\substack{p \in P(i) \\ v_j^t \geq c_{pj}^t}} \eta_{ip}^t$ e $SA_{it}^\xi \leftarrow SA_{it}^\xi - \Delta_j^t \sum_{\substack{p \in P(i) \\ v_j^t \geq c_{pj}^t}} \eta_{ip}^t, \tau \leq t \leq \xi$. $v_j^t \leftarrow v_j^t + \Delta_j^t$.
 7. Se $q \neq |J^+ \times T|, (j,t) \leftarrow (j,t)_{q+1}; q \leftarrow q + 1$ e ir para 3. Caso contrário, ir para 8.
 8. Se $\delta = 1$, ir para 2; se não, parar.
-

O cálculo dos parâmetros η_{ijp}^t pode ser feito de vários modos diferentes. Uma maneira muito simples de encontrar valores para estes parâmetros consiste em fazer $\eta_{ijp}^t = 1/n_p$, em que n_p é igual ao número de serviços que pertencem ao caminho p . Outra maneira de calcular estes valores resulta da análise do passo 4 deste procedimento. É imediato concluir que η_{ijp}^t deverá tomar valores mais pequenos para serviços i , tais que o quociente entre $S_{it}^\xi = \min\{S_{it}^\xi\}$, $\tau \leq t \leq \xi$, e o número de caminhos a que i pertence, com $v_j^t \geq c_{pj}^t$, é menor. Esta opção de calcular os valores η_{ijp}^t é mais exigente em termos do tempo computacional necessário mas, em geral, é também responsável por se encontrarem soluções duais de melhor qualidade. O Algoritmo 3-40 descreve este procedimento de cálculo.

Algoritmo 3-40

PROCEDIMENTO DE CÁLCULO DO VALOR η_{ijp}^t

c_{pj}^t - Custo de afectar o cliente j ao caminho p no período t , $P(i)$ - conjunto de caminhos a que pertence o serviço i , m - número de possíveis localizações para serviços, q - número de caminhos existentes.

1. $p \leftarrow 1$;
2. Se $v_j^t \geq c_{pj}^t$, então ir para 3; se não, ir para 9.
3. Se $n_p = 1$, então $\eta_{ijp}^t \leftarrow 1, i \in P(p)$. Ir para 8.
4. $i \leftarrow 1$.
5. Se $p \in P(i)$, então ir para 6; se não, ir para 7.
6. $e_{ij}^t \leftarrow \begin{cases} 1, & \text{se } v_j^t \geq c_{pj}^t \\ 0, & \text{em caso contrário} \end{cases}, n_{ij}^t \leftarrow \sum_{p \in P(i)} e_{ij}^t, \eta_{ijp}^t \leftarrow \min_{\tau \leq t \leq \xi} \frac{S_{it}^\xi}{n_{ij}^t}$.
7. $i \leftarrow i + 1$. Se $i > m$, então ir para 8. Caso contrário, ir para 5.

8. $D \leftarrow \sum_{i \in p} \eta_{ip}^i$. Actualizar $\eta_{ip}^i \leftarrow \frac{\eta_{ip}^i}{D}, \forall i \in p$.

9. $p \leftarrow p + 1$. Se $p > q$, então parar; se não, ir para 2.

3.3.1.2. Procedimento de Ajuste Primal-Dual para as Variáveis v_j^i

O procedimento de ajuste dual para as variáveis v_j^i detecta violações das condições de complementaridade (2-94) e diminui o valor de algumas variáveis v_j^i , aumentando assim o valor de algumas *slacks* e permitindo o aumento de outras variáveis v_j^i . Este procedimento tenta, simultaneamente, diminuir o número de violações das condições de complementaridade e melhorar o valor da função objectivo dual.

A condição de complementaridade (2-94) será violada se existirem pelo menos dois caminhos abertos ou parcialmente abertos p_1 e p_2 , tais que $v_j^i > c_{p_1,j}^i$ e $v_j^i > c_{p_2,j}^i$. Note-se que não é necessário que os caminhos estejam abertos; basta que estejam parcialmente abertos para que haja violação da condição de complementaridade. Se se diminuir o valor da variável v_j^i , aumentam-se as *slacks* $SA_{i\tau}^{\xi}$ e $SR_{i\tau}^{\xi}$, com $\tau \leq t \leq \xi$, tais que $v_j^i > c_{pj}^i$, com $p \in P(i)$. Se existirem outras duas variáveis v_j^i , que estejam bloqueadas exclusivamente devido a *slacks* cujo valor vai ser aumentado, então será possível melhorar o valor da função objectivo dual. É também interessante notar que a violação da condição de complementaridade pode ocorrer devido a um único serviço que pertence a mais do que um caminho.

No caso em que existe apenas um nível de serviços a violação da condição envolve, pelo menos, dois serviços. O procedimento dual ascendente tenta aumentar primeiro variáveis v_j^i bloqueadas exclusivamente por *slacks* que correspondem a um único serviço i . No caso multi-nível, tentam aumentar-se variáveis v_j^i bloqueadas exclusivamente por *slacks* cujo valor vai ser aumentado, mesmo que correspondam a mais do que um serviço.

Considerem-se as seguintes definições:

$$I_j^{i*} = \{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \text{ e } p \in P(i) \mid (i, \tau, \xi) \in I^* \wedge v_j^i \geq c_{pj}^i \}$$

$$I_j^{i+} = \{ i : i \in I_j^{i*} \text{ e } \exists p \in P(i), \text{ tal que } v_j^i > c_{pj}^i \}$$

$$P_j^{i+} = \{ p : \exists i \in p, \text{ tal que } i \in I_j^{i+} \wedge v_j^i > c_{pj}^i \}$$

$$J^{i+} = \{ (j, \tau) : I_j^{i*} \subset I_j^{i+} \text{ e } (i, \gamma, \xi) \notin I^*, \forall i \in I_j^{i+}, \gamma \leq \tau \leq \xi < t \text{ ou } t < \gamma \leq \tau \leq \xi \} \cup \\ \cup \{ (j, t) : I_j^{i*} \subset I_j^{i+} \}$$

$$c_j^{i-} = \max_p \{ c_{pj}^i : v_j^i > c_{pj}^i \}$$

O conjunto P_j^{i+} indica, para cada cliente j , todos os caminhos abertos ou parcialmente abertos, tais que v_j^i é maior do que o custo de afectação c_{pj}^i . A violação da condição de complementaridade (2-94) é detectada pela existência de, pelo menos, um par (j, t) , tal que o número de elementos de P_j^{i+} é maior do

que um. O conjunto I_j^{t+} indica, para cada cliente j , todos os serviços em funcionamento durante o período t que pertencem a qualquer caminho p , tal que v_j^t é maior do que o custo de afectação c_{pj}^t . Isto significa que todas as *slacks* $SA_{i\tau}^\xi$ e $SR_{i\tau}^\xi$, $\tau \leq t \leq \xi$, com $i \in I_j^{t+}$, serão aumentadas com a diminuição do valor da variável v_j^t . O conjunto I_j^{t*} corresponde ao conjunto de serviços i , tais que existe pelo menos uma *slack* $SA_{i\tau}^\xi$ ou $SR_{i\tau}^\xi$, com $\tau \leq t \leq \xi$, a bloquear uma variável v_j^t . O conjunto J^{t+} representa todas as variáveis duais que podem ser aumentadas com a diminuição de v_j^t . É possível que a própria variável v_j^t pertença a este conjunto J^{t+} . No entanto, se for esse o caso, esta variável nunca será considerada no conjunto J^* na primeira vez que é executado o procedimento dual ascendente.

Algoritmo 3-41

PROCEDIMENTO DE AJUSTE PRIMAL-DUAL PARA AS VARIÁVEIS v_j^t - CASO MULTI-NÍVEL

As variáveis v_j^t estão organizadas (sem nenhuma ordem especial) num conjunto formado por pares (j, t) , tal que $(j, t)_k$ representa o k -ésimo elemento desse conjunto; $I^* = \{ (i, \tau, \xi) : S_{i\tau}^\xi = 0 \}$; $I_t^+ = \{ i : \text{o serviço } i \text{ está em funcionamento durante o período } t \}$;

$$I_j^{t+} = \left\{ i : \exists (\tau, \xi) \text{ com } \tau \leq t \leq \xi \text{ e } p \in P(i) \mid (i, \tau, \xi) \in I^* \wedge v_j^t \geq c_{pj}^t \right\} ; \quad c_j^{t-} = \max_p \left\{ c_{pj}^t : v_j^t > c_{pj}^t \right\} ;$$

$$P_j^{t+} = \left\{ p : \exists i \in p \text{ tal que } i \in I_t^+ \wedge v_j^t > c_{pj}^t \right\}$$

$$J^{t+} = \left\{ (j, \tau) : I_j^{\tau*} \subset I_j^{t+} \text{ e } (i, \gamma, \xi) \notin I^*, \forall i \in I_j^{\tau*}, \gamma \leq \tau \leq \xi < t \text{ ou } t < \gamma \leq \tau \leq \xi \right\} \cup$$

$$\cup \{ (j, t) : I_j^{t*} \subset I_j^{t+} \}$$

1. Inicializar $(j, t) \leftarrow (j, 1)$; $q \leftarrow 1$; $\delta \leftarrow 0$.
 2. Se $\#P_j^{t+} \leq 1$, então ir para 9.
 3. Se $J^{t+} \setminus \{(j, t)\} = \emptyset$, então ir para 9.
 4. Para cada (i, τ, ξ) , $\tau \leq t \leq \xi$, $SA_{i\tau}^\xi \leftarrow SA_{i\tau}^\xi + \sum_{\substack{p \in P(i) \\ v_j^t > c_{pj}^t}} \eta_{ip}^t (v_j^t - c_j^{t-})$, $SR_{i\tau}^\xi \leftarrow SR_{i\tau}^\xi + \sum_{\substack{p \in P(i) \\ v_j^t > c_{pj}^t}} \eta_{ip}^t (v_j^t - c_j^{t-})$;
 $v_j^t \leftarrow c_j^{t-}$.
 5. $J^* \leftarrow J^* \setminus \{(j, t)\}$. Executar o procedimento dual ascendente para as variáveis v_j^t (Algoritmo 3-39).
 $J^+ \leftarrow J^+ \cup \{(j, t)\}$. Executar o procedimento dual ascendente para as variáveis v_j^t (Algoritmo 3-39).
 $J^* \leftarrow J^*$. Executar o procedimento dual ascendente para as variáveis v_j^t (Algoritmo 3-39).
 6. Se v_j^t tiver sido alterado ir para 2, se não ir para 7.
 7. Executar o procedimento primal.
 8. Se tiver havido melhorias nos valores das funções objectivo primal ou dual, então $\delta \leftarrow 0$. Caso contrário $\delta \leftarrow \delta + 1$.
 9. Se o valor da função objectivo primal igualar o valor da função objectivo dual, ou $\delta = \delta_{max}$, ou $q \neq \#J \times T$, então parar; se não, $q \leftarrow q + 1$, $(j, t) \leftarrow (j, t)_q$, e ir para 2.
-

3.3.1.3. Procedimento Primal

O procedimento primal constrói soluções primais admissíveis baseadas em soluções duais admissíveis, tentando forçar a satisfação das condições de complementaridade. Defina-se o conjunto $P_t^+ = \{p: p \text{ está aberto durante o período } t\}$.

Definição 3.3: Um caminho p é considerado essencial durante o período t se existe pelo menos um cliente j que tenha de ser afecto ao caminho p durante t , o que acontece se e só se $\exists j \in J: v_j' \geq c_{pj}' \wedge v_j' < c_{p',j}', \forall p' \in P, p' \neq p$.

Os caminhos considerados essenciais são os primeiros a serem abertos. Abrir um caminho no período t significa que se tem de garantir que todos os serviços que constituem p estão em funcionamento durante t . Tal é conseguido através da introdução de todos estes serviços no conjunto I_t^+ . Quando se abre um caminho, é comum violarem-se restrições de complementaridade (2-25) e (2-26), porque se incluem em I_t^+ serviços i , tais que $i \notin I_t^*$. Sempre que se abre um caminho p , todos os outros caminhos p' , tais que se $i \in p'$ então $i \in p$, também serão automaticamente abertos.

Algoritmo 3-42

PROCEDIMENTO PRIMAL – PROBLEMA MULTI-NÍVEL SEM CAPACIDADES

$I^* = \{(i, \tau, \xi) : S_{it}^* = 0\}$; $I_t^* = \{i : (i, \tau, \xi) \in I^* \wedge \tau \leq t \leq \xi\}$; $I_t^+ = \{i : i \text{ está aberto durante } t\}$;
 $I_A^* = \{(i, \tau, \xi) : a_{it}^* = 1\}$; $I_R^* = \{(i, \tau, \xi) : r_{it}^* = 1\}$; $P_t^+ = \{p: p \text{ está aberto durante o período } t\}$; c_{pj}' - custo de afectar o cliente j ao caminho p durante t .

1. $I_A^+ \leftarrow I_R^+ \leftarrow \emptyset, I_t^+ \leftarrow \emptyset, \forall t$. Construir conjuntos I^+ e I_t^+ . $Num \leftarrow 0$.
 2. Para $t=1$ até T , incluir em P_t^+ todos os caminhos p tais que $\exists j: v_j' \geq c_{pj}'$ e $v_j' < c_{p',j}', \forall p' \neq p$.
 Actualizar os conjuntos P_t^+ , $\forall t$, incluindo em P_t^+ todos os caminhos p abertos tais que $p \notin P_t^+$.
 3. Para cada cliente j tal que $v_j' < c_{pj}', \forall p \in P_t^+$, incluir em P_t^+ o caminho p tal que $c_{pj}' = \min_{v_j' \geq c_{p',j}'} c_{p',j}'$.
 $Num \leftarrow Num + 1$.
 4. Incluir em I_t^+ todos os serviços i que pertencem a um caminho $p \in P_t^+, \forall t$.
 5. Se $Num = 1$, então $I_t^+ \leftarrow I_t^*$ e $I_t^+ \leftarrow \emptyset, P_t^+ \leftarrow \emptyset, \forall t$, ir para 2. Caso contrário ir para 6.
 6. Construir os conjuntos I_A^+ e I_R^+ (Algoritmo 3-5). Actualizar I_t^+ e P_t^+ .
 7. Para $t=1$ até T , afectar cada cliente j a um caminho $p' \in P_t^+$, tal que $c_{p',j}' = \min_{p \in P_t^+} \{c_{pj}'\}$. Calcular Z como sendo o valor da função objectivo primal correspondente à solução actual.
 8. $G \leftarrow I_A^+ \cup I_R^+$.
 9. Escolher arbitrariamente uma variável a_{it}^* ou r_{it}^* tal que $(i, \tau, \xi) \in G$ e modificar o seu valor de um para zero. Se a solução se mantiver admissível recalculer as afectações de clientes a caminhos abertos. Calcular Z como o valor da função objectivo primal correspondente à nova solução. Se $Z < Z$, então remover (i, τ, ξ) de I_A^+ ou I_R^+ (a respectiva variável a_{it}^* ou r_{it}^* ficará igual a zero) e $Z \leftarrow Z$.
 10. $G \leftarrow G \setminus \{(i, \tau, \xi)\}$. Se $G = \emptyset$, ir para 11; se não, ir para 9.
 11. Verificar a satisfação das condições de complementaridade (2-21) a (2-28) (Algoritmo 3-8).
-

Os caminhos que não são considerados essenciais só serão explicitamente abertos no período t se existirem clientes j que não possam ser afectos a caminhos já abertos. Se este for o caso, então o procedimento abre o caminho que corresponde ao menor custo de afectação.

Os passos 8 a 10 tentam melhorar a solução primal construída através da diminuição do número de serviços abertos. Nos testes preliminares levados a cabo, estes passos provaram conseguir melhorar significativamente o valor da função objectivo primal. Estes passos poderiam também ser substituídos por uma heurística “*drop*”: de todas as variáveis de localização iguais a um, escolher aquela que, quando o seu valor é alterado para zero, mais contribui para a melhoria do valor da função objectivo primal; repetir o processo até que não seja mais possível melhorar a função objectivo desta forma. A utilização desta heurística foi experimentada, mas nos testes preliminares efectuados mostrou ser muito mais exigente em termos de tempo computacional, sendo o resultado obtido igual ao obtido com a execução dos passos 8 a 10. Tal pode ser justificado pelo facto do procedimento primal considerar iguais a 1 um número de variáveis de localização maior do que o estritamente necessário (isto porque a abertura de um caminho consiste na abertura de um conjunto de serviços, abrindo-se implicitamente outros caminhos). O valor destas variáveis que estão “a mais” será sempre alterado, mesmo que as variáveis sejam escolhidas arbitrariamente.

3.3.2. Introdução de Restrições de Capacidade Máxima

No caso do problema multi-nível em que se consideram restrições de capacidade máxima, o esquema de funcionamento da heurística é o mesmo descrito como Algoritmo 3-1, com dois passos adicionais:

-
10. Procedimento Dual Ascendente para as Variáveis λ_i^t . Se a solução dual sofrer alterações ir para 2.
 11. Procedimento Dual Descendente para as Variáveis λ_i^t . Se a solução dual sofrer alterações ir para 2.
-

Um aumento ou diminuição do valor de λ_i^t irá influenciar as *slacks* $S_{i\tau}^\xi$, com $\tau \leq \xi$, tais que $P(i) \cap P(\tau) \neq \emptyset, \forall i \in I$ (no cálculo de $S_{i\tau}^\xi$, a soma, considerando todos os caminhos $p \in P(i)$, tem em conta todos os valores λ_i^t , tais que i pertence a p). Os procedimentos dual ascendente e descendente para as variáveis λ_i^t serão descritos seguidamente. Estes procedimentos decorrem dos já desenvolvidos para os problemas com apenas um nível de serviços. Também o procedimento primal terá de ser alterado, pois torna-se necessário testar as novas restrições de capacidades máximas. Os procedimentos dual ascendente e ajuste primal-dual para as variáveis v_j^t (Algoritmo 3-39 e Algoritmo 3-41) mantêm-se válidos, sendo apenas necessário considerar custos de afectação dados por $c_{pj}^t + d_j^t \sum_{i_i \in p} \lambda_{i_i}^t$. Todos os restantes procedimentos não necessitam de

ser alterados.

3.3.2.1. Procedimento Dual Ascendente para as Variáveis λ_i^t

Se o valor da variável λ_i^t é aumentado, então a parte esquerda das restrições (2-95) e (2-96) irá diminuir. A máxima alteração que pode ser considerada deve ser tal que todos esses valores se tornem iguais ou menores do que zero. Se se considerar que a variável λ_i^t é aumentada em δ , então δ deve ser menor ou igual a $\Delta(i)$ definido em (3-22).

$$\Delta(i) = \max_{\substack{j \in J \\ p \in P(i)}} \left\{ \frac{v_j^t - c_{pj}^t - d_j^t \sum_{i_i \in p} \lambda_{i_i}^t}{d_j^t} \right\} \quad (3-22)$$

Defina-se:

$$JP(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^t - c_{pj}^t - d_j^t \sum_{i_i \in p} \lambda_{i_i}^t}{d_j^t} \leq \delta \right\}, \quad (3-23)$$

$$\overline{JP}(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^t - c_{pj}^t - d_j^t \sum_{i_i \in p} \lambda_{i_i}^t}{d_j^t} > \delta \right\}. \quad (3-24)$$

Proposição 3-11: Se a variável dual λ_i^t é aumentada em $\delta \in]0, \Delta(i)]$, então as *slacks* $SA_{i', \tau}^{\xi}$ e $SR_{i', \tau}^{\xi}$, $\tau \leq \xi$, $\forall i' \in I$, tais que $P(i) \cap P(i') \neq \emptyset$, serão alteradas em:

$$\Omega(\delta, i, i') = \sum_{(j, p) \in JP(i, i', \delta)} \eta_{i', jp}^t \max \left\{ 0, v_j^t - c_{pj}^t - d_j^t \sum_{i_i \in p} \lambda_{i_i}^t \right\} + \sum_{(j, p) \in \overline{JP}(i, i', \delta)} \eta_{i', jp}^t d_j^t \delta - E \quad (3-25)$$

$$\text{com: } E = \begin{cases} Q_i \delta, & \text{se } i = i' \\ 0, & \text{em caso contrário} \end{cases}$$

Esta proposição segue resultados similares que podem ser encontrados em Guignard & Spielberg, 1979, pelo que se omite a sua prova, e motiva o procedimento dual ascendente para as variáveis λ_i^t (Algoritmo 3-43).

Algoritmo 3-43

PROCEDIMENTO DUAL ASCENDENTE PARA AS VARIÁVEIS λ_i^t

d_j^t - procura do cliente j no período t ; c_{pj}^t - custo de afectar o cliente j ao caminho p no período t ; m - número de possíveis localizações dos serviços; I - conjunto das possíveis localizações para os serviços; T - número de períodos temporais; $P(i)$ - conjunto de caminhos a que pertence o serviço i ;

$$\Omega(\delta, i, i') = \sum_{(j, p) \in JP(i, i', \delta)} \eta_{i', jp}^t \max \left\{ 0, v_j^t - c_{pj}^t - d_j^t \sum_{i_i \in p} \lambda_{i_i}^t \right\} + \sum_{(j, p) \in \overline{JP}(i, i', \delta)} \eta_{i', jp}^t d_j^t \delta - E \quad \text{com:}$$

$$E = \begin{cases} Q_i \delta, & \text{se } i = i' \\ 0, & \text{em caso contrário} \end{cases};$$

$$\mathcal{JP}(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i}{d_j^i} \leq \delta \right\};$$

$$\overline{\mathcal{JP}}(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i}{d_j^i} > \delta \right\}.$$

1. $t \leftarrow 1$.

2. $i \leftarrow 1$.

3. $\delta \leftarrow 0$; $\delta' \leftarrow +\infty$.

4. $\delta \leftarrow \max_{\substack{j \in J \\ p \in P(i)}} \left\{ \frac{1}{d_j^i} \max \left\{ 0, v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i \right\} \right\}$
 $\frac{1}{d_j^i} \max \left\{ 0, v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i \right\} < \delta'$.

5. Calcular $\mathcal{JP}(i, i', \delta)$ e $\overline{\mathcal{JP}}(i, i', \delta)$ e $\Omega(\delta, i, i')$, $\forall i' \in I$, com $P(i) \cap P(i') \neq \emptyset$.

6. Se $\Omega(\delta, i, i') < 0$, $\forall i' \in I$, com $P(i) \cap P(i') \neq \emptyset$, então ir para 11, se não ir para 7.

7. Se $\exists i' \in I$, com $P(i) \cap P(i') \neq \emptyset$, tal que $SA_{i', \tau}^{\xi} + \Omega(\delta, i, i') < 0$ ou $SR_{i', \tau}^{\xi} + \Omega(\delta, i, i') < 0$, $\tau \leq \xi$, então ir para 8; se não, ir para 9.

8. Se $\delta' = 0$, então ir para 11. Caso contrário, $\delta' \leftarrow \delta$ e ir para 4.

9. $\lambda_i^t \leftarrow \lambda_i^t + \delta$, $SA_{i', \tau}^{\xi} \leftarrow SA_{i', \tau}^{\xi} + \Omega(\delta, i, i')$ e $SR_{i', \tau}^{\xi} \leftarrow SR_{i', \tau}^{\xi} + \Omega(\delta, i, i')$, $\tau \leq \xi$, $\forall i' \in I$, tal que $P(i) \cap P(i') \neq \emptyset$.

10. Executar o algoritmo dual ascendente para as variáveis v_j^i , com $J^i = J$ (Algoritmo 3-39).

11. $i \leftarrow i + 1$. Se $i > m$, ir para 12; se não, ir para 3.

12. $t \leftarrow t + 1$. Se $t > T$, então parar; se não, ir para 2.

3.3.2.1. Procedimento Dual Descendente para as Variáveis λ_i^t

Se a variável dual λ_i^t diminui, então todos os valores $v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i$, com $p \in P(i)$, irão ser aumentados. Considere-se uma diminuição de λ_i^t , tal que todos aqueles valores que são menores do que zero assim se mantêm. Isto significa que λ_i^t pode ser diminuído em δ , tal que (3-26) se verifica.

$$0 < \delta \leq \min_{\substack{j \in J, p \in P(i) \\ v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i < 0}} \left\{ -\frac{1}{d_j^i} \left(v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i \right) \right\} \quad (3-26)$$

Mais uma vez todas as *slacks* $SA_{i', \tau}^{\xi}$ e $SR_{i', \tau}^{\xi}$, $\tau \leq \xi$, $\forall i' \in I$ tal que $P(i) \cap P(i') \neq \emptyset$, serão alteradas.

Proposição 3-12: Se o valor da variável λ_i^t diminui em δ , tal que δ satisfaz (3-26), então $SA_{i', \tau}^{\xi}$ e $SR_{i', \tau}^{\xi}$, $\tau \leq \xi$, $\forall i' \in I$, tal que $P(i) \cap P(i') \neq \emptyset$, serão alteradas em:

$$\Phi(\delta, i', i) = \delta \left(E - \sum_{j \in J} \sum_{\substack{p \in P(i) \cap P(i') \\ v_j^i - c_{pj}^i - d_j^i \sum_{i_1 \in p} \lambda_{i_1}^i \geq 0}} d_j^i n_{i', jp}^i \right), \quad (3-27)$$

$$\text{com } E = \begin{cases} Q_i, & \text{se } i = i' \\ 0, & \text{em caso contrário} \end{cases}.$$

Esta proposição segue resultados similares que se encontram em Guignard & Spielberg, 1979, pelo que não é feita a prova. Esta proposição motiva o procedimento dual ascendente para as variáveis λ_i^t (Algoritmo 3-44).

Algoritmo 3-44

PROCEDIMENTO DUAL DESCENDENTE PARA AS VARIÁVEIS λ_i^t - CASO MULTI-NÍVEL

d_j^t - procura do cliente j no período t ; c_{pj}^t - custo de afectar o cliente j ao caminho p no período t ; m - número de possíveis localizações dos serviços; I - conjunto das possíveis localizações para os serviços; T - número de períodos temporais; $P(i)$ - conjunto de caminhos a que pertence o serviço i ;

$$\Phi(\delta, i', i) = \delta \left(E - \sum_{j \in J} \sum_{p \in P(i) \cap P(i')} d_j^t n_{i'pj}^t \right), \text{ com } E = \begin{cases} Q_i, & \text{se } i = i' \\ 0, & \text{em caso contrário} \end{cases}.$$

$$v_j^t - c_{pj}^t - d_j^t \sum_{i \in p} \lambda_i^t \geq 0$$

1. $t \leftarrow 1$.
2. $i \leftarrow 1$.

$$3. \delta \leftarrow \min \left\{ \lambda_i^t, \min_{\substack{j \in J, p \in P(i) \\ v_j^t - c_{pj}^t - d_j^t \sum_{i \in p} \lambda_i^t < 0}} \left\{ -\frac{1}{d_j^t} \left(v_j^t - c_{pj}^t - d_j^t \sum_{i \in p} \lambda_i^t \right) \right\} \right\}.$$

4. Calcular $\Phi(\delta, i', i)$, $\forall i' \in I$, com $P(i) \cap P(i') \neq \emptyset$.
 5. Se $\Phi(\delta, i', i) < 0$, $\forall i' \in I$, com $P(i) \cap P(i') \neq \emptyset$, então ir para 10; caso contrário, ir para 6.
 6. Se $\exists i' \in I$, com $P(i) \cap P(i') \neq \emptyset$, tal que $SA_{i'\tau}^\xi + \Phi(\delta, i', i) < 0$ ou $SR_{i'\tau}^\xi + \Phi(\delta, i', i) < 0$, $\tau \leq \xi$, então ir para 7; se não, ir para 8.
 7. $\delta \leftarrow \min_{\substack{\tau \leq i \leq \xi \\ S_{i'\tau}^\xi + \Phi(\delta, i', i) < 0}} \left\{ -\frac{S_{i'\tau}^\xi}{\Phi(\delta, i', i)/\delta} \right\}$. Se $\delta = 0$, então ir para 10; se não, ir para 8.
 8. $\lambda_i^t \leftarrow \lambda_i^t - \delta$; $SA_{i'\tau}^\xi \leftarrow SA_{i'\tau}^\xi + \Phi(\delta, i', i)$ e $SR_{i'\tau}^\xi \leftarrow SR_{i'\tau}^\xi + \Phi(\delta, i', i)$, $\tau \leq \xi$, $\forall i' \in I$, tal que $P(i) \cap P(i') \neq \emptyset$.
 9. Executar o procedimento dual ascendente para as variáveis v_j^t , com $J^* = J$ (Algoritmo 3-39).
 10. $i \leftarrow i + 1$. Se $i > m$, então ir para 11; se não, ir para 3.
 11. $t \leftarrow t + 1$. Se $t > T$, então parar; se não, ir para 2.
-
-

3.3.2.2. Procedimento Primal

O procedimento primal constrói soluções admissíveis de uma maneira muito similar à dos procedimentos primais já descritos. Após a construção dos conjuntos I_A^+ e I_R^+ , torna-se necessário testar a satisfação das restrições de capacidade. A afectação de clientes a caminhos é feita através da resolução de T problemas de *transshipment* (entrepósitos). Se o problema de entrepostos não é admissível para algum período t , então será necessário abrir mais caminhos. É interessante notar que neste problema não é possível estabelecer uma condição necessária e suficiente do tipo "somatório das capacidades dos serviços abertos maior ou igual à procura total dos clientes" que permita aferir de forma rápida a

admissibilidade de uma dada solução em relação às restrições de capacidades máximas. Para que se escolha qual o caminho a abrir, o procedimento calcula o custo mínimo de abrir o caminho p através do cálculo dos custos de abrir todos os serviços que pertencem a p mas que não estão em funcionamento em t . Defina-se:

H_p^t = custo mínimo de abrir o caminho $p \notin P_t^+$ durante o período t .

$$e_p^t = \begin{cases} 1, & \text{se } p \in P_t^+ \\ 0, & \text{em caso contrário} \end{cases}, \forall p, t; n_i^t = \sum_{p \in P^t(i)} e_p^t.$$

Pode, de seguida, descrever-se o procedimento primal (Algoritmo 3-45).

Algoritmo 3-45

PROCEDIMENTO PRIMAL PARA O PROBLEMA MULTI-NÍVEL COM CAPACIDADES MÁXIMAS

F_i^t - custo mínimo de abrir o serviço $i \in I_t^+$ durante o período t ; $P_t^+ = \{p: p \text{ está aberto durante o período } t\}$; $I_t^+ = \{i: i \text{ está aberto durante } t\}$; d_j^t - procura do cliente j no período t ; Q_i - capacidade máxima do serviço i ; T - número de períodos temporais; K - número máximo de níveis.

1. Executar os passos 1-6 do Algoritmo 3-42.

2. $t \leftarrow 1$.

3. Resolver um problema de entrepostos, considerando o conjunto J de clientes como origens (com ofertas iguais a d_j^t), o conjunto $i \in I_t^+$ de serviços pertencentes ao nível K como destinos (com procuras Q_i) e os serviços $i \in I_t^+$ pertencentes a níveis 1 a $K-1$ como entrepostos (com ofertas iguais às procuras e dadas por Q_i). Se o problema for impossível então ir para 4, caso contrário ir para 8.

4. Se $p \in P_t^+, \forall p$, então parar porque o problema é impossível; caso contrário, ir para 5.

5. Calcular $H_p^t = \sum_{\substack{i \in p \\ i \in I_t^+}} F_i^t$ e $Cap_p = \min_{i \in p} \left\{ \frac{Q_i}{n_i^t} \right\}, \forall p \notin P_t^+$.

6. Escolher p' tal que $\frac{H_{p'}^t}{Cap_{p'}} = \min_{p \in P_t^+} \left\{ \frac{H_p^t}{Cap_p} \right\}$.

7. Abrir o caminho p' , incluindo p' no conjunto P_t^+ e i em $I_t^+, \forall i \in p'$. Ir para 3.

8. $t \leftarrow t + 1$. Se $t > T$ então ir para 9, se não ir para 3.

9. Calcular Z igual ao valor da função objectivo primal da solução primal actual.

10. Executar os passos 8-10 do Algoritmo 3-42.

O cálculo de Cap_p tenta ilustrar a capacidade máxima do caminho p . A capacidade máxima de p é determinada pelo serviço $i \in p$ com a menor capacidade máxima. Se i pertencer a mais do que um caminho que esteja aberto, a sua capacidade é considerada como estando dividida em iguais partes por todos esses caminhos.

3.3.3. Introdução de Restrições de Capacidade Mínima

Se se considerarem restrições de capacidades máxima e mínima, a heurística não se altera de forma significativa. É necessário considerar as variáveis duais β_i^t que, por apresentarem um comportamento simétrico em relação às variáveis duais λ_i^t , permitem que o procedimento dual ascendente para λ_i^t seja o

procedimento dual descendente para β_i^t e que o procedimento dual descendente para λ_i^t seja o procedimento dual ascendente para β_i^t . A única alteração digna de nota ocorre no procedimento primal. Para que se possa calcular a afectação óptima de clientes a caminhos, considerando fixos os valores das variáveis de localização, é necessário resolver um problema de entrepostos, mas que tem de ter em conta as capacidades mínimas dos serviços. Para tal basta considerar cada ponto de entreposto como correspondendo a dois pontos: um que tem a procura igual à oferta e ambas iguais à sua capacidade mínima, e outro tendo procura e oferta igual à sua capacidade máxima menos a capacidade mínima. Os custos de transporte de uma possível origem fictícia para os primeiros entrepostos terão de ser iguais a um valor arbitrariamente elevado.

Relembre-se a seguinte notação:

f_i^t = menor custo em que se incorre por fechar $i \in I_i^+$ durante o período t .

g_i^t = menor custo em que se incorre por fechar $i \in I_i^+$ durante o período t e abrir um ou mais serviços $i' \in I_i^+$ tal que os problemas de afectação para t' , $t' \leq t$, são admissíveis. Designe-se por $\mathcal{I}(i, t)$ o conjunto de serviços $i' \in I_i^+$ que irão ser abertos.

Algoritmo 3-46

PROCEDIMENTO PRIMAL – CAPACIDADES MÁXIMAS E MÍNIMAS – CASO MULTI-NÍVEL

$I_i^+ = \{i : i \text{ está aberto durante } t\}$; $I_A^+ = \{(i, \tau, \xi) : a_{i\tau}^\xi = 1\}$; $I_R^+ = \{(i, \tau, \xi) : r_{i\tau}^\xi = 1\}$; $P_i^+ = \{p : p \text{ está aberto durante o período } t\}$; f_i^t = menor custo em que se incorre por fechar $i \in I_i^+$ durante o período t ; g_i^t = menor custo em que se incorre por fechar $i \in I_i^+$ durante o período t e abrir um ou mais serviços $i' \in I_i^+$ tal que os problemas de afectação para t' , $t' \leq t$, são admissíveis. Designe-se por $\mathcal{I}(i, t)$ o conjunto de serviços $i' \in I_i^+$ que irão ser abertos.

1. Executar os passos 1 a 8 do Algoritmo 3-45.
2. Se a solução satisfaz as restrições de capacidade mínima, então parar. Caso contrário, ir para 3.
3. $t \leftarrow 1$.
4. Se o problema de afectação é admissível, ir para 14; se não, ir para 5.
5. Calcular $f_i^t, \forall i \in I_i^+$.
6. Escolher o serviço \tilde{i} , tal que $f_{\tilde{i}}^t = \min_{i \in I_i^+} \{f_i^t\}$. Retirar o serviço \tilde{i} do conjunto I_i^+ . Actualizar o conjunto P_i^+ .
7. Se o problema de afectação é admissível, então ir para 14. Caso contrário, inserir \tilde{i} em I_i^+ e $f_{\tilde{i}}^t \leftarrow +\infty$.
8. Se $f_{\tilde{i}}^t = +\infty, \forall i \in I_i^+$, então actualizar P_i^+ e ir para 9. Caso contrário, ir para 6.
9. Calcular $g_i^t, \forall i \in I_i^+$.
10. Considere-se \tilde{i} , tal que $g_{\tilde{i}}^t = \min_{i \in I_i^+} \{g_i^t\}$. Retire-se \tilde{i} do conjunto I_i^+ . Insira-se em I_i^+ o conjunto $\mathcal{I}(\tilde{i}, t)$. Actualizar P_i^+ .
11. Se o problema de afectação é admissível, então ir para 14. Caso contrário, $g_{\tilde{i}}^t \leftarrow +\infty$ e ir para 12.
12. Se $g_{\tilde{i}}^t = +\infty, \forall i \in I_i^+$, então parar: o procedimento não é capaz de encontrar uma solução admissível para o problema de afectação. Caso contrário, ir para 10.
13. Reconstruir $I_A^+, I_R^+, I_i^+, P_i^+, \forall t$.

14. $t \leftarrow t + 1$. Se $t \leq T$, ir para 4; se não, parar.

No passo 5 deste procedimento, o cálculo de f_i^t é feito como descrito no Algoritmo 3-21. O cálculo de g_i^t (Algoritmo 3-47) é feito de um modo ligeiramente diferente do apresentado no Algoritmo 3-22.

Algoritmo 3-47

PROCEDIMENTO DE CÁLCULO DE $g_i^t, i \in I_t^+$

$I_t^+ = \{i : i \text{ está aberto durante } t\}$; $P_t^+ = \{p : p \text{ está aberto durante o período } t\}$; $f_i^t =$ menor custo em que se incorre por fechar $i \in I_t^+$ durante o período t ; F_i^t - custo mínimo de abrir o serviço $i \notin I_t^+$ durante o período t .

1. $g_i^t \leftarrow f_i^t, \forall i \in I_t^+ . I(i,t) \leftarrow \emptyset . I_t^+ \leftarrow I_t^+ \setminus \{i\}$.
 2. Calcular $F_i^t, \forall i \notin I_t^+$.
 3. Se $F_i^t = +\infty, \forall i \notin I_t^+$ então $g_i^t \leftarrow +\infty$ e parar.
 4. Escolher \bar{i} , tal que $F_{\bar{i}}^t = \min_{i \notin I_t^+} \{F_i^t\}$. $I_t^+ \leftarrow I_t^+ \cup \{\bar{i}\}$; $I(i,t) \leftarrow I(i,t) \cup \{\bar{i}\}$. Atualizar $P_t^+ . g_i^t \leftarrow g_i^t + F_i^t . F_{\bar{i}}^t \leftarrow +\infty$.
 5. Se o problema de afectação é admissível, então $I_t^+ \leftarrow I_t^+ \cup \{\bar{i}\}$ e $I_t^+ \leftarrow I_t^+ \setminus I(i,t)$. Atualizar P_t^+ e parar. Caso contrário, ir para 3.
-

O cálculo de $F_i^t, \forall i \notin I_t^+$, é feito como descrito no Algoritmo 3-19, mas garantindo que o problema de afectação se mantém admissível para todos os períodos temporais $t' < t$.

É importante notar que a introdução ou remoção de um único serviço i do conjunto I_t^+ pode ser responsável pela introdução ou remoção de mais do que um caminho do conjunto P_t^+ . Sempre que se remove um serviço i do conjunto I_t^+ todos os caminhos $p \in P(i)$ terão de ser retirados de P_t^+ . A introdução de i em I_t^+ faz com que todos os caminhos $p \in P(i)$, tais que p está parcialmente aberto (ou é apenas constituído por i) e i é o único serviço fechado, sejam incluídos em P_t^+ .

Tal como acontece no problema com apenas um nível de serviços, não é possível garantir que o procedimento primal consiga encontrar uma solução admissível em todos os casos em que tal solução exista.

3.3.4. Serviços sem Conservação de Fluxo

Para problemas em que se considera a existência de serviços sem conservação de fluxo, a heurística primal-dual apresenta o mesmo esquema de funcionamento indicado no Algoritmo III-1, com os dois passos adicionais descritos em 3.3.2, e ainda com o passo adicional que agora se indica:

-
12. Calcular a afectação óptima dos clientes a caminhos considerando os conjuntos I_t^* e P_t^* , que correspondem à melhor solução primal calculada até ao momento.
-

Alguns dos procedimentos descritos têm de ser ligeiramente alterados. Defina-se $d_{ijp}^t = d_j^t \prod_{\substack{i' < i \\ p}} \theta_{i'}$, $\forall i, p \in P(i)$, t . Os procedimentos dual ascendente e o de ajuste primal-dual para as variáveis v_j^t (Algoritmo 3-39 e Algoritmo 3-41) não terão de ser alterados desde que se considerem custos de afectação dados por $c_{ijp}^t = c_{pj}^t + d_{ijp}^t \sum_{i' \in p} \lambda_{i'}^t$, $\forall i, p \in P(i)$, t . As alterações nos procedimentos duais ascendente e descendente para as variáveis λ_i^t são uma consequência directa da definição de d_{ijp}^t . No procedimento dual ascendente as definições (3-22) a (3-24) devem ser substituídas por (3-28) a (3-30):

$$\Delta(i) = \max_{\substack{j \in J \\ p \in P(i) \\ i \in p}} \left\{ \frac{v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t}{d_{i'jp}^t} \right\}, \quad (3-28)$$

$$JP(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t}{d_{i'jp}^t} \leq \delta \right\}, \quad (3-29)$$

$$\overline{JP}(i, i', \delta) = \left\{ (j, p) : j \in J \wedge p \in P(i) \cap P(i') \wedge \frac{v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t}{d_{i'jp}^t} > \delta \right\}. \quad (3-30)$$

Neste caso, a definição de $\Omega(\delta, i, i')$ é dada por (3-31).

$$\Omega(\delta, i, i') = \sum_{(j,p) \in JP(i,i',\delta)} \eta_{i'jp}^t \max \left\{ 0, v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t \right\} + \sum_{(j,p) \in \overline{JP}(i,i',\delta)} \eta_{i'jp}^t d_{i'jp}^t \delta - E \quad (3-31)$$

$$\text{com: } E = \begin{cases} Q_i \delta, & \text{se } i = i' \\ 0, & \text{em caso contrário} \end{cases}$$

Assim, no passo 4 do procedimento dual ascendente para as variáveis λ_i^t (Algoritmo 3-43), o valor de δ deverá ser calculado como indicado por (3-32).

$$\delta \leftarrow \max_{\substack{j \in J \\ p \in P(i) \\ i \in p}} \left\{ \frac{1}{d_{i'jp}^t} \max \left\{ 0, v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t \right\} \right\} \quad (3-32)$$

$$\frac{1}{d_j^t} \max_{i' \in p} \left\{ 0, v_j^t - c_{pj}^t - d_{i'jp}^t \sum_{i_1 \in p} \lambda_{i_1}^t \right\} < \delta$$

No procedimento dual descendente para as variáveis λ_i^t (Algoritmo 3-44), devem ser consideradas as alterações que a seguir se indicam por forma a que o algoritmo se mantenha válido.

$$\Phi(\delta, i^t, i) = \delta \left(E - \sum_{j \in J} \sum_{\substack{p \in P(i) \cap P(i^t) \\ v_j^t - c_{pj}^t - d_{i^t, p}^t \sum_{h \in p} \lambda_h^t \geq 0}} d_{i^t, p}^t n_{i^t, p}^t \right), \quad (3-33)$$

com:

$$0 < \delta \leq \min_{\substack{j \in J, p \in P(i) \\ i^t \in p \\ v_j^t - c_{pj}^t - d_{i^t, p}^t \sum_{h \in p} \lambda_h^t < 0}} \left\{ -\frac{1}{d_{i^t, p}^t} \left(v_j^t - c_{pj}^t - d_{i^t, p}^t \sum_{h \in p} \lambda_h^t \right) \right\}. \quad (3-34)$$

O passo 3 deve ser alterado para que o valor de δ seja calculado como a seguir se apresenta:

$$\delta \leftarrow \min \left\{ \lambda_i^t, \min_{\substack{j \in J, p \in P(i) \\ i^t \in p \\ v_j^t - c_{pj}^t - d_{i^t, p}^t \sum_{h \in p} \lambda_h^t < 0}} \left\{ -\frac{1}{d_{i^t, p}^t} \left(v_j^t - c_{pj}^t - d_{i^t, p}^t \sum_{h \in p} \lambda_h^t \right) \right\} \right\}. \quad (3-35)$$

O procedimento primal é muito semelhante ao Algoritmo 3-45. Considere-se, em primeiro lugar, a situação em que se introduzem somente restrições de capacidades máximas. A única alteração ocorre no passo 5 em que os valores de H_p^t e Cap_p são calculados de forma ligeiramente diferente: $H_p^t = \sum_{\substack{i \in p \\ i \in I_t^+}} F_i^t$ e

$$Cap_p = \min_{i \in p} \left\{ \frac{Q_i}{n_i^t \prod_{\substack{i' < i \\ p}} \theta_{i'}} \right\}, \quad \forall p \notin P_t^+. \text{ Esta alteração tenta ter em conta não apenas o}$$

número de caminhos a que o serviço i pertence, mas também a posição relativa do serviço i nesses caminhos (porque esta posição relativa irá influenciar a quantidade de fluxo afecto a um caminho que chega ao serviço i).

A principal diferença na resolução deste problema reside, de facto, no cálculo dos valores das variáveis de afectação. A afectação óptima, considerando fixos os valores das variáveis de localização, não pode ser encontrada através da resolução de um problema de entrepostos. Em vez disto, após o cálculo de I_t^+ e P_t^+ , torna-se necessário resolver o problema de programação linear $AP(t)$ para cada período temporal t .

$$\begin{aligned} & \mathbf{AP}(t) \\ \text{Min} \quad & \sum_j \sum_p c_{pj}^t x_{pj}^t \end{aligned} \quad (3-36)$$

sujeito a:

$$\sum_{p \in P_i^+} x_{pj}^t = 1, \quad \forall j \quad (3-37)$$

$$\sum_j \sum_{p \in P(i) \cap P_i^+} x_{pj}^t d_j^t \prod_{i' < i} \theta_{i'} \leq Q_i, \quad \forall i \in I_i^+ \quad (3-38)$$

$$x_{pj}^t \geq 0, \quad \forall j, p, t \quad (3-39)$$

Resolver o problema $AP(t)$ de forma óptima (utilizando um *general solver*, por exemplo), em cada execução do procedimento primal e para cada período temporal é demasiado exigente em termos de tempo computacional necessário. Por isso, optou-se por desenvolver uma heurística (Algoritmo 3-48) que encontra soluções admissíveis para $AP(t)$. Esta heurística é utilizada no passo 3 do procedimento primal (em vez da resolução do problema de entrepostos). O problema $AP(t)$, $\forall t$, é apenas resolvido de forma óptima no passo 12 da heurística primal-dual, utilizando um *general solver*, e considerando os valores das variáveis de localização dados pela melhor solução primal encontrada até ao momento.

Algoritmo 3-48

HEURÍSTICA PARA O PROBLEMA $AP(t)$

d_j^t - procura do cliente j no período t ; Q_i - capacidade máxima do serviço i ; c_{pj}^t - custo de afectar o cliente j ao caminho p no período t ; $P_i^+ = \{p: p \text{ está aberto durante o período } t\}$; θ_i - percentagem de fluxo que o serviço i reencaminha para os serviços em níveis superiores.

1. $D_j \leftarrow d_j^t, \forall j$, $Q_p \leftarrow \prod_{i' < i} \theta_{i'}$, $\forall i, p \in P(i)$, $Cap_p \leftarrow \min_{i \in p} \{Q_{i'}\}$, $\forall p \in P_i^+, \forall p \in P_i^+$. $x_{pj}^t \leftarrow 0, \forall j, p$.
 2. Se $Cap_p = 0, \forall p \in P_i^+$ e existe um cliente j tal que $D_j > 0$, então parar: o problema é impossível. Caso contrário, ir para 3.
 3. Para cada cliente $j \in J$, com $D_j > 0$, calcular $c_{p_1}^t \leftarrow \min_{\substack{p \in P_i^+ \\ Cap_p > 0}} \{c_{pj}^t\}$ e $c_{p_2}^t \leftarrow \min_{\substack{p \in P_i^+ \\ Cap_p > 0}} \{c_{pj}^t : c_{pj}^t \geq c_{p_1}^t \wedge p_1 \neq p_2\}$.
 4. Escolher o cliente j e o caminho p_1 , tal que $c_{p_2}^t - c_{p_1}^t = \max_{j \in J} \{c_{p_2}^t - c_{p_1}^t\}$.
 5. $\delta \leftarrow \min\{D_j, Q_{p_1}\}$, $x_{p_1 j}^t \leftarrow x_{p_1 j}^t + \delta$, $Q_{p_1} \leftarrow Q_{p_1} - \delta, \forall i \in p_1$, $Cap_{p_1} \leftarrow Cap_{p_1} - \delta$, $D_j \leftarrow D_j - \delta$.
 6. $\forall i, p \in P_i^+, p \neq p_1$, tal que $i \in p_1 \wedge i \in p$, $Q_p \leftarrow Q_p - \frac{\delta \prod_{i' < i} \theta_{i'}}{\prod_{i' < i} \theta_{i'}}$ e $Cap_p \leftarrow \min\{Cap_p, Q_p\}$.
 7. Se $D_j = 0, \forall j$, então parar; se não, ir para 2.
-

O cálculo de Q_p corresponde ao fluxo máximo que pode ser afecto ao caminho p devido à capacidade máxima do serviço i . A menor destas capacidades corresponde à capacidade máxima do caminho p , representada por Cap_p .

Observação 1: Sejam (S_1 e S_2) dois conjuntos de variáveis de localização a_{it}^e e r_{it}^e . Considerem-se duas soluções admissíveis para o problema, construídas através da fixação das variáveis pertencentes a S_1 e S_2 , respectivamente, a um, e todas as outras variáveis de localização iguais a zero. Suponha-se que $Z_1^* < Z_2^*$, em que Z_1^* e Z_2^* representam o valor óptimo da função objectivo relativo às soluções S_1 e S_2 , respectivamente. Se Z_1' e Z_2' representarem os valores da função objectivo obtidos quando se consideram as soluções S_1 e S_2 , respectivamente e, resolvendo os problemas de afectação $AP(t)$ de forma heurística, tal como descrito no Algoritmo 3-48, então $Z_1' \geq Z_1^*$ e $Z_2' \geq Z_2^*$, no entanto não é possível garantir que $Z_1' < Z_2'$. Esta observação tem uma importante consequência imediata: é possível que a heurística primal-dual encontre soluções de melhor qualidade do que a solução final, soluções essas que a heurística não consegue identificar como sendo melhores. Esta desvantagem não pode ser ultrapassada, a não ser que se opte por resolver de forma óptima os problemas $AP(t)$ em cada execução do procedimento primal e para cada período t . No entanto, tal opção acarreta um aumento proibitivo em termos de tempos computacionais necessários.

Foi também testada uma variação ao Algoritmo 3-48, em que no passo 3, além de se calcular a diferença entre os dois menores custos de afectação para cada cliente, se calculava também a diferença entre os dois menores custos de afectação para cada caminho e se escolhia o par (cliente-caminho) a que corresponderia a maior das diferenças (à semelhança do que acontece no método das penalidades para encontrar uma solução admissível inicial para o problema de transportes). Esta variação deu sempre piores resultados do que os obtidos com o Algoritmo 3-48 em todos os testes preliminares realizados.

Considere-se agora o caso em que existem simultaneamente restrições de capacidade máxima e mínima. Como se pode observar no problema dual condensado, as variáveis duais β_i^t e λ_i^t apresentam um comportamento simétrico. Por este motivo, o procedimento dual ascendente descrito para as variáveis duais λ_i^t constitui o procedimento dual descendente para β_i^t , e o procedimento dual descendente para as variáveis duais λ_i^t constitui o procedimento dual ascendente para β_i^t (esta situação é semelhante à que ocorre nos problemas com um único nível). O procedimento primal terá de ser alterado, porque se torna necessário verificar a satisfação de um novo conjunto de restrições. No entanto, o novo procedimento primal construído é, na sua essência, igual ao já descrito. A única diferença consiste no facto de se ter de resolver um problema de afectação diferente pois o problema $AP(t)$ tem um novo conjunto de restrições. São também introduzidos alguns passos adicionais que tentam fechar ou “trocar” serviços abertos e fechados, sempre que uma solução admissível não é encontrada recorrendo apenas à abertura de novos serviços. Considerando fixas as variáveis de localização, a afectação óptima de clientes a caminhos corresponde à solução óptima do problema de programação linear $AP1(t)$. O Algoritmo 3-49 descreve a heurística que encontra uma solução admissível para $AP1(t)$.

$$AP1(t) \quad \text{Min} \sum_j \sum_p c_{pj}^t x_{pj}^t \quad (3-36)$$

sujeito a:

$$\sum_{p \in P_i^+} x_{pj}^t = 1, \quad \forall j \quad (3-37)$$

$$Q_i' \leq \sum_j \sum_{p \in P(i) \cap P_i^+} x_{pj}^t d_j^t \prod_{i' < i} \theta_{i'} \leq Q_i, \quad \forall i \in I_i^+ \quad (3-40)$$

$$x_{pj}^t \geq 0, \quad \forall j, p, t \quad (3-39)$$

Algoritmo 3-49

HEURÍSTICA PARA O PROBLEMA AP1(t)

d_j^t - procura do cliente j no período t ; Q_i - capacidade máxima do serviço i ; Q_i' - capacidade mínima do serviço i ; c_{pj}^t - custo de afectar o cliente j ao caminho p no período t ; $P_i^+ = \{p: p \text{ está aberto durante o período } t\}$; $P(i)$ - conjunto de caminhos a que pertence o serviço i ; $I_i^+ = \{i: i \text{ está aberto durante } t\}$; θ_i - percentagem de fluxo que o serviço i reencaminha para os serviços em níveis superiores.

1. Calcular uma solução admissível para $AP(t)$ através do Algoritmo 3-48. Se a solução calculada é admissível para $AP1(t)$, então parar. Caso contrário ir para 2.
2. $x_{pj}^t \leftarrow 0, \forall p, j$; $Capmin_i \leftarrow Q_i'$ and $Capmax_i \leftarrow Q_i, \forall i \in I_i^+$; $D_j \leftarrow d_j^t, \forall j$.
3. Se $Capmin_i \leq 0, \forall i \in I_i^+$, então ir para 10; se não, ir para 4.
4. Se $D_j = 0, \forall j$ e $\exists i \in I_i^+$ tal que $Capmin_i > 0$, então parar: a heurística não consegue encontrar uma solução admissível. Caso contrário, ir para 5.

5. Escolher o serviço i , com $Capmin_i > 0$, tal que $\frac{Capmin_i}{Capmax_i} = \max_{i' \in I_i^+} \left\{ \frac{Capmin_{i'}}{Capmax_{i'}} \right\}$.

6. Se $\exists p \in P_i^+ \cap P(i)$, tal que $Capmin_i > 0, \forall i \in p$, então ir para 7; se não, ir para 8.

7. Escolher o caminho $p \in P_i^+ \cap P(i)$ e o cliente j , tal que $\frac{c_{pj}^t}{\prod_{i' < i} \theta_{i'}} = \min_{\substack{p \in P(i) \cap P_i^+ \\ Capmin_i > 0, \forall i \in p \\ j: D_j > 0}} \left\{ \frac{c_{p'j'}^t}{\prod_{i' < i'} \theta_{i'}} \right\}$. Ir para 9.

8. Escolher o caminho $p \in P_i^+ \cap P(i)$ e o cliente j , tal que $\frac{c_{pj}^t}{\prod_{i' < i} \theta_{i'}} = \min_{\substack{p \in P(i) \cap P_i^+ \\ j: D_j > 0}} \left\{ \frac{c_{p'j'}^t}{\prod_{i' < i'} \theta_{i'}} \right\}$.

9. $\delta \leftarrow \min \left\{ D_j, \min_{i' \in p} \left\{ \frac{Capmax_{i'}}{\prod_{i'' < i'} \theta_{i''}} \right\}, \frac{Capmin_i}{\prod_{i'' < i} \theta_{i''}} \right\}$; $x_{pj}^t \leftarrow x_{pj}^t + \delta$; $D_j \leftarrow D_j - \delta$; $Capmin_i \leftarrow Capmin_i - \delta \prod_{i'' < i} \theta_{i''}$,

e $Capmax_i \leftarrow Capmax_i - \delta \prod_{i'' < i} \theta_{i''}, \forall i \in p$. Ir para 3.

10. Se existir um cliente j , tal que $D_j > 0$, então resolver o problema $AP(t)$, utilizando o Algoritmo 3-48, e considerando Q_i igual a $Capmax_i, \forall i \in I_i^+$ e d_j^t igual a $D_j, \forall j$. Caso contrário, parar.

Se a solução calculada no passo 1 não é admissível, então a heurística tenta afectar fluxos a caminhos por forma a satisfazer as restrições de capacidades mínimas nos serviços abertos. No passo 5, a heurística escolhe o serviço i , tal que a relação existente entre as suas capacidades máxima e mínima torna mais difícil a satisfação simultânea de ambas as restrições. A heurística experimenta, em primeiro lugar, afectar fluxos a caminhos $p \in P(i)$ tais que nenhum serviço i'

pertencente a p tem satisfeita a restrição de capacidade mínima (passo 6). Quando tal não é possível, a heurística considera todos os caminhos abertos $p \in P(i)$ (passo 7). Nos passos 6 ou 7 a heurística escolhe um cliente j e abre um caminho p que corresponde ao custo mínimo unitário do fluxo que chega ao serviço i . O aumento do fluxo que passa pelo caminho p é calculado como sendo o mínimo entre a procura ainda não satisfeita do cliente j e o fluxo necessário para satisfazer as restrições de capacidade do serviço i , garantindo que as restrições de capacidade máxima para todos os outros serviços pertencentes a p se mantêm admissíveis. Quando a heurística chega ao passo 10, todas as restrições de capacidades mínimas para serviços abertos estão a ser satisfeitas. Se ainda existirem clientes j com procuras não satisfeitas, então torna-se necessário afectar estas procuras a caminhos. Para tal utiliza-se o Algoritmo 3-48, considerando como capacidade máxima de um serviço a sua capacidade ainda disponível.

A heurística descrita não consegue garantir a construção de uma solução admissível em todas as situações em que $API(i)$ é um problema possível. Nestes casos, é sempre possível recorrer à utilização de um *general solver* para calcular a solução óptima do problema ou para concluir acerca da inexistência de soluções admissíveis.

3.3.5. Resultados Computacionais

À semelhança do que foi feito no caso dos problemas de localização com apenas um nível de serviços, também no caso multi-nível se testaram as heurísticas desenvolvidas através da resolução de problemas de teste aleatoriamente gerados. Considerou-se que o número de possíveis localizações para serviços do nível l é igual a metade do número de possíveis localizações para serviços no nível $l-1$. Optou-se por fazer a geração dos problemas desta forma porque, na maioria dos problemas reais multi-nível, o número de serviços diminui com o nível da rede a que pertencem. Deve, no entanto, deixar-se aqui a observação que tal comportamento não pode ser generalizado. São considerados admissíveis todos os caminhos entre serviços. Os problemas de teste foram gerados como indicado em III.2.1.12. As capacidades máximas e a procura dos clientes são geradas aleatoriamente como descrito em Saldanha-da-Gama, 2002, garantindo que serviços no nível k têm capacidades máximas maiores do que as capacidades máximas de serviços no nível $k-1$. Esta é a situação mais comum, em que um serviço de nível superior tem de satisfazer uma procura que é, na sua globalidade, maior do que as afectas aos serviços em níveis inferiores. Nos problemas em que se considera a possibilidade de não conservação de fluxo, cada serviço i , em níveis intermédios da rede, tem uma probabilidade de 80% de ter um valor para θ_i diferente de um. Neste caso θ_i é aleatoriamente gerado dentro do intervalo [0.20,0.99].

Todos os testes foram executados num computador Pentium 4, 1.80 Ghz, com o sistema operativo Windows 2000, com o máximo de 2000 MB de memória virtual e 256Mb de Ram. As heurísticas foram programadas usando linguagem C e compilador Microsoft Visual C++. O desempenho das heurísticas foi comparado com o desempenho do Cplex, versão 9.0. Também aqui se optou

por executar uma pesquisa local, após a execução das heurísticas primais-duais.

Para os problemas sem capacidades, foram geradas instâncias com as seguintes dimensões: 5 ou 10 períodos temporais; 25, 50 ou 100 clientes; 2, 5 ou 10 possíveis localizações para serviços no último nível da rede (este valor é representado nas tabelas por m_k). Foram considerados 2 ou 3 níveis de serviços. Apenas foram geradas instâncias com $n \geq m$. Para cada combinação destes valores cinco instâncias foram geradas e resolvidas. Foi estabelecido um limite de 200 000 segundos para o Cplex.

O Cplex teve muitas dificuldades em resolver problemas com 3 níveis e 10 possíveis localizações para serviços no último nível. Mesmo para $(T, n) = (5, 50)$ só foi capaz de calcular uma solução primal em 4 das 5 instâncias. A heurística primal-dual não foi capaz de resolver problemas com 10 períodos temporais, 3 níveis, 100 clientes e 10 serviços no último nível. Como pode ser observado na Tabela 3.28, os piores resultados em termos da qualidade da solução primal calculada pela heurística aparecem nos problemas que o Cplex não foi capaz de resolver. Nestes casos, o melhor limite inferior conhecido é o calculado pela própria heurística (a qualidade destes limites inferiores pode ser observada na 3.29). Estes maus resultados podem, assim, ser justificados pela má qualidade do melhor limite inferior conhecido e não pela má qualidade da solução primal calculada. Na Tabela 3.30 podem observar-se os tempos computacionais gastos pela heurística e pelo Cplex.

Para problemas com capacidades foram considerados problemas da mesma dimensão do caso sem capacidades. Na Tabela 3.31 é mostrada a qualidade da solução primal calculada pela heurística e pelo Cplex, isto porque, neste caso, o Cplex é incapaz de encontrar a solução ótima para a maioria das instâncias, dentro do limite dos 200000 segundos. No caso com capacidades apenas 5% dos problemas têm uma relaxação linear cuja solução ótima é admissível para o problema não relaxado. Os limites inferiores calculados pela heurística são, em geral, de má qualidade. Mais uma vez se testou a utilização do Cplex como procedimento heurístico, terminando a sua execução sempre que encontra uma solução de qualidade semelhante à encontrada, em média, pela heurística (para problemas de igual dimensão), ou se atinja o limite de tempo de 12000 segundos (mais de duas vezes superior ao maior tempo computacional gasto pela heurística). Os resultados apresentam-se na Tabela III-103. O Cplex não consegue, dentro do limite de tempo considerado, encontrar soluções com a qualidade pretendida para muitas das instâncias consideradas e precisa, em média, de mais de cinquenta vezes o tempo computacional gasto pela heurística.

No caso em que se consideraram serviços sem conservação de fluxo, o limite de tempo estabelecido para a execução do Cplex foi de 80000 segundos. Este valor é igual ao maior tempo computacional gasto pela heurística. O Cplex poderia encontrar melhores soluções se o limite de tempo considerado fosse maior. Como o objectivo principal destes testes computacionais é a aferição do comportamento da heurística, e a melhoria da qualidade da solução primal por unidade de tempo a partir dos 80000 segundos é praticamente insignificante, o aumento do limite de tempo não iria trazer nenhuma mais-valia. Como se pode ver, neste caso, a qualidade das soluções primais calculadas é nitidamente pior

do que no caso em que existe conservação de fluxo. No entanto, a heurística continua a apresentar um bom compromisso entre o tempo computacional e a qualidade da solução, quando comparada com o Cplex. Mais uma vez os limites inferiores calculados pela heurística são de má qualidade.

Nenhum dos problemas gerados apresenta uma relaxação linear cuja solução ótima seja admissível para o problema original. Na Tabela 3.35 apresenta-se a qualidade da solução primal calculada. Ao contrário do que aconteceu em tabelas semelhantes anteriores, optou-se por mostrar também a qualidade da solução primal encontrada pelo Cplex porque, na larga maioria dos casos, o Cplex não consegue encontrar a solução ótima dentro do limite de tempo estabelecido. Como seria de esperar, as maiores dificuldades são observadas nos problemas que consideram um maior número de caminhos admissíveis.

Na Tabela 3.36 pode observar-se a qualidade da solução dual calculada pela heurística. Como se pode ver, a solução dual é de má qualidade. A Tabela 3.37 apresenta os tempos computacionais gastos por ambas as técnicas de optimização.

				Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local		
<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.00	1.66	5.82	0.00	1.15	5.75
5	2	25	5	0.00	0.70	2.08	0.00	0.36	1.25
5	2	50	2	0.00	0.26	0.81	0.00	0.00	0.02
5	2	50	5	0.00	1.99	5.39	0.00	0.53	1.78
5	2	50	10	0.00	0.98	4.92	0.00	0.27	1.35
5	2	100	2	0.00	0.01	0.07	0.00	0.00	0.00
5	2	100	5	0.00	0.27	0.80	0.00	0.11	0.53
5	2	100	10	0.72	1.65	4.68	0.00	0.50	0.91
5	3	25	2	0.00	4.07	11.75	0.00	1.87	7.58
5	3	25	5	0.00	1.09	4.63	0.00	0.14	0.41
5	3	50	2	0.00	2.19	5.99	0.00	1.98	5.57
5	3	50	5	0.00	2.72	9.54	0.00	2.10	8.43
5	3	50	10	0.00	2.44	6.36	0.00	1.04	4.12
5	3	100	2	0.00	0.18	0.73	0.00	0.15	0.73
5	3	100	5	0.00	0.98	2.90	0.00	0.02	0.10
5	3	100	10	8.11	13.95	19.26	4.41	8.20	12.22
10	2	25	2	0.00	1.10	2.14	0.00	0.52	1.27
10	2	25	5	0.16	2.19	7.37	0.00	1.12	2.31
10	2	50	2	0.00	0.36	1.22	0.00	0.20	1.02
10	2	50	5	0.00	0.86	2.90	0.00	0.49	2.11
10	2	50	10	0.09	0.87	2.00	0.00	0.43	1.40
10	2	100	2	0.00	0.98	2.43	0.00	0.49	2.22
10	2	100	5	0.78	1.87	3.62	0.34	0.81	1.57
10	2	100	10	0.21	1.42	3.69	0.00	0.36	0.88
10	3	25	2	0.00	3.01	6.63	0.00	1.69	5.72
10	3	25	5	1.75	5.24	16.07	0.00	0.15	0.42
10	3	50	2	0.57	1.25	2.31	0.00	0.82	2.31
10	3	50	5	0.37	2.32	3.85	0.07	0.74	2.25
10	3	50	10	5.91	10.57	19.84	5.46	9.44	18.61
10	3	100	2	0.00	0.90	3.67	0.00	0.27	1.36
10	3	100	5	0.03	0.89	2.22	0.03	0.42	1.12
10	3	100	10	---	---	---	---	---	---
Resumo dos Valores				0.00	2.22	19.84	0.00	1.17	18.61

Tabela 3.28 – Qualidade da solução primal (em percentagem) – problemas multi-nível sem capacidades

Heurística Primal-Dual						
<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.94	3.87	5.72
5	2	25	5	2.57	5.49	8.38
5	2	50	2	0.06	1.54	2.50
5	2	50	5	2.74	4.49	6.50
5	2	50	10	2.28	5.06	7.50
5	2	100	2	0.25	0.73	1.72
5	2	100	5	1.33	2.19	2.91
5	2	100	10	4.28	5.25	6.36
5	3	25	2	3.21	6.74	9.23
5	3	25	5	7.28	10.05	13.72
5	3	50	2	1.70	4.89	9.24
5	3	50	5	3.03	6.30	9.39
5	3	50	10	2.14	6.43	8.72
5	3	100	2	0.00	2.03	5.52
5	3	100	5	2.74	4.30	5.24
5	3	100	10	4.23	7.52	10.89
10	2	25	2	1.04	2.99	5.06
10	2	25	5	3.38	5.57	7.25
10	2	50	2	0.71	1.58	3.60
10	2	50	5	2.75	3.04	3.65
10	2	50	10	3.54	4.45	5.27
10	2	100	2	0.72	1.09	1.74
10	2	100	5	2.16	2.77	3.69
10	2	100	10	3.34	3.89	4.61
10	3	25	2	0.94	3.16	4.86
10	3	25	5	5.88	7.46	10.97
10	3	50	2	1.69	2.77	4.60
10	3	50	5	3.62	5.75	8.93
10	3	50	10	5.18	8.47	15.69
10	3	100	2	1.11	1.96	2.87
10	3	100	5	3.16	3.71	4.15
10	3	100	10	---	---	---
Resumo dos Valores				0.00	4.37	15.69

3.29 – Qualidade do limite inferior –problema multi-nível sem capacidades

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Cplex		
				<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.01	0.02	0.06	0.01	0.03	0.07	0.06	0.10	0.25
5	2	25	5	0.03	0.03	0.04	0.03	0.04	0.04	0.23	0.24	0.25
5	2	50	2	0.00	0.00	0.01	0.00	0.01	0.02	0.09	0.10	0.14
5	2	50	5	0.03	0.05	0.10	0.04	0.07	0.11	0.47	0.49	0.53
5	2	50	10	0.03	0.20	0.38	0.06	0.24	0.41	1.72	11.94	52.67
5	2	100	2	0.01	0.02	0.03	0.01	0.02	0.03	0.20	0.21	0.23
5	2	100	5	0.05	0.09	0.17	0.06	0.10	0.18	1.00	1.18	1.69
5	2	100	10	0.31	0.47	1.02	0.37	0.56	1.12	4.11	33.05	148.23
5	3	25	2	0.02	0.05	0.08	0.03	0.06	0.09	0.33	0.34	0.38
5	3	25	5	0.51	4.17	8.29	0.77	4.36	8.58	6.66	11.52	30.36
5	3	50	2	0.02	0.11	0.18	0.03	0.13	0.20	0.66	0.69	0.76
5	3	50	5	0.73	2.27	5.07	0.91	2.46	5.24	13.72	17.78	32.08
5	3	50	10	17.87	49.70	119.57	20.39	53.13	124.60	3241.64	3557.85	4286.41
5	3	100	2	0.04	0.23	0.45	0.04	0.24	0.48	1.34	1.44	1.66
5	3	100	5	2.32	4.59	10.41	2.46	4.82	10.59	97.05	117.16	146.55
5	3	100	10	19.24	29.84	46.92	27.99	37.01	50.50	---	---	---
10	2	25	2	0.01	0.02	0.05	0.03	0.04	0.06	0.20	0.27	0.39
10	2	25	5	0.04	0.08	0.15	0.08	0.12	0.23	0.73	0.98	1.81
10	2	50	2	0.01	0.02	0.05	0.03	0.04	0.08	0.39	0.41	0.42
10	2	50	5	0.11	0.75	2.82	0.16	0.82	2.88	1.50	5.55	18.41
10	2	50	10	0.49	0.81	1.35	0.85	1.14	1.56	5.06	6.19	9.72
10	2	100	2	0.01	0.04	0.11	0.05	0.08	0.13	0.86	0.89	0.95
10	2	100	5	0.30	0.49	0.69	0.42	0.66	0.94	3.33	7.27	16.02
10	2	100	10	1.37	5.26	16.31	1.92	5.91	16.70	13.38	49.61	123.34
10	3	25	2	0.09	0.17	0.32	0.14	0.23	0.44	0.91	0.93	0.95
10	3	25	5	1.35	3.04	5.95	3.43	5.08	7.08	14.83	42.12	83.42
10	3	50	2	0.08	0.95	1.45	0.14	1.03	1.53	1.83	2.15	3.34
10	3	50	5	4.06	6.65	9.75	5.31	8.23	11.11	131.81	1309.64	4377.34
10	3	50	10	46.38	86.39	137.60	80.50	113.04	158.16	---	---	---
10	3	100	2	0.67	1.48	2.87	0.77	1.60	2.95	4.17	5.27	8.94
10	3	100	5	7.18	26.44	81.46	8.24	28.13	82.96	1236.61	1379.63	1468.06
Resumo dos Valores				0.00	7.24	137.60	0.00	8.69	158.16	0.06	226.38	4377.34

Tabela 3.30 – Tempos computacionais (em segundos) – problema multi-nível sem capacidades

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_s</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Cplex		
				<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.00	0.91	2.39	0.00	0.00	0.00	0.00	0.00	0.00
5	2	25	5	0.46	2.52	6.43	0.00	1.74	5.34	0.00	0.00	0.01
5	2	50	2	0.00	0.78	1.81	0.00	0.00	0.00	0.00	0.00	0.00
5	2	50	5	0.07	1.21	2.14	0.00	0.33	0.71	0.00	0.00	0.00
5	2	50	10	1.77	4.69	6.10	0.90	3.06	4.86	0.63	2.78	5.30
5	2	100	2	0.00	0.25	0.79	0.00	0.16	0.68	0.00	0.00	0.00
5	2	100	5	0.08	3.62	7.68	0.01	1.88	4.34	0.00	0.01	0.01
5	2	100	10	4.59	6.70	9.21	2.45	4.48	8.06	1.80	4.32	7.77
5	3	25	2	0.00	1.91	4.08	0.00	1.05	3.58	0.00	0.00	0.00
5	3	25	5	4.22	7.23	10.22	1.18	3.20	5.60	0.01	2.23	3.59
5	3	50	2	0.00	2.07	3.55	0.00	0.39	1.92	0.00	0.00	0.01
5	3	50	5	2.76	7.55	12.57	0.00	2.70	10.98	0.00	2.41	10.06
5	3	50	10	---	---	---	---	---	---	---	---	---
5	3	100	2	0.78	5.73	11.90	0.00	1.19	4.43	0.00	0.00	0.01
5	3	100	5	7.70	10.84	14.36	0.00	6.12	9.45	3.34	5.41	7.93
5	3	100	10	---	---	---	---	---	---	---	---	---
10	2	25	2	0.00	1.21	2.83	0.00	0.44	1.19	0.00	0.00	0.01
10	2	25	5	3.21	4.20	5.48	0.23	1.51	2.90	0.00	0.01	0.01
10	2	50	2	0.00	0.91	1.81	0.00	0.46	1.81	0.00	0.00	0.01
10	2	50	5	0.17	1.67	2.42	0.00	1.11	1.97	0.00	0.01	0.01
10	2	50	10	4.21	4.98	5.36	2.94	3.59	4.25	2.50	3.74	5.17
10	2	100	2	0.00	2.71	5.46	0.00	0.95	4.16	0.00	0.00	0.01
10	2	100	5	1.99	2.60	3.08	0.57	1.76	2.30	0.00	0.01	0.01
10	2	100	10	3.19	5.22	7.45	2.40	4.21	6.79	2.08	5.00	11.15
10	3	25	2	0.83	3.01	5.49	0.27	0.86	1.40	0.00	0.01	0.01
10	3	25	5	5.45	10.13	14.64	2.94	6.06	8.80	2.60	4.61	6.06
10	3	50	2	1.66	6.56	9.85	0.64	1.71	4.07	0.00	0.01	0.01
10	3	50	5	2.89	10.61	21.62	1.67	5.76	13.38	1.29	7.41	16.80
10	3	50	10	---	---	---	---	---	---	---	---	---
10	3	100	2	0.62	2.91	6.76	0.27	1.03	1.87	0.00	0.01	0.01
10	3	100	5	4.38	9.65	14.69	2.58	6.16	10.28	1.32	4.15	6.75
10	3	100	10	---	---	---	---	---	---	---	---	---
Resumo dos Valores				0.00	4.37	21.62	0.00	2.21	13.38	0.00	1.50	16.80

Tabela 3.31 – Qualidade da solução primal (em percentagem) – problemas multi-nível com capacidades máximas

Heurística Primal-Dual						
<i>T</i>	<i>K</i>	<i>n</i>	<i>m_b</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	2.25	9.60	14.23
5	2	25	5	12.32	14.17	17.29
5	2	50	2	8.30	11.82	16.15
5	2	50	5	9.33	14.73	20.41
5	2	50	10	12.42	14.90	17.15
5	2	100	2	5.68	13.35	19.57
5	2	100	5	8.83	11.20	12.54
5	2	100	10	12.00	15.24	20.99
5	3	25	2	8.45	19.40	38.75
5	3	25	5	15.08	19.68	25.96
5	3	50	2	14.46	18.13	22.10
5	3	50	5	21.62	22.95	25.32
5	3	100	2	17.69	19.92	23.53
5	3	100	5	14.38	17.34	19.30
10	2	25	2	2.52	7.32	11.47
10	2	25	5	6.43	9.58	13.73
10	2	50	2	3.70	11.73	22.77
10	2	50	5	7.65	8.90	9.86
10	2	50	10	9.16	11.43	13.09
10	2	100	2	4.67	8.42	11.94
10	2	100	5	6.30	9.92	12.52
10	2	100	10	9.73	11.85	13.65
10	3	25	2	14.50	18.74	24.11
10	3	25	5	11.60	14.04	17.13
10	3	50	2	11.02	15.86	19.42
10	3	50	5	8.70	14.04	21.57
10	3	100	2	11.84	17.59	21.28
10	3	100	5	9.72	13.93	19.65
Resumo dos Valores				2.25	14.14	38.75

Tabela 3.32 – Qualidade da solução dual (em percentagem) – problemas multi-nível com capacidades máximas

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_b</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Cplex		
				<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.08	0.18	0.25	0.15	0.23	0.32	0.05	0.32	0.72
5	2	25	5	0.64	0.79	1.14	1.00	1.20	1.48	16.59	38.08	50.84
5	2	50	2	0.19	0.23	0.29	0.35	0.41	0.50	0.61	1.29	2.17
5	2	50	5	0.65	1.59	2.19	1.73	2.69	3.35	33.31	517.81	951.67
5	2	50	10	10.48	12.77	15.05	18.49	22.61	26.58	200000.75	206468.79	232334.22
5	2	100	2	0.32	0.42	0.52	0.67	0.78	0.92	1.41	10.55	25.25
5	2	100	5	2.84	3.58	5.25	5.65	6.63	9.50	83.86	618.86	1158.03
5	2	100	10	40.95	47.57	54.39	64.80	70.06	74.43	120100.78	184026.70	200010.83
5	3	25	2	0.53	0.84	1.20	1.04	1.34	1.51	3.20	10.95	20.97
5	3	25	5	129.07	152.22	178.02	134.93	159.57	187.20	20903.95	30412.03	52673.70
5	3	50	2	1.08	2.35	4.29	2.14	3.56	5.45	34.92	109.21	239.03
5	3	50	5	38.08	833.24	1190.07	62.19	857.07	1212.58	49697.09	117018.40	222695.70
5	3	50	10	---	---	---	---	---	---	---	---	---
5	3	100	2	3.91	11.76	16.36	8.04	15.92	20.10	193.33	385.45	481.09
5	3	100	5	4041.55	4780.70	5586.15	4091.74	4825.37	5629.60	129352.80	169917.49	200483.42
5	3	100	10	---	---	---	---	---	---	---	---	---
10	2	25	2	0.20	0.30	0.47	0.46	0.58	0.73	0.45	3.94	9.91
10	2	25	5	0.76	1.05	1.36	2.56	4.46	5.71	20.41	1570.96	6784.53
10	2	50	2	0.19	0.44	0.63	0.94	1.10	1.38	1.45	13.12	28.95
10	2	50	5	0.38	1.79	2.30	2.64	6.03	7.79	14.97	3152.88	11040.03
10	2	50	10	16.97	20.95	26.36	43.71	57.93	71.47	52781.30	120585.28	200017.11
10	2	100	2	0.64	0.82	1.10	2.36	2.91	3.92	3.55	29.50	64.30
10	2	100	5	3.86	5.68	7.85	11.28	18.69	22.57	207.81	16433.43	41849.42
10	2	100	10	55.51	96.92	185.79	145.41	188.22	302.07	168137.67	193632.67	200007.84
10	3	25	2	1.69	2.36	2.82	4.54	5.86	6.88	59.70	361.68	692.95
10	3	25	5	216.55	231.07	259.26	219.59	266.27	303.43	13414.86	26325.06	43485.02
10	3	50	2	4.20	6.30	8.61	10.96	13.63	16.82	121.63	409.00	851.14
10	3	50	5	1131.42	1304.53	1592.82	1213.45	1406.67	1709.48	61491.13	101196.99	121325.98
10	3	50	10	---	---	---	---	---	---	---	---	---
10	3	100	2	2.66	5.18	10.88	12.50	14.91	20.75	150.98	17629.35	63419.45
10	3	100	5	643.05	827.58	1090.95	750.42	965.87	1252.08	200054.78	200087.05	200131.09
10	3	100	10	---	---	---	---	---	---	---	---	---
Resumo dos Valores				0.08	298.33	5586.15	0.15	5629.60	389.24	0.05	49677.39	232334.22

Tabela 3.33 – Tempos computacionais (em segundos) – problemas multi-nível com capacidades máximas

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	<i>Tempo Computacional Médio</i>	<i>Número de problemas não resolvidos</i>	<i>Relação</i>
5	2	25	5	31.94	0	26.63
5	2	50	5	487.01	0	181.10
5	2	50	10	---	5	---
5	2	100	2	10.17	0	12.98
5	2	100	5	256.08	0	38.61
5	2	100	10	1963.61	4	28.03
5	3	25	2	9.27	0	6.91
5	3	25	5	774.89	4	4.86
5	3	50	2	103.12	0	28.94
5	3	50	5	---	5	---
5	3	100	2	331.28	0	20.80
5	3	100	5	---	5	---
10	2	25	2	3.78	0	6.54
10	2	25	5	414.26	0	92.79
10	2	50	2	11.69	0	10.66
10	2	50	5	1714.19	0	284.48
10	2	50	10	1693.73	4	29.24
10	2	100	2	24.21	0	8.30
10	2	100	5	3631.22	1	194.25
10	2	100	10	8262.55	4	43.90
10	3	25	2	254.78	0	43.51
10	3	25	5	517.64	2	1.94
10	3	50	2	154.76	0	11.35
10	3	50	5	2912.34	4	2.07
10	3	100	2	1638.29	2	109.91
10	3	100	5	---	5	---
Valores Médios Globais				1145.49		53.99

Tabela 3.34 – Tempos computacionais quando o Cplex é utilizado como procedimento heurístico – problemas multi-nível com restrições de capacidade máxima

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Cplex		
				<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.00	2.45	5.03	0.00	1.21	2.46	0.00	0.00	0.00
5	2	25	5	5.81	7.35	9.31	3.17	4.82	8.38	0.00	0.00	0.01
5	2	50	2	0.00	1.41	4.01	0.00	1.53	5.66	0.00	0.00	0.00
5	2	50	5	0.22	2.97	5.62	0.22	2.84	4.62	0.01	0.01	0.01
5	2	50	10	0.58	3.04	6.90	0.58	2.77	7.10	0.00	0.95	3.53
5	2	100	2	0.00	0.18	0.57	0.00	0.59	2.96	0.00	0.00	0.00
5	2	100	5	0.10	1.90	3.43	0.50	1.42	2.48	0.00	0.01	0.01
5	2	100	10	6.00	9.30	13.32	6.98	8.63	10.09	4.34	7.32	10.13
5	3	25	2	0.00	3.23	10.55	0.00	2.07	4.21	0.00	0.00	0.00
5	3	25	5	4.04	8.93	13.05	3.66	7.83	11.87	0.00	4.05	7.98
5	3	50	2	0.02	6.17	16.78	1.71	4.54	11.67	0.00	0.00	0.00
5	3	50	5	2.84	18.74	44.57	2.66	18.48	44.98	1.79	10.09	26.41
5	3	50	10	---	---	---	---	---	---	---	---	---
5	3	100	2	0.01	4.27	11.02	0.54	2.31	5.25	0.00	0.00	0.00
5	3	100	5	8.14	16.00	26.13	6.78	14.39	22.91	5.06	8.05	14.47
5	3	100	10	---	---	---	---	---	---	---	---	---
10	2	25	2	0.00	0.38	0.95	0.00	0.07	0.35	0.00	0.00	0.00
10	2	25	5	1.21	2.11	4.62	0.80	1.29	2.29	0.01	0.01	0.01
10	2	50	2	0.00	0.43	1.10	0.00	0.38	1.10	0.00	0.00	0.00
10	2	50	5	1.27	2.49	5.35	0.59	1.80	3.28	0.00	0.01	0.01
10	2	50	10	3.45	5.61	9.25	3.14	4.99	7.70	1.26	2.93	4.47
10	2	100	2	0.29	2.06	3.27	0.00	0.77	1.86	0.00	0.00	0.00
10	2	100	5	0.23	1.53	2.92	0.32	1.39	2.43	0.01	0.01	0.01
10	2	100	10	2.82	4.60	5.58	2.63	4.45	5.83	1.74	3.49	5.63
10	3	25	2	1.03	6.18	10.19	0.00	4.76	9.95	0.00	0.00	0.01
10	3	25	5	10.05	13.34	18.93	9.71	11.45	13.27	4.53	6.88	12.71
10	3	50	2	0.53	5.66	13.18	0.50	3.51	9.07	0.01	0.01	0.01
10	3	50	5	6.48	12.56	16.45	3.24	8.56	12.63	3.41	4.95	6.83
10	3	50	10	---	---	---	---	---	---	---	---	---
10	3	100	2	0.10	8.41	17.38	0.59	3.46	11.54	0.00	0.01	0.01
10	3	100	5	---	---	---	---	---	---	---	---	---
10	3	100	10	---	---	---	---	---	---	---	---	---
Resumo dos Valores				0.00	5.60	44.57	0.00	4.46	44.98	0.00	1.81	26.41

Tabela 3.35 – Qualidade da Solução Primal (em porcentagem) – Problemas multi-nível sem conservação de fluxo

Heurística Primal-Dual						
<i>T</i>	<i>K</i>	<i>n</i>	<i>m_k</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	9.30	11.11	14.67
5	2	25	5	13.21	15.31	17.46
5	2	50	2	11.31	15.88	23.47
5	2	50	5	10.57	13.71	18.72
5	2	50	10	11.10	15.25	18.64
5	2	100	2	10.42	13.20	16.26
5	2	100	5	13.61	14.29	15.15
5	2	100	10	17.35	19.88	23.61
5	3	25	2	12.41	14.51	15.85
5	3	25	5	22.60	25.45	29.86
5	3	50	2	15.52	18.56	20.93
5	3	50	5	24.30	28.65	32.41
5	3	50	10	---	---	---
5	3	100	2	16.95	22.95	26.49
5	3	100	5	21.94	26.48	32.20
5	3	100	10	---	---	---
10	2	25	2	4.30	10.66	15.95
10	2	25	5	7.67	10.54	13.21
10	2	50	2	5.02	7.87	10.90
10	2	50	5	8.75	10.77	12.15
10	2	50	10	9.20	12.50	17.40
10	2	100	2	5.27	8.73	11.65
10	2	100	5	8.09	9.44	10.80
10	2	100	10	11.01	13.14	14.18
10	3	25	2	10.39	13.59	20.07
10	3	25	5	18.47	22.20	27.87
10	3	50	2	10.86	14.98	16.87
10	3	50	5	16.10	23.56	32.67
10	3	50	10	---	---	---
10	3	100	2	9.26	13.33	17.82
10	3	100	5	---	---	---
10	3	100	10	---	---	---
Resumo dos Valores				4.30	15.80	32.67

Tabela 3.36 – Qualidade da Solução Dual (em percentagem) – Problemas multi-nível sem conservação de fluxo

<i>T</i>	<i>K</i>	<i>n</i>	<i>m_s</i>	Heurística Primal-Dual			Heurística Primal-Dual + Pesquisa Local			Cplex			
				<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	<i>Melbor</i>	<i>Média</i>	<i>Pior</i>	
5	2	25	2	0.06	0.13	0.23	0.12	0.18	0.30	0.49	1.45	3.33	
5	2	25	5	0.33	0.35	0.38	0.58	0.61	0.66	8.69	185.55	494.98	
5	2	50	2	0.08	0.19	0.31	0.16	0.28	0.38	2.16	6.30	8.72	
5	2	50	5	0.66	0.70	0.84	1.09	1.29	1.52	71.08	981.09	2577.78	
5	2	50	10	0.33	30.86	74.38	0.52	33.33	77.96	6.58	184148.90	509626.50	
5	2	100	2	0.34	0.46	0.56	0.64	0.75	0.88	5.11	11.42	30.25	
5	2	100	5	1.42	2.48	5.14	2.20	3.91	7.16	375.33	1033.13	1861.99	
5	2	100	10	6.13	9.12	16.06	12.44	19.53	31.49	80001.19	80008.01	80020.84	
5	3	25	2	0.28	3.26	12.49	0.56	3.61	13.20	7.16	17.96	24.52	
5	3	25	5	37.66	29908.99	80000.00	82.79	29937.52	80027.70	55350.39	75089.03	80027.47	
5	3	50	2	2.13	7.38	15.91	2.55	8.28	16.47	33.64	191.88	484.09	
5	3	50	5	132.46	37025.52	78644.39	167.94	37055.60	78671.10	80001.20	80008.66	80014.05	
5	3	50	10	---	---	---	---	---	---	---	---	---	
5	3	100	2	5.58	18.58	32.75	7.06	21.96	35.33	135.34	1057.68	3436.05	
5	3	100	5	214.99	3331.74	10583.09	260.27	3398.64	10649.91	80004.77	80017.07	80058.95	
5	3	100	10	---	---	---	---	---	---	---	---	---	
10	2	25	2	0.11	0.23	0.53	0.28	0.40	0.64	0.20	2.81	4.94	
10	2	25	5	0.59	0.75	1.00	1.63	1.93	2.20	288.52	1956.35	5276.19	
10	2	50	2	0.23	0.29	0.39	0.70	0.83	1.06	3.34	6.02	14.16	
10	2	50	5	1.16	1.29	1.47	3.84	4.50	5.14	171.53	15375.36	39945.20	
10	2	50	10	5.28	6.52	9.30	18.58	26.02	35.67	59261.03	72304.86	80002.28	
10	2	100	2	0.70	1.54	3.24	2.05	3.29	5.19	7.78	16.24	32.61	
10	2	100	5	3.05	3.46	4.20	8.39	11.42	14.23	388.16	11822.06	41287.84	
10	2	100	10	17.52	25.14	41.78	39.18	49.17	71.31	80004.09	80004.81	80006.92	
10	3	25	2	1.20	1.87	3.03	2.69	3.55	4.69	16.06	1132.39	4769.31	
10	3	25	5	44.47	2599.39	12638.50	162.33	2755.00	12750.86	40638.84	56490.41	68009.84	
10	3	50	2	2.69	82.77	243.00	9.72	87.80	247.22	73.58	2848.65	12977.69	
10	3	50	5	3.99	29.67	148.34	180.29	258.86	361.14	80023.52	80075.73	80102.91	
10	3	50	10	---	---	---	---	---	---	---	---	---	
10	3	100	2	7.42	61.18	118.20	16.41	78.30	144.77	310.19	16080.93	51504.70	
10	3	100	5	---	---	---	---	---	---	---	---	---	
10	3	100	10	---	---	---	---	---	---	---	---	---	
Resumo dos Valores				0.06	2709.40	80000.00	0.12	2732.09	80027.70	0.20	31143.51	509262.50	

Tabela 3.37 – Tempo Computacional (em segundos) – Problemas multi-nível sem conservação de fluxo

(Página deixada propositadamente em branco)

Capítulo 4

METAHEURÍSTICAS

As desvantagens sentidas aquando do desenvolvimento das heurísticas primais-duais levaram ao desenvolvimento de um algoritmo memético para a resolução dos problemas formulados no capítulo 2. Descrevem-se todos os elementos constituintes do algoritmo, realçando-se o papel fulcral do procedimento de pesquisa local no bom desempenho do algoritmo. Apontam-se as vantagens e desvantagens deste tipo de abordagem, e salienta-se ser esta apenas uma proposta, dentro de um vasto conjunto de propostas possíveis, da construção de uma metaheurística. Consideram-se os problemas com mais do que um objectivo e apontam-se os desenvolvimentos que ainda é necessário realizar nesta vertente do trabalho. Apresentam-se resultados computacionais.

Whatever the reason, there is something profoundly moving about linking a genetic algorithm to a difficult problem and returning later to find that the algorithm has evolved a solution that is better than the one a human found.

Davis, 1991

4.1. Motivação

Sempre que é preciso resolver um problema combinatório difícil, a primeira questão a colocar é qual ou quais os algoritmos que devem ser utilizados para encontrar a solução óptima do problema. Na maior parte das vezes, a melhor maneira de resolver um problema depende fortemente do problema em si e, mesmo, da instância particular com que se está a trabalhar. Muitas vezes, os melhores resultados são obtidos, não à custa de uma única abordagem, mas antes graças à combinação de várias abordagens diferentes (Toth, 2000). E será verdadeiramente necessário em todos os casos encontrar a solução óptima do problema, ou será suficiente encontrar uma boa solução, mesmo que não óptima? E quanto tempo é que se poderá esperar até ter acesso a uma solução óptima ou quase óptima? E será que o caminho até à descoberta de uma boa solução não é tão importante como a solução em si? Será que não ajuda o decisor, aumentando o seu conhecimento acerca do problema, levantando questões e dando ideias em que não se tinha pensado numa fase inicial? Se se pensar num contexto em que existem não um mas vários agentes de decisão, então estas interrogações são ainda mais pertinentes e de mais difícil resposta.

A utilização de métodos exactos, como *branch-and-bound*, têm muitas

provas dadas na resolução de problemas combinatórios, mas muitas vezes à custa de grandes esforços computacionais. Quando confrontados com problemas de muito difícil resolução, em que por vezes só o simples facto de encontrar uma solução admissível é difícil, a utilização de métodos exactos torna-se impraticável, mesmo tendo em conta todos os avanços tecnológicos que aumentaram várias vezes a capacidade dos computadores e, logo, diminuíram tempos computacionais. Citando D. Goldberg, 1989:

...convergence to the best is not an issue in business or in most walks of life; we are only concerned with doing better relative to others. (...) Attainment of the optimum is much less important for complex systems. It would be nice to be perfect, meanwhile we can only strive to improve.

A consciência desta realidade motivou o desenvolvimento de várias heurísticas primais-duais descritas nos capítulos anteriores e que se mostraram eficientes no cálculo de soluções de qualidade para os problemas de localização dinâmicos descritos no capítulo 2.

As metaheurísticas são técnicas de optimização combinatória, desenhadas com o intuito de resolverem o maior número possível de problemas combinatórios (Hertz & Widmer, 2003). As metaheurísticas são naturalmente discretas e apresentam como principal desvantagem a necessidade de, em geral, ser necessário configurar um conjunto de parâmetros (Jones, Mirrazavi & Tamiz, 2002). Apesar disso, provaram já ser capazes de encontrar soluções de elevada qualidade para problemas complexos. No entanto, a construção de uma metaheurística continua, segundo Osman & Kelly, 1996, a ser uma arte. Quando se pretende trabalhar com algoritmos genéticos, por exemplo, existe um grande conjunto de diferentes alternativas de modos de proceder sem que haja linhas de orientação teóricas que possam ajudar a fazer as escolhas mais acertadas (Mitchell, 1996), sendo certo que não é também possível experimentar e comparar todas as alternativas. Exemplos de metaheurísticas largamente utilizadas são a pesquisa tabu, *simulated annealing*, *scatter search*, *evolutionary computing* (incluindo algoritmos genéticos, estratégias evolutivas, programação evolutiva e programação genética). Taillard, Gambardella, Gendreau & Potvin, 2001, consideram que a maioria destas metaheurísticas têm algumas linhas estruturais em comum, nomeadamente a existência de memória ao longo da execução dos algoritmos, pelo que se deve evoluir para a criação de uma abordagem unificadora. Estas metaheurísticas têm sido largamente utilizadas na resolução de problemas combinatórios (como, por exemplo, em C. R. Reeves, 1993a e Taillard *et al.*, 2001), e em problemas de localização em particular.

Hosage & Goodchild, 1986, estudam as possibilidades de se aplicarem algoritmos genéticos a problemas de localização. Concluem que a representação é um elemento crucial no sucesso destes algoritmos. Alves & Almeida, 1992, utilizam *simulated annealing* na resolução de PLS. Pierre Hansen, Filho & Ribeiro, 1992, utilizam pesquisa tabu na resolução de um problema de localização com capacidades de plataformas num sistema de exploração de petróleo. Houck, Joines & Kay, 1996, utilizam algoritmos genéticos para resolverem problemas de

localização-afecção de grandes dimensões, em que cada localização é um ponto no espaço contínuo bidimensional. Rolland, Schilling & Current, 1996, aplicam pesquisa tabu ao problema da p -mediana. J. E. Beasley & Chu, 1996, aplicam algoritmos genéticos ao problema de cobertura, desenvolvendo operadores específicos que garantem a admissibilidade dos indivíduos. Josef Kratica, 1999; Jozef Kratica, Filipovic, Sesum & Tosic, 1996; Josef Kratica, Tosic, Filipovic & Ljubic, 2001, utilizam algoritmos genéticos para resolverem o PLS e propõem uma hibridização de algoritmos genéticos com uma heurística *ADD*. Owen & Daskin, 1998a, 1998b, utilizam algoritmos evolutivos num problema de p -mediana, com a consideração de vários cenários diferentes possíveis. Filipovic, Kratica, Tosic & Ljubic, 2000, melhoram o desempenho do algoritmo genético, aplicando um operador de selecção de *grained tournament*. Vaithyanathan, Burke & Magent, 1996, desenvolvem um algoritmo combinando redes neuronais com pesquisa tabu para problemas combinatórios e aplicam-no ao PLS. L. Lorena & Lopes, 1997, aplicam algoritmos genéticos a problemas de cobertura computacionalmente difíceis. P. Hansen & Mladenovic, 1997, aplicam *variable neighborhood search* ao problema da p -mediana. Bornstein & Azlan, 1998, estudam o problema de localização com capacidades e utilizam *simulated annealing* para o cálculo de valores óptimos para as variáveis de localização cujos valores não é possível fixar através de testes de redução. Owen & Daskin, 1998a, 1998c, utilizam programas evolutivos para resolverem problemas de localização estratégicos, no contexto de planeamento de cenários. Sun, Aronson, McKeown & Drinka, 1998, desenvolvem uma heurística baseada em pesquisa tabu para o problema de transportes de custo fixo, mostrando alguns resultados computacionais. Abdinnour-Helm, 1998, descreve uma heurística híbrida que faz uso de algoritmos genéticos para resolver o problema de localização, e pesquisa tabu para resolver os problemas de afecção em problemas de *hubs* sem capacidades. A ideia de considerar dois processos diferentes de optimização (um para a localização e outro para a afecção) está também presente no trabalho de Righini, 1995. Este autor descreve um algoritmo de duplo *annealing* que trabalha com dois conjuntos de variáveis mutuamente dependentes. J. Jaramillo, Bhadury & Batta, 2002; J. H. Jaramillo, 1998, estudam algoritmos genéticos como um processo alternativo de gerar soluções óptimas, ou quase óptimas, para problemas de localização. Os autores estudam o problema de localização com custos fixos com e sem capacidades, o problema da cobertura máxima e problemas de localização com competição. Concluem que os algoritmos genéticos não deverão ser usados na resolução de problemas de localização com custos fixos e capacidades. V.J.M.F.Filho & Galvão, 1998, desenvolvem uma heurística baseada em pesquisa tabu para o problema de localização de concentradores. Rosing, ReVelle & Schilling, 1999, descrevem uma metaheurística com dois níveis que é particularmente recomendada para resolver problemas de localização como a p -mediana, em que o número de serviços a localizar é pré-definido. Maniezzo, Mingozzi & Baldacci, 1998, propõem uma heurística *bionomic* como um processo eficiente de resolver o problema da p -mediana, e Alp, Erkut & Drezner, 2003, propõem um algoritmo genético para o mesmo problema. A heurística *bionomic* enquadra-se nos algoritmos

evolutivos, possibilitando a utilização de múltiplos progenitores na construção dos filhos, e a existência de conjuntos de soluções de dimensão variável, sendo essencial a execução de um procedimento de otimização local de soluções. Aydin *et al.*, 2002a, b, descrevem um algoritmo de *simulated annealing* com implementação em paralelo para a resolução de PLS de grande dimensão. O algoritmo proposto pelos autores pode ser visto como um algoritmo evolutivo em que uma solução é escolhida aleatoriamente da população e é alterada através da execução de um algoritmo de *simulated annealing*, sendo assim uma abordagem híbrida. Syam, 2002, compara o desempenho de um algoritmo de *simulated annealing* e de uma heurística lagrangeana na resolução de um problema de p -mediana que inclui um sub-problema de distribuição multi-produto. Shimizu & Wada, 2003, formulam um problema de localização e de rotas ótimas como um problema p -hub capacitado e desenvolvem um algoritmo híbrido baseado em pesquisa tabu. Shimizu, 1999, combina um algoritmo genético com programação matemática e considera o problema de localização de aterros para resíduos perigosos, num ambiente multi-objectivo. P. Chardaire, Lutton & Sutter, 1999, calcula limites inferiores para um problema de localização de dois níveis utilizando a relaxação linear e relaxações lagrangeanas do problema que é motivado por uma situação real na indústria de telecomunicações. Soluções admissíveis são calculadas através da utilização de *simulated annealing*. Al-Sultan & Al-Fawzan, 1999, utilizam com sucesso um algoritmo de pesquisa tabu para resolução do problema de localização sem capacidades. Correa, Steiner, Freitas & Carnieri, 2001, consideram um problema real (a seleção de locais para a realização de exames de admissão numa universidade) que formulam como um problema de p -mediana com capacidades e desenvolvem um algoritmo genético com uma abordagem heurística. A heurística é responsável por determinar afectações admissíveis. Antunes & Peeters, 2001, aplicam um algoritmo de *simulated annealing* a um problema de localização dinâmico. Cheung, Langevin & Villeneuve, 2001, descrevem um algoritmo genético, parcialmente implementado como algoritmo paralelo, que aplicam à resolução de problemas de localização de *hubs* e de problemas de localização-afectação na indústria do petróleo. Ghosh, 2003, estuda a aplicação de pesquisa local, pesquisa tabu e pesquisa local com memória ao PLS. Cortinhal & Captivo, 2003a, estudam o problema de localização com capacidades e afectação total e aplicam algoritmos genéticos. Salhi & Gamal, 2003, aplicam algoritmos genéticos a um problema de localização-afectação, onde se pretende encontrar a localização de p serviços no plano para que sirvam n clientes, minimizando os custos totais de transporte. Os autores optam por utilizar uma representação com números reais. Berman, Drezner & Wesolowsky, 2003, aplicam cinco algoritmos baseados em metaheurísticas (entre os quais algoritmos genéticos, pesquisa tabu e *simulated annealing*) a um problema em que se considera a existência de uma probabilidade (função da distância) de que um dado serviço não seja capaz de satisfazer um dado cliente. Resende & Werneck, 2004, propõem uma heurística híbrida para resolução do problema da p -mediana, que é na sua essência um método de *multi-start* em que cada iteração consiste na construção aleatória de uma solução que é depois

submetida a um processo de pesquisa local. Barahona & Chudak, 1999, apresentam uma heurística capaz de lidar com problemas de localização capacitados de grandes dimensões (até 1000 serviços e 1000 clientes), utilizando a relaxação lagrangeana para calcular limites inferiores. Aydin & Fogarty, 2004, utilizam um algoritmo evolutivo em que os operadores de reprodução habituais são substituídos por um algoritmo de *simulated annealing*, na resolução de problemas de localização. Crainic, Gendreau, Hansen & Mladenovic, 2004, apresentam um algoritmo em que várias *variable neighborhood search* metaheurísticas são executadas em paralelo na resolução do problema da p -mediana. Michel & Hentenryck, 2004, apresentam um algoritmo de pesquisa tabu para a resolução do problema de localização simples. Díaz & Fernández, 2004, comparam várias heurísticas que utilizam GRASP, *path relinking* e *scatter search* na resolução do problema da p -mediana com capacidades. Pacheco & Casado, 2005, aplicam *scatter search* na resolução de dois problemas de localização (p -centro e máxima cobertura) com, no máximo, 10 localidades. Domínguez-Marín, Nickel, Hansen & Mladenovic, 2005, resolvem problemas de localização como o p -centro e a p -mediana utilizando algoritmos genéticos e *variable neighborhood search*. Melechovský, Prins & Calvo, 2005, apresentam um modelo para o problema de localização e determinação de rotas óptimas, e resolvem-no através de um algoritmo híbrido, que inclui pesquisa tabu e *variable neighborhood search*. Doerner *et al.*, 2009, utilizam algoritmos genéticos para cálculo de soluções num problema de localização multiobjectivo não linear.

Que razão terá levado ao desenvolvimento de metaheurísticas, após a obtenção de bons resultados com as heurísticas primais-duais? Para além de ser um campo de pesquisa aliciante e que muito se tem desenvolvido nos últimos anos, a principal atracção desta vertente do trabalho prende-se com as características destas metaheurísticas que são opostas às das heurísticas primais-duais, principalmente a rigidez de umas versus a flexibilidade das outras. Como já se referiu anteriormente, as heurísticas primais-duais desenvolvidas são específicas para trabalharem com determinados modelos. Qualquer alteração, mesmo que pequena, nos modelos definidos faz com que seja necessário estudar novos procedimentos ou alterar os já existentes, muitas vezes acarretando largos períodos de desenvolvimento. Note-se que foram desenvolvidas diferentes heurísticas para diferentes problemas, não sendo fácil conjugá-las todas numa única heurística. E é fácil imaginar algumas alterações aos modelos: introdução de novas restrições, imposições por parte do agente de decisão na fixação do valor de uma ou mais variáveis de localização, a consideração de mais do que um objectivo, ou a consideração de vários equipamentos de diferentes características no mesmo modelo. O outro lado da questão prende-se com o ter de deixar um mundo com uma teoria forte, conhecida e que fornece linhas de orientação muito definidas e entrar num mundo onde os resultados experimentais se sobrepõem ainda aos resultados teóricos, e em que não há possibilidade de ter certezas. Em relação aos algoritmos genéticos, o que há de mais parecido com uma teoria que explique o seu funcionamento é a teoria desenvolvida por Hamming: os algoritmos genéticos procuram soluções de boa qualidade através da combinação de blocos

pequenos mas que representam as componentes essenciais de uma boa solução (ver, por exemplo, D. Goldberg, 1989). Não existem garantias de convergência, nem cálculos de limites inferiores que permitam aferir a qualidade da solução. Á. E. Eiben, Hinterding & Michalewicz, 1999, diz mesmo que a teoria relacionada com os algoritmos evolutivos não é, hoje em dia, tida como uma base útil para quem os utiliza na prática.

The development and implementation of meta-heuristic procedures usually entails a fair amount of experimentation and reliance on past experiences. Meta-heuristic methodology is based on principles and not necessarily on theory that can be spelled out with theorems and proofs. (...) The acceptance of the so-called modern optimization techniques is mainly due to the important contributions that meta-heuristic research has made in the area of solving practical optimization problems.

Laguna & Martí, 2003

O principal objectivo desta vertente do trabalho pode enunciar-se como sendo o desenvolvimento de uma heurística capaz de lidar com todos os modelos desenvolvidos (com um ou mais níveis de serviços, com e sem capacidades), permitindo a introdução de restrições adicionais, de preferências por parte do decisor ou, mesmo, mais do que um objectivo.

4.2. O Problema de Localização

O problema de localização que se considerou no desenvolvimento da metaheurística é uma generalização de todos os problemas descritos no capítulo 2. Assim, consideram-se problemas com um ou mais níveis de serviços, em que cada serviço pode ser aberto, fechado e reaberto mais do que uma vez durante o horizonte temporal considerado, e em que os serviços podem ter as seguintes características:

- Tipo 1. Não terem capacidades;
- Tipo 2. Terem capacidades mínimas e/ou máximas com valores de $\theta_i \in [0,1]$.
- Tipo 3. Terem capacidades máximas decrescentes;
- Tipo 4. Serem compostos por um ou mais equipamentos de iguais ou diferentes dimensões.

Definam-se os seguintes conjuntos: I = conjunto de todas as possíveis localizações para serviços; $I_1 = \{i \in I \text{ e } i \text{ é do tipo 1}\}$; $I_2 = \{i \in I \text{ e } i \text{ é do tipo 2}\}$; $I_3 = \{i \in I \text{ e } i \text{ é do tipo 3}\}$; $I_4 = \{i \in I \text{ e } i \text{ é do tipo 4}\}$.

É interessante notar que, após se terem fixado as variáveis de localização (que são da forma a_{it}^s , r_{it}^s , a_{ist}^s e r_{ist}^s como definido no capítulo 2) o valor das variáveis de afectação pode ser calculado de forma mais ou menos fácil, dependendo de se estar perante um problema mono ou multi-nível e dependendo do tipo de serviços existentes. Assim, no caso de problemas com apenas um nível de serviços, tem-se que:

1. Se nenhum serviço tem capacidades, então cada cliente é afecto a exactamente um serviço:

- o mais barato;
2. Se todos os serviços são de tipo 2 e/ou 4, então é necessário resolver um problema de entrepostos (*transshipment*) em cada período temporal;
 3. Se existe pelo menos um serviço do tipo 3, então é necessário resolver um problema de programação linear, ou um procedimento heurístico semelhante ao Algoritmo 3-28. A afectação óptima é mais difícil de calcular, porque não é possível desagregar o problema em T problemas independentes.

No caso multi-nível acontece algo muito semelhante:

1. Se nenhum serviço tem capacidades, então cada cliente é afecto a exactamente um caminho: o mais barato;
2. Se todos os serviços são de tipo 2 e/ou 4, e $\theta_i = 1, \forall i$, então é necessário resolver um problema de entrepostos em cada período temporal;
3. Se existe pelo menos um serviço do tipo 3, então é necessário resolver um problema de programação linear, porque não é possível desagregar o problema em T problemas independentes;
4. Se existe pelo menos um serviço i , tal que $\theta_i \neq 1$, então é também necessário resolver um problema de programação linear.

4.3. O Algoritmo Memético

O algoritmo desenvolvido integra um algoritmo genético e pesquisa local (Joana Dias, Captivo & Clímaco, 2005b; J. Dias, Captivo & Clímaco, 2007d). Tenta-se, deste modo, tirar partido das características dos algoritmos genéticos, incorporando todo o conhecimento existente acerca do problema de localização.

As primeiras experiências efectuadas consideraram a utilização de um algoritmo genético simples, sem pesquisa local. No entanto, os resultados não foram satisfatórios. Por esse motivo, seguiu-se o conselho de C. R. Reeves, 1993a: *Hybridize whenever possible!* Moscato & Cotta, 2003, explicam o sucesso dos algoritmos meméticos como sendo uma consequência directa das sinergias das diferentes abordagens de pesquisa que incorporam.

Segundo Osman & Kelly, 1996, um algoritmo evolutivo é composto por cinco componentes básicas:

1. Uma representação de soluções para o problema.
2. Uma maneira de criar uma população inicial de soluções.
3. Uma função que permita avaliar a qualidade das soluções.
4. Operadores genéticos que alteram a composição genética dos filhos durante o processo de reprodução.
5. Valores para todos os parâmetros do algoritmo.

Estes autores consideram que a estrutura de dados que suporta a representação das soluções para o problema e o conjunto de operadores genéticos utilizados constituem as componentes essenciais de qualquer algoritmo evolutivo. Nas secções seguintes todas as particularidades do algoritmo desenvolvido serão descritas.

4.3.1. Representação de Soluções

Everything should be made as simple as possible, but not simpler. Albert Einstein

Os algoritmos genéticos trabalham com populações de indivíduos, em que cada um representa uma dada solução. Por esse motivo, o primeiro passo na construção do algoritmo tem de passar pela definição de um esquema de representação adequado ao problema (J. Jaramillo *et al.*, 2002). A maneira como as soluções são codificadas é factor central no sucesso de um algoritmo genético. Muitas vezes, conseguir encontrar a melhor representação é quase tão difícil como resolver o problema propriamente dito (Mitchell, 1996), especialmente porque a representação das soluções é uma componente de qualquer algoritmo genético que é limitado apenas pela imaginação de quem está a pôr em prática o algoritmo (Van Veldhuizen, 1999).

Neste campo de pesquisa, a maior parte dos investigadores pede emprestada à Biologia as noções e definições que se referem à codificação de soluções tais como *cromossomas*, *genes* (elementos individuais que constituem um cromossoma), e *alelos* (os possíveis valores que um determinado gene pode tomar). Cada gene está localizado numa dada posição (*locus*) do cromossoma. Se a solução é codificada recorrendo a um único cromossoma, então cada indivíduo da população é *haploide*. Se a solução é codificada usando dois cromossomas (como acontece com o ser humano) é chamado *diploide*. A maioria das aplicações de algoritmos genéticos utiliza indivíduos haploides (Mitchell, 1996), havendo excepções (Sakawa & Kato, 2003). É também usual definirem-se dois espaços diferentes: o espaço do genotipo e do fenotipo. O espaço do genotipo é o espaço em que se processa toda a pesquisa evolucionária (Davis, 1991). O espaço do fenotipo pode ser distinto e é o espaço das soluções do problema (onde, segundo Schuster, 1997, se encontram as verdadeiras fontes da complexidade dos processos evolutivos). Os cromossomas têm geralmente um número fixo de genes e cada gene tem uma localização fixa dentro do cromossoma (uma excepção a esta regra é, por exemplo, o algoritmo genético *messy* descrito em D. E. Goldberg, Deb, Kargupta & Harik, 1993).

Na maior parte dos algoritmos genéticos aplicados a problemas de localização as soluções são representadas por cromossomas, tais que os genes são uma tradução directa das variáveis de decisão de localização. Como exemplo, no caso do PLS com variáveis de localização y_i (iguais a um se o serviço i está aberto e zero em caso contrário), as soluções podem ser codificadas em cromossomas, tais que cada gene corresponde a uma variável y_i .

Como indicado na secção anterior, uma vez fixados os valores das variáveis de localização, é geralmente mais ou menos simples fazer o cálculo do valor óptimo das variáveis de localização¹². Esta simples observação justifica a opção de se codificar apenas parte da solução, a que respeita às variáveis de localização.

Se todos os serviços são dos tipos 1 a 3, então as variáveis de localização são binárias. Se existir pelo menos um serviço do tipo 4, e N_{max} for maior do que

¹² Não se consideram, neste trabalho, os problemas em que as próprias variáveis de afectação têm de tomar valores inteiros. Estes problemas são, geralmente, mais difíceis de resolver (ver, por exemplo, Cortinhal & Captivo, 2003a).

um, então as variáveis r_{sr}^{ξ} podem tomar valores inteiros diferentes de um e zero, o que é mais um factor que dificulta a pesquisa de uma representação válida. De momento, e para facilitar a exposição, vai considerar-se que não existem serviços de tipo 4, o que significa que todas as variáveis de localização são binárias. Voltar-se-á a esta questão mais tarde.

A primeira tentativa que foi feita para encontrar uma codificação para as soluções do problema foi considerando que cada variável a_{ir}^{ξ} e r_{ir}^{ξ} seria representada por um gene (igual a um se a variável correspondente fosse igual a um e zero de outro modo). As variáveis de afectação não eram codificadas, sendo calculadas após a fixação das variáveis de localização. Esta representação apresenta, logo à partida, uma grande desvantagem. Com simples cálculos é imediato concluir que, mesmo pequenas instâncias do problema, necessitariam de cromossomas com um enorme número de genes¹³, a maioria dos quais seria igual a zero. Outra desvantagem desta representação prende-se com a grande dificuldade em gerar soluções admissíveis, quer na fase de inicialização da população, quer no processo geracional.

A observação do comportamento do algoritmo motivou uma outra representação: cada indivíduo é constituído por dois cromossomas, cada um dos quais com mT genes. O primeiro cromossoma, a que se irá chamar *L*-cromossoma, é composto por mT genes que podem tomar o valor zero ou um. O gene na posição $(t-1)m+i$ é igual a um, se o serviço i está aberto durante o período t , e igual a zero, em caso contrário¹⁴. A informação codificada neste cromossoma não é suficiente para construir uma solução para o problema, porque é necessário determinar se houve reaberturas em períodos de tempo consecutivos em que um serviço está em funcionamento: um serviço i pode estar em funcionamento de τ até ξ mas ter sido fechado no fim de χ e reaberto no início de $\chi+1$, com $\tau \leq \chi \leq \xi$. Tal informação não é obtida através deste *L*-cromossoma, pelo que se optou por considerar um segundo cromossoma a que se irá chamar *F*-cromossoma. Neste cromossoma, o gene na posição $(t-1)m+i$ ¹⁵ será igual a um, se o serviço i é reaberto no início do período t , e zero, em caso contrário. O *F*-cromossoma é “menos importante” do que o *L*-cromossoma, pois os *F*-cromossomas complementam a informação dada pelos *L*-cromossomas. Os valores dos seus genes só são tidos em conta quando estritamente necessário. Ir-se-á designar o gene $(t-1)m+i$ no cromossoma *F* ou *L* como *F* ou *L*- $[t,i]$ gene.

Considere-se o seguinte par de cromossomas indicados na Tabela 4.1, que codificam uma solução com m igual a 5 e T igual a 3.

¹³ Para um problema com m possíveis localizações para serviços e T períodos temporais, o número de variáveis de localização pode atingir as $2mT \frac{1+T}{2}$.

¹⁴ Cada *L*-cromossoma pode ser interpretado como uma matriz com T linhas e m colunas, tal que o gene na posição $[t,i]$ é igual a 1 se o serviço i está aberto durante t e 0 em caso contrário.

¹⁵ Tal como acontece com o *L*-cromossoma, este cromossoma pode ser considerado como uma matriz com T linhas e m colunas.

		<i>L</i> – Cromossoma							<i>F</i> – Cromossoma				
<i>t</i> \ <i>i</i>		1	2	3	4	5	<i>t</i> \ <i>i</i>		1	2	3	4	5
1		1	0	0	1	1	1		1	1	0	0	1
2		1	1	0	0	0	2		0	0	0	1	1
3		1	1	0	1	0	3		1	0	1	0	1

Tabela 4.1 – Representação de uma solução

Em termos de variáveis de localização, estes dois cromossomas serão interpretados como uma solução em que todas as variáveis são iguais a zero, à exceção de a_{11}^2 , r_{13}^3 , a_{22}^3 , a_{41}^1 , r_{43}^3 , a_{51}^1 . Os genes do *F*-cromossoma, que estão representados a negrito, são os únicos que realmente têm de ser tidos em conta para construir a solução.

Definição 4.1: Considerem-se dois indivíduos que diferem apenas num único gene do *L(F)*- cromossoma. Se as soluções que eles representam no espaço do fenotipo são diferentes, então o gene é chamado de determinante; caso contrário, é designado como não-determinante.

Proposição 4-1: Todos os genes dos *L*-cromossomas são determinantes.

Proposição 4-2: De todos os genes de um *F*-cromossoma, os únicos determinantes são aqueles na posição $(t-1)m+i$, para algum i e $t>1$, tal que os genes do *L*-cromossoma nas posições $(t-1)m+i$ e $(t-2)m+i$ são iguais a um.

Proposição 4-3: É possível representar toda e qualquer solução de um problema de localização pertencente à classe de problemas descritos em 4.2 através de um par de *L* e *F*- cromossomas.

Reconhece-se, de imediato, que esta representação é redundante, de acordo com a definição de Rothlauf & Goldberg, 2002: as representações são redundantes se o número de genótipos excede o número de fenótipos. Observando o exemplo anterior, existe um número de indivíduos diferentes entre si (no espaço genótipo) que serão mapeados na mesma solução (no espaço fenótipo): todos os indivíduos que diferem do indivíduo representado na Tabela 4.1 em, pelo menos, um gene não determinante do cromossoma *F*.¹⁶ No entanto dois indivíduos só poderão corresponder à mesma solução se os seus *L*-cromossomas forem exactamente iguais (devido à Proposição 4-1). Rothlauf & Goldberg, 2002, estudam o efeito das representações redundantes no desempenho dos algoritmos genéticos. Alguns autores partilham a opinião que todas as soluções deveriam ser codificadas com exactamente o mesmo número de indivíduos diferentes. A justificação é óbvia: se algumas soluções estão “super-representadas”, então a pesquisa através do algoritmo genético pode ser viciada. Rothlauf & Goldberg, 2002, são da opinião que representações redundantes designadas por “*synonymously*”, isto é, representações tais que os genótipos que representam os mesmos fenótipos são muito similares entre si, não alteram o desempenho do algoritmo desde que todos os fenótipos sejam representados, em média, pelo mesmo número de genótipos diferentes. No caso

¹⁶ Se existirem n genes determinantes no *F*-cromossoma, isto significa que poderão existir $2^{(Tm-n)}$ indivíduos diferentes que codificam exactamente a mesma solução primal.

da representação adoptada, os indivíduos que representam a mesma solução podem ser considerados semelhantes entre si, pois partilham o mesmo L -cromossoma. Note-se ainda que, no caso da representação descrita, pode evitar-se a redundância se no processo de geração de soluções, se optar por igualar a zero todos os genes não determinantes do F -cromossoma.

A representação escolhida apresenta uma vantagem que generaliza os problemas de localização estudados. Em todos os modelos de localização desenvolvidos considera-se possível que um dado serviço possa ser aberto, fechado e reaberto mais do que uma vez durante o horizonte temporal considerado. O decisor pode ainda determinar períodos mínimos de funcionamento para um serviço que é aberto. O que não é permitido fazer-se é impedir que um serviço seja reaberto durante um intervalo de tempo em que está consecutivamente em funcionamento. Imagine-se que o decisor decide que não faz sentido fechar um serviço no fim de t e reabri-lo no início de $t+1$. Tal só poderá ser incluído na formulação do problema através de restrições adicionais da forma (4-1), que irão complicar o desenvolvimento das heurísticas primais-duais.

$$a_{ir}^t + r_{ir}^t + \sum_{\xi=t+1}^T r_{i(t+1)}^{\xi} \leq 1 \quad \forall 1 \leq \tau \leq t \leq \xi \leq T \quad (4-1)$$

No caso do algoritmo genético, e em particular da representação escolhida, tal imposição por parte do decisor implica apenas que todos os genes do cromossoma F são não determinantes, ou seja, que este cromossoma não influencia em nada a solução pois todos os seus genes são considerados iguais a zero. Como “não há bela sem senão”, a questão da imposição de períodos mínimos de funcionamento para serviços abertos não pode ser tratada de forma tão simples como nas heurísticas primais-duais, em que é suficiente considerar variáveis de localização a_{ir}^{ξ} e r_{ir}^{ξ} tais que $\xi - \tau$ seja superior a esse limiar mínimo de períodos consecutivos de funcionamento. A esta questão se voltará mais tarde.

Observe-se, agora, a adequabilidade desta representação em relação às restrições do problema. Todas as restrições do problema, que se prendem com as relações das variáveis de localização entre si, são automaticamente satisfeitas por qualquer indivíduo. No entanto, é possível que sejam violadas as restrições de capacidade máxima, mínima ou máxima decrescente.

Olhe-se novamente para o problema e considere-se a existência de serviços do tipo 4. A complicação adicional que estes serviços acarretam prende-se com o facto de que existem variáveis de localização inteiras não binárias. Em vez de criar uma outra representação para as variáveis não binárias, o que iria complicar a utilização deste algoritmo em problemas que incluam serviços de diferentes tipos, optou-se por adaptar a representação já descrita a esta nova situação. Assim, cada serviço i do tipo 4 pode ser constituído por até $Nmax$ elementos de iguais ou diferentes dimensões. Isto significa que, em cada período temporal, existem no máximo $Nmax$ elementos de cada possível dimensão. Por este motivo, cada serviço i é representado por $q \times Nmax$ genes,

em cada período t . Cada serviço $i \in I_4$ é transformado em $q \times Nmax$ serviços chamados *serviços fictícios*. Como exemplo, considere-se que todos os serviços são do tipo 4, T é igual a um, m é igual a 3, q é igual a 4 e $Nmax$ é igual a 2. Então, cada cromossoma terá $m \times q \times Nmax$ genes, isto é, 24 genes organizados como se representa na Figura 4.1.

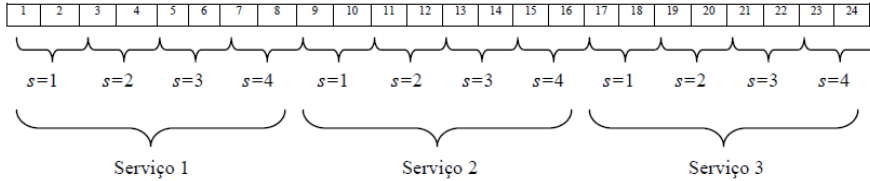


Figura 4.1 – Representação esquemática de um cromossoma na presença de serviços do tipo 4

Esta representação é facilmente traduzida para variáveis de decisão: as variáveis $r_{i\tau}^{\xi}$ são iguais ao número de *serviços fictícios* que correspondem ao serviço i e dimensão s que funcionam de τ a ξ . De todas as variáveis $r_{i\tau}^{\xi}$, com valores maiores do que zero, escolhe-se $r_{i\tau'}^{\xi}$, tal que $\tau' = \min_{\tau, s} \{r_{i\tau}^{\xi} : r_{i\tau}^{\xi} > 0\}$, diminui-se o valor desta variável em uma unidade e altera-se a variável $a_{i\tau'}^{\xi}$ de zero para um. Note-se que, por cada *serviço fictício* i , correspondente a um serviço $i \in I_4$ e dimensão $s \in S$, pode definir-se $FR_{i\tau'}^{\xi}$ igual a $FR_{i\tau}^{\xi}$ e $FA_{i\tau'}^{\xi}$ igual a $FA_{i\tau}^{\xi}$. O problema de localização pode ser formulado, recorrendo apenas a variáveis $a_{i\tau}^{\xi}$ e $r_{i\tau}^{\xi}$, com $i=1, \dots, \#I_1+\#I_2+\#I_3+q \times Nmax \times \#I_4$.

No caso da existência de serviços de tipo 4, esta representação não garante a satisfação das restrições de capacidade máxima, nem a satisfação das restrições que impõem o número máximo de elementos que podem estar simultaneamente em funcionamento em i . Apresenta como maior vantagem a uniformidade de representação entre todos os tipos de serviços, o que facilita a definição dos operadores genéticos, e como principal desvantagem o aumento significativo no número de genes de cada cromossoma. Esta desvantagem pode mesmo impossibilitar a aplicação do algoritmo a problemas com elevados valores para $Nmax$ e com um elevado número de possíveis dimensões para os equipamentos. No entanto, também a heurística primal-dual não consegue lidar bem com esta situação.

4.3.2. Função de Avaliação de Indivíduos

Neste algoritmo, o valor de desempenho (*fitness*) de cada indivíduo da população é igual ao valor da função objectivo da solução correspondente no espaço do fenotipo. O cálculo dos custos fixos de abertura e reabertura faz-se directamente a partir da representação de cada solução (Algoritmo 4-1). Se o serviço $i \in I_4$, então este algoritmo é utilizado para cada um dos $q \times Nmax$ serviços fictícios, sendo os custos somados. Após se ter decidido qual a variável $a_{i\tau'}^{\xi}$ que é igual a um (como se descreveu já na secção anterior), o custo total é alterado,

somando FA_{isr}^{ξ} e subtraindo FR_{isr}^{ξ} . Os custos de afectação são depois calculados, dependendo este cálculo do tipo de problema. É no cálculo das afectações de clientes a serviços ou caminhos que se pode descobrir a não admissibilidade da solução representada por um determinado indivíduo. Se tal acontece, o valor de desempenho do indivíduo iguala-se a $+\infty$. O mesmo acontece se o número máximo de elementos simultaneamente em funcionamento numa dada localidade ($Nmax$) é excedido.

Algoritmo 4-1

PROCEDIMENTO DE CÁLCULO DOS CUSTOS FIXOS

T – número de períodos temporais; i – serviço em relação ao qual se vão calcular os custos fixos; FA_i^{ξ} - custo de abertura do serviço i quando este está aberto de t a ξ ; FR_i^{ξ} - custo de reabertura do serviço i quando este está aberto de t a ξ .

-
1. $custo \leftarrow 0$; $open \leftarrow false$; $t \leftarrow 1$.
 2. Se $t > T$ então parar, se não ir para 3.
 3. Se $L[t, i] = false$ então $t \leftarrow t + 1$ e ir para 2; se não ir para 4.
 4. $tin \leftarrow t$; $t \leftarrow t + 1$;
 5. Se $L[t, i] = 1$ então ir para 6; se não ir para 7.
 6. Se $F[t, i] = 1$ e $tin \neq t$ então ir para 7, se não $t \leftarrow t + 1$ e ir para 5.
 7. $tend \leftarrow t - 1$.
 8. Se $open = false$ e $i \notin I_4$ então $open \leftarrow true$, $cost \leftarrow cost + FA_{i\ tin}^{tend}$; se não $cost \leftarrow cost + FR_{i\ tin}^{tend}$. Ir para 2.
-

4.3.3. Operadores Genéticos

O algoritmo genético desenvolvido utiliza os operadores comuns encontrados na literatura: selecção, *crossover* e mutação. Também foram desenvolvidos alguns operadores especiais que tiram proveito do conhecimento da estrutura do problema, nomeadamente um algoritmo de reparação (que diminui o número de indivíduos não admissíveis) e um procedimento que altera os genes do F -cromossoma (que tenta diminuir os custos fixos de abertura e reabertura).

4.3.3.1. Selecção

O operador de selecção é baseado no “*binary tournament selection* com *sharing*” (D. Goldberg, 1989, Oei, Goldberg & Chang, 1991, Deb, 2001). A. E. Eiben & Smith, 2003, apresentam como principais vantagens da selecção por torneio o facto de não ser necessário recorrer a um conhecimento global da população, pois é apenas necessário comparar um número pequeno de indivíduos e ser conceptualmente simples e rápido de executar.

Em cada geração, dois indivíduos são aleatoriamente seleccionados a partir da população “pai”. Um valor de *sharing* é calculado para cada um deles. Estes valores de *sharing* são utilizados para prevenir a convergência prematura da população em torno de uma única solução, e são calculados da seguinte forma: dados dois indivíduos i e j , a distância entre eles é dada pelo número de genes do L -cromossoma que são diferentes em ambos os indivíduos.

$$d_{ij}^{\vartheta} = \begin{cases} 1, & \text{se o } \vartheta\text{-gene no } L\text{-cromossoma é diferente em } i \text{ e } j \\ 0, & \text{em caso contrário} \end{cases} \quad (4-2)$$

$$d_{ij} = \sum_{\vartheta=1}^{mT} d_{ij}^{\vartheta} \quad (4-3)$$

$$sb(i, j) = \begin{cases} 1 - \left(\frac{d_{ij}}{\alpha_{share}} \right)^{\alpha}, & \text{se } d_{ij} \leq \alpha_{share} \\ 0, & \text{em caso contrário} \end{cases} \quad (4-4)$$

Para cada indivíduo i seleccionado, todos os valores $sb(i, j)$ ¹⁷ (calculado considerando todos os indivíduos j que pertencem à nova população – população filha) são somados obtendo-se o valor nc_i (4-5).

Se, no momento da selecção, existem já num indivíduos na nova população, então o valor de nc_i é alterado como indicado em (4-6). O valor de desempenho de cada indivíduo é dividido por nc_i e o valor resultante ($f(i)$) é usado no mecanismo de selecção. Na presença de dois indivíduos aleatoriamente seleccionados x_1 e x_2 , se $f(x_1) < f(x_2)$, então o indivíduo x_1 ganha o torneio com uma dada probabilidade p_{bt} ¹⁸.

$$nc_i = \sum_{j \text{ pertence à nova população}} sb(i, j) \quad (4-5)$$

$$nc_i = num - nc_i \quad (4-6)$$

4.3.3.2. Crossover

O operador de *crossover* utilizado é uma adaptação do *one-point crossover*. Duas soluções pai serão recombinadas, dando origem a dois filhos. Um valor κ entre 1 e T é aleatoriamente escolhido. O primeiro filho terá todos os genes dos cromossomas L e F nas posições $(t-1)m+i$, com $t < \kappa$, iguais ao primeiro progenitor e todos os outros genes iguais ao segundo progenitor. O oposto acontece com o segundo filho. Este operador garante que, se os pais satisfazem a restrição de número máximo de elementos simultaneamente em funcionamento em i ser inferior a $Nmax$, o mesmo acontece com os seus filhos. No entanto, este operador não garante que os filhos de dois indivíduos admissíveis sejam também admissíveis.

Note-se que a recombinação de soluções não precisa de estar limitada a dois progenitores. Exemplos de algoritmos que utilizam mais do que dois progenitores podem encontrar-se em A. E. Eiben, 2000, 2002b; Tsutsui & Ghosh, 1998.

4.3.3.3. Mutação

O operador de mutação é responsável por alterações aleatórias no genotipo

¹⁷ O valor de α_{share} é calculado como descrito em Deb, 2001, e é considerado igual a um.

¹⁸ Esta probabilidade deverá ser próxima de um.

de um indivíduo. Segundo D. Goldberg, 1989, a mutação é simplesmente uma apólice de seguro contra a perda de material genético.

Cada gene de L e F -cromossomas é alterado com base numa dada probabilidade de mutação p_{μ} , que é normalmente muito próxima de zero. Para cada serviço de tipo 4, apenas um gene de cada L e F -cromossomas pode ser alterado em cada período.

4.3.3.4. Operadores Genéticos Adicionais

O algoritmo genético desenvolvido tenta tirar partido de todo o conhecimento existente acerca do problema. Por este motivo, dois operadores adicionais foram desenhados. Um operador tenta diminuir o número de indivíduos não admissíveis em cada geração. Este operador é um procedimento de “reparação” que foi motivado pelo enorme número de indivíduos não admissíveis em cada população (especialmente quando em presença de serviços do tipo 3 e 4), que eram responsáveis pelo fraco desempenho do algoritmo (que conseguia apenas encontrar soluções de má qualidade). Tal observação está de acordo com Davis, 1991: um algoritmo genético que gera muitas soluções “ilegais” – não admissíveis – irá sempre ter um desempenho pior do que um algoritmo que não gere soluções ilegais. O outro operador desenvolvido tenta alterar os genes determinantes do F -cromossoma por forma a diminuir os custos fixos de abertura e reabertura.

4.3.3.4.1. Operador Reparador

Um indivíduo representa uma solução não admissível se:

1. Viola alguma restrição de capacidade;
2. O número total de elementos localizados num serviço i do tipo 4 é maior do que N_{max} .

Estas violações são causadas apenas por genes de L -cromossomas.

O operador reparador altera de forma aleatória, mas guiada, os genes do L -cromossoma. O Algoritmo 4-2 exemplifica o esquema de funcionamento deste operador, quando em presença de um problema com um nível. Se as restrições de capacidade máxima são violadas em t , o algoritmo abre de forma aleatória mais serviços, modificando genes de zero para um, tal que as restrições de capacidade mínima continuem a ser satisfeitas. Se são estas restrições de capacidade mínima a serem violadas em t , então fecham-se serviços aleatoriamente escolhidos (alterando genes de um para zero) de forma a que as restrições de capacidade máxima continuem a ser satisfeitas. Se existem serviços de tipo 3, então a afectação óptima dos clientes não pode ser calculada através da resolução de T problemas independentes. Neste caso, é utilizada uma heurística para testar e tentar atingir a admissibilidade do indivíduo. Se o número máximo de elementos localizados em i (sendo i um serviço do tipo 4) é excedido, então o operador escolhe aleatoriamente genes i' iguais a um, que correspondem a elementos de i e altera o seu valor para zero.

No caso de se estar perante um problema multi-nível, o processo é muito

semelhante: no caso de não existirem serviços com capacidade, se não existe nenhum caminho aberto em t , então o operador escolhe aleatoriamente um caminho e abre-o. Se existirem serviços de tipo 4 que violem a restrição do número máximo de elementos em funcionamento num dado período t , o operador repete o procedimento descrito anteriormente. Se existirem serviços com capacidade no último nível, então o operador verifica as relações existentes entre a oferta total e as capacidades máxima e mínima totais dos serviços abertos no último nível, fechando ou abrindo serviços por forma a encontrar um equilíbrio admissível entre estes valores. No caso de existirem serviços com valores para θ_i diferentes de um, ou se existirem serviços com capacidade máxima decrescente, o operador não altera a solução (torna-se muito complicado conseguir encontrar um algoritmo que, de uma forma aleatória e rápida, tente encontrar a admissibilidade da solução)¹⁹.

Como todas estas alterações são levadas a cabo de forma aleatória, este operador não garante a reparação total de um indivíduo, ou seja, o indivíduo pode continuar inadmissível. É por este motivo que existe a necessidade de se impôr um limite máximo ao número de iterações executadas. A escolha de se desenvolver um algoritmo fortemente baseado em alterações aleatórias em vez da utilização de algoritmos mais estruturados (tais como heurísticas *greedy* ou mesmo procedimentos primais semelhantes aos descritos em III) prende-se com o facto de se pretender evitar a introdução de desvios no processo de pesquisa genética (tal como sugere Coello Coello, 2002).

Algoritmo 4-2

OPERADOR REPARADOR

Parâmetros pré-definidos: NMAXTRY – número total de iterações em cada período temporal.

d'_j – procura do cliente j durante t ; S – conjunto de possíveis dimensões para elementos de serviços de tipo 4; Q_i – capacidade máxima do serviço i ; Q'_i – capacidade mínima do serviço i .

1. Se existirem serviços de tipo 3 então ir para 2, se não ir para 3.
2. Resolver o problema de afectação, recorrendo a um *general solver*. Se o problema é impossível, então ir para 3; se não, parar (o indivíduo corresponde a uma solução admissível).
3. $t \leftarrow 1$. Para cada serviço $i \in I_3$, calcular $Cap_i^0 \leftarrow 0$.
4. $ntries \leftarrow 1$. Se $t > T$ então parar, se não ir para 5.
5. Para cada serviço $i \in I_3$ actualizar $Cap_i^t \leftarrow \begin{cases} Cap_i^{t-1} + Q_i, & \text{se } L-[t,i] = 1 \text{ e } F-[t,i] = 1 \\ Cap_i^{t-1}, & \text{em caso contrário} \end{cases}$. Este valor representa a capacidade máxima de i durante o período t .
6. Calcular $D \leftarrow \sum_j d'_j$, $C_{max} \leftarrow$ somatório das capacidades disponíveis de serviços em funcionamento durante t ²⁰, $C_{min} \leftarrow$ somatório das capacidades mínimas dos serviços abertos em t , $Num_i \leftarrow$ número total de elementos abertos em $i \in I_4$, durante t , $\forall i \in I_4$.

¹⁹ Não se testou a possibilidade de utilizar os procedimentos primais desenvolvidos no capítulo 2 neste operador reparador. É uma possibilidade interessante, pois os procedimentos primais baseiam-se numa solução primal, que pode não ser admissível, e tentam construir uma solução admissível. A maior dificuldade neste ponto prende-se com o número de vezes que estes procedimentos teriam de ser executados, muito superior ao número de vezes que são chamados nas heurísticas primais-duais. Não deixa, no entanto, de ser uma ideia a explorar.

²⁰ Se existe pelo menos um serviço do tipo 1 em funcionamento em t , então $C_{max} \leftarrow +\infty$.

7. $ntries \leftarrow ntries + 1$. Se $ntries > NMAXTRY$ então parar, se não ir para 8.
 8. Se $C_{min} > D$, então ir para 9. Se $C_{max} < D$, então ir para 10. Se $\exists i \in I_1: Num_i > Nmax$, então ir para 11. Caso contrário, ir para 13.
 9. Escolher aleatoriamente um serviço $i \in I_2$, tal que $L-[t, i] = 1$ e $C_{max} - Q_i \geq D$. $L-[t, i] \leftarrow 0$, $C_{min} \leftarrow C_{min} - Q_i$, $C_{max} \leftarrow C_{max} - Q_i$. Ir para 7.
 10. Escolher aleatoriamente um serviço i (incluindo serviços fictícios), tal que $L-[t, i] = 0$ e $C_{min} + Q_i \leq D$ ²¹. $L-[t, i] \leftarrow 1$, $C_{min} \leftarrow C_{min} + Q_i$. Se $i \in I_3$, então $C_{max} \leftarrow C_{max} + Cap_i$; se não, $C_{max} \leftarrow C_{max} + Q_i$ ²². Se i é um serviço fictício, então $Num_i \leftarrow Num_i + 1$, com i a representar o serviço de tipo 4 correspondente. Ir para 7.
 11. Escolher aleatoriamente um serviço $i \in I_4$ tal que $Num_i > Nmax$.
 12. Escolher aleatoriamente um serviço fictício \tilde{i} , que corresponde a um elemento aberto em i de dimensão s , $s \in S$. $L-[t, \tilde{i}] \leftarrow 0$, $Num_i \leftarrow Num_i - 1$, $C_{max} \leftarrow C_{max} - Q_s$.
 13. Se não existem serviços de tipo 3, então $t \leftarrow t + 1$ e ir para 4. Caso contrário, ir para 14.
 14. Resolver um problema de transportes (ou entrepostos), em que se consideram os serviços de tipo 3 como sendo de tipo 2, com capacidades máximas iguais a Cap_i e capacidades mínimas iguais a zero. Atualizar $Cap_i, \forall i \in I_3$, subtraindo o fluxo total que chega a cada serviço i . $t \leftarrow t + 1$ e ir para 4.
-

4.3.3.4.2. O Procedimento “Altera Aberturas”

Este procedimento estuda o efeito de alterar os períodos de reabertura definidos pelos genes determinantes do F -cromossoma. Tal como indicado na Proposição 4-2, podem identificar-se os genes determinantes no F -cromossoma, se os genes no L -cromossoma são iguais a um para t e para o período imediatamente anterior, para algum serviço i .

O procedimento não experimenta alterar todo e qualquer gene determinante do F -cromossoma, porque tal seria demasiado pesado em termos do tempo computacional necessário. O procedimento identifica situações, tais que um serviço i está aberto desde o início de τ até ao fim de ξ , $\xi > \tau$, e é reaberto durante esse intervalo (num período $t \leq \xi$). Isto significa que existe um gene determinante no F -cromossoma na posição $(t-1)m+i$ que é igual a um, e é exactamente o valor deste gene que o procedimento tenta alterar de um para zero. Se os custos fixos diminuem, então o gene é alterado, caso contrário mantém o seu valor original.

O procedimento não altera genes correspondentes a serviços de tipo 3, pois alterações deste género levariam não só a uma alteração dos custos, mas também a alterações no valor das capacidades disponíveis em cada período temporal, o que poderia pôr em risco a admissibilidade do indivíduo.

²¹ Se $i \in I_3$ ou i é um serviço fictício, então Q_i é igual a zero.

²² Se i é um serviço fictício, que corresponde a um elemento de dimensão s , então Q_i será igual a Q_s .

Algoritmo 4-3

PROCEDIMENTO "ALTERA ABERTURAS"

I_3 – conjunto de serviços do tipo 3; I_4 – conjunto de serviços do tipo 4; m – número total de possíveis localizações para serviços; FA_{it}^{ξ} – custo de abertura do serviço i quando este está aberto de t a ξ ; FR_{it}^{ξ} – custo de abertura do serviço i quando este está aberto de t a ξ .

1. $i \leftarrow -1$;
 2. Se $i \in I_3$ então $i \leftarrow i + 1$.
 3. Se $i > m$, então parar. Se $i \notin I_4$ ir para 4, se não ir para 6.
 4. Detectar um par de variáveis de localização iguais a 1 do tipo $(a_{ir}^{\xi}, r_{i\xi+1}^{\nu})$ ou $(r_{ir}^{\xi}, r_{i\xi+1}^{\nu})$. Se não existe qualquer par de variáveis nesta situação então $i \leftarrow i+1$ e ir para 3. Caso contrário, ir para 5.
 5. $\Delta \leftarrow FA_{ir}^{\nu}$ (or FR_{ir}^{ν}) – FA_{ir}^{ξ} (or FR_{ir}^{ξ}) – $FR_{i\xi+1}^{\nu}$. Se $\Delta < 0$ então $F\text{-}[\xi+1, i] \leftarrow 0$, a_{ir}^{ξ} (ou r_{ir}^{ξ}) $\leftarrow 0$, $r_{i\xi+1}^{\nu} \leftarrow 0$ e a_{ir}^{ν} (ou r_{ir}^{ν}) $\leftarrow 1$. Ir para 4.
 6. Detectar um par de variáveis de localização maiores do que zero do tipo $(a_{ist}^{\xi}, r_{is\xi+1}^{\nu})$ ou $(r_{ist}^{\xi}, r_{is\xi+1}^{\nu})$. Se não existe qualquer par de variáveis nesta situação, então $i \leftarrow i+1$ e ir para 3. Caso contrário, ir para 7.
 7. $\Delta \leftarrow FA_{ist}^{\nu}$ (or FR_{ist}^{ν}) – FA_{ist}^{ξ} (or FR_{ist}^{ξ}) – $FR_{is\xi+1}^{\nu}$. Se $\Delta < 0$ então $F\text{-}[\xi+1, i] \leftarrow 0$, tal que i é um serviço fictício, correspondendo ao serviço i , com dimensão s , e $F\text{-}[\xi+1, i] = 1$, $a_{ist}^{\xi} \leftarrow 0$ ou $r_{ist}^{\xi} \leftarrow r_{ist}^{\xi} - 1$, $r_{is\xi+1}^{\nu} \leftarrow r_{is\xi+1}^{\nu} - 1$, e $a_{ist}^{\nu} \leftarrow 1$ ou $r_{ist}^{\nu} \leftarrow r_{ist}^{\nu} + 1$. Ir para 4.
-

4.3.4. Pesquisa Local

A pesquisa local revela-se de importância fundamental no algoritmo desenvolvido. As primeiras versões deste algoritmo não incluíam nenhum tipo de pesquisa local, e os resultados obtidos foram claramente maus. Vários são os exemplos encontrados na literatura que referem algoritmos genéticos hibridizados com pesquisa local (Huntley & Brown, 1996; Murata, Ishibuchi & Gen, 1998; C. Reeves & Höhn, 1996; Yagiura & Ibaraki, 1996; Krasnogor & Smith, 2005).

A pesquisa local não foi incluída na secção dedicada aos operadores genéticos pois, apesar de poder ser interpretada como um operador (Ishibuchi & Yoshida, 2002, dizem que pode mesmo ser interpretada como um caso especial de mutação), tem características próprias. Na realidade, enquanto que operadores como a selecção, mutação e *crossover* influenciam a população como um todo e são verdadeiramente operadores evolutivos, a pesquisa local trabalha com base em soluções individuais sem ter uma perspectiva global da população em que o indivíduo está inserido. Sinha & Goldberg, 2003, são da opinião que operadores genéticos como *crossover* são responsáveis por uma pesquisa global, enquanto que a pesquisa local corresponde apenas a um caminho individual de aprendizagem.

Neste algoritmo, a pesquisa local é executada após o *crossover* e a mutação. Cada indivíduo pertencente à nova população é uma potencial solução de partida para o procedimento de pesquisa local, que é executado com uma dada probabilidade p_{ls} . Se um filho é igual a um dos seus pais que, por sua vez, é o resultado do procedimento de uma pesquisa local, então esta probabilidade será

igual a zero, pois não se ganhará nada voltar a executar este procedimento partindo do filho.

Considere-se a seguinte definição:

Definição 4.2: Um indivíduo x' está na k -vizinhança de um indivíduo x se e só se x' difere de x pela introdução ou remoção de, no máximo, k períodos contínuos de funcionamento relativos a um único serviço i ²³. Isto significa que o genotipo de x' e x diferem apenas em k genes de um L -cromossoma e no máximo k genes no F -cromossoma.

A definição da vizinhança é factor extremamente importante no desempenho de qualquer procedimento de pesquisa local, juntamente com a solução inicial e com o tempo computacional necessário para calcular o valor da função objectivo (Laguna & Martí, 2003). O algoritmo genético, ele próprio, pode ser considerado como um procedimento de pesquisa local em que se exploram simultaneamente as vizinhanças de várias soluções sendo que as interações existentes fazem com que as vizinhanças visitadas sejam diferentes da simples união de vizinhanças (Pirlot, 1996).

Entre todos os indivíduos x' que pertencem a uma k -vizinhança de x com os mesmos valores nos k genes do L -cromossoma, a pesquisa local visita apenas os indivíduos, tais que os correspondentes genes do F -cromossoma representam o custo mínimo de abertura e reabertura. Considere-se um indivíduo x , tal que o serviço i está em funcionamento de τ até ξ e de $\xi+k+1$ até φ . Existem diversos indivíduos x' na k -vizinhança de x , tais que i está continuamente em funcionamento de τ até φ . Estes indivíduos apresentam, uns em relação aos outros, diferenças nos genes determinantes do F -cromossoma em posições $[\xi+1, i]$ a $[\xi+k+1, i]$. A pesquisa local irá apenas considerar o indivíduo x' que corresponde à solução com o menor custo fixo.

Exemplificando com k igual a dois, a situação é ilustrada na Figura 4.2, em que a pesquisa local irá experimentar colocar o serviço i em funcionamento em $\xi+1$ e $\xi+2$. A pesquisa local considera quatro possibilidades, dependendo dos genes do L -cromossoma: 1) funcionar em contínuo de τ a φ sem quaisquer reaberturas; 2) reabrir em $\xi + 1$ e $\xi + 3$; 3) reabrir apenas em $\xi + 1$; 4) reabrir apenas em $\xi + 3$.

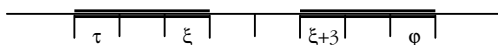


Figura 4.2 – Períodos não contínuos de funcionamento. ■ representa os períodos de funcionamento

O indivíduo que corresponder à solução com o menor custo, em termos de custos fixos, será o único visitado. Se não existirem serviços do tipo 3, visitar apenas este subconjunto dos vizinhos dará exactamente o mesmo resultado do que visitar toda a vizinhança. No entanto, na presença de serviços do tipo 3, os

²³ Nesta definição, se o serviço i é de tipo 4, então *serviço* deve ser substituído por *elemento de um serviço*.

períodos de reabertura não afectam apenas os custos fixos mas também os custos de afectação (porque as capacidades máximas disponíveis se alteram). Por este motivo, é possível que a pesquisa local não visite alguns k -vizinhos interessantes. Esta desvantagem é compensada pela diminuição do peso computacional do procedimento.

O procedimento de pesquisa local tenta melhorar o desempenho de um indivíduo através da pesquisa das suas k -vizinhanças, com $k = 1$ até T ²⁴. Sempre que a função de avaliação do indivíduo é melhorada, o seu genotipo é imediatamente alterado, e o processo de pesquisa local continua, considerando este “novo” indivíduo como ponto de partida (optou-se por seguir a estratégia de *best improvement*). Se a pesquisa local não encontra nenhum indivíduo melhor do que o actual entre os primeiros z k_k -vizinhos visitados, então pára de procurar na k_k -vizinhança e continua na k_{k+1} -vizinhança.

A pesquisa local é realizada no espaço dos fenótipos, mas se se encontra um indivíduo melhor do que o actual, o genotipo deste indivíduo é alterado. Isto corresponde a optar pelo paradigma Lamarckiano (Sinha & Goldberg, 2003). Existem autores que optam por considerar que a pesquisa local, interpretada como processo de aprendizagem, não deve influenciar a estrutura genética de um indivíduo (paradigma Baldwiniano). Neste caso, o indivíduo adquire como valor para a função de avaliação o melhor valor encontrado nas suas vizinhanças, mas o seu genotipo não é alterado, ou é alterado com uma probabilidade mínima. A vantagem deste procedimento é a de, possivelmente, garantir uma maior diversidade genética. Obriga, no entanto, a que sejam gastos mais recursos, nomeadamente porque é necessário guardar, para cada indivíduo, não só a sua representação genética, mas também a representação genética que é responsável pelo valor actual da função de avaliação.

O procedimento de pesquisa local é muito exigente em termos de tempos computacionais. Na maior parte dos casos é responsável por 95%, ou mais, do tempo total de execução do algoritmo. Isto torna obrigatório que sejam levados a cabo todos os esforços para melhorar o seu desempenho. Como defendido em Ishibuchi, Yoshida & Murata, 2002, é muito importante, mas na maioria dos casos muito difícil, encontrar um equilíbrio razoável entre a pesquisa levada a cabo pelo algoritmo genético e a pesquisa local na construção de algoritmos evolutivos híbridos.

Em cada iteração da pesquisa local, k genes do L -cromossoma e l genes, $0 \leq l \leq k$, do F -cromossoma são alterados, sendo que todos se referem ao mesmo serviço i . Isto significa que é necessário recalcular os custos fixos de abertura e reabertura e também recalcular a solução óptima para os k problemas de afectação²⁵. O cálculo das afectações óptimas contribui de forma preponderante para o aumento do tempo computacional, especialmente quando em presença

²⁴ A presente implementação do algoritmo considera k -vizinhanças de $k=1$ até T , sequencialmente. É possível que o procedimento possa ser melhorado se se considerarem outras alterações de estruturas de vizinhança, como na *Variable Neighbourhood Search Metabeuristic* (Glover & Kochenberger, 2003).

²⁵ Note-se que, na presença de serviços do tipo 3, há necessidade de recalcular a solução óptima para o problema de afectação global (que inclui os T períodos temporais).

de serviços com capacidades²⁶. Por este motivo, a pesquisa local foi alterada para que, antes de se calcularem as novas k soluções de afectação²⁷ (na presença de serviços com capacidades), se levasse a cabo uma análise de sensibilidade que tenta estimar se se está perante um vizinho melhor do que a solução actual ou não. Esta análise de sensibilidade é feita tendo como base a solução dual dos problemas de afectação já resolvidos.

Considere-se que a pesquisa local está a visitar um k -vizinho, e estude-se o efeito de abrir (fechar) o serviço i do período τ ao $\tau + k - 1$. Comece por assumir-se que não existem serviços de tipo 3 e que o problema tem apenas um nível de serviços. Neste caso, a pesquisa local começa por testar a admissibilidade do vizinho: é necessário testar as restrições de capacidade máxima e as restrições ao número máximo de elementos no caso do serviço i ir ser aberto, ou então as restrições de capacidade máxima no caso de ele ir ser fechado. Se se conclui pela não admissibilidade do vizinho, então a pesquisa local continua visitando outro vizinho. Caso contrário, realiza uma análise de sensibilidade a cada um dos k problemas de afectação.

No caso de existir apenas um nível de serviços, cada problema de afectação pode ser resolvido, recorrendo a um problema de transportes. Dado um período t considerem-se, como habitualmente, variáveis duais u_j^t associadas às restrições da origem j e variáveis v_d^t associadas às restrições do destino d . A alteração no valor da função objectivo, que resulta do aumento (diminuição) da procura do destino d em δ unidades, é dado por $\delta(u_{n+1}^t + v_d^t)$, desde que a solução dual se mantenha a única solução dual óptima para o problema com esta modificação nos dados (Murty, 1983)²⁸.

O procedimento de pesquisa local estuda o efeito de aumentar (diminuir) uma unidade na procura de um serviço quando este é considerado aberto (fechado). Se o serviço tem restrições de capacidade mínima então, para cada $\tau \leq t \leq \tau + k - 1$, o procedimento calcula $\Delta_t = u_{n+1}^t + v_{i+m}^t$. Se o serviço tem apenas restrições de capacidade máxima, então calcula $\Delta_t = u_{n+1}^t + v_i^t$. Depois, para cada k -vizinho, o procedimento calcula $\Delta_A = \sum_{t=\tau}^{\tau+k-1} \Delta_t$ e compara este valor com a alteração, por unidade de capacidade máxima do serviço, do custo fixo total de abertura e reabertura (Δ_F). Se $\Delta_F + \Delta_A$ (ou $\Delta_F - \Delta_A$) for maior do que zero, então a pesquisa local visita o vizinho com uma probabilidade p_v . Caso contrário, visita o vizinho com uma probabilidade p_w , tal que $p_w \gg p_v$.

No caso de existir pelo menos um serviço do tipo 3, o valor de Δ_A é

²⁶ Se todos os serviços são não capacitados, então é muito simples e imediato encontrar as novas afectações óptimas: se o serviço é fechado em t , então todos os clientes afectos a este serviço serão reafectos ao serviço aberto mais barato, sendo que mais nenhuma afectação é alterada. Se o serviço i é aberto em t , todos os clientes j , tais que $c_{ij}^t < c_{i'j}^t$ (sendo i' o serviço a que j está afecto na solução actual), será afecto ao serviço i .

²⁷ Ou a solução óptima para o problema de afectação total, se for esse o caso.

²⁸ Note-se que o aumento (diminuição) da procura num destino terá de ser acompanhado por igual aumento (diminuição) na oferta da origem fictícia.

calculado considerando a soma dos valores das variáveis duais associados às restrições de capacidade máxima ou capacidade máxima decrescente de cada um dos serviços.

Em comparação com o procedimento de pesquisa local em que não se recorre à análise de sensibilidade, o procedimento dá, em praticamente todos os casos²⁹, os mesmos valores para a função objectivo, verificando-se uma significativa redução no tempo total de computação. Mesmo assim, a pesquisa local continua a ser o procedimento responsável pela maior parte deste tempo.

É interessante notar que é muito mais difícil aferir as alterações que são provocadas na função objectivo resultantes do fecho ou abertura de um dado serviço num dado período t , quando comparado com o caso estático. No problema de localização estático, sabe-se que a abertura de um serviço aumenta o custo fixo e diminui os custos de afectação. O fecho de um serviço tem o efeito oposto. No problema de localização dinâmico com as características do apresentado, e particularmente quando se consideram serviços com capacidades mínimas maiores do que zero, a abertura ou fecho de um serviço durante o período t pode aumentar ou diminuir os custos fixos e pode aumentar ou diminuir os custos de afectação. Tal observação inviabiliza a utilização dos métodos tradicionais de se fixarem *a priori* os valores de algumas variáveis de localização iguais a zero ou um, através dos chamados testes de redução (consultar, por exemplo, Bornstein & Azlan, 1998; F. Saldanha-da-Gama & Captivo, 2002). Estes testes de redução baseiam-se numa propriedade do problema de transportes que se perde quando na presença de capacidades mínimas maiores do que zero. Considere-se um conjunto O_1 de serviços abertos (destinos) e defina-se $W(O_1)$ como sendo o valor da função objectivo do respectivo problema de afectação (como em Bornstein e Azlan, 1998). Considere-se ainda um serviço $i \notin O_1$. Se o serviço i é incluído no conjunto O_1 , então: $W(O_1) - W(O_1 \cup \{i\}) \geq 0$ e $W(O_1) - W(O_1 \cup \{i\}) \leq W(O_2) - W(O_2 \cup \{i\})$, $\forall O_2 \subseteq O_1$ ³⁰. É fácil demonstrar que se existem serviços com capacidades mínimas maiores do que zero, nenhuma destas inequações se pode garantir que seja satisfeita. A possibilidade de se estenderem os testes de redução e heurísticas *ADD* e *DROP* por forma a que seja possível fixar o valor de algumas variáveis de localização estará limitada a problemas com serviços sem capacidades ou na presença de restrições de capacidades máximas. Esta possibilidade não foi estudada.

Algoritmo 4-4

PROCEDIMENTO DE PESQUISA LOCAL

x – solução inicial; $f(x)$ – *fitness* do indivíduo x .

1. $k \leftarrow 1$.
2. Se $k > T$ então, parar; se não, $count \leftarrow 0$ e ir para 3.
3. $flag(i) \leftarrow false, \forall i$.

²⁹ Com base nos resultados preliminares obtidos enquanto se desenvolvia o algoritmo, e que não têm um carácter suficientemente sistemático, nem exaustivo para que esta afirmação possa ser generalizada. Apenas se refere a tendência notada.

³⁰ Esta propriedade é chamada de supermodularidade.

4. Se $flag(i) = true$, $\forall i$ ou $count > z$, então $k \leftarrow k + 1$ e ir para 2. Caso contrário, escolher aleatoriamente i tal que $flag(i) = false$.
 5. $flag(i) \leftarrow true$; $t \leftarrow 1$;
 6. Se $t > T - k + 1$ ou $count > z$, então ir para 4; se não, ir para 7.
 7. Se o serviço i está em funcionamento nos períodos t ao $t + k - 1$, então estudar o k -vizinho x' obtido a partir de x pela remoção dos períodos de funcionamento t a $t + k - 1$ e ir para 9. Caso contrário, ir para 8.
 8. Se o serviço i não está em funcionamento de t a $t + k - 1$, então estudar o k -vizinho x' obtido a partir de x pela introdução dos períodos de funcionamento t a $t + k - 1$ e ir para 9. Caso contrário, $t \leftarrow t + 1$ e ir para 6.
 9. Se $f(x) > f(x')$, então $x \leftarrow x'$ e $count \leftarrow 0$; se não, $count \leftarrow count + 1$. $t \leftarrow t + 1$ e ir para 6.
-

Em vez de se resolverem todos os k problemas de transporte (ou o problema de afectação total) para cada um dos k -vizinhos visitados, é também possível considerar o cálculo de valores aproximados, nomeadamente recorrendo ao cálculo de limites inferiores e superiores para os valores óptimos das funções objectivo. Quando se abre um novo serviço (que não seja de tipo 3), podem calcular-se os fluxos que irão ser redireccionados para este serviço através da resolução de um problema de saco de mochila modificado³¹. O valor óptimo da função objectivo deste problema fornece um limite superior válido para o problema de transportes correspondente. Se um serviço i não está em funcionamento durante t , então é possível utilizar uma relaxação lagrangeana para calcular um limite inferior para o valor óptimo da função objectivo do problema de afectação correspondente. Esta relaxação lagrangeana pode mesmo ser usada na presença de serviços de tipo 3. Os valores óptimos para as variáveis de afectação podiam apenas ser calculados no fim do procedimento de pesquisa local, e apenas para a melhor solução calculada até ao momento. Estas possibilidades não foram testadas, mas crê-se que possam reduzir de forma significativa o tempo computacional utilizado pela pesquisa local à custa, obviamente, de um possível decréscimo na qualidade do indivíduo devolvido por este procedimento.

4.3.5. População

A população inicial é constituída por indivíduos aleatoriamente gerados que são depois modificados pelos Algoritmo 4-2, Algoritmo 4-3 e Algoritmo 4-4. Não foi experimentada a inicialização da população através de indivíduos escolhidos e que garantissem, à partida, uma diversidade genética considerável. Espera-se que a inicialização da população afecte apenas as primeiras gerações, e que o processo evolutivo seja suficientemente rico para criar a diversidade genética necessária ao bom desempenho do algoritmo. Davis, 1991, refere alguns trabalhos que sugerem que a inicialização da população feita utilizando soluções de boa qualidade (obtidas, por exemplo, pela aplicação de uma outra heurística), pode ajudar o algoritmo genético a encontrar melhores soluções em menos tempo do que se a inicialização for feita de forma puramente aleatória. No entanto, chama também a atenção para o facto de tal opção levar à

³¹ Jacobsen, 1983, utiliza um procedimento semelhante na resolução do problema estático de localização com capacidades.

convergência prematura do algoritmo. Seria, sem dúvida, interessante comparar o desempenho deste algoritmo, considerando a inicialização aleatória da população ou inicializá-la com indivíduos calculados usando, por exemplo, algumas das heurísticas primais-duais descritas no capítulo 3.

O algoritmo desenvolvido é muito exigente em relação a tempos computacionais, principalmente devido ao procedimento de pesquisa local. Trabalhar com populações muito grandes, como acontece na maior parte dos algoritmos genéticos³², poderia pôr em risco a possibilidade de se obterem desempenhos razoáveis. Por esse motivo optou-se por trabalhar com populações pequenas. Quando se toma este caminho, o algoritmo vai simular um maior número de gerações por unidade de tempo. No entanto, existe um risco real de convergência prematura para uma solução de má qualidade. Existem resultados empíricos que mostram ser possível obter bons resultados com populações construídas com 30 ou menos elementos (C. R. Reeves, 1993b; Sörensen & Sevaux, 2006).

Uma das maneiras possíveis de combater esta desvantagem é considerar uma população pequena mas estruturada como é descrito em Cortinhal & Captivo, 2003a. Nesse trabalho, a população está estruturada como uma árvore treanária, com relações hierárquicas estabelecidas entre indivíduos através do seu valor de *fitness*. Esta abordagem foi experimentada mas deu, para os problemas estudados, maus resultados, pelo que foi abandonada.

Outro modo de tentar evitar a convergência prematura do método é pela alteração dinâmica e *on-line* do número total de indivíduos na população actual. O número de indivíduos é aumentado sempre que, num número *nimp* pré-definido de gerações, não existe melhoria no melhor valor conhecido para a função objectivo. Sempre que a população é aumentada, os novos indivíduos que se introduzem são aleatoriamente gerados. Grefenstette, 1992, apresenta um algoritmo em que se considera a substituição de parte da população por indivíduos gerados aleatoriamente. O autor interpreta esta técnica como representando uma migração de indivíduos novos provenientes de uma outra população com um elevado rácio de mutação, tendo como efeito principal manter um nível contínuo de capacidade de pesquisa. No caso do presente algoritmo, não é a população que é substituída, mas antes aumentada. O efeito que se pretende é o mesmo: aumentar a capacidade de pesquisa da população, tentando encontrar outras regiões de soluções interessantes, quando o algoritmo parece ter convergido para uma determinada área que não há maneira de saber se é a melhor ou não. Também Arabas, Michalewicz & Mulawka, 1994, consideram uma população em que o número de indivíduos não é constante, mas as alterações no número de indivíduos na população deve-se ao envelhecimento dos cromossomas: a cada indivíduo, quando nasce, é atribuído o número de gerações em que permanecerá na população.

A população é inicializada com *npop* indivíduos. O valor de *npop* é calculado como descrito em C. R. Reeves, 1993b: é igual ao menor valor, tal que

³² Note-se que a questão de qual o número óptimo de indivíduos de uma população não está respondida, existindo trabalhos que se debruçam unicamente sobre esta matéria (por exemplo, Alander, 1992 e Odetayo, 1993).

$$\left(1 - \left(\frac{1}{2}\right)^{n_{pop}-1}\right)^l \geq 0.99^{33}, \text{ em que } l \text{ representa o número de genes de cada}$$

indivíduo. Como cada indivíduo tem dois cromossomas, cada um com mT genes, então l deveria ser igual a $2mT$. No entanto, como o F -cromossoma tem muito poucos genes determinantes, optou-se por considerar l igual a mT no cálculo do valor inicial de n_{pop} , e igual a $2mT$ no cálculo do valor máximo que n_{pop} pode tomar.

Em cada geração, toda a população é substituída pelos filhos gerados, à excepção de um indivíduo, que passa automaticamente para a geração seguinte: o indivíduo com o melhor valor de *fitness*. O algoritmo integra, assim, um tipo de elitismo que garante que, em cada geração, só pode haver melhorias no valor da função objectivo do problema. É também permitida a existência de indivíduos duplicados na população. Escolheu-se a substituição geracional em vez da substituição *steady-state* fundamentalmente por razões que se prendem com o tempo computacional necessário, uma vez que a segunda opção obriga a que sejam feitas muito mais comparações entre indivíduos.

4.3.6. Valores para os Parâmetros

Como se pode observar da descrição feita do funcionamento do algoritmo até ao momento, existe um conjunto de parâmetros cujos valores têm de ser fixados e que podem influenciar de forma determinante o comportamento do algoritmo. É muitas vezes difícil compreender a influência de um único parâmetro no desempenho do algoritmo, e ainda mais complicado compreender a interacção existente entre parâmetros. Esta é uma das principais desvantagens da utilização destes métodos de resolução de problemas: na maioria das situações, os valores dos parâmetros são encontrados com base em experiências que não conseguem testar todas as combinações possíveis.

Existem duas formas de estabelecer os valores para os parâmetros (Á. E. Eiben *et al.*, 1999): controlo dinâmico de parâmetros e pré-fixação dos seus valores. O controlo dinâmico dos parâmetros permite que os valores se alterem durante a execução do algoritmo (ver, por exemplo, o trabalho de Bäck, A.E.Eiben & Vaart, 2000). No algoritmo desenvolvido todos os valores para os parâmetros são fixados antes da execução do algoritmo, com a excepção, já referida, do número de indivíduos que compõem a população.

Não foi estudada de forma sistemática a influência dos parâmetros no algoritmo. Na Tabela 4.2 listam-se todos os parâmetros utilizados e descreve-se a influência que se pensa que cada um terá no comportamento do algoritmo (com base nos resultados e nas experiências feitas até ao momento).

Genetic Algorithms are not simple systems; they are complicated and non-linear, and their behaviour is difficult to formalize and predict. Modifying the parameter settings or the components of a genetic algorithm may greatly impact the algorithm's performance. (...) In fact, the time and resources required to find optimal parameter settings for a

³³ O valor 0.99 representa a probabilidade de, pelo menos um alelo, estar presente em cada locus na população inicial.

problem domain are often orders of magnitude greater than the time one plans to spend solving problems in the domain.

Davis, 1991

4.3.7. Perspectiva Global

Todos os procedimentos descritos são agora considerados globalmente para se descrever o algoritmo genético. O Algoritmo 4-5 descreve a simulação de uma geração e o Algoritmo 4-6 descreve o funcionamento de todo o algoritmo memético.

Algoritmo 4-5

SIMULAÇÃO DE UMA GERAÇÃO

x_{best} – representa o melhor indivíduo da geração anterior; $flag(x)$ – é igual a true se x já foi sujeito ao procedimento de pesquisa local, false em caso contrário; $P_{current}$ – população actual; $f(x_j)$ – valor de *fitness* de x .

1. $x_1 \leftarrow x_{best}$; $\mathcal{X}_{best} \leftarrow x_1$; $j \leftarrow 2$; $Newpop \leftarrow \{x_1\}$.
2. Se $j > npop$ então $P_{current} \leftarrow Newpop$, se não ir para 3.
3. Seleccionar pais x_A e x_B , usando a selecção por torneio binário.
4. Aplicar o crossover por forma a gerar dois filhos: x_j and x_{j+1} . $flag(x_j) \leftarrow false$; $flag(x_{j+1}) \leftarrow false$.
5. Aplicar o operador de mutação a x_j .
6. Se $x_j = x_A$, então $flag(x_j) \leftarrow flag(x_A)$; se $x_j = x_B$, então $flag(x_j) \leftarrow flag(x_B)$; ³⁴
7. Calcular o valor de *fitness* de $x_j : f(x_j)$. Se $f(x_j) = +\infty$, então executar o procedimento reparador (Algoritmo 4-2) a x_j . Se $f(x_j) < f(\mathcal{X}_{best})$, então $x_{best} \leftarrow x_j$.
8. Executar o procedimento de alterar reaberturas (Algoritmo 4-3) a x_j .
9. Se $flag(x_j) = false$, então executar o procedimento de pesquisa local (Algoritmo 4-4).
10. Se $(j+1) \leq npop$, então repetir os passos 5 a 9 com o filho x_{j+1} .
11. $Newpop \leftarrow Newpop \cup \{x_j\}$. Se $(j+1) \leq npop$, então $Newpop \leftarrow Newpop \cup \{x_{j+1}\}$. $j \leftarrow j + 2$. Ir para 2.

Algoritmo 4-6

ALGORITMO MEMÉTICO

$P_{current}$ – população actual; x_{best} – melhor indivíduo.

1. Inicializar $P_{current}$.
2. Inicializar x_{best} , $best \leftarrow f(x_{best})$, $ngen \leftarrow 1$, $count \leftarrow 0$.
3. Se $ngen > nger$ ou $count > nimp$, então parar, se não ir para 4.
4. $ngen \leftarrow ngen + 1$. Executar a geração.
5. Se $f(x_{best}) \geq best$, então $count \leftarrow count + 1$; se não, $count \leftarrow 0$.
6. Se $count < nimp$ então $ngen \leftarrow ngen + 1$, ir para 3. Se $\min\{ \lceil npop(1 + \beta) \rceil, Nmaxpop \} > npop$, então $npop \leftarrow \min\{ \lceil npop(1 + \beta) \rceil, Nmaxpop \}$, inicializar aleatoriamente os novos indivíduos e $count \leftarrow 0$. Ir para 3.

³⁴ x_j é considerado igual a x_A se todos os genes do L -cromossoma são iguais.

Parâmetro	Descrição	Influência no algoritmo e valores recomendados ³⁵
P_{bt}	Probabilidade de que seja escolhido o indivíduo com o melhor desempenho no processo de selecção por torneio binário	Quanto maior a probabilidade, mais difícil se torna para os menos aptos passarem a sua informação genética para a próxima geração. Assim, pode ser usado para influenciar a diversidade genética da população. Se fosse controlado dinamicamente, este parâmetro poderia aumentar com o número de gerações, para assegurar diversidade no início do método e convergência no fim. Também se poderia diminuir, sempre que se suspeitasse de convergência prematura. Neste algoritmo, o valor é fixado em 0.9.
P_{μ}	Probabilidade de que um gene seja alterado pelo operador mutação	Este parâmetro pode influenciar a diversidade da população. Se controlado <i>on-line</i> , pode ser diminuído com o aumento do número de gerações, ou aumentado quando o valor da função objectivo não melhora num dado número de gerações. Neste algoritmo, este valor é fixo e igual a 0.002.
P_b	Probabilidade de que seja executado o procedimento de pesquisa local para um dado indivíduo	Este parâmetro influencia fortemente quer o tempo computacional, quer a qualidade da melhor solução encontrada. Com valores perto de 1, o algoritmo converge mais depressa (num menor número de gerações) e é possível obter soluções de boa qualidade. É difícil aferir com certeza como é que este parâmetro influencia o tempo computacional: valores mais pequenos fazem com que cada geração seja simulada em menos tempo mas a convergência para uma boa solução é mais demorada, o que pode levar a um aumento do tempo computacional total. É aconselhável que p_b seja igual a 1 pelo menos na última geração. No algoritmo, este valor é sempre igual a 1.
Z	Número máximo de k -vizinhos visitados, sem que se melhore o desempenho do indivíduo	Este parâmetro influencia o algoritmo de um modo semelhante ao parâmetro anterior. Deveria ter em conta o número total de k -vizinhos de um dado indivíduo, valor esse que é difícil de calcular. No algoritmo, este valor é igual a 10000.
P_v	Probabilidade de visitar um vizinho que se espera vir a melhorar o valor da função objectivo	Este parâmetro influencia o algoritmo de modo semelhante aos dois anteriores. Esta probabilidade deverá estar sempre próxima de 1. No algoritmo, é fixa a 1.
P_{nv}	Probabilidade de visitar um vizinho que não se espera que venha a melhorar a função objectivo	Esta probabilidade influencia o tempo computacional e também a qualidade da solução final. Deve tentar obter-se um bom compromisso entre estes dois factores. Recomenda-se que este parâmetro seja escolhido no intervalo entre 0 e 0.1.
N_{pop}	Número de indivíduos na população actual	Este parâmetro influencia o tempo computacional e a qualidade da solução final: populações com mais indivíduos fazem com que cada geração precise de mais tempo para ser simulada, mas são geneticamente mais poderosas. Populações pequenas correm o risco de sub-cobrirem o espaço das soluções. Este é o único parâmetro do algoritmo que se altera de forma dinâmica, sendo aumentado sempre que se simulam <i>nimp</i> gerações sem que haja melhorias no valor da função objectivo.

Tabela 4.2 – Lista de parâmetros do algoritmo

³⁵ Estes “valores recomendados” têm apenas por base as experiências computacionais feitas até ao momento.

Parâmetro	Descrição	Influência no algoritmo e valores recomendados
$Nmaxpop$	Número máximo de indivíduos que podem pertencer à população	Este parâmetro influencia o tempo total de execução e pode influenciar a qualidade da melhor solução gerada. Os valores utilizados são calculados como indicado em IV.3.5.
β	Porcentagem de aumento no número de indivíduos da população	Este parâmetro, juntamente com o $Nmaxpop$, controla o número de vezes que o número de indivíduos da população pode aumentar. Não é fácil estimar de que modo é que irão influenciar o tempo total de execução e a qualidade da solução calculada: maiores valores levarão a um menor número de gerações, mas com tempos de execução por geração mais elevados. No algoritmo, este valor foi considerado igual a 25%.
$Nger$	Número máximo de gerações	É um parâmetro que pode ser usado como condição de terminação do algoritmo, sendo completamente cega em relação ao desempenho do algoritmo. Pode ser responsável pela terminação prematura do algoritmo, mas também pela simulação desnecessária de gerações inconsequentes. No algoritmo, este parâmetro não é importante, porque são consideradas outras condições de terminação.
$nimp$	Número máximo de gerações sem que se melhore o valor da função objectivo	Este parâmetro é utilizado como medida da convergência do algoritmo. O número de indivíduos na população é aumentado sempre que a função objectivo não melhora durante $nimp$ gerações, de forma a aumentar a diversidade genética e impedir que o algoritmo fique preso em óptimos locais. Se a população actual tiver $Nmaxpop$ indivíduos, então o algoritmo termina. Este número é fixado em 5.

Tabela 4.2 – Lista de parâmetros do algoritmo (cont)

4.3.8. Restrições Adicionais

Existe um número de restrições adicionais que decorrem directamente da definição do problema de localização: estabelecer um número máximo de serviços abertos em cada período, estabelecer restrições de orçamento, garantir que um determinado serviço está aberto ou fechado num dado período temporal, etc. Este tipo de restrições são difíceis de lidar utilizando as heurísticas primais-duais: as restrições adicionais alteram o problema primal, logo alteram o problema dual e torna-se necessário modificar todos os procedimentos desenvolvidos ou integrar a heurística nalgum mecanismo de pesquisa em árvore.

Existem muitos modos diferentes de lidar com restrições no contexto de algoritmos genéticos. Em Michalewicz, 1995, encontram-se referências aos métodos mais utilizados para lidar com restrições. A. E. Eiben, 2001, apresenta um tutorial acerca de como lidar com restrições no contexto de algoritmos evolutivos. Distingue duas situações: a situação em que as restrições são tratadas directamente e indirectamente. No primeiro caso, os algoritmos são alterados por forma a que se obrigue à satisfação das restrições, significando que se têm de criar e manter apenas indivíduos admissíveis. Na segunda situação, o problema de satisfação das restrições é transformado, incorporando-o no cálculo

do desempenho de cada indivíduo. Coello Coello, 1999c, 2000a, 2000c; Coello Coello & Montes, 2002, introduzem o conceito de não-dominância para lidar com problemas com restrições através da resolução de problemas multi-objectivo. O mesmo autor apresenta uma recolha dos métodos mais utilizados para lidar com restrições em Coello Coello, 2002, sendo que considera como método mais utilizado pela comunidade a utilização de funções penalidade. Nesta técnica, distingue dois tipos: métodos exteriores e interiores. Nos primeiros parte-se de uma solução não admissível e tenta atingir-se a admissibilidade. Nos segundos, a penalidade é calculada tendo como base a distância da solução não admissível à região admissível. Coello Coello & Cortés, 2001, utilizam um algoritmo baseado no comportamento do sistema imunitário para lidarem com restrições dentro dos algoritmos genéticos e Coello Coello & Becerra, 2002, propõem a utilização de um “algoritmo cultural”. Neste algoritmo considera-se que um indivíduo é fruto da sua carga genética mas também do contexto cultural em que se insere. Este contexto cultural é chamado de *belief space*, e é aqui que se insere informação acerca da admissibilidade dos indivíduos. Barbosa, 1999, introduz multiplicadores de Lagrange associados às restrições e desenvolve um algoritmo evolutivo que trabalha com duas populações.

No algoritmo descrito, foram utilizadas duas formas distintas para lidar com as restrições originais do problema: através de uma representação genética adequada e através do desenvolvimento do Algoritmo 4-2. Existe um tipo de restrições adicionais que são tratadas de forma muito eficiente pelo presente algoritmo: fixar alguns serviços abertos ou fechados durante alguns períodos de tempo. Todas as outras restrições adicionais são tratadas, considerando uma função de penalidade.

4.3.8.1. Fixar Serviços Abertos ou Fechados

Considere-se que se pretendem fixar abertos ou fechados alguns serviços durante alguns períodos de tempo. Este tipo de preferências podem ser introduzidas nos modelos através das restrições (4-7) e (4-8)³⁶:

1. Fixar um serviço i aberto durante o período t

$$\sum_{\tau=1}^I \sum_{\xi=t}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) = 1 \quad (4-7)$$

2. Fixar um serviço fechado durante o período t

$$\sum_{\tau=1}^I \sum_{\xi=t}^T (a_{i\tau}^{\xi} + r_{i\tau}^{\xi}) = 0 \quad (4-8)$$

Se se pretender ter em conta restrições deste tipo e resolver o problema utilizando alguma das heurísticas primais-duais desenvolvidas, seria necessário fixar a zero ou a um todas as variáveis $a_{i\tau}^{\xi}$ e $r_{i\tau}^{\xi}$, tais que $\tau \leq t \leq \xi$. A fixação do

³⁶ Restrições semelhantes podem ser usadas para fixar o número possível, em cada serviço do tipo 4, de elementos de cada dimensão.

valor destas variáveis pode ser feita como descrito em 3.2.1.11.

Graças à representação escolhida, estas restrições adicionais são tratadas de forma muito eficiente pelo algoritmo: se se pretende que o serviço i esteja aberto (fechado) durante o período t , então o gene na posição $(t-1)m+i$ no L -cromossoma terá de ser igual a um (zero), para todos os indivíduos da população. Isto é conseguido de forma simples: todos os indivíduos são obrigados a passar num filtro que altera o seu genotipo de acordo com as restrições (4-7) e (4-8). Este filtro deverá ser utilizado na inicialização da população, no operador de mutação (não será possível alterar um gene que tenha sido fixado pelo filtro) e no procedimento de pesquisa local (em que se visitam apenas os vizinhos que satisfazem as restrições (4-7) e (4-8)). Os testes computacionais preliminares realizados mostram que a introdução destes tipos de restrições não altera em nada o desempenho do algoritmo, nem em termos de tempos computacionais, nem em termos de qualidade da solução.

De forma semelhante, se se pretender estabelecer um período mínimo de funcionamento para um dado serviço i , tal que se i é aberto em t , então tem de estar em funcionamento pelo menos até $t+b$, aplica-se um filtro que obriga a que os genes do L -cromossoma em posições $(t_1-1)m+i$ com t_1 igual a $t+1$ até $t+b$ sejam iguais a um.

4.3.8.2. Restrições Lineares Adicionais

Considera-se seguidamente a possibilidade de se inserirem uma ou mais restrições adicionais r do tipo representado por (4-9).

$$\begin{aligned} & \sum_t \sum_{i \in I_4} \sum_j \sum_s c^{rt}_{ijs} x^t_{ijs} + \sum_t \sum_{i \in I_4} \sum_s \sum_{\xi=t}^T FA^{r\xi}_{ist} a^{\xi}_{ist} + \sum_t \sum_{i \in I_4} \sum_s \sum_{\xi=t}^T FR^{r\xi}_{ist} r^{\xi}_{ist} + \\ & \sum_t \sum_{i \in I / I_4} \sum_j c^{rt}_{ij} x^t_{ij} + \sum_t \sum_{i \in I / I_4} \sum_{\xi=t}^T FA^{r\xi}_{it} a^{\xi}_{it} + \sum_t \sum_{i \in I / I_4} \sum_{\xi=t}^T FR^{r\xi}_{it} r^{\xi}_{it} \geq B^r \leq \end{aligned} \quad (4-9)$$

Estas restrições são mais difíceis de tratar do que as referidas na secção anterior, pois levantam dois tipos diferentes de questões: a) como lidar com soluções que são admissíveis para o problema original, mas que não satisfazem as restrições adicionais (4-9); b) como resolver os problemas de afectação, quando $\exists r: c^{rt}_{ijs} > 0$ ou $c^{rt}_{ij} > 0$.

Em 4.3.2, aos indivíduos não admissíveis foram atribuídos valores de *fitness* iguais a $+\infty$: são considerados indivíduos piores do que qualquer outro na população. Esta opção pode ser extendida a indivíduos que violem, pelo menos, uma restrição adicional. No entanto pode também optar-se por penalizar estes indivíduos de uma outra forma: o seu valor de *fitness* é igual ao respectivo valor da função objectivo penalizado com: a) o valor exacto em que cada restrição adicional está a ser violada; b) uma percentagem fixa em relação ao valor da função objectivo; c) o indivíduo não é penalizado³⁷.

³⁷ Esta opção pode ser útil em fases iniciais do processo de resolução de problemas complexos, em que o decisor sabe que terá de ser considerada uma determinada restrição mas não quer que a

A representação que foi escolhida para codificar as soluções considera um conjunto único de valores óptimos para as variáveis de afectação associadas a cada indivíduo. Se o cálculo do valor óptimo das variáveis de afectação for feito exactamente do modo descrito em 4.2, então existem algumas situações em que o algoritmo não conseguirá identificar a solução óptima, mesmo se na presença de um indivíduo que codifica o valor óptimo para as variáveis de localização. Se se considerarem as restrições adicionais no cálculo do valor óptimo das variáveis de afectação, (note-se que o lado direito das restrições adicionais são alteradas considerando os valores das variáveis de localização fixos), então o problema de afectação torna-se muito mais difícil de resolver e terá de ser resolvido como um problema de programação linear, utilizando um *general solver* (mesmo sem existirem serviços de tipo 3). Outra maneira de lidar com este problema é através da resolução de vários problemas de afectação, em vez de apenas um único, considerando funções objectivo diferentes, tentando encontrar uma solução que melhor se adequa às restrições adicionais que foram incluídas. Isto pode ser feito, utilizando multiplicadores de lagrange associados às restrições adicionais, ajustando os seus valores através de um procedimento de *multiplier adjustment* (C. R. Reeves, 1993a) como indicado no Algoritmo 4-7. Neste algoritmo considera-se o caso de um único nível, sendo trivial a sua adaptação, para que possam ser considerados vários níveis.

Algoritmo 4-7

Considera-se que existem R restrições adicionais (4-9), todas do tipo \leq .³⁸ Considera-se cada restrição (4-9) representada por $A_r^1(l) + A_r^2(x) \leq B^r$, tal que l e x representam o vector de variáveis de localização e afectação, respectivamente. Os valores das variáveis de localização são fixos e conhecidos (dados pelo valor dos genes dos L e F -chromossomas); c_{ij}^t - custo de afectar o cliente j ao serviço i no período t ; c_{ijs}^t - custo de afectar o cliente i a um elemento de dimensão s localizado em i durante o período t ; T - número períodos temporais; $Nmaxtries$ - número máximo de iterações.

1. $r \leftarrow 1$.
 2. Para cada restrição adicional r calcular $A_r^1(l)$, $B^r \leftarrow B^r - A_r^1(l)$, $\mu_r \leftarrow 0$.
 3. $admin \leftarrow false$, $c_{ij}^t \leftarrow c_{ij}^t + \sum_r \mu_r c_{ij}^{rt}$, $\forall i \notin I_3, j, t$, $c_{ijs}^t \leftarrow c_{ijs}^t + \sum_r \mu_r c_{ijs}^{rt}$, $\forall i \in I_3, s, j, t$.
 4. Resolver T problemas de transporte, considerando os custos de afectação modificados c_{ij}^t e c_{ijs}^t . Calcular $A_r^2(x)$, $\forall r$.
 5. Se $A_r^2(x) \leq B^r$, $\forall r$, então $admin \leftarrow true$ e ir para 8. Caso contrário, ir para 6.
 6. Se $A_r^2(x) > B^r$, então $\delta_r \leftarrow \min \left\{ \min_{i,j,t} \left\{ \frac{c_{ij}^t}{c_{ij}^{rt}} \right\}, \min_{i,j,s,t} \left\{ \frac{c_{ijs}^t}{c_{ijs}^{rt}} \right\} \right\}$; se não, $\delta_r \leftarrow 0$, $\forall r$.
 7. $\mu_r \leftarrow \mu_r + \delta_r$, $\forall r$, $count \leftarrow count + 1$.
 8. Se $count > Nmaxtries$ ou $admin$, então parar; se não, ir para 3.
-
-

pesquisa inicial seja limitada para que possa conhecer melhor o problema. Também pode ser utilizada para se sinalizarem restrições "soft" que, segundo Deb, 2001, são aquelas que, até certo ponto, podem ser relaxadas para que uma determinada solução seja aceite.

³⁸ A alteração deste procedimento, para que se considerem restrições do tipo \geq , é trivial.

O algoritmo desenvolvido permite que se opte por lidar com as restrições adicionais de uma forma ou de outra. Os testes computacionais preliminares efectuados apontam uma tendência: a primeira alternativa leva a melhores soluções à custa de maiores tempos computacionais.

Em comparação com a situação em que não existem restrições adicionais, o desempenho do algoritmo é claramente pior, quer em termos da qualidade da solução quer em termos de tempos computacionais. Este facto é especialmente notado quando o lado direito das restrições adicionais é muito diferente do lado esquerdo das mesmas, quando se considera a solução óptima do problema sem restrições adicionais. É igualmente interessante notar que a deterioração do desempenho do algoritmo dá-se, apenas, quando os coeficientes relativos às variáveis de afectação nas restrições adicionais são diferentes de zero para, pelo menos, uma variável. Caso contrário, não se nota qualquer tipo de alteração no desempenho do algoritmo, pois o cálculo do valor óptimo das variáveis de afectação não é afectado.

4.3.9. Resultados Computacionais

Davis, 1991, define três maneiras de se observar o desempenho de um algoritmo genético: *on-line*, *off-line* e *best-so-far*. As medidas de desempenho *on-line* consideram a qualidade média de todos os indivíduos gerados, enquanto que a medida de desempenho *off-line* se relaciona com a capacidade do algoritmo genético convergir para uma solução óptima. Enquanto que esta última realça a importância de se chegar a uma boa (e possivelmente isolada) solução, a anterior considera a capacidade do algoritmo encontrar uma área do espaço de procura em que a qualidade média das soluções é, em geral, boa. Como se pretende comparar o desempenho deste algoritmo com o desempenho das heurísticas primais-duais, optou-se por considerar o desempenho *off-line*, pelo que se vão mostrar resultados que consideram a melhor solução calculada pelo algoritmo para cada problema resolvido. Os problemas resolvidos são os mesmos gerados anteriormente, para aferir da qualidade das heurísticas primais duais, mantendo-se todas as condições experimentais descritas. A execução da metaheurística é interrompida sempre que o tempo de execução excede os 200 000 segundos, ou existem problemas de insuficiência de memória. Em cada tabela mostram-se dois tempos computacionais (em segundos): o *tempo computacional 1* refere-se ao tempo total de execução da heurística, enquanto que o *tempo computacional 2* mostra o tempo necessário até que a heurística encontre aquela que vai ser a sua melhor solução. É também mostrada a qualidade da melhor solução calculada, em percentagem. A Tabela 4.3 mostra os resultados computacionais referentes aos problemas com um nível e sem capacidades. A tabela seguinte ilustra os resultados para problemas com capacidade máxima. O algoritmo não foi bem sucedido na resolução de problemas com capacidade máxima e mínima, e também na presença de restrições de capacidade máxima decrescente. Tal facto pode ser explicado pelo aumento da complexidade dos problemas de afectação que é necessário resolver, e que leva a necessidades acrescidas de memória e tempos computacionais várias vezes superiores aos piores tempos apresentados pelo Cplex.

T	m	n	Tempo Computacional 1			Tempo Computacional 2			Qualidade da Solução Primal		
			Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
5	5	25	1.97	2.19	2.42	0.06	0.16	0.27	0.00	0.00	0.00
5	5	50	2.58	3.51	4.99	0.11	0.22	0.32	0.00	0.00	0.00
5	5	100	6.42	7.28	8.90	0.27	0.38	0.60	0.00	0.00	0.00
5	5	200	30.54	35.09	38.72	2.36	4.50	8.02	0.00	0.00	0.00
5	5	500	73.93	88.31	105.67	5.27	9.85	16.15	0.00	0.00	0.00
5	5	1000	156.54	170.21	187.25	9.45	12.34	15.93	0.00	0.00	0.00
5	10	25	10.43	11.28	12.75	0.65	0.67	0.72	0.00	0.00	0.00
5	10	50	19.11	22.11	24.11	1.15	1.41	1.76	0.00	0.00	0.00
5	10	100	54.26	65.22	74.26	3.29	3.98	5.77	0.00	0.00	0.00
5	10	200	176.20	184.76	191.03	14.66	48.18	95.08	0.00	0.00	0.00
5	10	500	445.33	493.67	550.96	25.16	48.94	88.43	0.00	0.00	0.00
5	10	1000	868.43	967.53	1077.20	109.52	412.38	752.81	0.00	0.00	0.00
5	50	25	549.53	590.40	628.62	51.36	67.06	96.34	0.00	0.07	0.35
5	50	50	1476.29	1511.59	1625.30	140.66	184.89	323.51	0.00	0.00	0.00
5	50	100	3349.96	3717.04	4648.51	541.29	1099.40	2916.60	0.00	0.09	0.44
5	50	200	6266.78	7717.36	10506.80	602.76	3724.68	8368.01	0.00	0.05	0.13
5	50	500	15640.00	16910.26	19150.40	1886.25	6372.27	13573.00	0.00	0.07	0.18
5	50	1000	30507.40	33957.02	41909.90	4402.06	11978.96	31965.50	0.00	0.03	0.07
5	100	25	3190.13	3479.19	4213.17	436.55	1030.52	3018.65	0.00	0.42	2.09
5	100	50	6798.62	7737.80	8291.17	480.21	1559.78	2787.41	0.00	0.00	0.00
5	100	100	14430.60	17925.56	21248.00	5938.71	11200.10	15871.20	0.24	0.81	1.85
5	100	200	29868.90	33449.22	36853.00	13099.10	21422.60	26149.50	0.03	0.43	1.54
5	100	500	65257.20	76531.52	90610.50	4643.34	31624.03	64522.60	0.10	0.84	3.07
5	100	1000	130489.00	147134.00	159419.00	33148.90	58450.72	81241.00	1.63	1.96	2.08
20	5	25	55.53	95.12	115.29	5.05	15.77	44.59	0.00	0.04	0.19
20	5	50	207.67	236.96	270.79	16.36	46.21	107.16	0.00	0.00	0.00
20	5	100	569.09	605.05	679.43	58.50	216.80	316.53	0.00	0.62	1.27
20	5	200	988.27	1000.74	1022.83	89.75	189.47	361.96	0.00	1.10	3.29
20	5	500	1831.76	3589.93	4715.08	338.18	1284.81	3468.10	0.00	0.13	0.46
20	5	1000	7057.65	9014.24	11862.10	540.03	4276.58	9517.54	0.00	0.01	0.04
20	10	25	637.74	820.13	995.85	61.68	490.09	747.42	0.00	0.21	0.99
20	10	50	1516.28	1843.16	2055.36	131.88	835.54	1477.06	0.00	0.41	0.99
20	10	100	3736.58	4525.04	5134.88	950.37	2163.48	2787.53	0.00	0.70	1.02
20	10	200	8201.92	9470.29	12982.90	796.04	4311.96	9995.01	0.00	0.57	0.96
20	10	500	20951.20	23506.68	27142.10	6540.97	11443.03	19565.30	0.00	0.17	0.62
20	10	1000	27916.50	30114.88	34274.20	6923.47	11855.76	18185.50	0.00	0.02	0.10
20	50	25	18896.90	19609.86	20123.90	2339.39	2748.19	4310.17	2.59	9.32	13.79
20	50	50	38332.20	42472.66	50729.20	1159.91	7044.99	24930.20	1.68	8.76	17.45
20	50	100	79632.20	83202.28	87685.20	3939.10	6581.03	9650.74	6.51	7.79	9.04
50	5	25	133.47	196.94	249.64	31.58	82.83	150.56	0.00	3.09	10.97
50	5	50	473.18	533.88	599.95	80.14	207.49	334.50	1.24	2.73	7.74
50	5	100	693.27	931.16	1172.77	64.59	322.54	671.47	1.42	5.19	10.41
50	5	200	1944.58	2574.63	3148.82	182.46	494.59	1227.53	0.47	1.49	2.59
50	5	500	4511.36	6270.19	7869.94	565.57	2056.53	4442.92	0.35	0.73	1.19
50	5	1000	11497.10	13617.86	15494.40	1120.10	2530.93	5429.55	0.08	0.50	0.79
50	10	25	2553.05	2841.36	3179.53	231.29	1162.50	1906.02	3.55	6.62	14.06
50	10	50	4305.56	6983.32	9639.15	313.90	3948.66	7189.69	1.59	7.13	13.87
50	10	100	8907.92	11666.06	13474.60	1112.35	3925.25	7455.48	2.55	4.86	7.97
50	10	200	21999.50	27557.82	32771.50	2269.08	3857.47	6757.43	1.47	3.52	7.48
50	10	500	54181.90	65213.46	81165.90	3703.52	21151.65	68029.90	1.84	2.31	3.34
50	10	1000	128741.00	149904.80	160187.00	18280.60	55343.04	113574.00	0.55	0.86	1.36
50	50	25	209385.00	219252.20	227093.00	7863.46	22457.28	32790.20	17.07	38.09	58.51
Resumo dos Valores			1.97	20968.52	227093.00	0.06	6159.66	113574.00	0.00	2.15	58.51

Tabela 4.3 – Problemas de localização sem capacidades – tempos computacionais em segundos e qualidade da solução em percentagem

<i>T</i>	<i>m</i>	<i>n</i>	Tempo Computacional 1			Tempo Computacional 2			Qualidade da Solução Primal		
			<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	25	5	17.02	25.70	30.65	0.44	0.67	1.32	0.00	0.00	0.00
5	25	10	96.88	147.18	231.07	2.36	16.02	34.22	0.00	0.82	3.70
5	25	20	456.48	552.54	732.26	113.48	312.24	489.77	1.26	2.64	4.82
5	50	5	19.89	48.34	76.90	1.21	1.47	1.87	0.00	0.00	0.00
5	50	10	234.20	322.95	418.37	14.12	68.92	145.55	0.00	0.03	0.15
5	50	20	1168.21	1647.49	2226.29	283.74	901.58	1522.76	1.53	2.66	5.57
5	100	5	85.63	163.87	244.26	1.82	5.56	11.32	0.00	0.00	0.00
5	100	10	696.29	902.50	1152.51	104.03	365.97	818.50	0.00	0.55	2.75
5	100	20	3362.81	4072.52	5201.01	457.69	1973.87	3456.90	2.06	2.64	3.39
5	100	50	23266.00	25394.04	27177.00	13372.00	15109.78	16453.60	4.28	6.22	7.45
5	200	5	273.04	416.14	745.06	5.44	45.19	188.07	0.00	0.04	0.22
5	200	10	1550.60	1851.04	2321.59	109.46	452.88	982.40	0.32	1.72	3.35
5	200	20	8999.55	12640.71	16816.40	650.10	6124.64	12233.30	0.68	2.36	3.34
5	200	50	89324.20	121565.04	151836.00	23143.20	76304.26	113899.00	4.18	5.94	8.11
5	500	5	378.55	1276.95	1808.47	22.63	58.53	120.68	0.00	1.08	4.45
5	500	10	3492.43	9392.75	16903.10	646.30	3070.73	10366.10	0.00	0.69	3.28
5	500	20	33729.90	53807.74	85932.30	8660.82	25216.87	56123.10	2.32	4.10	6.70
10	25	5	64.81	95.03	113.09	1.43	3.66	9.89	0.00	0.00	0.00
10	25	10	482.80	926.56	1308.43	30.60	111.69	240.80	0.00	0.04	0.19
10	25	20	2329.45	3603.11	5925.53	153.57	2140.87	4568.15	0.60	1.07	1.78
10	50	5	124.47	292.07	375.80	4.23	57.17	151.70	0.00	0.00	0.00
10	50	10	1218.80	1671.13	2257.66	155.88	656.02	1257.96	0.00	0.14	0.42
10	50	20	5462.62	8257.58	11578.50	2496.75	5130.20	8444.25	0.71	1.80	2.76
10	100	5	366.90	672.95	972.62	9.61	62.22	189.39	0.00	0.00	0.00
10	100	10	3236.38	3802.32	4364.60	259.20	1484.71	2860.96	0.00	0.15	0.46
10	100	20	19215.00	31651.20	41673.40	6691.51	20313.10	33502.70	0.23	1.54	3.64
10	100	50	97938.40	150728.28	211820.00	10449.20	72466.14	150930.00	2.33	3.62	4.83
10	200	5	1646.77	2425.99	3706.31	25.32	635.56	1847.42	0.00	0.02	0.09
10	200	10	8682.57	13266.09	22946.70	1168.49	5868.51	15957.40	0.00	0.32	0.76
10	200	20	48940.20	76893.04	111592.00	8383.61	26761.60	75154.70	0.05	0.59	1.32
Resumo dos Valores			17.02	17617.10	211820.00	0.44	8857.35	150930.00	0.00	1.36	8.11

Tabela 4.4 – Problemas de localização com capacidade máxima – tempos computacionais em segundos e qualidade da solução em percentagem

A Tabela 4.5 mostra os resultados obtidos com problemas multi-nível sem capacidades. É interessante notar que o algoritmo encontra a solução óptima em todos os problemas que consegue resolver. Mais uma vez as dificuldades aparecem nos problemas de maior dimensão, devido à dificuldade de resolver, para cada indivíduo, T problemas de afectação de dimensão considerável. A Tabela 4.6 apresenta os resultados obtidos quando se introduzem as restrições de capacidade máxima. Novamente se detecta a impossibilidade do algoritmo resolver problemas com um número de caminhos admissíveis muito elevado. O algoritmo não consegue lidar com os problemas multi-nível sem conservação de fluxo em alguns nodos intermédios da rede.

<i>T</i>	<i>N</i>	<i>n</i>	<i>m_s</i>	Tempo Computacional 1			Tempo Computacional 2			Qualidade da Solução Primal		
				<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>	<i>Melhor</i>	<i>Média</i>	<i>Pior</i>
5	2	25	2	0.81	0.97	1.06	0.74	0.89	0.97	0.00	0.00	0.00
5	2	25	5	11.44	13.33	14.86	11.22	13.12	14.65	0.00	0.00	0.00
5	2	50	2	1.67	1.82	1.92	1.55	1.74	1.81	0.00	0.00	0.00
5	2	50	5	26.48	27.53	30.08	26.20	27.25	29.83	0.00	0.00	0.00
5	2	50	10	239.19	255.50	287.34	238.39	254.65	286.55	0.00	0.00	0.00
5	2	100	2	2.80	2.99	3.28	2.67	2.87	3.19	0.00	0.00	0.00
5	2	100	5	57.03	60.86	63.72	56.39	60.25	62.77	0.00	0.00	0.00
5	2	100	10	517.06	550.80	578.41	515.28	548.96	576.53	0.00	0.00	0.01
5	3	25	2	19.81	20.84	21.45	19.50	20.61	21.31	0.00	0.00	0.00
5	3	25	5	828.22	895.86	960.22	824.41	892.06	955.59	0.00	0.00	0.00
5	3	50	2	40.27	41.81	44.27	40.02	41.46	43.83	0.00	0.00	0.00
5	3	50	5	1485.34	1626.51	1763.20	1480.01	1621.45	1758.32	0.00	0.00	0.00
5	3	100	2	94.83	20225.81	100734.00	93.98	20048.89	99853.00	0.00	0.00	0.00
5	3	100	5	3416.88	3615.75	3983.45	3408.81	3607.07	3973.93	0.00	0.00	0.00
10	2	25	2	6.95	8.56	13.06	6.78	8.40	12.78	0.00	0.00	0.00
10	2	25	5	85.72	92.50	100.58	85.33	92.13	100.18	0.00	0.00	0.00
10	2	50	2	12.88	13.59	14.52	12.66	13.44	14.31	0.00	0.00	0.00
10	2	50	5	196.91	203.26	208.56	196.25	202.53	207.81	0.00	0.00	0.00
10	2	50	10	1696.47	1922.58	2587.84	1694.44	1919.47	2583.16	0.00	0.00	0.00
10	2	100	2	24.25	26.65	29.34	24.03	26.35	29.03	0.00	0.00	0.00
10	2	100	5	416.63	470.33	519.88	415.21	469.00	518.35	0.00	0.00	0.00
10	2	100	10	3242.59	3801.18	5532.42	3238.63	3796.40	5526.08	0.00	0.00	0.00
10	3	25	2	149.34	153.11	157.70	148.856	152.669	157.251	0.00	0.00	0.00
10	3	50	2	316.89	358.12	428.91	315.80	356.98	427.49	0.00	0.00	0.00
10	3	100	2	731.64	760.93	789.14	130.00	639.07	787.36	0.00	0.00	0.00
Resumo dos Valores				0.81	1406.05	100734.00	0.74	7493.36	157251.00	0.00	0.00	0.00

Tabela 4.5 – Problemas multi-nível sem capacidades – tempos computacionais em segundos e qualidade da solução em percentagem

<i>T</i>	<i>N</i>	<i>n</i>	<i>m_s</i>	Tempo Computacional 1			Tempo Computacional 2			Qualidade da Solução Primal		
				<i>melhor</i>	<i>média</i>	<i>pioor</i>	<i>melhor</i>	<i>média</i>	<i>pioor</i>	<i>melhor</i>	<i>média</i>	<i>pioor</i>
5	2	25	2	2.41	4.92	8.16	0.16	0.71	1.25	0.00	0.32	0.97
5	2	25	5	40.30	137.64	279.11	27.27	100.41	219.66	0.43	1.67	2.90
5	2	50	2	11.78	33.74	72.66	0.70	12.83	45.25	0.16	0.95	1.92
5	2	50	5	81.53	301.42	399.69	54.89	205.57	311.08	0.14	0.89	2.06
5	2	50	10	1149.55	1614.11	2233.72	23.00	829.53	1476.36	2.34	4.42	7.10
5	2	100	2	11.83	54.37	118.41	2.34	22.09	70.25	0.00	0.86	3.06
5	2	100	5	642.88	948.87	1283.83	423.80	628.80	885.50	0.42	1.98	6.94
5	2	100	10	4309.45	6139.57	9884.03	3077.89	4543.87	8120.81	3.67	6.08	8.27
5	3	25	2	26.42	73.88	124.11	9.88	42.19	88.13	0.00	2.11	4.45
5	3	25	5	2250.06	4553.54	7065.33	467.02	1653.68	3340.41	5.26	7.97	11.93
5	3	50	2	38.39	104.40	202.97	13.27	56.59	117.66	0.33	1.71	3.07
5	3	50	5	736.08	1237.78	1713.14	144.77	499.85	1171.56	3.22	8.10	15.88
5	3	100	2	103.11	182.39	237.38	43.28	125.30	179.91	1.70	2.38	3.61
10	2	25	2	28.22	80.05	228.56	18.47	54.94	153.83	0.61	1.60	3.33
10	2	25	5	160.91	870.17	1781.89	97.84	475.45	1127.02	1.81	4.51	9.75
10	2	50	2	39.99	146.29	471.91	18.45	103.43	335.27	0.28	1.43	2.08
10	2	50	5	1107.13	3434.84	7263.03	566.11	2568.32	6240.17	2.99	4.06	4.76
10	2	50	10	7267.00	11647.20	17786.90	3200.98	7119.02	12529.90	6.69	8.17	9.08
10	2	100	2	72.16	510.20	1342.69	35.33	259.99	836.97	0.44	1.14	1.80
10	2	100	5	7309.02	14402.60	34007.60	928.33	3835.08	6404.58	0.90	3.35	5.38
10	2	100	10	14614.00	22980.68	30347.60	2733.64	11334.35	18835.60	7.08	8.14	10.32
10	3	25	2	113.78	341.81	1148.45	49.95	218.13	769.94	1.40	5.04	11.28
10	3	50	2	294.72	781.67	1411.73	132.30	446.87	1006.56	2.51	4.50	5.84
10	3	100	2	485.88	2224.01	4360.48	186.44	1062.34	1920.53	2.76	5.38	6.80
Resumo dos Valores				2.41	3033.59	34007.6	0.16	1508.31	18835.60	0.00	3.61	15.88

Tabela 4.6 – Problemas de localização multi-nível com capacidade máxima – tempos computacionais em segundos e qualidade da solução em percentagem

4.4. Conclusões

A primeira conclusão que se pode retirar do trabalho desenvolvido é que o procedimento de pesquisa local é indispensável para que se consigam obter resultados razoáveis com o algoritmo genético. Outra conclusão que se pode retirar do que foi exposto até agora diz respeito à opção natural de codificar apenas variáveis de localização, e não variáveis de afectação. Como se pode observar, se numa primeira abordagem tal parece óbvio e não parece trazer quaisquer desvantagens (a não ser o tempo computacional necessário para o cálculo da solução óptima de um problema, mas que é também compensado pelo espaço que não é gasto na codificação destas variáveis), quando se pensa em introduzir restrições adicionais ou então considerar mais do que um objectivo o cálculo do valor das variáveis de afectação aparece como um problema real para o qual é necessário imaginar soluções. Apesar desta clara desvantagem, as vantagens que esta representação apresenta, nomeadamente em relação ao espaço necessário para as estruturas de dados e a maior facilidade em gerar soluções admissíveis, continuam a torná-la uma representação aliciente. É, no entanto, necessário continuar a desenvolver esforços com vista a encontrar novas codificações, ou optar por trabalhar simultaneamente com duas populações, como já fazem alguns autores.

Das experiências feitas pode concluir-se que:

- 1- Quando comparado com as heurísticas primais-duais, o algoritmo necessita de tempos computacionais muito maiores, conseguindo por vezes encontrar melhores soluções. No entanto, as heurísticas primais-duais continuam a constituir um melhor compromisso entre a qualidade da solução encontrada e o tempo dispendido. Esta conclusão está de acordo com o que afirma A. E. Eiben, 2002a:

"Common wisdom within Evolutionary Computing also states that Evolutionary Algorithms are seldomly superior on problems that have been subjects of intensive study and have sophisticated solution methods. The same common wisdom says, however, that it costs only little effort (in comparison with those sophisticated methods: much less effort) to develop an EA delivering an acceptable solution in an acceptable running time. An EA is the second best algorithm for any problem."

No entanto, o algoritmo não apresenta ainda tempos computacionais "aceitáveis", pois chegam mesmo a ser mais elevados do que os tempos computacionais gastos pelo Cplex para encontrar a solução óptima. Deve realçar-se uma diferença fundamental entre as duas abordagens: enquanto que se executam heurísticas primais-duais dedicadas a cada problema, o algoritmo memético é um único para a resolução de todos os diferentes tipos de problemas.³⁹

- 2- A consideração de serviços com capacidades torna o problema mais difícil, devido à necessidade de se resolverem problemas de afectação mais complexos e de aumentar o número de indivíduos não admissíveis.

³⁹ Relembre-se que, para além dos problemas que foram utilizados nos testes computacionais, o algoritmo foi programado para que possa lidar com quaisquer combinações de serviços dos tipos apresentados, em problemas mono ou multi-nível.

- 3- O algoritmo lida de maneira muito eficiente com restrições adicionais que fixem alguns serviços fechados ou abertos durante alguns períodos temporais.
- 4- O algoritmo tem algumas dificuldades em lidar com restrições adicionais lineares genéricas, especialmente devido à representação que foi escolhida. Se as restrições adicionais envolverem apenas variáveis de localização, então o desempenho do algoritmo não se deteriora na presença destas restrições.
- 5- O algoritmo é facilmente adaptado para que consiga lidar com mais do que uma função objectivo, como se irá ver na secção seguinte.
- 6- Ainda há muito a fazer para melhorar o desempenho do algoritmo memético desenvolvido:
 - a. O operador de *crossover* pode ser alterado para que sejam gerados filhos de diferentes maneiras. Na realidade, com o operador escolhido, e que é uma adaptação directa do *one-point crossover*, o que se verifica é que os filhos gerados terão sempre o mesmo padrão de serviços abertos e fechados, em cada período temporal, de um dos pais. A matriz que representa cada cromossoma é “partida” em linhas (em que cada linha representa um período temporal), e o operador de *crossover* apenas procede a trocas de linhas para geração dos dois filhos. Se se considerar uma adaptação do *3-point crossover*, então cada matriz será subdividida em quatro submatrizes, como ilustra a Figura 4.3. Neste caso, existem mais graus de liberdade na criação dos filhos (algumas possibilidades são ilustradas na Figura 4.4). Este operador de *crossover* pode permitir não só aumentar a diversidade da população, mas também ajudar a que o método convirja em menor tempo: de entre um conjunto de filhos possíveis, escolhem-se dois utilizando, por exemplo, um torneio em que os que têm melhor desempenho têm maior probabilidade de passarem à geração seguinte. Este tipo de *crossover* pode ainda ser utilizado na situação em que se pretende aumentar o número de indivíduos da população, em vez de se gerarem os indivíduos a introduzir na população de forma aleatória.
 - b. Toma-se imperativo melhorar o desempenho, em termos de tempo computacional, do procedimento de pesquisa local. Tal pode ser feito pela exploração de novas definições de vizinhança. No desenvolvimento deste algoritmo, e em particular do procedimento de pesquisa local, a influência do trabalho efectuado com as heurísticas primais-duais reflecte-se, por exemplo, na definição do mesmo tipo de vizinhanças. No entanto, como a solução é construída de modo completamente diferente, faz todo o sentido definir vizinhanças de outros modos diferentes.
 - c. Em vez de se optar por aumentar o número de indivíduos na população, quando esta não está a evoluir para soluções melhores, pode optar-se por manter o número de indivíduos na população constante, mas aplicar um esquema de gestão da população, como, por exemplo, é apresentado em Sörensen & Sevaux, 2006. Esta gestão da população consiste na utilização de uma função que determina se um filho vai ser ou não incluído na nova geração. Esta função pode ter em conta factores como a qualidade da solução e a sua contribuição para a diversidade da nova população.
 - d. O procedimento que tenta reparar soluções não admissíveis tem uma forte componente aleatória, por razões oportunamente explicadas. No entanto, e mais uma vez reflectindo o trabalho anteriormente realizado, fica a vontade de experimentar utilizar procedimentos, semelhantes aos procedimentos primais desenvolvidos ao longo do capítulo 3, para reparar estas soluções.

- e. Com base nos bons resultados computacionais obtidos com as heurísticas primais-duais, pode também equacionar-se a hipótese de interligação entre estas heurísticas e o algoritmo memético, aproveitando os pontos mais fortes de cada uma das abordagens. Assim, a utilização das heurísticas primais-duais na inicialização das populações, considerando, por exemplo, a perturbação de alguns dos dados do problema para que se possa encontrar um conjunto de soluções com características diferentes, poderá diminuir a dependência do algoritmo do seu procedimento de pesquisa local.

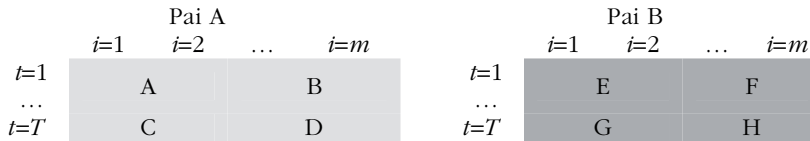


Figura 4.3 – Divisão possível da matriz que representa um cromossoma

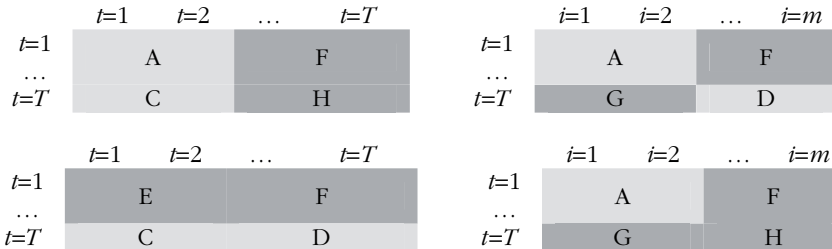


Figura 4.4 – Exemplo de filhos obtidos a partir de um operador de *crossover* com três pontos

4.5. Problemas Multi-objectivo

The notion of optimal solution is extremely unstable and fragile. The notion of optimum solution is very weak, probably the weakest ever produced in Operations Research. Change the objective function and you have another optimal solution.

Brans, 2004

Desde o trabalho pioneiro de Schaffer, 1985, responsável pelo desenvolvimento do algoritmo VEGA, muito tem sido feito e dito acerca da utilização de algoritmos genéticos (ou algoritmos evolutivos, em geral) num cenário em que se considera mais do que um objectivo. O facto de existir uma população de indivíduos motiva de imediato a possibilidade de aproximar simultaneamente várias soluções não dominadas. Jones *et al.*, 2002, apresentam um resultado que indica que cerca de 70% dos artigos que descrevem o uso de meta-heurísticas em problemas multi-objectivo são sobre algoritmos genéticos. Segundo os autores, o sucesso das aplicações destas técnicas prende-se com o facto de serem ferramentas ideais para gerarem representações do conjunto de

soluções eficientes. Alguns dos mais citados algoritmos evolutivos multi-objectivo são o MOGA (Fonseca e Fleming, 1993), NSGA (Srinivas & Deb, 1994), SPEA (Zitzler & Thiele, 1999). No algoritmo MOGA, o desempenho de cada indivíduo é calculado tendo como base o número de indivíduos que o dominam, na população. Utilizam, também, o conceito de *sharing* no espaço dos objectivos. O algoritmo NSGA utiliza, também, o conceito de não-dominância para determinar o desempenho de cada indivíduo: primeiro, são seleccionados os indivíduos não-dominados na população actual, aos quais é atribuído um valor de *fitness* muito elevado. Tal como no algoritmo anterior, o conceito de *sharing* é aplicado, por forma a que o valor de *fitness* de um indivíduo é diminuído, proporcionalmente ao número de indivíduos que lhe estão próximos. Esta primeira frente de indivíduos não-dominados é então ignorada, e encontram-se os próximos indivíduos não-dominados, aos quais é também atribuído um valor de *fitness*, forçosamente pior do que aos da primeira frente, e assim sucessivamente. No algoritmo SPEA todas as soluções não-dominadas são armazenadas numa população exterior, e podem participar na geração da população filha. O desempenho de cada indivíduo é calculado tendo como base um valor real, pertencente ao intervalo $[0,1]$, que é proporcional ao número de indivíduos dominados por ele na população actual. Também são calculados valores de *fitness* para os indivíduos que pertencem à população externa, sempre melhores do que os valores calculados para a população actual.

Muitos outros exemplos de aplicações de metaheurísticas em contextos multi-objectivo podem ser encontrados. Horn, Nafpliotis & Goldberg, 1994, apresentam um algoritmo genético multi-objectivo, em que indivíduos seleccionados aleatoriamente são comparados com os indivíduos pertencentes a um conjunto de comparação (também escolhidos de forma aleatória). Este conjunto serve de base para comparar soluções, utilizando o número de soluções, nesse conjunto, que dominam ou são dominadas. Em caso de empate, utilizam o conceito de *sharing*. Fonseca & Fleming, 1995a, introduzem os algoritmos genéticos multi-objectivo, considerando que estes algoritmos, quando combinados com um interface gráfico adequado, podem tornar-se métodos interactivos com grande potencial, permitindo ao decisor aprender acerca do problema antes de optar pela solução final. Em Fonseca & Fleming, 1995b, os autores discutem os algoritmos multi-objectivo descritos até à altura na literatura. Consideram que cada etapa de selecção de soluções pode ser interpretada como um problema de decisão por si só, e que é importante que o valor de *fitness* atribuído a cada indivíduo reflecta o entendimento que o decisor tem acerca da qualidade de cada solução. Defendem que algoritmos evolutivos baseados unicamente no conceito de óptimo de Pareto terão sempre dificuldades em problemas com vários objectivos conflituosos, devido à dimensão da fronteira não-dominada. Lis & Eiben, 1996, apresentam um algoritmo genético multi-objectivo em que a cada objectivo corresponde um sexo diferente, sendo que cada indivíduo é avaliado unicamente em relação ao objectivo que corresponde ao sexo que tem. Aplicam um procedimento de *crossover* com múltiplos pais. Binh, 1999; Binh & Korn, 1996, 1997, descrevem um algoritmo

evolutivo multi-objectivo, mas não introduzem nenhum elemento novo ao que é descrito na literatura. Valenzuela-Rendón & Uresti-Charre, 1997, apresentam um algoritmo genético não geracional. Isto significa que não existe o conceito de geração: em cada ciclo apenas uma pequena parte dos indivíduos existentes (normalmente apenas um) é substituída por novos indivíduos. M. P. Hansen, 1997, apresenta um algoritmo de pesquisa tabu aplicado a problemas multi-objectivo. Loughlin & Ranjithan, 1997, propõem um algoritmo genético multi-objectivo a que dão o nome de *neighborhood constraint method*. Este algoritmo não faz uso explícito das relações de não-dominância entre as soluções para avaliar cada indivíduo, e é inspirado no fenómeno natural de adaptação dos indivíduos às condições ambientais. Para lidar com problemas com k objectivos, converte $k-1$ objectivos em restrições, otimizando o objectivo restante. Köppen & Rudlof, 1998, descrevem um algoritmo que resulta da integração de um algoritmo evolutivo com uma arquitectura de redes neuronais, em que o estado de cada neurónio equivale a um cromossoma. Zitzler & Thiele, 1998, aplicam quatro algoritmos evolutivos a problemas de saco de mochila multi-objectivo, e comparam a qualidade dos métodos tendo como base o tamanho da fronteira não dominada coberta. Coello Coello & Christiansen, 1998, descrevem uma série de métodos evolutivos desenvolvidos para a resolução de problemas multi-objectivo e propõem dois novos métodos baseados em optimizações do tipo *min-max*. Neste trabalho descrevem também uma aplicação (MOSES) que integra todas as metodologias descritas. Borges & Hansen, 1998, trabalham com o problema do caixeiro-viajante multi-objectivo, e estudam questões de convexidade global. Laumanns, Rudolph & Schwefel, 1998, apresentam um novo algoritmo que faz uso de um modelo de caçador-presa. Hanne, 1999, estuda o problema da convergência dos algoritmos evolutivos multi-objectivo, mas não refere resultados para aqueles que utilizam uma segunda população que guarda as soluções não-dominadas. Rekiek, 1999, descreve um algoritmo genético em que os indivíduos são comparados entre si não recorrendo a um valor único de *fitness* mas recorrendo ao resultado da aplicação do método multi-critério PROMETHEE II. Deb, 1999, apresenta algumas justificações para o facto de, algumas vezes, os algoritmos genéticos não serem capazes de convergir para a fronteira não-dominada. Coello Coello, 1999a, 1999b, apresenta uma introdução aos principais algoritmos evolutivos multi-objectivo, dando especial importância às suas raízes e motivando o desenvolvimento de novas abordagens. Para cada técnica descrita, são apresentadas as suas vantagens e desvantagens e as aplicações conhecidas. O autor chama a atenção para a dificuldade de comparação de métodos numa área em que a “melhor” solução é, em último caso, sempre determinada pelo decisor. Em Rudolph, 1998, o autor mostra como alguns resultados teóricos conhecidos para os casos de aplicação de algoritmos evolutivos a problemas com um único objectivo não são extensíveis ao caso multi-objectivo. Sbalzarini, Müller & Koumoutsakos, 2000, modificam ligeiramente o algoritmo SPEA para que possa ser possível considerar um ponto de referência. Desta forma, o algoritmo deve calcular soluções o mais próximas possível desse ponto, em vez de calcular soluções ao longo de toda a fronteira não-dominada. Rudolph & Agapie, 2000,

apresentam quatro algoritmos evolutivos e alguns resultados teóricos acerca da convergência destes métodos. Van Veldhuizen & Lamont, 2000, apresentam um esquema de classificação para algoritmos evolutivos multi-objectivo, e avaliam a variedade de algoritmos existentes. Os autores consideram imprescindível a existência de uma segunda população composta pelas soluções não dominadas que vão sendo calculadas, pois o carácter estocástico do algoritmo não garante que essas soluções se mantenham na população até ao fim da execução do algoritmo. Consideram também importante que se opte por considerar o menor número possível de funções objectivo, inserindo novas funções apenas se tal for necessário para capturar algumas características relevantes do problema. Teghem, J., Tuytens & Ulungu, 2000, desenvolvem uma metodologia em que aplicam *simulated annealing* a problemas multi-objectivo. Cvetkovic & Parmee, 2001; Parmee, Cvetkovic, Watson & Bonham, 2000, apresentam a aplicação a um algoritmo evolutivo multi-objectivo de um método de aferição de preferências que se baseia no desenvolvimento de um modelo matemático de preferências que evita a existência de contradições, no espírito da escola americana. Os autores chamam a atenção para a importância de não “confundir” o decisor disponibilizando-lhe uma quantidade de informação, ou obrigando-o a estabelecer valores para um conjunto de parâmetros, superior à sua capacidade cognitiva. Zitzler, Thiele & Deb, 2000, comparam, de forma empírica, alguns algoritmos evolutivos tendo como base seis funções de optimização. Concluem que o melhor desempenho é obtido com a aplicação do algoritmo SPEA, desenvolvido pelo próprio autor, que utiliza explicitamente o conceito de elitismo. Dumitrescu, Grosan & Oltean, 2000, apresentam um algoritmo evolutivo capaz de trabalhar com problemas em que as soluções são ou um ponto ou um intervalo fechado. Coello Coello & Pulido, 2000, 2001, propõem um algoritmo genético que trabalha com uma população muito pequena e com um processo de reinicialização. Num trabalho posterior, apresenta-se uma versão revista deste algoritmo em que não existe necessidade de ajustar parâmetros. Branke, Kaubler & Schmek, 2001, apresentam um algoritmo evolutivo em que as preferências do decisor são introduzidas no algoritmo através da possibilidade de definir funções que caracterizam os compromissos existentes entre os objectivos. Os autores consideram que, mesmo que o decisor não seja capaz de definir as suas preferências na forma de uma função utilidade, ele terá sempre uma ideia acerca de quais são os compromissos aceitáveis. Neste trabalho consideram apenas os casos em que existem dois objectivos, e os compromissos definidos pelo decisor são lineares. Barbosa & Barreto, 2001, descrevem um algoritmo genético em que duas populações evoluem, sendo que uma delas é uma população de conjuntos de pesos para ponderar as funções objectivo. A evolução entre as duas populações está ligada pelo cálculo do *fitness* de cada indivíduo, que envolve interacção com o utilizador. Laumanns, Zitzler & Thiele, 2001, estudam a influência dos mecanismos de arquivo (que permitem controlar as soluções não dominadas calculadas), de elitismo e de manutenção de diversidade na qualidade das soluções calculadas por algoritmos evolutivos multi-objectivo. Apresentam resultados válidos para o problema de saco de mochila bicritério. Tan, Lee & Khor, 2001, apresentam um algoritmo genético

multi-objectivo em que a dimensão da população é um parâmetro dinâmico, que é aumentado ou diminuído consoante as soluções descobertas em cada instante. Os autores propõem ainda uma medida da convergência do método, introduzindo os conceitos de domínio da população e de rácio de progresso. Jazskiewicz, 2002, apresenta um algoritmo que tem como objectivo gerar, num curto espaço de tempo, um conjunto de soluções potencialmente eficientes que permita ao decisor escolher uma solução de compromisso. A ideia base do algoritmo é gerar aleatoriamente vectores de pesos em cada iteração do método que são depois utilizados na construção de uma função utilidade. Em termos conceptuais, isto corresponde à tentativa de otimizar simultaneamente todas as funções objectivo que se obtêm por ponderação das funções objectivo originais do problema. Um procedimento de pesquisa local é aplicado a cada filho. Laumanns, Thiele, Deb & Zitzler, 2002a e 2002b, apresentam um algoritmo evolutivo multi-objectivo em que desenvolvem estratégias de arquivo e selecção que permitam, simultaneamente, a convergência para o conjunto de soluções não dominadas, e a cobertura de toda a gama de soluções. Os autores definem o conceito de ϵ -dominância e defendem que o arquivo deverá ser de dimensão constante, independente do número de gerações executadas. Lu & Yen, 2002, apresentam um algoritmo evolutivo que converte qualquer problema multi-objectivo num problema com dois objectivos: minimização, para cada indivíduo, dos valores de *fitness* dos indivíduos que o dominam e minimização da densidade de soluções. Kumar & Rockett, 2002, desenvolvem o algoritmo PCGA (Pareto Converging Genetic Algorithm). Segundo os autores, este algoritmo elimina a necessidade de trabalhar com parâmetros e procedimentos escolhidos de forma heurística. Definem um histograma de ordenação de Pareto que representa a frequência de empates na ordenação de soluções, mantendo as soluções não dominadas entre gerações. Zitzler, Laumanns & Thiele, 2002a, apresentam uma versão melhorada do algoritmo SPEA. Sarker, Liang & Newton, 2002, descrevem um algoritmo que preserva apenas indivíduos não-dominados, podendo haver variações no número de indivíduos na população, de uma geração para outra. Os autores optam por utilizarem uma codificação com número reais e não binários. Knowles & Corne, 2002, apresentam o algoritmo PAES (*Pareto Archived Evolution Strategy*), que mantém um registo limitado de indivíduos não-dominados, escolhidos com o objectivo de manter a diversidade das soluções. Para tal a fronteira não-dominada é dividida em regiões, e calcula-se o grau de densidade de soluções em cada uma delas. O algoritmo assenta, sobretudo, no operador de mutação. Coello Coello & Sierra, 2003, apresentam um algoritmo baseado em co-evolução: o desempenho de um determinado indivíduo depende dos indivíduos de uma outra população, podendo definir-se relacionamentos de competitividade ou cooperação entre populações. Coello Coello & Becerra, 2003, apresentam aquilo que os autores consideram ser a primeira tentativa de introduzir o conceito de *cultura* em algoritmos evolutivos, que não é mais do que a incorporação de conhecimentos obtidos durante o processo evolutivo. São definidos dois espaços (o espaço da população e o espaço das *crenças*), sendo definido um protocolo que controla a interacção entre eles. Na realidade, este segundo espaço não é mais do que uma

matriz que permite aferir até que ponto é que as soluções estão a ser calculadas de forma a estarem bem dispersas ao longo da região não-dominada. Yen & Lu, 2003, propõem um algoritmo evolutivo dinâmico multi-objectivo, baseado numa estratégia de estimação de densidade e numa ordenação baseada em células (que dividem o espaço dos objectivos) para calcular de forma eficiente a diversidade e a dominância entre indivíduos, numa população cuja dimensão varia dinamicamente. O algoritmo tem ainda em conta a idade e a “saúde” de cada indivíduo. Cortés & Coello Coello, 2003, desenvolvem um novo algoritmo multi-objectivo baseado na criação de clones: cada indivíduo é considerado um “anti-corpo”, sendo que os melhores são os que correspondem a soluções não-dominadas. Estes irão ser clonados, aumentando o tamanho da população. Os sobreviventes sofrem depois mutações. O operador de *crossover* é apenas utilizado na população secundária, onde se guardam todos os indivíduos não-dominados calculados. Q. Chen & Guan, 2004, apresentam um algoritmo genético em que o processo de evolução está dividido em etapas, tantas quantas o número de objectivos. Em cada etapa considera-se incrementalmente mais um objectivo. Hu, Coello Coello & Huang, 2005, apresentam um algoritmo evolutivo multi-objectivo a que dão o nome de NEES (*Neighborhood Exploring Evolution Strategy*). Neste algoritmo, os indivíduos são ordenados por ordem ascendente do seu valor de *fitness* (calculado tendo em conta as relações de não-dominância existentes entre os indivíduos). O algoritmo reconhece uma vizinhança para cada indivíduo, que é tanto maior quanto maior a posição do indivíduo na lista. Cada indivíduo sofre, então, uma mutação λ vezes, produzindo λ filhos dentro da sua vizinhança. Este indivíduo e respectivos filhos constituem uma família. Apenas o melhor indivíduo de cada família sobrevive como pai para a próxima geração.

A maior parte dos autores considera como a característica mais importante de qualquer algoritmo multi-objectivo a capacidade de se conseguir calcular um subconjunto representativo do conjunto de soluções não dominadas. É importante que o algoritmo convirja para o conjunto de soluções não dominadas, sem que perca diversidade (caso contrário apenas uma pequena parte deste conjunto seria conhecido). Deb, 2001, considera que estes dois objectivos (descobrir soluções o mais perto possível das soluções óptimas de Pareto e o mais diversificadas possíveis) são objectivos “ortogonais”, no sentido em que o primeiro requer uma procura em direcção à região não-dominada, enquanto que o segundo requer uma procura ao longo da fronteira não-dominada.

Na maior parte dos casos a interacção com o decisor acontece *a priori* ou *a posteriori*, isto é, antes ou depois da execução do algoritmo e do cálculo de soluções não dominadas (excepção é, por exemplo, o trabalho de Branke *et al.*, 2001; Shibuya, Kita & Kobayashi, 1999; Barbosa & Barreto, 2001; Sakawa & Kato, 2002). Zitzler, 2002, defende que no campo de optimização de problemas multi-objectivo recorrendo a algoritmos evolutivos é geralmente assumido que a optimização toma lugar antes do processo de decisão, pelo que o objectivo primeiro é o de encontrar a melhor aproximação possível para o conjunto de soluções óptimas de Pareto. Deb, 2001, afirma mesmo que o esforço maior deve ser colocado no cálculo das soluções de compromisso, definindo como

procedimento ideal de optimização para problemas multi-objectivo aquele em que primeiro se faz o cálculo de soluções e depois se escolhe uma das soluções. Apesar de descrever alguns métodos que permitem incluir as preferências do decisor por forma a “viciar” a pesquisa para que incida sobre um determinado conjunto de soluções, diminuindo o esforço do algoritmo, todos estes métodos são não interactivos. Podem, também, considerar-se métodos que interagem com o decisor, tendo como base um conjunto de soluções não-dominadas calculadas *a priori*, em vez de soluções que vão sendo calculadas progressivamente e fortemente dependentes das interacções com o decisor. Jaszkiwicz, 2000, chama a esta geração de soluções a geração *off-line*, e apresenta algumas vantagens em relação à geração *on-line*: permite que sejam efectuadas análises estatísticas que aumentem o conhecimento que o decisor tem do problema; garante uma interacção rápida com o decisor porque não há optimizações a decorrer durante a interacção; garante que todas as soluções mostradas ao decisor são não-dominadas entre si, enquanto que numa interacção *on-line* podem mostrar-se ao decisor soluções que mais tarde se descobre serem dominadas por outras entretanto calculadas. Estas vantagens apresentadas podem ser facilmente criticadas. Está, à partida, a limitar-se a gama de opções do decisor e a partir-se do princípio que o conhecimento das características do problema é feito somente a partir do conhecimento de soluções não-dominadas. As análises estatísticas podem sempre ser feitas à medida que o método interactivo avança. Coello Coello, 2000b, afirma que existe muito pouco trabalho feito na área dos algoritmos evolutivos que considere explicitamente as preferências do decisor, e que permita que essas preferências sofram alterações ao longo do tempo. Também Van Veldhuizen, 1999, considera surpreendente a escassez de esforços em métodos interactivos. Segundo o autor, qualquer que seja a técnica utilizada, a interacção entre o decisor e os algoritmos de pesquisa pode apenas levar a melhores soluções. Coello Coello, Van Veldhuizen & Lamont, 2002, defendem que, quando é desejável conduzir a pesquisa para uma determinada área de interesse, algo comum em problemas reais complexos, os métodos interactivos são a melhor escolha constituindo mesmo uma área de investigação futura promissora (Coello Coello, 2001). É também importante reconhecer a limitação dos modelos matemáticos utilizados como base para qualquer algoritmo. Na realidade, nada garante que, mesmo que centenas de soluções não-dominadas sejam calculadas por um dado algoritmo, se encontre entre elas uma solução aceitável pelo decisor (Bentley & Wakefield, 1998).

Neste trabalho parte-se do pressuposto que é benéfico para o decisor que o algoritmo seja interactivo, e que convirja para soluções não dominadas mas que se encontrem em áreas de interesse do decisor, áreas essas que são geralmente definidas no espaço dos objectivos. O algoritmo deve permitir “saltar” de um área de interesse para outra de forma fácil e eficiente, por forma a permitir a exploração de toda a região não dominada. As áreas de interesse podem ser estabelecidas pelo decisor de forma interactiva, através da imposição de limites inferiores ou superiores aos valores das funções objectivo⁴⁰, da explícita

⁴⁰ Considerando que todos os objectivos são de minimização.

definição de pesos para cada função objectivo ou pela definição de pontos de referência. No algoritmo desenvolvido as três possibilidades podem ser consideradas. Deve também dar-se sempre ao decisor a possibilidade de ele conhecer e seleccionar soluções que são dominadas (quando se consideram apenas as funções objectivo explicitamente definidas no problema). A justificação para tal possibilidade prende-se com o facto de ser possível que o decisor tenha dois tipos de preferências com características diferentes: aquelas para as quais ele é capaz de explicitar razões racionais para preferir uma solução a outra (e também para definir funções objectivo que reflectam essas preferências), mas também outras preferências intrínsecas para as quais não é de todo possível estabelecer razões racionais para a escolha (Cvetkovic, 1997).

A. E. Eiben & Smith, 2003, são da opinião que, num mundo ideal, existiriam a tecnologia e os algoritmos capazes de providenciar uma solução óptima para qualquer problema que fosse colocado ao sistema. E onde ficam os problemas multi-objectivo neste mundo ideal? Será que num mundo ideal não existem problemas com mais do que um objectivo conflituosos entre si?

4.5.1. Detalhes do Algoritmo

Vai agora apresentar-se o algoritmo desenvolvido (Joana Dias, Captivo & Clímaco, 2008b). Considere-se uma população P de indivíduos x . Considere-se que $f_i(x)$ representa o desempenho de x com respeito à i -ésima função objectivo, X é o conjunto de todas as soluções admissíveis para o problema de localização.

Definição 4.3: Um indivíduo $x \in P$ é P -eficiente se e só se não existe nenhum outro indivíduo $x' \in P$ tal que $f_i(x') \leq f_i(x)$, para todos os objectivos i , e $f_i(x') < f_i(x)$ para, pelo menos, um objectivo i . Caso contrário o indivíduo diz-se P -não eficiente.

Definição 4.4: Considerem-se dois indivíduos, x e x' , tais que $f_i(x) \leq f_i(x')$, para todos os objectivos i , e $f_i(x) < f_i(x')$ para pelo menos um objectivo i . Então, x domina x' .

Definição 4.5: Considere-se uma população P tal que todas as soluções $y \in X$ são representadas por, pelo menos, um indivíduo $x \in P$. Se x é P -eficiente, então x é eficiente e a solução respectiva y é eficiente. Caso contrário y não é eficiente.

Definição 4.6: Se $y \in X$ é eficiente, a sua imagem no espaço dos objectivos, z , é não dominada.

Proposição 4-4: Se x é eficiente, então x é P -eficiente para toda e qualquer população P considerada.

Proposição 4-5: Um indivíduo x pode ser P_1 -eficiente e P_2 -não eficiente se e só se P_2 tiver pelo menos um indivíduo $x' \notin P_1$, tal que x' domina x .

4.5.1.1. O Desempenho de Cada Indivíduo e os Problemas de Afectação

Quando se pretende adaptar um algoritmo evolutivo ao caso multi-objectivo, um dos primeiros problemas a resolver é o de como calcular o valor de *fitness* de cada indivíduo pertencente à população. Na realidade, não só é difícil decidir como se avalia o desempenho de cada indivíduo, como pode também ser

computacionalmente penoso ter de calcular, para cada indivíduo, os valores correspondentes de cada função objectivo (existe mesmo a possibilidade de se considerar que, além dos genes, os filhos podem também herdar dos pais o valor de *fitness*, para que se consiga diminuir o tempo computacional que se gasta nestes cálculos - J.-H. Chen, Goldberg, Ho & Sastry, 2002; Sastry, Goldberg & Pelikan, 2001). Enquanto que operadores como o de mutação ou *crossover* não têm de ser alterados, a selecção dos indivíduos é, forçosamente, afectada pelo carácter multi-objectivo do problema (Hanne, 1999). No caso do algoritmo memético desenvolvido, torna-se também necessário decidir como se irão calcular os valores óptimos dos problemas de afectação. Para se efectuar este cálculo é necessário recorrer a, apenas, uma função objectivo. Seria também possível considerar vários objectivos nos problemas de afectação, mas tal opção iria acarretar um incremento significativo da complexidade computacional e obrigaria à utilização de estruturas de dados extra (pois indivíduos idênticos poderiam ter variáveis de afectação com valores distintos – a representação de cada indivíduo não está preparada para lidar com esta possibilidade).

O valor de *fitness* de cada indivíduo x pode ser calculado de várias formas: usando funções de agregação lineares ou não lineares, considerando o número de indivíduos que dominam ou são dominados por x na população actual, pedir ao decisor que faça comparações entre indivíduos diferentes, etc.⁴¹

Neste trabalho optou-se por seguir duas possibilidades distintas: considerar o desempenho de um indivíduo igual ao valor de uma função objectivo ponderada, ou então considerar a distância a um ponto de referência.

4.5.1.1.1. Função Objectivo Ponderada

Neste caso utiliza-se uma função objectivo ponderada para calcular os valores óptimos das variáveis de afectação. Esta mesma função é utilizada no procedimento de pesquisa local, para guiar o processo de procura de melhores soluções. No procedimento de pesquisa local, os pesos utilizados na definição de uma única função objectivo são iguais aos que foram utilizados aquando da escolha dos progenitores. Ishibuchi, Yoshida & Murata, 2003, não consideram esta opção, em que se utilizam os mesmos pesos na escolha dos pais e na pesquisa local, uma boa abordagem. Os autores consideram que a direcção da pesquisa deve ser dependente da posição relativa, no espaço dos objectivos, da solução de partida. Descrevem um modo alternativo de levar a cabo a pesquisa local: gerar um conjunto de pesos de forma aleatória e independente da forma como os pais foram escolhidos; aplicar a pesquisa local a um subconjunto de filhos escolhidos dentro da população, através de selecção por torneio em que se utiliza a função objectivo ponderada. Os autores defendem que não valerá a pena dispendir tempo com procedimentos de pesquisa local junto a soluções de muito má qualidade. Por este motivo, apenas aplicam pesquisa local a indivíduos que já passaram o teste da selecção por torneio. Tal opção não foi ainda testada no algoritmo que foi desenvolvido.

⁴¹ Ver Coello Coello *et al.*, 2002, por exemplo.

Existem numerosos procedimentos que permitem a determinação de pesos a afectar às funções objectivo (Weber & Borchering, 1993). Neste trabalho, os pesos que são utilizados podem ter duas interpretações distintas: se o decisor assim o desejar, os pesos podem ser explicitamente afectos a cada função objectivo e, de algum modo, serem uma representação das preferências do decisor. Não é este, contudo, o melhor uso que se entende poder ser dado aos pesos. Atendendo às preferências do decisor expressas, por exemplo, pelo estabelecimento de limites aos valores das funções objectivo, os pesos podem ser calculados automaticamente pelo algoritmo, sendo assim, apenas, interpretados como ferramenta técnica necessária no cálculo de soluções não dominadas. Voltar-se-á a este assunto mais tarde, quando da definição dos possíveis protocolos de interacção com o decisor.

4.5.1.1.2. Pontos de Referência

A metodologia dos pontos de referência na resolução de problemas com mais do que um objectivo pode ser visto como uma generalização da programação por metas (Wierzbicki, 2000b). Pontos de referência são metas, interpretadas de forma consistente em relação aos conceitos básicos de optimalidade vectorial: a ideia de “estar perto” da meta é especial e não é equivalente à minimização de uma norma. Segundo Wierzbicki, 2000b, se se concordar com a definição de óptimos de Pareto, então a ideia de “estar próximo” de um ponto de referência deve significar que se pretende encontrar: 1) soluções uniformemente próximas do ponto de referência, se este não for alcançável; 2) soluções que correspondam exactamente ao ponto de referência, se este for atingível; 3) soluções que são melhores do que o ponto de referência, se este é possível de atingir e de ser melhorado. Desta forma, pretende-se minimizar uma distância, não do ponto de vista geométrico mas do ponto de vista das preferências do decisor (Romero, Tamiz & Jones, 1998).

No algoritmo desenvolvido considerou-se que o decisor pode interagir com o algoritmo através da definição de pontos de referência: pontos de aspiração e de reserva. Desta forma, o valor de desempenho de cada indivíduo é calculado com base nestes dois pontos.

Defina-se uma função $\sigma_i(q_i, \bar{q}_i, \underline{q}_i)$ (Wierzbicki, 2000a), tal que q_i representa o valor da função objectivo i correspondente à solução codificada pelo indivíduo q , \bar{q}_i e \underline{q}_i os valores de aspiração e reserva, respectivamente, definidos para a função objectivo i , $q_{i,lo}$ e $q_{i,up}$ limites inferior e superior conhecidos para o valor da função objectivo i .

$$\sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) = \begin{cases} 1 + \alpha(\bar{q}_i - q_i)/(\bar{q}_i - q_{i,lo}), & q_{i,lo} \leq q_i \leq \bar{q}_i \\ (\bar{\bar{q}}_i - q_i)/(\bar{\bar{q}}_i - \bar{q}_i), & \bar{q}_i < q_i < \bar{\bar{q}}_i \\ \beta(\bar{\bar{q}}_i - q_i)/(\bar{\bar{q}}_i - q_{i,up}), & q_i \leq q_i \leq q_{i,up} \end{cases} \quad (4-10)$$

Esta função mostra a que distância um determinado indivíduo se encontra dos pontos de referência definidos. Os coeficientes α e β devem ser maiores do que zero e escolhidos por forma a que esta função seja monótona e côncava. No presente algoritmo, estes parâmetros são calculados por tentativa e erro, como descrito no Algoritmo 4-8.

O valor de desempenho de um dado indivíduo q , em relação aos pontos de aspiração e reserva \bar{q} e $\bar{\bar{q}}$, respectivamente, é calculado usando a função $\sigma(q, \bar{q}, \bar{\bar{q}})$ definida em (4-11), em que k representa o número de funções objectivo consideradas.

$$\sigma(q, \bar{q}, \bar{\bar{q}}) = \left(\min_{1 \leq i \leq k} \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) + \varepsilon \sum_{i=1}^k \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) \right) / (1 + k\varepsilon), \varepsilon > 0 \quad (4-11)$$

O valor obtido caracteriza um indivíduo em relação à distância a que se encontra dos pontos de referência, considerando as possíveis situações: o indivíduo pode dominar ou ser dominado pelos pontos de referência definidos. Um indivíduo que se encontre distante de um ponto de aspiração, mas que o domine, é considerado melhor do que um indivíduo próximo, ou exactamente igual, a este ponto.

Note-se que o cálculo do desempenho de um indivíduo feito deste modo pressupõe que os objectivos podem tomar, todos eles, valores de grandeza semelhante. Se tal não acontecer, é necessário garantir que todos os objectivos irão tomar valores dentro de intervalos da mesma gama, executando algumas operações simples de conversão (pois este método de classificação de soluções insere-se no grupo dos métodos *range dependent*, como definido em Bentley & Wakefield, 1998).

Ao efectuar-se a selecção dos indivíduos, estes são escolhidos através de torneio binário em que se dá preferência aos que apresentam um valor de *fitness* maior.

No procedimento de pesquisa local, é necessário trabalhar com uma função objectivo ponderada, por forma a que seja possível aplicar o procedimento descrito em 4.3.4. Para tal optou-se por gerar de forma aleatória pesos para atribuir a cada função objectivo e utilizar a função objectiva ponderada como guia na pesquisa local.

Esta forma de aplicar a metodologia dos pontos de referência a um algoritmo evolutivo não é única. Szczepanski & Wierzbicki, 2002, por exemplo, utilizam a

função $\sigma(q, \bar{q}, \bar{\bar{q}})$ para definirem um *ranking* de indivíduos que pretende ter em conta quer o número de indivíduos que domina q , quer a sua distância aos pontos de referência.

Algoritmo 4-8

CÁLCULO DOS PARÂMETROS α E β

\bar{q}_i - valor de aspiração; $\bar{\bar{q}}_i$ - valor de reserva; $q_{i,lo}$ - limite inferior conhecido; $q_{i,up}$ - limite superior conhecido.

1. $\alpha \leftarrow -1$; $\beta \leftarrow 0$; $\varphi \leftarrow -1$.
 2. Se $\varphi < 1$, então ir para 6. Se não, $\alpha \leftarrow \alpha/2$ e ir para 3.
 3. $a \leftarrow (\bar{q}_i + q_{i,lo})/2$; $b \leftarrow \sigma_i(a, \bar{q}_i, \bar{\bar{q}}_i)$, considerando o actual valor de α .
 4. $c \leftarrow \bar{q}_i + (\bar{\bar{q}}_i + \bar{q}_i)/2$; $d \leftarrow \sigma_i(c, \bar{q}_i, \bar{\bar{q}}_i)$.
 5. $\lambda \leftarrow \frac{\bar{q}_i - c}{a - c}$; $\varphi \leftarrow \lambda b + (1 - \lambda)d$. Ir para 2.
 6. $a \leftarrow (\bar{q}_i + \bar{q}_i)/2$; $b \leftarrow \sigma_i(a, \bar{q}_i, \bar{\bar{q}}_i)$, considerando o actual valor de β .
 7. $c \leftarrow (q_{i,up} + \bar{q}_i)/2$; $d \leftarrow \sigma_i(c, \bar{q}_i, \bar{\bar{q}}_i)$.
 8. $\lambda \leftarrow \frac{\bar{q}_i - c}{a - c}$; $\varphi \leftarrow \lambda b + (1 - \lambda)d$.
 9. Se $\varphi < 0$, então parar. Caso contrário, $\beta \leftarrow \beta \times 1.25$ e ir para 6.
-
-

4.5.1.2. Populações

Na versão actual, o algoritmo trabalha com duas populações distintas. Utilizando a notação definida por Van Veldhuizen & Lamont, 2000, a população $P_{current}(ger)$ representa a população na geração ger . A população $P_{known}(ger)$ é composta por todos os indivíduos eficientes no conjunto $\bigcup_{g=1}^{ger} P_{current}(g)$. Van Veldhuizen, 1999, considera imprescindível a existência de uma segunda população composta pelas soluções não dominadas encontradas até ao momento. Note-se que é possível que haja um indivíduo que seja eficiente com respeito a $P_{current}$ e não eficiente com respeito a P_{known} . Mais ainda: podem existir indivíduos eficientes com respeito a $P_{known}(ger)$ e não eficientes com respeito a $P_{known}(ger+g)$, $g > 0$. Isto significa que para construir $P_{known}(ger)$ não basta copiar todos os indivíduos eficientes de $P_{current}(ger)$; é também necessário verificar a eficiência de todos os elementos de $P_{current}(ger)$ em relação a $P_{known}(ger-1)$ e vice-versa. Este procedimento consiste num algoritmo de complexidade $O(n^2)$, em que n representa o número de indivíduos na população (Van Veldhuizen & Lamont, 2000), não devendo, por isso, ser executado com muita frequência. É também importante notar que a população P_{known} tem uma capacidade limitada: se o algoritmo encontra muitas soluções eficientes, então é necessário ir actualizando este conjunto, com a possível eliminação de alguns indivíduos e introdução de outros. Mesmo que se considerasse que P_{known} poderia conter um número ilimitado de indivíduos, não seria razoável mostrar um tão grande

número de soluções ao decisor. Por este motivo, será sempre necessário definir-se algum tipo de método que permita escolher um subconjunto de soluções eficientes de P_{known} ⁴². Outra questão interessante é a definição da interacção destas duas populações entre si e com o algoritmo.

No algoritmo desenvolvido considera-se que o conjunto $P_{known}(ger)$ é igual a $P_{known}(ger-1) \cup P_{current}(ger)$. Os indivíduos não eficientes de P_{known} são apenas eliminados quando o decisor pretende ver todas as soluções P_{known} -eficientes calculadas até ao momento, ou, então, o número de elementos em $P_{current}(ger)$ é maior do que a capacidade ainda disponível de P_{known} . Neste caso, apenas são inseridos indivíduos eficientes e todos os indivíduos não eficientes são removidos de P_{known} ⁴³. Se P_{known} atingiu o seu número máximo de elementos e todos os indivíduos são eficientes, alguns destes terão de ser removidos⁴⁴. Proceder-se-á do seguinte modo: se existem indivíduos em P_{known} que estão a violar alguma restrição imposta pelo decisor (restrição essa que o decisor decidiu não penalizar – cf 4.3.8), estes são os primeiros candidatos a serem removidos. O algoritmo escolhe aquele que se encontra em P_{known} há um maior número de gerações. Se nenhum dos indivíduos se encontra nesta situação, então o algoritmo escolhe aleatoriamente um indivíduo. Muitos outros procedimentos podiam ser levados à prática: perguntar directamente ao decisor quais os indivíduos que ele deseja manter ou remover; aplicar algoritmos de *clustering* por forma a que o conjunto P_{known} se mantenha composto por soluções dispersas, etc.

Neste algoritmo, a população P_{known} não participa de nenhuma forma nos processos de reprodução e selecção. Nenhum dos seus membros interage com indivíduos de $P_{current}$. Há autores que pensam que uma interacção próxima entre estas duas populações pode ser benéfica para o desempenho do algoritmo (Zitzler & Thiele, 1999). Uma possível interacção entre estas duas populações poderia ser facilmente integrada do seguinte modo: sempre que o número de indivíduos em $P_{current}$ é aumentado, os novos indivíduos podem ser aleatoriamente escolhidos a partir de P_{known} , em vez de serem aleatoriamente inicializados. Pode, também, definir-se uma probabilidade de escolher, em cada operação de selecção, um dos progenitores oriundos de P_{known} .

Os indivíduos P_{known} -eficientes são, em cada geração, uma aproximação do conjunto das soluções verdadeiramente eficientes (que não é conhecido). Existem várias, e não consensuais, maneiras de avaliar a qualidade das aproximações ao conjunto de soluções não dominadas (M. Hansen & Jaszkiewicz, 1998). Neste algoritmo, como se considera sempre que existe uma interacção directa com o decisor, tais medidas revestem-se de importância secundária. É sempre possível aferir se um dado indivíduo x representa uma solução eficiente

⁴² Em Laumanns, Thiele, Deb, & Zitzler, 2001, os autores apresentam e discutem maneiras diferentes de se fazer a actualização das populações de arquivo.

⁴³ Não se admite a existência de indivíduos replicados em P_{known} .

⁴⁴ A questão de como fazer a actualização da população de arquivo quando esta atingiu a sua dimensão máxima e existe uma nova solução não-dominada, tal que não domina nenhuma das soluções pertencentes à população de arquivo, é uma questão difícil e objecto de estudo por parte de alguns autores (Laumanns *et al.*, 2002a; Laumanns, Thiele, Zitzler, & Deb, 2002b; Zitzler, 2002; Corne & Knowles, 2003; Knowles & Corne, 2002).

$y \in X$ através da resolução de um problema de programação linear inteira mista, que se obtém, por exemplo, através da introdução de uma restrição adicional p por cada função objectivo, e considerando uma qualquer função objectivo ponderada. Esta restrição adicional impõe um limite máximo (no caso da minimização) ao valor de cada função objectivo o que, no caso presente, em que se pretende avaliar uma solução, corresponderia ao valor que essa solução atinge na função objectivo p menos um valor ξ tendencialmente próximo de zero. Se o problema resultante tem uma solução admissível, então y não é uma solução eficiente e deve ser substituída pela solução óptima do problema construído. Se o problema é impossível, então y é uma solução eficiente (devido ao resultado de Ross & Soland, 1980).

A população P_{known} é inicializada como um conjunto vazio. A população $P_{current}$ é inicializada de forma aleatória, como acontece no caso mono-objectivo, mas considerando diferentes conjuntos de pesos gerados aleatoriamente pelo algoritmo.

A forma de elitismo presente considera somente a passagem automática do melhor indivíduo (definido tendo como base o modo como, nessa geração, se está a calcular o desempenho de cada um) para a população seguinte. Como é lógico, quando se trabalha com problemas multi-objectivo, existem várias maneiras diferentes de se introduzir o conceito de elitismo (tantas quantas as maneiras diferentes de comparar soluções⁴⁵). Há que ter sempre o cuidado de considerar as alterações sentidas nas preferências do decisor, por forma a evitar que a consideração de elitismo tenha sérias desvantagens (como consistir numa âncora a que o decisor fica preso por escolhas feitas em fases iniciais do processo).

4.5.1.3. Interacção com o Decisor

Após a inicialização da população $P_{current}$, esta população evolui durante um número de gerações, sendo esta evolução fortemente dependente da interacção com o decisor. Assim, numa fase inicial, são mostradas ao decisor um conjunto de soluções. Mostrar, numa fase inicial do método interactivo, um conjunto de soluções ao decisor, em detrimento de mostrar apenas uma solução, é vantajoso pois diminui o efeito de *anchoring*. Se o decisor for de opinião que já encontrou uma solução de compromisso que considera de boa qualidade, então o algoritmo pára. Caso contrário, o algoritmo irá pedir-lhe para que ele expresse as suas preferências na forma de estabelecimento de limites superiores aos valores da função objectivo, estabelecimento de pesos para cada uma das funções objectivo ou através da indicação de pontos de referência. A interacção possível entre o decisor e o método é ilustrada através de um fluxograma presente na Figura IV-5, e é motivado pelos trabalhos de Joana Dias, 2000; Joana Dias, Captivo & Clímaco, 2003; Ferreira, 1997. Nenhuma decisão tomada em qualquer das fases do processo interactivo é irrevogável. Considera-se que as preferências do decisor são sempre relativas às circunstâncias (Cvetkovic, 1997), pelo que o método deve permitir que o decisor explore o espaço das soluções

⁴⁵ Para uma discussão desta questão consultar, por exemplo, Laumanns, Zitzler, & Thiele, 2000.

de forma livre, mesmo que isso signifique que o decisor esteja a apresentar comportamentos contraditórios à medida que o método interactivo é executado. Esta interacção pode ser melhorada através do uso de ferramentas visuais, principalmente no caso biobjectivo.

4.5.1.3.1. Estabelecimento de Limites Superiores ou de Pesos

Sempre que o decisor opta por limitar superiormente o valor das funções objectivo (considerando que são todas de minimização), então esta opção traduzir-se-á na introdução de restrições adicionais no problema, que são tratadas como descrito em 4.3.8.2.

Se existem apenas dois objectivos, o estabelecimento de limites superiores para cada função objectivo pode ser feita de forma fácil como descrito em Ferreira, Santos, Captivo, Clímaco & Silva, 1996; Ferreira, 1997; Joana Dias, 2000; Joana Dias *et al.*, 2003. Neste caso o algoritmo calcula os pesos de forma automática, considerando os limites impostos pelo decisor e tentando melhorar o desempenho do algoritmo (Joana Dias, 2000; Joana Dias *et al.*, 2003). Os pesos são calculados considerando os limites já estabelecidos anteriormente e os valores das funções objectivo dos melhores indivíduos gerados. Em todos os casos com três ou mais objectivos, os pesos são aleatoriamente gerados.

Em cada iteração do algoritmo, um número pré-definido de gerações são executadas e a melhor solução encontrada, considerando o conjunto actual de pesos, é mostrada ao decisor. Sempre que o decisor assim o desejar, pode conhecer todo o conjunto de soluções P_{known} -eficientes.

O algoritmo nunca considera a existência de decisões irrevogáveis por parte do decisor: ele pode mudar as suas preferências, pesquisar áreas que tinha optado por abandonar em fases anteriores, e apresentar um comportamento incoerente durante a execução do método interactivo.

4.5.1.3.2. Pontos de Referência

De acordo com Simon, 1957⁴⁶, (citado em Wierzbicki, 2000b), os agentes de decisão não optimizam uma função utilidade quando tomam decisões. Os decisores, através da aprendizagem, vão desenvolvendo de forma dinâmica níveis de aspiração em relação a vários objectivos e procuram soluções que estejam tão próximas quanto possível desses níveis de aspiração, ou que os alcancem quando tal é atingível.

Num algoritmo interactivo que utilize como ferramenta principal os pontos de referência, é pedido ao utilizador que este forneça níveis de aspiração e de reserva: os primeiros representam situações que o decisor gostaria de alcançar, enquanto que os segundos representam aquilo de que o decisor se quer afastar como, por exemplo, valores máximos admissíveis para as funções objectivo. Um exemplo de aplicação de um software interactivo que faz uso de pontos de referência pode ser encontrado em Granat & Makowski, 1998, 2000. Miettinen &

⁴⁶ Simon, H. A., 1957, *Models of Man*, Macmillan, New York, USA

Mäkelä, 1999, apresentam três métodos interactivos baseados na ideia de ponto de referência. Fonseca e Fleming, 1993, apresentam um método interactivo baseado na programação por metas.

Num algoritmo evolutivo, a população evolui concentrando-se, tendencialmente, na área potencialmente não-dominada mais interessante aos olhos do agente de decisão e definida pelos pontos de referência obtidos na fase do diálogo. Note-se que, tal como anteriormente, não se considera a existência de decisões irrevogáveis por parte do decisor.

O Algoritmo 4-9 ilustra o funcionamento geral do algoritmo genético para o caso multi-objectivo.

Algoritmo 4-9

ALGORITMO MEMÉTICO PARA PROBLEMAS COM MAIS DO QUE UM OBJECTIVO

$P_{current}$ – população actual; P_{known} – conjunto conhecido de soluções candidatas a não dominadas; $n_{gerações}$ – número máximo de gerações simuladas para $P_{current}$ em cada iteração do algoritmo na sua fase inicial; N – número máximo de iterações na fase inicial do algoritmo; x_{best} – melhor solução encontrada na iteração presente do algoritmo.

1. $P_{current} \leftarrow \emptyset, P_{known} \leftarrow \emptyset$.
 2. Inicializar aleatoriamente a população $P_{current}$, calculando o valor de *fitness* de cada indivíduo.
 3. $n \leftarrow 1$.
 4. Gerar aleatoriamente um conjunto admissível de pesos para as funções objectivo.
 5. Evoluir $P_{current}$ durante n gerações, utilizando o Algoritmo 4-6, e actualizar a população P_{known} .
 6. $n \leftarrow n + 1$. Se n é maior do que N , então ir para 7; se não, ir para 4.
 7. Mostrar a população P_{known} ao decisor. Se este ficar satisfeito, parar, se não ir para 8.
 8. Pedir ao decisor que estabeleça:
 - a. Limites superiores para cada função objectivo;
 - b. Pesos para cada função objectivo;
 - c. Pontos de referência.
 9. Evoluir $P_{current}$, utilizando o Algoritmo 4-6, e actualizar P_{known} .
 10. Mostrar ao decisor a solução representada por x_{best} . Se o decisor desejar, mostrar todas as soluções P_{known} -eficientes. Se o decisor estiver satisfeito, parar; se não, ir para 8.
-

4.5.2. Resultados Computacionais

Se a aferição da qualidade de um algoritmo evolutivo aplicado a problemas com um único objectivo não é trivial, muito mais complicado é aferir a qualidade de um algoritmo evolutivo aplicado a problemas com mais do que um objectivo. Zitzler, Laumanns, Thiele, Fonseca & Fonseca, 2002b, defendem que não é, em geral, possível encontrar um conjunto de medidas como, por exemplo, distância ou diversidade, que descrevam de forma única a qualidade de um conjunto de soluções candidatas a não-dominadas. Mostram, também, que a maioria das medidas existentes servem apenas para concluir que uma dada aproximação não é pior do que outra. A questão de aferir a qualidade de um algoritmo multi-objectivo é, em si mesmo, um outro problema multi-objectivo (Zitzler *et al.*, 2002b).

Quando se consideram métodos interactivos esta dificuldade aumenta ainda mais. No pressuposto de que o decisor escolhe uma solução de entre o conjunto de soluções candidatas a não dominadas utilizando uma função utilidade, que ele próprio pode desconhecer, existe a possibilidade de se executar o algoritmo interactivo utilizando os valores de uma determinada função para fazer progredir o método. No entanto, neste caso, não se admite a possibilidade de existência

de contradições nas respostas do decisor, e realizam-se testes que vão contra o princípio base que leva à opção de construção de um método interactivo: a capacidade de aprendizagem do decisor, e que o pode levar a mudar de opinião, a incluir características subjectivas e não racionalizáveis na sua escolha, etc. Como aferir, então, a qualidade de um método evolutivo e interactivo na resolução de um problema multi-objectivo? E como comparar duas abordagens diferentes?

O ideal seria poder contar com a ajuda de decisores que, dispendendo o seu tempo e paciência, se dispusessem a colaborar e a participar um elevado número de vezes na resolução de problemas multi-objectivo utilizando métodos diferentes. Tais decisores não existem, ou são realmente muito difíceis de encontrar. Como tal, uma solução de compromisso, que não a óptima, terá de ser adoptada. Zitzler *et al.*, 2000, consideram que, na definição da qualidade de um algoritmo evolutivo, se devem ter em conta os seguintes factores: uma medida da distância do conjunto de soluções não dominadas calculados até à fronteira de óptimos de Pareto (que deve ser a menor possível); a distribuição de soluções (que deve ser uniforme) e ainda a extensão de soluções não dominadas calculadas (para cada objectivo, uma gama larga de valores deve estar coberta pelas soluções calculadas).

A primeira dificuldade aparece no cálculo do conjunto de soluções não-dominadas. Os elementos que constituem este conjunto estão totalmente dependentes da interacção com o decisor, visto este poder preferir fazer uma pesquisa ao longo de toda a região não-dominada, ou escolher fixar-se, desde muito cedo no processo interactivo, numa determinada área desta região. Tais opções podem alterar significativamente a avaliação que é feita do ponto de vista da diversidade de soluções encontradas, por exemplo.

Neste capítulo interessa, sobretudo, retirar algumas conclusões sobre dois aspectos do algoritmo memético desenvolvido: verificar se é capaz de calcular soluções não-dominadas dentro da área de interesse definida pelo decisor, e verificar ainda se existem diferenças significativas, em termos do número e da qualidade das soluções encontradas, entre os dois modos diferentes de interacção com o utilizador descritos em 4.5.1.3.1 e em 4.5.1.3.2 Para tal, vão resolver-se um conjunto de problemas de localização, comparando-se os resultados obtidos com as duas abordagens. O número de soluções que são verdadeiramente não-dominadas e a distância das que o não são à solução não-dominada mais próxima (considerando a distância euclideana), podem ser utilizadas para caracterizar os algoritmos quanto à sua convergência. Algumas medidas como, por exemplo, a métrica da onda⁴⁷ (Van Veldhuizen, 1999), não fazem sentido ser aplicadas a algoritmos que trabalhem com duas populações, e em que se dá a conhecer ao decisor apenas soluções candidatas a não dominadas. Para caracterizar os algoritmos quanto à sua capacidade de encontrarem soluções que representem adequadamente toda a região não

⁴⁷ Nesta métrica, considera-se i igual a 1, e verificam-se todas as soluções candidatas a não-dominadas pertencentes ao conjunto de soluções. Estas soluções retiram-se do conjunto. Se o conjunto fica vazio, então a métrica devolve o valor i ; caso contrário, incrementa-se o valor de i em uma unidade e repete-se o processo. Um bom método de cálculo de soluções deverá ter valores baixos para esta métrica.

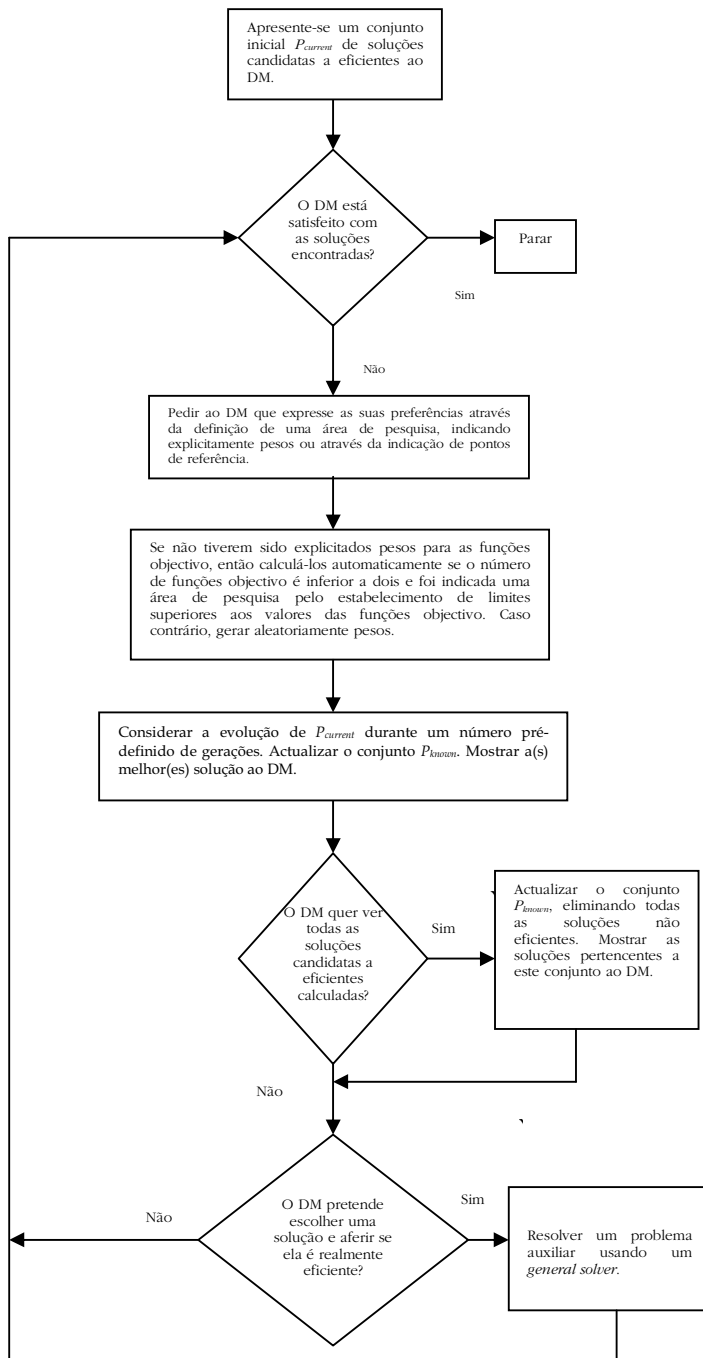


Figura 4.5 – Esquema da interacção com o Decisor

dominada, pode calcular-se a métrica espacial modificada definida em Collette &

Siarry, 2005: $medida = \left[\frac{1}{N-1} \sum_{i=1}^{N-1} \left(1 - \frac{d_i}{\bar{d}} \right)^2 \right]^{\frac{1}{2}}$, em que N é o número de soluções

candidatas a não dominadas, d_i representa uma medida da distância⁴⁸ entre a solução x_i e a solução seguinte (considerando-as ordenadas, por exemplo, de forma lexicográfica) e \bar{d} a média de todos os valores d_i .

Para cada uma das abordagens pode também calcular-se o número de soluções calculadas numa delas que são dominadas pelas soluções calculadas através da outra (como definido em Deb, 2001)⁴⁹.

Não se irão realizar comparações em relação aos tempos computacionais, pois serão muito semelhantes para ambas as abordagens interactivas, e já foram retiradas conclusões acerca da eficiência do algoritmo memético no caso mono-objectivo.

Três problemas de localização foram aleatoriamente gerados: um problema com um nível de serviços e sem capacidades, considerando um horizonte temporal de 10 períodos temporais, 25 possíveis localizações para os serviços e 100 clientes; um problema com um nível de serviços e restrições de capacidade máxima, da mesma dimensão do anterior; e um problema com dois níveis de serviços, sem capacidades, considerando um horizonte temporal de 10 períodos temporais, 100 clientes, 5 possíveis localizações para os serviços no segundo nível e 10 no primeiro nível. Estes problemas foram gerados utilizando os processos descritos no capítulo 3.

No caso do primeiro problema, o algoritmo começa por mostrar ao decisor um conjunto de soluções candidatas a não dominadas, como é ilustrado pela Figura 4.6. Como se pode ver, existem algumas áreas da região não-dominada que não estão a ser cobertas por este conjunto. O decisor interage com o algoritmo, estabelecendo limites superiores (e inferiores, se assim o desejar) aos valores das funções objectivo. Estes valores podem ser definidos explicitamente ou implicitamente através da selecção de duas soluções já conhecidas. Suponha-se que o decisor define os seguintes limites: a primeira função objectivo deve ter um valor inferior a 293191 e a segunda deve ter um valor inferior a 335809 (estes valores são implicitamente dados pela selecção de duas soluções, fornecendo simultaneamente limites superiores). Como pode ser observado na Figura 4.7, o algoritmo é capaz de calcular soluções na área de interesse definida, mas estas soluções não estão bem dispersas. Após mais duas interacções com o decisor, o conjunto de soluções encontrado é mostrado na Figura 4.8. O decisor

⁴⁸ Considerando que x_i e x_{i+1} são vectores que representam duas soluções no espaço dos objectivos, então $d_i = \sqrt{(f_1(x_i) - f_1(x_{i+1}))^2 + (f_2(x_i) - f_2(x_{i+1}))^2}$, em que $f_c(x_i)$ representa o valor da função objectivo c relativo a x_i .

⁴⁹ Muitas outras métricas podem ser utilizadas para caracterizar os algoritmos evolutivos multi-objectivos. Deb, 2001, dedica várias páginas do seu livro a descrever as mais utilizadas (pp 306-324).

pode, a qualquer momento, definir novas áreas de interesse, como ilustrado nas Figura 4.9 e Figura 4.10. Em todos os casos, o algoritmo é capaz de encontrar soluções candidatas a não dominadas na área de interesse, mas este conjunto é pobre em termos de dispersão de soluções.

Se o algoritmo pedir agora ao decisor que este forneça níveis de aspiração e reserva, os resultados são bastante diferentes. A Figura 4.11 mostra um primeiro conjunto de soluções e os pontos de aspiração e reserva definidos. A Figura 4.12 ilustra as soluções calculadas pelo algoritmo e a fixação de novos pontos de referência. Com apenas duas interações, a Figura 4.13 mostra o conjunto total de soluções calculadas e na Figura IV-14 compara-se este conjunto com o obtido através da abordagem anterior, em que a interação se fez com base em limites superiores.

Resultados semelhantes se mostram nas duas figuras seguintes, para o caso do problema com restrições de capacidade, em que se estabelecem limites superiores. Da Figura 4.19 à Figura 4.23 apresentam-se os resultados quando a interação se baseia nos pontos de referência. A Figura 4.24 compara os resultados obtidos com ambas as abordagens. Da Figura 4.25 à Figura 4.33 ilustram-se os resultados, também similares, obtidos aquando da resolução do problema com dois níveis. As três últimas figuras comparam os diferentes conjuntos obtidos para cada um dos problemas: o conjunto de óptimos de Pareto e os conjuntos de soluções candidatas a não-dominadas obtidos. A Tabela 4.7 faz um sumário dos resultados obtidos.

Como se pode justificar que a abordagem que assenta na indicação de pontos de referência presente, nos três exemplos, melhores resultados? À primeira vista podia pensar-se que a principal razão se prende com a utilização de pesos gerados aleatoriamente e que são frequentemente alterados. Isto pode, de facto, ser parte importante da resposta, mas a verdadeira justificação para estes resultados encontra-se na função de desempenho propriamente dita. Na realidade, se os pesos fossem escolhidos aleatoriamente, ou modificados com mais frequência, na abordagem que assenta no estabelecimento de limites superiores aos valores das funções objectivo, tal resultaria em piores resultados do que os apresentados. A qualidade de um dado indivíduo é dada pelo valor da função objectivo ponderada, considerando as restrições adicionais que correspondem aos limites estabelecidos. Se os pesos fossem calculados ou alterados de modo diferente, a maior parte dos indivíduos gerados não seriam admissíveis, em relação aos limites estabelecidos, levando a piores resultados globais. A mais-valia da interação que pede a definição de pontos de referência ao decisor reside na função de desempenho considerada, que não está dependente de nenhum conjunto de pesos. Os piores resultados, para ambas as abordagens, registam-se no problema em que se considera a existência de restrições de capacidade. Tal resultado está de acordo com os resultados obtidos para o caso mono-objectivo. O algoritmo memético apresentado pode ser utilizado com qualquer número de objectivos. No entanto, quando se considerem mais do que dois objectivos, a interação com o decisor não pode ser baseada apenas em gráficos bidimensionais.

		TNS	DS	ES	MA_ED	AED	MI_ED	SM
Problema com um nível e sem capacidades	RPS	417	32		5518.32	445.09	0.00	0.97
	UBS	134	70	3	7993.00	888.64	1.41	1.03
Problema com um nível, com capacidades	RPS	47	0		14973.90	9927.05	3373.00	0.48
	UBS	32	24	8	17911.70	11038.05	7082.50	0.83
Problema com dois níveis e sem capacidades	RPS	97	2		1623.28	65.85	0.00	0.59
	UBS	30	11	9	1811.2	549.16	0.00	0.62

Tabela 4.7 – Resultados computacionais

Legenda:

RPS: Soluções geradas pela abordagem baseada em pontos de referência;

UBS: Soluções geradas pela abordagem baseada na definição de limites superiores e/ou inferiores;

TNS: Número total de soluções calculadas;

DS: Número de soluções dominadas em relação ao outro conjunto (RPS é comparado com UBS e vice-versa);

ES: Número de soluções que aparecem simultaneamente em UBS e RPS;

AED, MA_ED, MI_ED: Distâncias euclidianas máxima, média e mínima entre a solução candidata a não-dominada e a fronteira de ótimos de Pareto;

SM: *spacing metric* (Collette & Siarry, 2005).

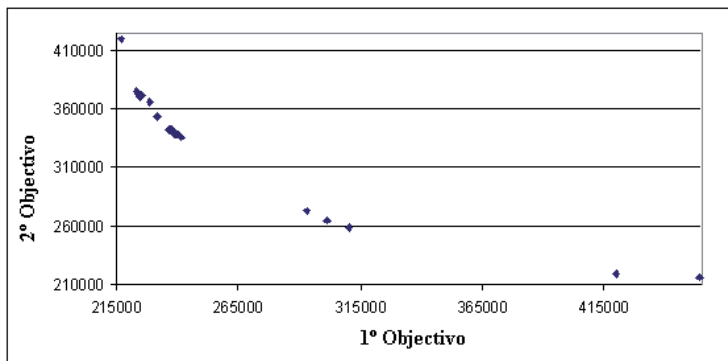


Figura 4.6 – Primeiro conjunto de soluções.

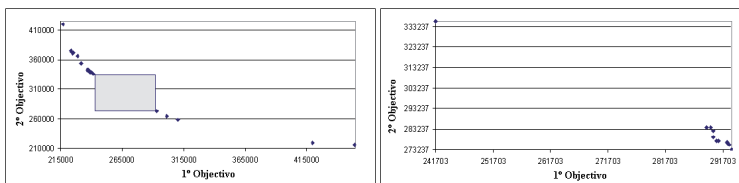


Figura 4.7 – Cálculo de soluções na área seleccionada.

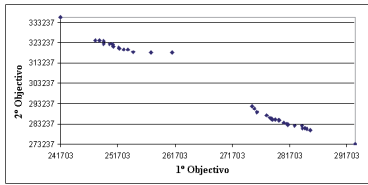


Figura 4.8 – Novas soluções após duas interações com o decisor.

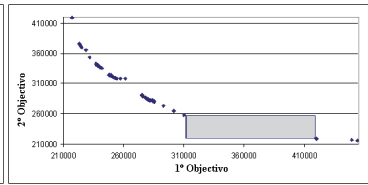


Figura 4.9 – Definição de uma nova área de interesse.

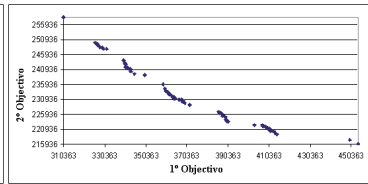
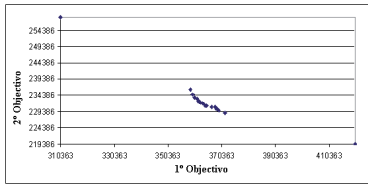


Figura 4.10 – Novas soluções na área de interesse após 1 e 5 interações.

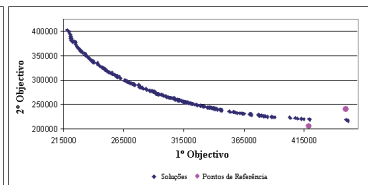
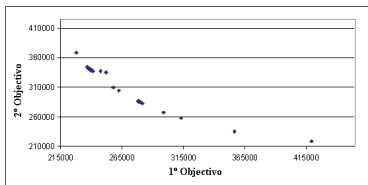


Figura 4.11 – Primeiro conjunto de soluções e definição dos pontos de referência.

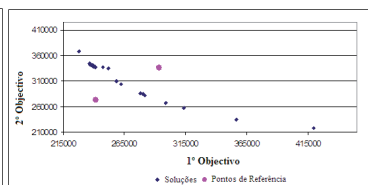
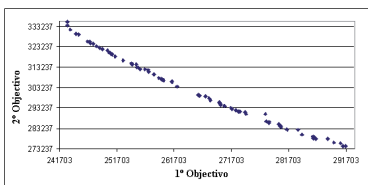


Figura 4.12 – Cálculo de novas soluções e definição de novos pontos de referência.

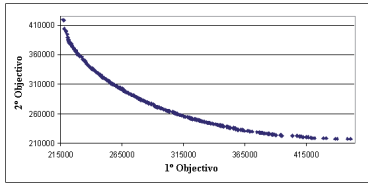


Figura 4.13 – Soluções calculadas através da definição de pontos de referência.

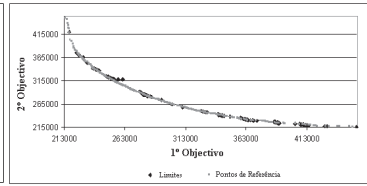


Figura 4.14 – Comparação com o conjunto calculado pela definição de limites superiores/inferiores.

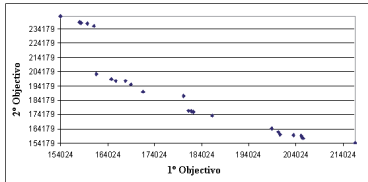


Figura 4.15 – Primeiro conjunto de soluções.

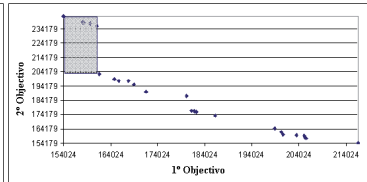


Figura 4.16 – Definição de área de interesse.

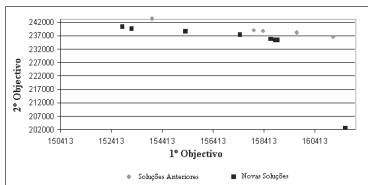


Figura 4.17 – Comparação entre as soluções existentes e as soluções calculadas.

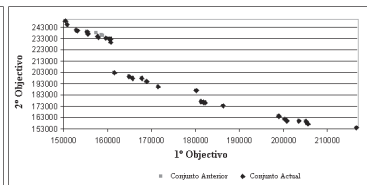


Figura 4.18 – Após 2 interações, o conjunto de soluções sofre poucas alterações.

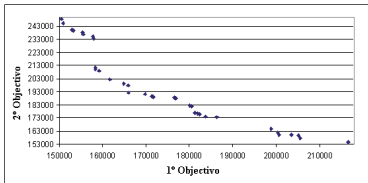


Figura 4.19 – Primeiro conjunto de soluções.

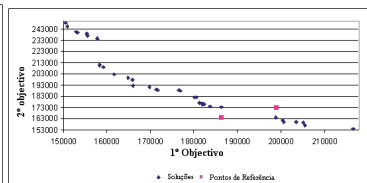


Figura 4.20 – Definição de pontos de referência.

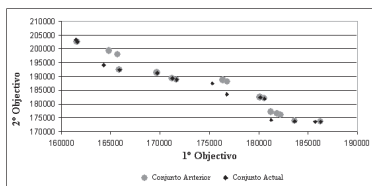


Figura 4.21 – Novo conjunto de soluções.

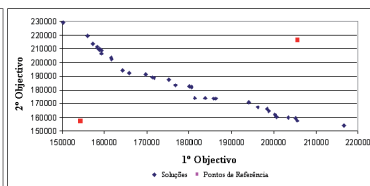


Figura 4.22 – Definição de novos pontos de referência.

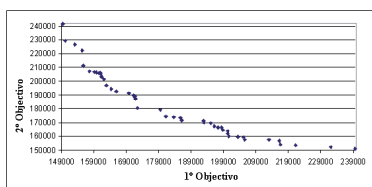


Figura 4.23 – Conjunto final de soluções.

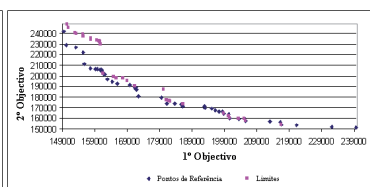


Figura 4.24 – Comparação entre os 2 conjuntos.

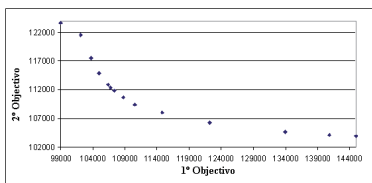


Figura 4.25 – Primeiro conjunto de soluções.

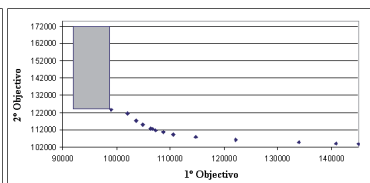


Figura 4.26 – Definição da área de interesse.

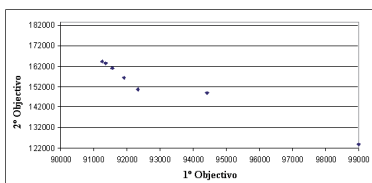


Figura 4.27 – Soluções calculadas na área de interesse.

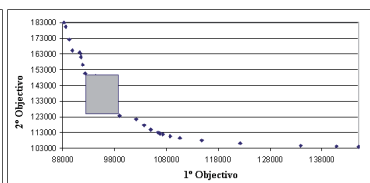


Figura 4.28 – Nova área de pesquisa.

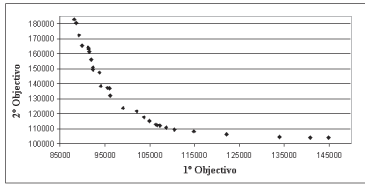


Figura 4.29 – Conjunto final de soluções.

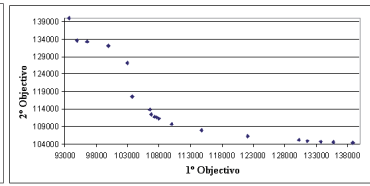


Figura 4.30 – Primeiro conjunto de soluções para a abordagem baseada em pontos de referência.

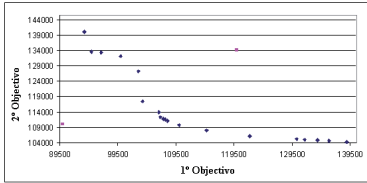


Figura 4.31 – Pontos de referência.

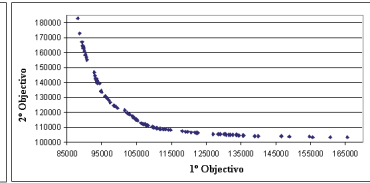


Figura 4.32 – Conjunto final de soluções.

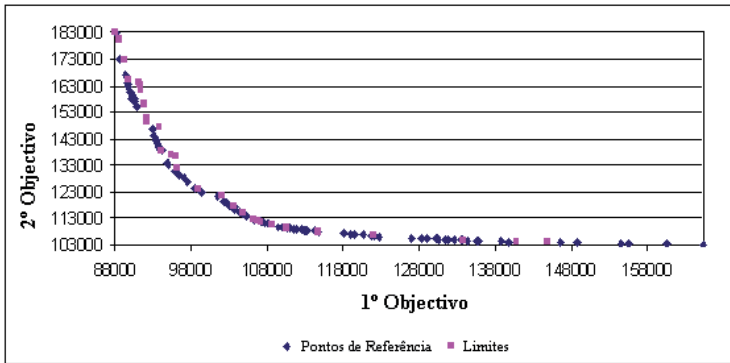


Figura 4.33 – Comparação entre os dois conjuntos de soluções.

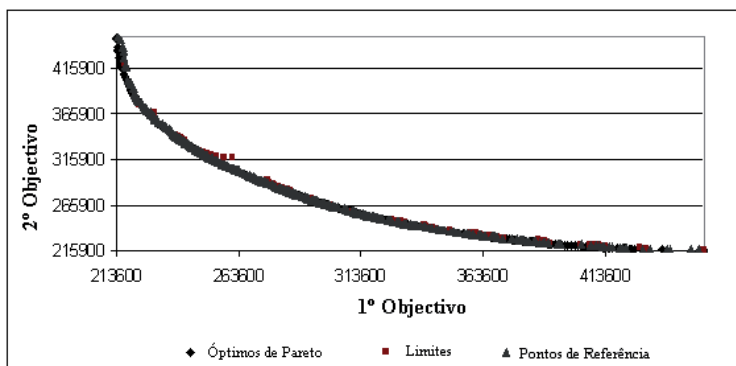


Figura 4.34 – Problema de localização com um nível e sem capacidades: comparação dos diferentes conjuntos de soluções.

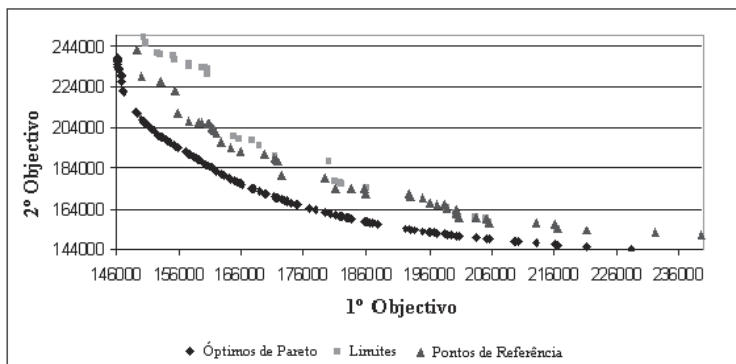


Figura 4.35 – Problema de localização com um nível e restrições de capacidade máxima: comparação entre os conjuntos de soluções.

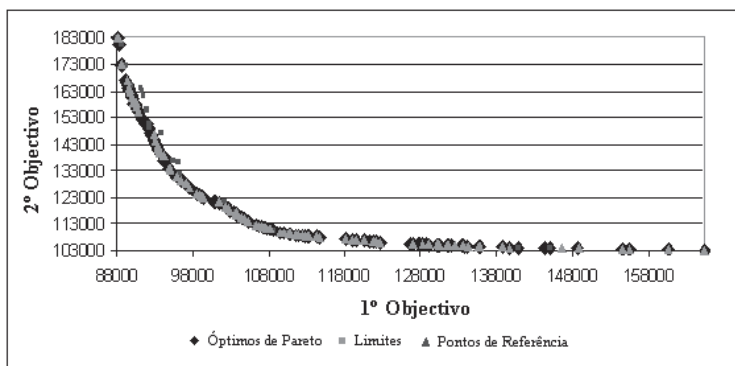


Figura 4.36 – Problema de localização de dois níveis e sem capacidades: comparação dos diferentes conjuntos de soluções.

(Página deixada propositadamente em branco)

Capítulo 5

CONCEPÇÃO E ESTRUTURAÇÃO DO SISTEMA DE APOIO À DECISÃO

Neste capítulo descreve-se, de forma sucinta, um possível sistema de apoio à decisão, capaz de integrar todos os modelos e algoritmos desenvolvidos ao longo deste trabalho. Descreve-se de que modo as diferentes abordagens apresentadas se podem complementar. Aborda-se a questão de se considerarem vários agentes de decisão, em lugar de um único. Considera-se que, num ambiente multi-decisor, há vários aspectos a considerar que não se limitam à utilização de modelos e de algoritmos, nem se resolvem apenas com a consideração de mais do que um objectivo (Joana Dias, Captivo & Clímaco, 2006).

Decision Support is not about tools per se but rather about making better decisions.

Alter, 2004

5.1. Introdução

Um problema de decisão pode nascer de uma situação em que um indivíduo, ou grupo de indivíduos, detecta a diferença existente entre o estado actual e o estado desejado. Deste modo, um problema pode emergir devido a alterações ocorridas no mundo real, mas também resultar de mudanças nos valores, percepções, preferências, convicções que se têm desse mundo real. Uma decisão é um comprometimento em agir, que conta com afectação de recursos. O objectivo primeiro de um processo de decisão é transformar informação, que constitui o *input*, em decisão, que é o *output* do processo (Janssen, 1992). As decisões têm de ser tomadas cada vez mais rapidamente, com mais frequência, e em contextos cada vez mais complexos e competitivos (Jessup & Valacich, 1992).

E o que será uma boa decisão? Aquela que leva a bons resultados? Winterfeldt & Edwards, 1986, defendem que a qualidade de uma determinada decisão é dada pela qualidade do processo que levou a essa decisão, e que tem em conta factores como a quantidade e qualidade da informação disponível. Se, em situações ideais, a informação disponível pode ser precisa, conhecida, exaustiva e objectiva, na realidade a informação disponível é insuficiente e tem um elevado grau de incerteza (Munda, Nijkamp & Rietveld, 1995; Nijkamp & Bergh, 1997).

Um sistema de apoio à decisão pode ser definido como um sistema informático interactivo, que ajuda à resolução de problemas muitas vezes mal

definidos, considerando um ou vários decisores que podem, ou não, trabalhar em equipa. Na realidade, em muitas situações de decisão em grupo, os agentes de decisão têm objectivos individuais que entram em conflito com objectivos de outros agentes de decisão. Assim, muitas vezes, o processo de apoio à decisão torna-se um processo de negociação e de procura de consensos. Na maior parte das situações, os participantes na tomada de decisão estão simultaneamente motivados para cooperar e competir uns com os outros (Anson & Jelassi, 1990). Segundo Bennett & Howard, 1996:

Conflict and cooperation become two sides of the same coin and cannot satisfactorily be dealt with separately.

O desacordo entre elementos de um grupo que tem de tomar uma decisão nem sempre deve ser visto como uma desvantagem. Na realidade, alguns estudos apontam estes desacordos como sendo vantajosos: podem levar a que os juízos acerca das várias alternativas possíveis sejam mais acertados do que os feitos por decisores individuais (Brill, Flach, Hopkins & Ranjithan, 1990). Quando a tarefa mais importante no processo de tomada de decisão é, na realidade, conseguir encontrar um consenso, então devem utilizar-se sistemas particularmente preocupados com o apoio à negociação e à resolução de conflitos: sistemas de apoio à negociação⁵⁰. Os processos de decisão em grupo são, actualmente, dos processos mais importantes e mais frequentemente encontrados em empresas e organizações públicas ou privadas (Matsatsinis & Samaras, 2001). Se se juntar a esta constatação o facto de que os gestores e profissionais passam cerca de 50% a 60% do seu tempo em reuniões que consideram ser, na maior parte das vezes, improdutivas e ineficientes (Pinsonneault & Kraemer, 1990), facilmente se concluirá sobre o efeito positivo que sistemas de apoio à decisão em grupo poderão trazer. Podem apontar-se quatro propriedades dos problemas de decisão em grupo que os tornam especialmente difíceis: a) são problemas sociais, mais do que matemáticos ou científicos; 2) é muito difícil satisfazer todas as restrições; c) são mais difíceis de modelar; d) existem poucas metodologias que permitam aferir a justeza, conceito com estreitas ligações à agregação de preferências (Matsatsinis & Samaras, 2001).

Porque é que os indivíduos em conjunto são diferentes daquilo que seriam isoladamente, a ponto de não ser possível prever a suas reacções a partir do momento em

⁵⁰ A estruturação de um sistema de apoio à negociação está para além dos objectivos deste capítulo. Os processos negociais revestem-se de algumas características específicas que os tornam um caso especial dos sistemas de apoio à decisão em geral (Raiffa, 1982). K. P. Sycara, 1990, é da opinião que *Negotiation is an ill-structured and complex process, that to date has defied all attempts at analysis. The outcome of the process depends on such intangibles as the negotiators' skills and experience, uncertain and changing information, the parties' perceptions and idiosyncratic behaviours and on the exhaustive and systematic analysis of the problem. The values that enter the search for a suitable compromise are so many that negotiators themselves have not been able to elucidate a well-defined procedure that a negotiator can follow.*

que estão integradas numa multidão, numa reunião, etc.? Este é o enigma que continua a espantar as nossas ciências humanas desde o seu início.

Moscovici & Doise, 1991

Wetmore III & Summers, 2003, afirmam mesmo que a decisão em grupo não deve ser considerada quando não existem objectivos quantificáveis, que possam ser avaliados independentemente das preferências pessoais. Neste trabalho avaliam-se um conjunto de questões que influenciam, de forma determinante, o processo de decisão em grupo, como sejam, a influência das preferências individuais, a pressão do tempo, a dinâmica de grupo⁵¹.

Um sistema de apoio à decisão não pode, em caso algum, ser visto como substituto do agente de decisão. São sistemas construídos com o intuito de apoiar e ajudar o agente de decisão na sua tarefa de tomar decisões responsáveis e fundamentadas, utilizando, da melhor maneira possível, todo o potencial disponível.

In any phase of a decision process, a DSS must not replace the user in their sovereign decision making: it only helps them in various tasks. The main aim of a DSS is to enhance the creativity and intuition of the user rather than to automate the decision making.

Makowski & Wierzbicki, 2000

Mendes *et al.*, 1998, partilham da opinião que os sistemas de apoio à decisão devem ser utilizados, não só como meio de justificar as opções tomadas, mas, principalmente, como meio de explorar alternativas e melhorar o processo de tomada de decisão. Â. G. Pereira & Quintana, 2002, defendem mesmo que os sistemas de apoio à decisão devem ser, cada vez mais, vistos como processos de apoio e de iniciação ao debate, devendo passar a ser designados como TIDDD: *tools to inform debates, dialogues and deliberations*. Neste trabalho, os autores analisam as alterações sentidas no desenvolvimento de sistemas de apoio à decisão no JRC (*Joint Research Center*), onde se passou de uma visão puramente tecnocrática e de apoio a especialistas, para sistemas distribuídos, com a participação de todos os actores na tomada de decisão.

Os actores que, na actualidade, entram em cena num conjunto vasto de tomadas de decisão são os *cidadãos*. A *Convention on Access to Information, Public Participation in Decision-Making and Access to Justice in Environmental Matter*, assinada em Aarhus, Dinamarca, em 1998, confere aos cidadãos o acesso à informação e o direito à participação nas tomadas de decisão em todos os assuntos que tenham implicações ambientais, defendendo que esta participação pública melhora a qualidade das decisões e facilita a sua implementação. Também Mika & Hämäläinen, 1995, defendem que as decisões poderão ser melhoradas, aumentando a participação pública, e Frade, 1999, afirma que a coordenação institucional e política entre os vários níveis institucionais, que

⁵¹ Estas questões, sem dúvida fundamentais no estudo dos fenómenos relacionados com a decisão em grupo, não se inserem no âmbito deste capítulo. Consultar, também, Moscovici & Doise, 1991.

tomam parte na tomada de decisão, deve ser reforçada pela participação dos próprios cidadãos. Hoje em dia, os cidadãos não aceitam uma determinada decisão, que os possa afectar, se tomada apenas com base em justificações fundamentadas na opinião de especialistas ou de resultados científicos. Gonçalves, 2000, descreve da seguinte forma a atitude dos cidadãos:

A crença na ciência como única fonte legítima do saber dominante no mundo ocidental tem vindo contudo a erodir-se sob o efeito combinado da multiplicação de controvérsias públicas amplamente mediatizadas, particularmente em domínios como o ambiente e a saúde pública, e da percepção, cada vez mais generalizada, de que os peritos divergem entre si, que o acesso a metodologias rigorosas e objectivas não lhes dá o dom da infalibilidade, e que a sua opinião, pretensamente desinteressada, não é dissociável de considerações profissionais, económicas ou políticas.

Por este motivo, é cada vez mais importante que os cidadãos sejam participantes de pleno direito na tomada de decisões, e que as decisões tomadas reflectam processos de decisão transparentes, construídos em bases sólidas. É politicamente aconselhável que todos os argumentos que suportam uma determinada decisão sejam bem definidos e explicados antes de se pôr em prática a alternativa escolhida, até porque tudo pode depender dos argumentos serem, ou não, efectivamente convincentes (Nutt, 2000). Um sistema de apoio à decisão não deve servir apenas para validar decisões tomadas *a priori*. Deve ser utilizado, por todos os actores envolvidos no processo, como uma ferramenta útil na procura da “melhor” solução. O que institui o consenso e o torna convincente não é o acordo, mas a participação daqueles que o concluíram (Moscovici & Doise, 1991). Espera-se que os diferendos entre os indivíduos se resolvam mediante concessões que os aproximam da média das suas posições e os afastam dos extremos.⁵²

Segundo Makowski & Wierzbicki, 2000, um sistema de apoio à decisão tem, pelo menos, três componentes principais: uma base de dados, uma base de modelos e um interface. No entanto, há dois componentes adicionais que são essenciais na estrutura actual de um sistema de apoio à decisão. Um deles, essencial quando se utilizam modelos analíticos, é uma biblioteca de algoritmos (uma base de algoritmos). A outra componente é a responsável pela organização de todo o processo de decisão. Este módulo de organização coordena todas as outras componentes do sistema.

Os processos de decisão têm um carácter intrinsecamente dinâmico. Isto obriga a que os sistemas de apoio à decisão sejam flexíveis, e permitam que o decisor se coloque em qualquer fase do processo de decisão, podendo voltar atrás, mudar de opinião, fazer novas escolhas.

E em que momento se inicia o processo da tomada de decisão? Quando existe um conjunto limitado de opções, entre as quais será preciso escolher

⁵² No entanto, a discussão e a exposição de ideias também os pode levar a atitudes mais extremas (Moscovici & Doise, 1991).

apenas uma? Ou será que começa no momento em que há a consciência de que é preciso tomar a decisão? Será que a escolha de um determinado modelo matemático, ou de um algoritmo de resolução, não pertencem já ao processo de escolha de uma solução? Se estas fases iniciais e preparatórias são, por vezes, tratadas separadamente devem ser sempre consideradas como parte integrante do processo de tomada de decisão, e deve-lhes ser reconhecida a sua real importância que têm, pois podem influenciar, de maneira decisiva, todo o processo daí para a frente.

Most common pitfall of analysis: producing a sophisticated solution to the wrong problem!

Winterfeldt & Edwards, 1986

Weick & Meader, 1992, apontam diferenças entre a maneira oriental e ocidental de interpretar o conceito de “tomar uma decisão”. No ocidente toda a ênfase é colocada na resposta à questão. No oriente, o elemento mais importante é a definição de qual será a questão. Os passos cruciais são os que decidem se há realmente necessidade de se tomar uma decisão, e sobre a substância dessa decisão. É na formulação da pergunta que se deve investir para encontrar consensos.

5.2. Sistema de Apoio à Decisão para a Resolução de Problemas de Localização

5.2.1. Base de Dados e Base de Modelos

Quando se considera a possibilidade da existência de mais do que um agente de decisão, é normalmente considerado que todos os agentes de decisão trabalham sobre o mesmo modelo, e que todos estão de acordo em que o modelo escolhido é o que melhor representa o problema real em causa. Não será este um pressuposto demasiado forte? Não será mais razoável e mais realista assumir que há um conjunto de características do problema reconhecidas e aceites por todos os intervenientes, e um conjunto de questões, algumas das quais correspondendo a interpretações subjectivas do problema, que não poderão, à partida, e de forma consensual, ser inseridas num modelo único? A escolha de um modelo matemático único que sirva de base ao cálculo de possíveis soluções para o problema é, em si mesma, uma escolha difícil e faz parte integrante do processo de decisão. Diferentes participantes na tomada de decisão podem formular e pensar o mesmo problema de forma distinta. Winterfeldt & Edwards, 1986, defendem que cada participante tem as suas opções e objectivos, e que a estruturação do problema é, de longe, o passo mais importante no processo de tomada de decisão.

Desta forma, e tendo como base os modelos matemáticos definidos no capítulo 2 do presente trabalho, considera-se que, na existência de vários agentes de decisão, estes deverão concordar nos aspectos fundamentais do problema, limitando-se a determinar: as possíveis localizações para os serviços;

o número e a localização de clientes; o número de períodos temporais e definição de período e horizonte temporais; as capacidades dos serviços e ofertas/procuras dos clientes (se aplicável). A partir desta base comum de trabalho, o decisor deve ser livre de escolher as suas próprias funções objectivo e restrições adicionais ao problema. Estas podem reflectir opções como, por exemplo, impedir um determinado serviço de estar aberto em determinado período temporal, ou obrigar à abertura de um serviço. Cada decisor poderá gerar soluções para o problema, que têm em conta apenas a sua visão pessoal. O decisor é também livre de dar, ou não, a conhecer ao grupo o modelo com que está a trabalhar. Convivem, desta forma, vários modelos: um modelo por cada decisor e um modelo de grupo, onde figuram apenas as características do problema que geram consensos entre os participantes.

Ao longo do processo de tomada de decisão, os modelos utilizados por cada agente de decisão pode sofrer alterações. Para que se consiga chegar a um consenso é muitas vezes necessário reestruturar o problema sobre o qual se está a trabalhar (K. Sycara, 1991). Essa reestruturação pode incluir a introdução, substituição ou remoção de objectivos e restrições. Aquilo que não era uma solução admissível numa determinada altura, pode tornar-se uma solução aceitável noutra altura do processo (K. P. Sycara, 1990). O ideal seria que os vários modelos dessem lugar a um único modelo consensual. No entanto, se tal não for alcançado, o processo de tomada de decisão continuará a desenrolar-se.

A base de dados do sistema de apoio à decisão terá de conter todos os dados necessários ao(s) modelo(s) matemático(s) escolhido(s) pelos agentes de decisão. Deverá ainda conter informação complementar como, por exemplo, legislação aplicável ou pareceres de peritos. Tais informações devem ser introduzidas no processo de decisão, mas não devem ser confundidas com o processo de procura de soluções, que deve ser deixada unicamente a cargo dos agentes de decisão. Os problemas de localização podem ser, em muitos casos, interpretados de um ponto de vista mais amplo como pertencendo a uma classe de problemas de ordenamento de território. A informação que deve estar, em cada momento, disponível no sistema dependerá essencialmente do problema particular que se pretende resolver.

Também não se deve descurar a importância que os sistemas de informação geográfica devem ter na ajuda à tomada de decisão na área dos problemas de localização. Os sistemas de informação geográfica poderão ser uma importante ferramenta na obtenção e organização de dados, para aplicação de modelos de localização (Church, 2002; A. G. Pereira, Munda & Paruccini, 1994; A. G. Pereira, Peckham & Antunes, 1993; Dibble & Densham, 1993; Jungthirapanich & Pratheepthaweophon, 1998).

Estudos mostram que os decisores podem alterar as suas preferências em resposta à maneira como a informação lhes é apresentada e aos argumentos defendidos por outros participantes no processo de decisão (King & Kugler, 2000). Nos problemas em grupo, para além do decisor poder não ter a certeza acerca dos seus próprios valores e preferências, estará ainda mais incerto acerca dos valores e preferências dos outros participantes (Mumpower, 1991). É, por isso, muito importante, que o sistema tenha capacidade para armazenar todos os

dados relevantes que se vão produzindo durante o processo de tomada de decisão. Deste modo, devem ser armazenadas informações relativas à evolução das preferências e opções de cada agente de decisão (que correspondem ao histórico de cada interveniente), e que permitam que o processo de tomada de decisão seja parado e retomado em qualquer momento, sem perdas de informação (Valacich, Vogel & Jr, 1989).

5.2.2. Base de Algoritmos

Um grupo de indivíduos que pretende resolver um dado problema pode ser visto como um grupo que procura uma solução óptima, ou de muito boa qualidade, num determinado espaço de soluções de elevada complexidade (Rees & Koehler, 2004). Para que seja capaz de realizar esta árdua tarefa, o grupo deve contar com a ajuda de algoritmos.

Makowski & Wierzbicki, 2000, defendem que, para que um algoritmo seja incluído num sistema de apoio à decisão, tem de ser robusto. Com isto pretende-se dizer que o algoritmo deve funcionar correctamente, e de forma eficiente, para um elevado número de instâncias de um determinado modelo e, no caso de falhar em instâncias mais difíceis, não deve deixar o decisor sem pistas para saber o que fazer a seguir. Segundo os autores:

Optimization should be treated as a reliable tool and not as an abstract goal in a decision support system.

A base de algoritmos de um sistema de apoio à decisão deve ter disponível, não só um conjunto de algoritmos com características diferentes, mas também ferramentas que permitam ao utilizador passar de um algoritmo para outro, aproveitando ao máximo a informação obtida com a execução de algoritmos diferentes.

No caso particular dos modelos e algoritmos desenvolvidos nos capítulos precedentes, como se poderá levar a cabo a interligação entre os diferentes métodos de resolução? E como escolher qual utilizar em cada situação? Como exemplo, pode dizer-se que (Figura 5.1, Figura 5.2):

- ⊕ Se o problema de localização é formulado, tendo como base um dos modelos desenvolvidos no capítulo 2, sem a existência de restrições adicionais, e considerando apenas uma função objectivo, então dever-se-á começar por utilizar uma das heurísticas primais-duais desenvolvidas. Após a execução da heurística, pode encontrar-se a solução óptima, caso tal seja pretendido, recorrendo, por exemplo, a um *general solver*, cuja execução será otimizada se se fornecer a este algoritmo a informação obtida com a heurística (como, por exemplo, um limite inferior válido para o valor óptimo da função objectivo). Este *general solver* poderá, também, ser utilizado se se pretender introduzir restrições adicionais.
- ⊕ Se o problema de localização é formulado com base num dos modelos desenvolvidos no capítulo 2, considerando a existência de restrições adicionais, e apenas um objectivo, então poder-se-á optar pela utilização de um *general solver* ou pela utilização de metaheurísticas. O algoritmo memético desenvolvido apresenta vantagens em relação ao *general solver* nas situações em que existe mais do que um agente de decisão. Neste caso, enquanto que o *general solver* apresenta como *output* apenas uma solução, o algoritmo memético apresenta um conjunto de soluções, o

que pode levar a um maior conhecimento, por parte do decisor, das características do problema, e, logo, potenciar consensos.

⊕ Se o problema de localização considera múltiplos objectivos, serão várias as abordagens possíveis: se se aceitar a limitação de não conhecer soluções não-dominadas não-suportadas, então podem executar-se as heurísticas primais-duais, considerando como função objectivo somas ponderadas com ponderações diferentes das várias funções objectivo do problema. Pode, de igual modo, optar-se pela utilização de um *general solver*. Neste caso, será possível encontrar mesmo as soluções não-dominadas não-suportadas (incluindo, por exemplo, um conjunto de restrições adicionais). A utilização do algoritmo memético tem, mais uma vez, a vantagem de apresentar uma família de soluções. O *general solver* poderá ser sempre utilizado como ferramenta auxiliar, que ajuda a confirmar se as soluções geradas pelo algoritmo candidatas a não-dominadas, o são ou não.

Qualquer que seja o sistema de apoio à decisão ou os algoritmos utilizados, deverá sempre assumir-se que as soluções provenientes de modelos matemáticos só muito raramente representam as melhores soluções para problemas complexos. Na realidade, elas representam apenas dados sobre os quais o decisor trabalha, incorporando informações provenientes de outras fontes, antes de tomar a sua decisão (Brill *et al.*, 1990). É, por isso, vantajoso, que várias soluções sejam calculadas utilizando algoritmos diferentes. É ainda importante a existência de algoritmos dedicados a realizar uma análise de sensibilidade e robustez das soluções à variação de um ou mais parâmetros.

Como se considera possível que vários decisores trabalhem sobre diferentes modelos, deve também ser tida em conta a existência de algoritmos que analisem as relações existentes entre estes vários modelos, encontrando consensos ou conflitos. Um decisor pode não querer dar a conhecer totalmente a sua visão do problema aos outros decisores. O sistema, no entanto, conhece, pelo menos, uma parte dessa visão.

Sendo assim, o sistema pode encontrar grupos de decisores que decidam incluir restrições ao problema que impossibilitem a descoberta de soluções admissíveis para todos, motivando a discussão em torno destas restrições. Também será possível desenvolver heurísticas especializadas que permitam encontrar soluções de compromisso “entre” soluções diferentes e aparentemente inconciliáveis (técnicas de optimização de *path relinking*, por exemplo, são ferramentas que imediatamente aparecem como indicadas para realizarem esta função), ou mesmo a criação de um “macro algoritmo genético”, que, em vez de se preocupar em encontrar soluções para um dado problema, se preocupa em trabalhar com soluções geradas por vários outros algoritmos, tentando maximizar o consenso entre decisores.

A ideia de utilizar algoritmos genéticos como ajuda à tomada de decisão em grupo aparece naturalmente, se se pensar neste processo como um processo evolutivo, em que o grupo vai sofrendo alterações. Rees & Koehler, 2002, propõem mesmo que o processo de decisão em grupo possa ser modelado por um algoritmo genético, utilizando selecção, mutação e *crossover*.

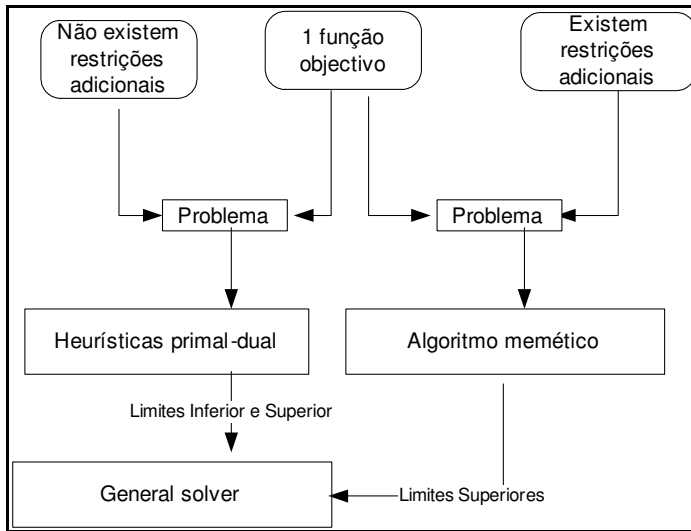


Figura 5.1 – Proposta de interação entre algoritmos – caso mono-objectivo

Yuhong *et al.*, 1997, fazem uso de algoritmos genéticos na resolução de conflitos, em que cada indivíduo representa uma possível solução para o problema e é avaliado por cada agente de decisão. As propostas e contra-propostas dos vários decisores podem ser inseridas na população, que evolui normalmente, utilizando os operadores habituais. Matwin, Szapiro & Haigh, 1991, desenvolvem um algoritmo genético para ser utilizado por uma das partes de um processo de negociação, algoritmo este que codifica regras de negociação e que ajuda o decisor na construção de contra-propostas.

Após se ter chegado a um grupo finito, e em número razoável, de alternativas possíveis, várias metodologias devem ainda ser aplicadas para que se possa escolher a solução. Nesta fase, reveste-se de particular importância a análise multi-critério, os modelos de argumentação e os modelos de simulação (Allett, 1986; Chi, Riordan & Bejamin, 1996; Karacapilidis, Papadias, Gordon & Voss, 1997; Easley, Valacich & Venkataramanan, 2000; Kim & Ahn, 1997; Matsatsinis & Samaras, 2001; Harper, Shahani, Gallagher & Bowie, 2005).

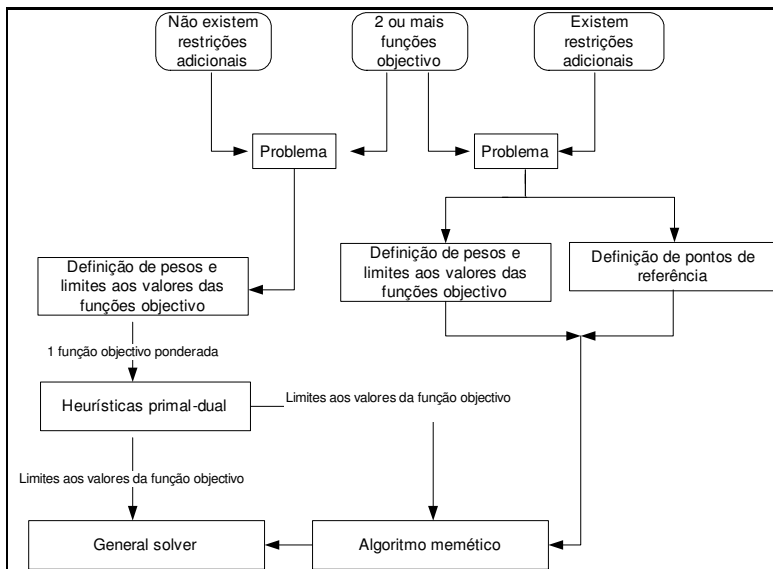


Figura 5.2 – Proposta de interacção entre algoritmos – caso multi-objectivo

5.2.3. Base de Organização

A base organizacional é responsável pela gestão do sistema de apoio à decisão. O esquema de funcionamento do sistema de apoio à decisão é mostrado na Figura 5.3, e é fundamentalmente constituído por duas áreas distintas: a área individual e a área comum.

Na área individual, cada decisor pode analisar o problema de forma independente dos restantes intervenientes no processo: pode adoptar os seus próprios modelos, calcular soluções, analisá-las, etc. Na área comum desenrolam-se os processos que implicam mais do que um decisor. Estas áreas podem e devem estar em permanente interacção, como se ilustra no esquema.

A interacção entre as áreas individuais e área comum estende-se a todos os componentes do sistema de apoio à decisão. Os dados do problema podem ser comuns a todos os intervenientes, mas considera-se possível a existência de dados conhecidos apenas por algum deles. A qualquer momento pode haver fluxo de dados de uma área para a outra. O mesmo se passa com os modelos matemáticos. Cada decisor pode dar a sua contribuição para aquilo que pensa ser o problema a resolver. As interacções a este nível são de extrema importância, como já foi realçado antes. A procura de consensos é um passo fundamental para a descoberta de soluções de compromisso em fases mais avançadas do processo. A análise dos modelos individuais permite salientar os pontos que podem gerar maior conflituosidade, e encontrar pontos em que grupos de decisores estão em acordo. Mesmo que os decisores nunca cheguem a acordo acerca de um modelo único representativo do problema, podem sempre partilhar soluções que cada um, através do cálculo de soluções baseado

no seu modelo individual, foi descobrindo. A descoberta destas soluções pode, também, levar à alteração do modelo individual ou do modelo comum. A base de algoritmos, constituída por várias técnicas optimizantes, pode igualmente trabalhar sobre o conjunto de soluções disponibilizadas pelos intervenientes e calcular novas soluções, por forma a maximizar os consensos.

Após se ter conseguido construir um conjunto formado por um número reduzido de soluções possíveis, poder-se-á recorrer a diversas técnicas que implicam, mais uma vez, interacção entre os decisores, para que se consiga acordar acerca da solução a escolher.

A Figura 5.4 representa um possível diagrama de fluxo de dados correspondente ao sistema proposto.

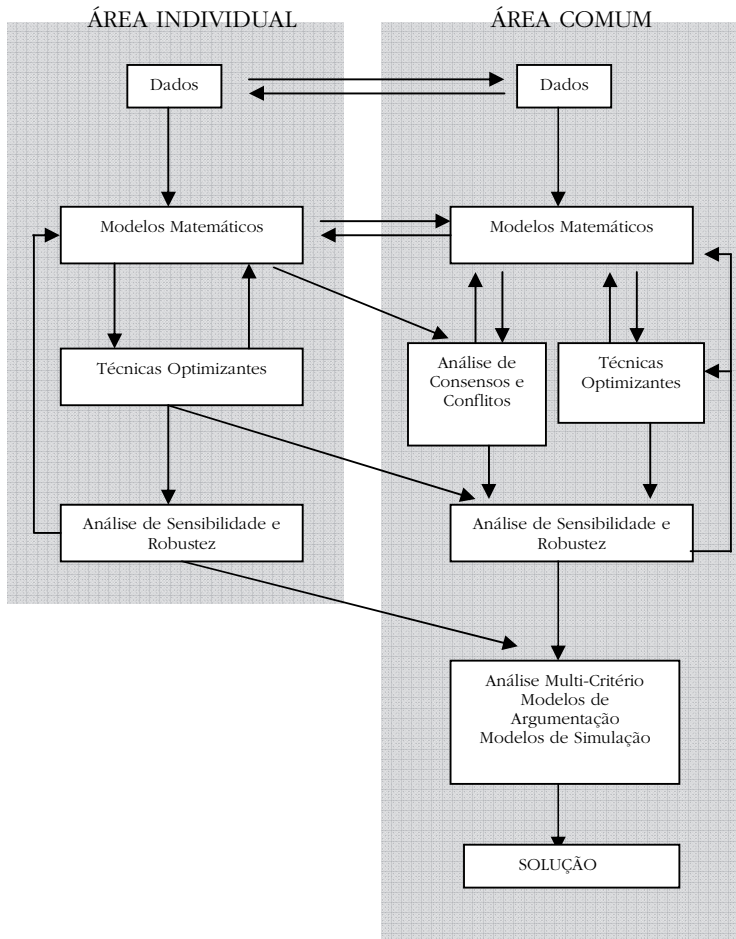


Figura 5.3 – Estrutura do sistema de apoio à decisão

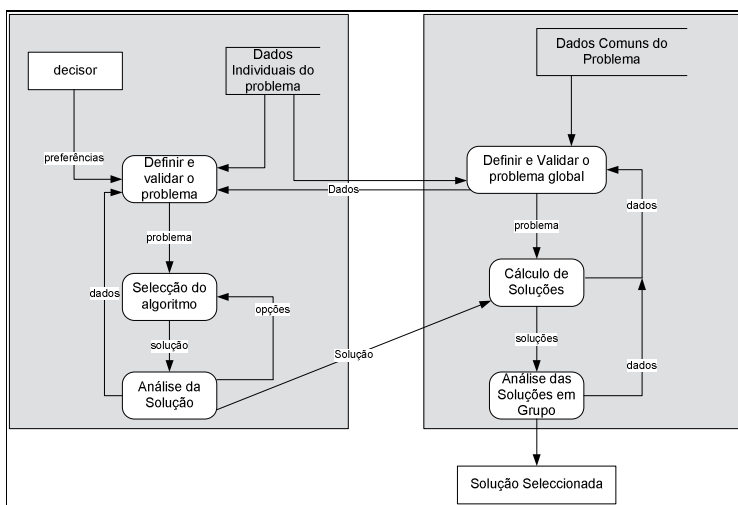


Figura 5.4 – Diagrama de Fluxo de Dados

Capítulo 6

CONCLUSÕES

Os problemas de localização são problemas complexos, em que a tomada de decisão implica normalmente a consideração de vários factores e impactes a curto, médio e longo prazo.

O trabalho desenvolvido e apresentado ao longo deste texto tenta contribuir para o processo de tomada de decisão nestas matérias através da utilização de modelos e de ferramentas de optimização que poderão ser um precioso auxílio na definição e análise do problema, bem como criar uma base de sustentação para as decisões a tomar.

No capítulo introdutório, definiu-se como âmbito deste texto o de contribuir para o estudo dos problemas de localização, dando enfoque ao desenvolvimento de modelos matemáticos e de métodos de resolução desses modelos que possam ser utilizados no apoio à tomada de decisão.

No capítulo 2 apresentou-se um conjunto de modelos de localização dinâmica que generalizam alguns problemas de localização estudados na literatura. Nestes modelos, introduziu-se a possibilidade de um serviço ser aberto, fechado e reaberto mais do que uma vez durante o horizonte temporal considerado; a possibilidade de se considerarem diferentes tipos de restrições de capacidade; a possibilidade de se considerarem problemas com um ou vários níveis de serviços. Os modelos apresentados são mais flexíveis do que outros conhecidos da literatura, e obviamente passíveis de serem melhorados: podem incorporar-se restrições adicionais que consideram limites máximos e/ou mínimos no número de serviços operacionais em cada período temporal; restrições relacionadas com valores orçamentais, que seja imperativo cumprir; restrições relacionadas com gestão de stocks; etc. No entanto, a mais-valia dos modelos desenvolvidos está, não só na generalização conseguida, mas também na possibilidade de se desenvolverem procedimentos heurísticos capazes de encontrarem soluções de boa qualidade, em tempos computacionais aceitáveis. Na realidade, o desenvolvimento de modelos mais complexos, sem o desenvolvimento paralelo de algoritmos capazes de os tornar úteis, apresenta uma mais-valia reduzida.

As heurísticas descritas no capítulo 3, baseadas em conceitos muito simples (a relação entre os problemas primal e dual), são, na sua maioria, meios eficientes de cálculo de soluções admissíveis. No entanto, poderão ser

melhoradas em alguns aspectos: as heurísticas que resolvem problemas com capacidades mínimas têm de considerar, como caso especial, a situação em que as capacidades máxima e mínima apresentam valores muito semelhantes. Este é o principal factor que justifica os resultados menos bons obtidos em algumas instâncias do problema. Também o modelo que considera a possibilidade de se abrirem vários equipamentos numa dada localidade merece atenção especial. Vale a pena estudar com mais profundidade os problemas de afectação resultantes da introdução de capacidades máximas decrescentes, e os que resultam dos problemas com mais do que um nível, sem conservação de fluxo em nodos intermédios. Conseguir resolver eficientemente, de forma heurística ou exacta, estes problemas de afectação, terá um efeito preponderante na qualidade das heurísticas que se possam vir a desenvolver.

No capítulo 3 todos os procedimentos são descritos formalmente, o que torna acessível a sua programação a qualquer leitor deste trabalho. A autora disponibiliza-se para fornecer quer o código fonte, quer os programas executáveis que foram desenvolvidos e que estiveram na base de todos os resultados computacionais aqui apresentados. Deverá, contudo, deixar-se aqui uma observação importante: no desenvolvimento dos vários programas que foram necessários para obter todos os resultados computacionais mostrados, não houve nenhuma preocupação especial em que o código fosse optimizado em termos de tempo ou de memória necessária⁵³. A preocupação principal era a de verificar se as heurísticas apresentavam um bom comportamento e se podiam ser consideradas alternativas viáveis à utilização de um *general solver*. É, por isso, possível que se consigam obter códigos mais eficientes que permitam resolver as instâncias de menor dimensão mais rapidamente, ou resolver instâncias de maior dimensão.

Na capítulo 4, a que foi dado o nome de *Metabeurísticas*, descreve-se um algoritmo memético capaz de resolver qualquer dos modelos desenvolvidos no capítulo 2, e mesmo um modelo mais geral em que se podem considerar, simultaneamente, serviços com diferentes características. A conclusão mais importante a retirar do trabalho efectuado nesta vertente do seu desenvolvimento é a de que vale a pena fazer muito mais... O algoritmo memético é capaz de encontrar soluções de boa qualidade para alguns dos problemas, mas apresenta tempos computacionais demasiadamente elevados, mesmo quando comparados com o *general solver*. Ora, o mais difícil era acreditar que seria possível encontrar boas soluções, utilizando um processo que está, basicamente, assente em acontecimentos aleatórios. Este cepticismo inicial era ainda reforçado por todo o trabalho anterior, praticamente antagónico a este: desenvolvimento de procedimentos muito bem estruturados, assentes em regras relativamente rígidas. Tudo aquilo que foi feito suscita o interesse de fazer muito mais. Tirar realmente partido das sinérgias de ambas as metodologias permitirá chegar a algoritmos mais flexíveis do que as heurísticas primais-duais e também mais eficientes do que o algoritmo memético. Interessa estudar novas maneiras

⁵³ Em qualquer programa computacional, deve sempre tentar encontrar-se o melhor compromisso para o binómio (tempo, memória): programas mais rápidos conseguem-no através de maior ocupação de memória e vice-versa.

de codificar soluções, de inicializar populações, de evoluir gerações, integrando todo o conhecimento adquirido durante o estudo dos modelos e construção das heurísticas primais-duais.

No capítulo 5 esboça-se, de forma sumária, o esquema de um possível sistema de apoio à decisão para problemas de localização. Na realidade, se se almeja ver, um dia, estes modelos e metodologias a serem aplicados a problemas reais, tem de se apresentar ao utilizador final uma ferramenta fácil e agradável de utilizar (e, para o utilizador, o interface *é* a ferramenta). Este capítulo pretende mostrar igualmente como todas as metodologias se podem complementar, mesmo parecendo, por vezes, inconciliáveis. Com o aumento crescente da complexidade dos problemas aos quais se tem de dar solução, interessa, sobretudo, conhecer todas as armas disponíveis, usá-las o melhor possível e, sempre que necessário, em conjunto.

De todos os elementos constituintes do sistema de apoio à decisão apresentado, as contribuições deste trabalho cingem-se à base de modelos e à base algoritmos. Muito há ainda para fazer, tendo particular interesse a exploração dos consensos e conflitos entre decisores, expressos na escolha dos modelos, e a ajuda que processos algorítmicos possam dar na procura de soluções de compromisso. A aplicação conjunta de técnicas de análise multi-critério, modelos de argumentação e modelos de simulação, numa fase final de discussão, é também uma vertente muito interessante de trabalho a desenvolver. As metaheurísticas poderão desempenhar, também nestes campos, um papel principal.

Chegados ao fim deste texto, considera-se que o objectivo foi alcançado, e sente-se a vontade de percorrer outros tantos caminhos de investigação e desenvolvimento. Cumpriu-se uma etapa de uma viagem que ainda agora teve o seu início.

In many ways the work that goes on all around the world developing algorithms for the solution of combinatorial problems is analogous to a large group of children playing. (...) One of the appeals of solving combinatorial optimization problems is that only a small amount of knowledge is required in order to join in the game with the other children and, through experience, one learns to build better houses with the building blocks at one's disposal.

Davis, 1991

BIBLIOGRAFIA

- "If you steal from one author, it is plagiarism; if you steal from many, it is research" Wilson Mizner (1876-1933), Dramaturgo Norte-americano. Citado em "The Legendary Mizners", Alva Johnston, 1953.
- Aardal, K. (1998a). Capacitated Facility Location: Separation Algorithms and Computational Experience. *Mathematical Programming*, 81, 149-175.
- Aardal, K. (1998b). Reformulation of Capacitated Facility Location Problems: How Redundant Information Can Help. *Annals of Operations Research*, 82, 289-308.
- Aardal, K., Labbé, M., Leung, A., & Queyranne, M. (1996). On the Two-Level Uncapacitated Facility Location Problem. *INFORMS Journal on Computing*, 8(3), 289-301.
- Abdinnour-Helm, S. (1998). A Hybrid Heuristic for the Uncapacitated Hub Location Problem. *European Journal of Operational Research*, 106, 489-499.
- Agar, M. C., & Salhi, S. (1998). Lagrangean Heuristics Applied to a Variety of Large Capacitated Plant Location Problems. *Journal of the Operational Research Society*, 49, 1072-1084.
- Ageev, A. A. (2002). Improved approximation algorithms for multilevel facility location problems. *Operations Research Letters*, 30, 327-332.
- Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network Flows-Theory, Algorithms and Applications*: Prentice Hall, Inc.
- Al-Sultan, K. S., & Al-Fawzan, M. A. (1999). A Tabu Search Approach to the Uncapacitated Facility Location Problem. *Annals of Operations Research*, 86, 91-103.
- Alander, J. T. (1992). On Optimal Population Size of Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*.
- Albareda-Sambola, M., Fernández, E., Hinojosa, Y., & Puerto, J. (2009). The multi-period incremental service facility location problem. *Computers and Operations Research*, 36(5), 1356-1375.
- Alçada-Almeida, L., Coutinho-Rodrigues, J., & Current, J. (2009). A multiobjective modeling approach to locating incinerators. *Socio-Economic Planning Sciences*, 43(2), 111-120.
- Allett, E. J. (1986). Environmental Impact Assessment and Decision Analysis. *Journal of the Operational Research Society*, 37(9), 901-910.
- Alminyana, M., Borrás, F., & Pastor, J. T. (1998). A New Directed Branching Heuristic for the p-q-median problem. *Location Science*, 6.
- Alp, O., Erkut, E., & Drezner, Z. (2003). An Efficient Genetic Algorithm for the p-Median Problem. *Annals of Operations Research*, 122, 21-42.
- Alter, S. (2004). A Work System View of DSS in its Fourth Decade. *Decision Support Systems*, 38, 319-327.
- Alves, M. L., & Almeida, M. T. (1992). Simulated Annealing Algorithm for the Simple Plant Location Problem: a Computational Study. *Investigação Operacional*, 12(2), 145-157.
- Anson, R. G., & Jelassi, M. T. (1990). A Development Framework for Computer-supported Conflict Resolution. *European Journal of Operational Research*, 46, 181-199.
- Antunes, A., & Peeters, D. (2001). On solving complex multi-period location models using simulated annealing. *European Journal of Operational Research*, 130, 190-201.
- Arabas, J., Michalewicz, Z., & Mulawka, J. (1994). *GAVaPS - a Genetic Algorithm with Varying Population Size*.

- Avella, P., Benati, S., Martinez, L. C., Dalby, K., Girolamo, D., Dimitrijevic, B., et al. (1998). Some Personal views on the Current State and the Future of Locational Analysis. *European Journal of Operational Research*, 104, 269-287.
- Aydin, M. E., & Fogarty, T. C. (2004). A Distributed Evolutionary Simulated Annealing Algorithm for Combinatorial Optimisation Problems. *Journal of Heuristics*, 10, 269-292.
- Bäck, T., A.E.Eiben, & Vaart, N. A. L. V. d. (2000). *An Empirical Study on Gas ""Without Parameters*. Paper presented at the PPSN VI.
- Baker, B. (1986). A Partial Dual Algorithm for the Capacitated Warehouse Location Problem. *European Journal of Operational Research*, 23, 48-56.
- Baker, B. M. (1982). Linear Relaxations of the Capacitated Warehouse Location Problem. *Journal of the Operational Research Society*, 33, 475-479.
- Baldaci, R., Hadjiconstantinou, E., Maniezzo, V., & Mingozzi, A. (2002). A New Method for Solving Capacitated Location Problems Based on a Set Partitioning Approach. *Computers & Operations Research*, 29, 365-386.
- Ballou, R. H. (1968). Dynamic Warehouse Location Analysis. *Journal of Marketing Research*, 5, 271-276.
- Barahona, F., & Chudak, F. (1999). *Near-Optimal Solutions to Large Scale Facility Location Problems* (No. RC21606).
- Barbosa, H. J. C. (1999). *A Coevolutionary Algorithm for Constrained Optimization*. Paper presented at the Proceedings of the 1999 Congress on Evolutionary Computation.
- Barbosa, H. J. C., & Barreto, A. M. S. (2001). *An Interactive Genetic Algorithm with Co-Evolution of Weights for Multiobjective Problems*. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference.
- Barceló, J., & Casanovas, J. (1984). A Heuristic Lagrangean Algorithm for the Capacitated Plant Location Problem. *European Journal of Operational Research*, 15, 212-226.
- Barcelo, J., Fernandez, E., & Jörnsten, K. (1991). Computational Results from a New Lagrangean Relaxation Algorithm for the Capacitated Plant Location Problem. *European Journal of Operational Research*, 53, 38-45.
- Barros, A. I., & Labbé, M. (1994). The Multi-Level Uncapacitated Facility Location Problem is not Submodular. *European Journal of Operational Research*, 72, 607-609.
- Batthacharya, U., Rao, J. R., & Tiwari, R. N. (1993). Bi-criteria Multi Facility Location Problem in Fuzzy Environmet. *Fuzzy Sets and Systems*, 56, 145-153.
- Beasley, J. E. (1988). An Algorithm for Solving Large Capacitated Warehouse Location Problems. *European Journal of Operational Research*, 33, 314-325.
- Beasley, J. E. (1993). Lagrangean Heuristics for Location Problems. *European Journal of Operational Research*, 65, 383-399.
- Beasley, J. E., & Chu, P. C. (1996). A Genetic Algorithm for the Set Covering Problem. *European Journal of Operational Research*, 94, 392-404.
- Bennett, P., & Howard, N. (1996). Rationality, Emotion and Preference Change, Drama-Theoretic Models of Choice. *European Journal of Operational Research*, 92, 603-614.
- Bentley, P. J., & Wakefield, J. P. (1998). Finding Acceptable Solutions in the Pareto-optimal range using Multiobjective Genetic Algorithms. In P. K. Chawdry, R. Roy & R. K. Pant (Eds.), *Soft Computing in Enginnering Design and Manufacturing* (Vol. 5, pp. 231-240).
- Berman, O., Drezner, Z., & Wesolowsky, G. (2003). Locating Service Facilities whose Reliability is

- Distance Dependent. *Computers & Operations Research*, 30, 1683-1695.
- Berman, O., & Kaplan, E. (1990). Equity Maximizing Facility Location Schemes. *Transportation Science*, 24(2), 137-144.
- Binh, T. T. (1999). *A Multiobjective Evolutionary Algorithm the Study Cases*. Paper presented at the Proceedings of the 1999 Genetic and Evolutionary Computation Conference.
- Binh, T. T., & Korn, U. (1996). *An Evolution Strategy for the Multiobjective Optimization*. Paper presented at the Proceedings of the Second International Conference o Genetic Algorithms MENDEL'96.
- Binh, T. T., & Korn, U. (1997). *Mutiobjective Evolution Strategy with Linear and Nonlinear Constraints*. Paper presented at the 15 IMACS World Congress, Berlin, August 1997.
- Bitran, G., Chandru, V., Sempolinski, D., & Shapiro, J. (1981). Inverse Optimization: An Application to the Capacitated Plant Location Problem. *Management Science*, 27(10), 1120-1141.
- Bloemhof-Ruwaard, J., Salomon, M., & Wassenhove, L. N. V. (1996). The Capacitated Distribution and Waste Disposal Problem. *European Journal of Operational Research*, 88, 490-503.
- Borges, P. C., & Hansen, M. P. (1998). *A Basis for Future Successes in Multiobjective Combinatorial Optimization* (No. IMM-REP-1998-8).
- Bornstein, C. T., & Azlan, H. B. (1998). The Use of Reduction Tests and Simulated Annealing for the Capacitated Plant Location Problem. *Location Science*, 6, 67-81.
- Branas, C. C., & ReVelle, C. S. (2001). An Iterative Switching Heuristic to Locate Hospitals and Helicopters. *Socio-Ecomic Planning Science*, 35, 11-30.
- Branke, J., Kaubler, T., & Schmek, H. (2001). Guidance in Evolutionary Multi-objective Optimization. *Advances in Engineering Software, Elsevier Publisher*, 32, 499-507.
- Brans, J.-P. (2004). The Management of the Future Ethics in OR: Respect, Multicriteria Management, Happiness. *European Journal of Operational Research*, 153, 466-467.
- Brill, E. D., Flach, J. M., Hopkins, L., & Ranjithan, S. (1990). MGA: A Decision Support System for Complex, Incompletely Defined Problems. *IEEE Transactions on Systems Man and Cybernetics*, 20(4), 745-757.
- Brimberg, J., & ReVelle, C. (1998). A Bi-Objective Plant Location Problem: Cost vs. Demand Served. *Location Science*, 6, 121-135.
- Bumb, A. (2001). An Approximation Algorithm for the Maximization Version of the Two-Level Uncapacitated Facility Location Problem. *Operations Research Letters*, 29, 155-161.
- Canel, C., Khumawala, B., Law, J., & Loh, A. (2001). An Algorithm for the Capacitated, Multi-Commodity, Multi-Period Facility Location Problem. *Computers & Operations Research*, 28, 411-427.
- Caprara, A., & González, J. J. S. (1994). A Branch and Cut Algorithm for a Generalization of the Uncapacitated Facility Location Problem. *TOP*, 4, 135-163.
- Ceselli, A., & Righini, G. (2005). A Branch-and-Price Algorithm for the Capacitated p-Median Problem. *Networks*, 45(3), 125-142.
- Chamayou, C., Moisan, A., & Scala, C. (2005). *An Exact Method for a Bicriteria Location Problem: Minimal Cost vs Load Balancing*. Paper presented at the International Network Optimization Conference INOC 2005 20-23 March Lisboa.
- Chang, N.-B., & Wang, S. F. (1997). A Fuzzy Goal Programming Approach for the Optimal Planning of Metropolitan Solid Waste Management Systems. *European Journal of Operational Research*, 99, 303-321.

- Chardaïre, P. (1999). Hierarchical Two-Level Location Problems. In Sansò & Soriano (Eds.), *Telecommunications Network Planning* (pp. 33-54): kluwer Academic Publishers.
- Chardaïre, P., Lutton, J.-L., & Sutter, A. (1999). Upper and Lower Bounds for the Two-level Simple Plant Location Problem. *Annals of Operations Research*, 86, 117-140.
- Chardaïre, P., Sutter, A., & Costa, M.-C. (1996). Solving the Dynamic Facility Location Problem. *Networks*, 28, 117-124.
- Chen, J.-H., Goldberg, D. E., Ho, S.-Y., & Sastry, K. (2002). *Fitness Inheritance in Multi-Objective Optimization* (No. 2002017).
- Chen, Q., & Guan, S.-U. (2004). Incremental Multiple Objective Genetic Algorithms. *IEEE Transactions on Systems Man and Cybernetics*, 34(3), 1325-1334.
- Chenoweth, T., Dowling, K. L., & Louis, R. D. S. (2004). Convincing DSS users that Complex Models are Worth the Effort. *Decision Support Systems*, 37, 71-82.
- Cheung, B. K.-S., Langevin, A., & Villeneuve, B. (2001). High Performing Techniques for Solving Complex Location Problems in Industrial System Design. *Journal of Intelligent Manufacturing*, 12, 455-466.
- Chhajed, D., & Lowe, T. J. (1998). Solving a Selected Class of Location Problems by Exploiting Problem Structure: A Decomposition Approach. *Naval Research Logistics*, 45, 791-815.
- Chi, S.-C., Riordan, C. A., & Bejamin, C. O. (1996). *The Development and Empirical Study of a Knowledge-Based Site Selection Group Decision Support System*. Paper presented at the IEEE International Conference on Systems, Man and Cybernetics, 1996 14-17 Oct.
- Christofides, N., & Beasley, J. E. (1983). Extensions to a Lagrangean Relaxation Approach for the Capacitated Warehouse Location Problem. *European Journal of Operational Research*, 12, 19-28.
- Church, R. (2002). Geographical Information Systems and Location Science. *Computers & Operations Research*, 29, 541-562.
- Church, R., & Garfinkel, R. (1978). Locating an Obnoxious Facility on a Network. *Transportation Science*, 12(2), 107-118.
- Coello Coello, C. A. (1999a). A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3), 269-308.
- Coello Coello, C. A. (1999b). *An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends*. Paper presented at the Special Session on Multiobjective Optimization at the 1999 Congress on Evolutionary Computation.
- Coello Coello, C. A. (1999c). *The Use of a Multiobjective Optimization Technique to Handle Constraints*. Paper presented at the Second International Symposium on Artificial Intelligence.
- Coello Coello, C. A. (2000a). Constraint Handling using an Evolutionary Multiobjective Optimization Technique. *Civil Engineering Systems, Gordon and Breach Science Publishers*, 17, 319-346.
- Coello Coello, C. A. (2000b). *Handling Preferences in Evolutionary Multiobjective Optimization: A Survey*. Paper presented at the 2000 Congress on Evolutionary Computation.
- Coello Coello, C. A. (2000c). Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3), 275-308.
- Coello Coello, C. A. (2001). *A Short Tutorial on Evolutionary Multiobjective Optimization*. Paper presented at the First International Conference on Evolutionary Multi-Criterion Optimization, Zurich, Switzerland, March 2001.
- Coello Coello, C. A. (2002). Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey and the State of the Art. *Computer Methods in Applied Mechanics*

and Engineering, 191(11-12), 1245-1287.

Coello Coello, C. A., & Becerra, R. L. (2002). Constrained Optimization Using an Evolutionary Programming-based Cultural Algorithm. In I. C. Parmee (Ed.), *Adaptive Computing in Design and Manufacture V* (pp. 317-328): Springer London.

Coello Coello, C. A., & Becerra, R. L. (2003). *Evolutionary Multiobjective Optimization using a Cultural Algorithm*. Paper presented at the 2003 IEEE Swarm Intelligence Symposium.

Coello Coello, C. A., & Christiansen, A. D. (1998). Two New GA-Based Methods for Multiobjective Optimization. *Civil Engineering Systems, Gordon and Breach Science Publishers*, 15(3), 207-243.

Coello Coello, C. A., & Cortés, N. C. (2001). *Constraint-Handling in Genetic Algorithms through Emulations of the Immune System*. Paper presented at the Tercer Encuentro Internacional de Ciencias de la Computación (ENC'01).

Coello Coello, C. A., & Montes, E. M. (2002). Constraint-Handling in Genetic Algorithms Through the Use of Dominance-based Tournament Selection. *Advanced Engineering Informatics*, 16(3), 193-203.

Coello Coello, C. A., & Pulido, G. T. (2000). *A Micro-Genetic Algorithm for Multiobjective Optimization* (No. Lania-RI-2000-06).

Coello Coello, C. A., & Pulido, G. T. (2001). *Multiobjective Optimization Using a Micro-Genetic Algorithm*. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2001.

Coello Coello, C. A., & Sierra, M. R. (2003). *A Multi-Objective Evolutionary Algorithm Based on Coevolutionary Concepts* (No. EVOCINV-03-2003).

Coello Coello, C. A., Van Veldhuizen, D., & Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*: Kluwer Academic/ Plenum Publishers.

Collette, Y., & Siarry, P. (2005). Three New Metrics to Measure the Convergence of Metaheuristics Towards the Pareto Frontier and the aesthetic of a set of Solutions in Biobjective Optimization. *Computers & Operations Research*, 32, 773-792.

Corne, D., & Knowles, J. (2003). *Some Multiobjective Optimizers are Better than Others*. Paper presented at the Proceedings of the IEEE Congress on Evolutionary Computation.

Cornuejols, G., Nemhauser, G., & Wolsey, L. (1990). The Uncapacitated Facility Location Problem. In *Discrete Location Theory - Wiley Interscience - Mirchandani e Francis Edition* (pp. 119-172).

Cornuejols, G., Sridharan, R., & Thizy, J. M. (1991). A Comparison of Heuristics and Relaxations for the Capacitated Plant Location Problem. *European Journal of Operational Research*, 50, 280-297.

Correa, E. S., Steiner, M. T. A., Freitas, A. A., & Carnieri, C. (2001). *A Genetic Algorithm for the P-Median Problem*. Paper presented at the Proceedings 2001 Genetic and Evolutionary Computation GECCO2001.

Correia, I., & Captivo, M. E. (2003). A Lagrangean Heuristic for a Modular Capacitated Location Problem. *Annals of Operations Research*, 122, 141-161.

Cortés, N. C., & Coello Coello, C. A. (2003). *Multiobjective Optimization using Ideas from the Clonal Selection Principle*. Paper presented at the Genetic and Evolutionary Computation Conference GECCO'03.

Cortinhal, M. J., & Captivo, M. E. (2003a). Genetic Algorithms for the Single Source Capacitated Location Problem. In M. Resende & J. P. d. Sousa (Eds.), *Metaheuristics: Computer Decision-Making* (pp. 187-216): Kluwer Academic.

Cortinhal, M. J., & Captivo, M. E. (2003b). Upper and Lower Bounds for the Single Source Capacitated Location Problem. *European Journal of Operational Research*, 151, 333-351.

- Crainic, T., Delorme, L., & Dejax, P. (1993). A Branch-and-Bound Method for Multicommodity Location with Balancing Requirements. *European Journal of Operational Research*, 65, 368-382.
- Crainic, T., Gendreau, M., Hansen, P., & Mladenovic, N. (2004). Cooperative Parallel Variable Neighborhood Search for the p-Median. *Journal of Heuristics*, 10, 293-314.
- Current, J., & Ratick, S. (1995). A Model to Assess Risk, Equity and Efficiency in Facility Location and Transportation of Hazardous Materials. *Location Science*, 3(3), 187-201.
- Current, J., Ratick, S., & ReVelle, C. (1997). Dynamic Facility Location When the Total Number of Facilities is Uncertain: A Decision Analysis Approach. *European Journal of Operational Research*, 110, 597-609.
- Cvetkovic, D. (1997). *The Logic of Preference and Decision Supporting Systems*. Unpublished MSc Thesis, Faculty of Mathematics, University of Belgrade, Yugoslavia, May 1993.
- Cvetkovic, D., & Parmee, I. C. (2001). Preferences and their Application in Evolutionary Multiobjective Optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1), 42-57.
- Darby-Dowman, K., & Lewis, H. S. (1988). Lagrangian Relaxation and the Single Source Capacitated Facility-Location Problem. *Journal of the Operational Research Society*, 39(11), 1035-1040.
- Daskin, M. S. (1995). *Network and Discrete Location Models, Algorithms and Applications*: John Wiley & Sons.
- Davis, L. (1991). *Handbook of Genetic Algorithms*: Van Nostrand Reinhold.
- Deb, K. (1999). Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3), 205-230.
- Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms*: John Wiley & Sons.
- Dias, J. (2000). *Localização Simples Multicritério: desenvolvimento de um algoritmo em ambiente interativo*. Faculdade de Ciências da Universidade de Lisboa, Lisboa.
- Dias, J., Captivo, M. E., & Clímaco, J. (2003). An interactive procedure dedicated to a bicriteria plant location model. *Computers & Operations Research*, 30(13), 1977-2002.
- Dias, J., Captivo, M. E., & Clímaco, J. (2004a). *Capacitated Dynamic Location Problems with Opening, Closure and Reopening of Facilities* (No. 02/2004): Inesc-Coimbra.
- Dias, J., Captivo, M. E., & Clímaco, J. (2004b). *Dynamic Location Problem with Discrete Expansion and Reduction Sizes of Available Capacities* (No. 18/2004): Inesc-Coimbra.
- Dias, J., Captivo, M. E., & Clímaco, J. (2004c). *Dynamic Multi-Level Capacitated and Uncapacitated Location Problems: an approach using primal-dual heuristics*: Inesc-Coimbra.
- Dias, J., Captivo, M. E., & Clímaco, J. (2005a). Efficient Primal-Dual Heuristic for a Dynamic Location Problem. *Computers & Operations Research*, 34(6), 1800-1823.
- Dias, J., Captivo, M. E., & Clímaco, J. (2005b). *A Hybrid Algorithm for Dynamic Location Problems* (No. 03-2005): Inesc-Coimbra.
- Dias, J., Captivo, M. E., & Clímaco, J. (2005c). *Planning a Solid Urban Waste Treatment System: Dynamic Location Model and Primal-Dual Heuristic* (No. 05/2005): Inesc-Coimbra.
- Dias, J., Captivo, M. E., & Clímaco, J. (2006). Decision Support System for Location Problems. In F. Adam, P. Brézillon, S. Carlsson & P. Humphreys (Eds.), *Creativity and Innovation in Decision Making and Decision Support* (Vol. 1, pp. 388-402): Decision Support Press.
- Dias, J., Captivo, M. E., & Clímaco, J. (2007a). Dynamic Location Problems with Discrete Expansion and Reduction Sizes of Available Capacities. *Investigação Operacional*, 27, 107-130.
- Dias, J., Captivo, M. E., & Clímaco, J. (2007b). Dynamic Multi-Level Capacitated and Uncapacitated

- Location Problems: an approach using primal-dual heuristics. *Operational Research An International Journal*, 345-380.
- Dias, J., Captivo, M. E., & Clímaco, J. (2007c). Efficient Primal-Dual Heuristic for a Dynamic Location Problem. *Computers & Operations Research*, 34(6), 1800-1823.
- Dias, J., Captivo, M. E., & Clímaco, J. (2007d). A Memetic Algorithm for Dynamic Location Problems. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. J. Gutjahr, R. F. Hartl & M. Reimann (Eds.), *Metaheuristics: Progress in Complex Systems Optimization* (Vol. 39).
- Dias, J., Captivo, M. E., & Clímaco, J. (2008a). Dynamic Location Problems with Maximum Decreasing Capacities. *Central European Journal of Operations Research*, 16, 251-280.
- Dias, J., Captivo, M. E., & Clímaco, J. (2008b). A Memetic Algorithm for Multi-objective Dynamic Location Problems. *Journal of Global Optimization*, 42(2), 221-253.
- Díaz, J. A., & Fernández, E. (2004). Hybrid Scatter Search and Path Relinking for the Capacitated p-Median Problem. *European Journal of Operational Research*, 0.
- Dibble, C., & Densham, P. J. (1993). *Generating Interesting Alternatives in GIS and SDSS using Genetic Algorithms*. Paper presented at the GIS/LIS Proceedings, 2-4 November 1993 Minneapolis, Minnesota.
- Doerner, K. F., Gutjahr, W. J., & Nolz, P. C. (2009). Multi-criteria location planning for public facilities in tsunami-prone coastal areas. *OR Spectrum*, 31(3), 651-678.
- Domínguez-Marín, P., Nickel, S., Hansen, P., & Mladenovic, N. (2005). Heuristic Procedures for Solving the Discrete Ordered Median Problem. *Annals of Operations Research*, 136, 145-173.
- Drezner, Z. (1995). Dynamic Facility Location: The Progressive p-Median Problem. *Location Science*, 3(1), 1-7.
- Drezner, Z., & Wesolowski, G. O. (1991). Facility Location When Demand is Time Dependent. *Naval Research Logistics*, 38, 763-777.
- Drezner, Z., & Wesolowsky, G. (2001). On the Collection Depots Location Problem. *European Journal of Operational Research*, 130, 510-518.
- Drezner, Z., & Wesolowsky, G. O. (1989). Location of an Obnoxious Route. *Journal of the Operational Research Society*, 40(11), 1011-1018.
- Dumitrescu, D., Grosan, C., & Oltean, M. (2000). A New Evolutionary Approach for Multiobjective Optimization. *Informatica*, 15(1), 51-68.
- Easley, R. F., Valacich, J. S., & Venkataramanan, M. A. (2000). Capturing group preferences in a multicriteria decision. *European Journal of Operational Research*, 125, 73-83.
- Edwards, N. (2001). *Approximation algorithms for the multi-level facility location problem*. Unpublished PhD Thesis, Cornell University.
- Efrosom, M. A., & Ray, T. L. (1966). A Branch-Bound Algorithm for Plant Location. *Operations Research*, 0, 361-369.
- Eiben, A. E. (2000). Multi-parent Recombination. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Evolutionary Computation 1: Basic Algorithms and Operators* (pp. 289-307): Institute of Physics Publishing.
- Eiben, A. E. (2001). Evolutionary Algorithms and Constraint Satisfaction: Definitions, Survey, Methodology and Research Directions. In Kalle, B. Nandts, A. Rogers (eds) *Theoretical Aspects of Evolutionary Computing, Natural Computing Series, Springer*, 0, 13-58.
- Eiben, A. E. (2002a). Evolutionary Computing: The Most Powerful Problem Solver in the Universe? *Dutch Mathematical Archive*(5/3, n°2), 126-131.

- Eiben, A. E. (2002b). Multiparent Recombination in Evolutionary Computing. In A. Ghosh & S. Tsutsui (Eds.), *Advances in Evolutionary Computing*.
- Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2), 124-141.
- Eiben, A. E., & Smith, J. E. (2003). *Introduction to Evolutionary Computing*: Springer.
- Eitan, Y., Narula, S. C., & Tien, J. (1991). A Generalized Approach to Modeling the Hierarchical Location-Allocation Problem. *IEEE Transactions on Systems Man and Cybernetics*, 21(1), 39-46.
- Erkut, E., & Neuman, S. (1992). A Multiobjective Model for Locating Undesirable Facilities. *Annals of Operations Research*, 40, 209-227.
- Erlenkotter, D. (1975). Capacity Planning for Large Multilocation Systems: Approximate and Incomplete Dynamic Programming Approaches. *Management Science*, 22(3), 274-285.
- Erlenkotter, D. (1978). A Dual-Based Procedure for Uncapacitated Facility Location. *Operations Research*, 26(6), 992-1009.
- Erlenkotter, D. (1981). A Comparative Study of Approaches to Dynamic Location Problems. *European Journal of Operational Research*, 6, 133-143.
- Espejo, L. G. A., & Galvão, R. D. (2004). Uma Aproximação da Fronteira Eficiente para um Problema de Localização Hierárquico de Máxima Cobertura. *Pesquisa Operacional*, 24(2), 303-321.
- Espejo, L. G. A., Galvão, R. D., & Boffey, B. (2003). Dual-based Heuristics for a Hierarchical Covering Location Problem. *Computers & Operations Research*, 30, 165-180.
- Fernández, E., & Puerto, J. (2003). Multiobjective solution of the uncapacitated plant location problem. *European Journal of Operational Research*, 145, 509-529.
- Ferreira, C. (1997). *Problemas de Localização e Distribuição Multicritério: aproximações e estudo de alguns casos com implicações ambientais*. Unpublished PhD, Universidade de Aveiro.
- Ferreira, C., Santos, B. S., Captivo, M. E., Clímaco, J., & Silva, C. C. (1996). Multiobjective location of Unwelcome or Central Facilities Involving Environmental Aspects: a Prototype of a Decision Support System. *Belgian Journal of Operations Research Statistics and Computer Science*, 36, 159-172.
- Filipovic, V., Kratica, J., Tosic, D., & Ljubic, I. (2000). *Fine Grained Tournament Selection for the Simple Plant Location Problem*. Paper presented at the Proceedings of the 5th Online World Conference on Soft Computing Methods in Industrial Applications WSC5, September 2000.
- Fishburn, P., & Sarin, R. (1991). Dispersive Equity and Social Risk. *Management Science*, 37(7), 751-769.
- Fong, C. O., & Srinivasan, V. (1981a). The Multiregion Dynamic Capacity Expansion Problem - Part I. *Operations Research*, 29(4), 787-799.
- Fong, C. O., & Srinivasan, V. (1981b). The Multiregion Dynamic Capacity Expansion Problem - Part II. *Operations Research*, 29(4), 800-816.
- Fonseca, C. M., & Fleming, P. J. (1995a). *Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction*. Paper presented at the The Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95).
- Fonseca, C. M., & Fleming, P. J. (1995b). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1), 1-16.
- Frade, C. (1999). *A Componente Territorial no Ordenamento do Território*: Conselho Económico e Social.
- Francis, R. L., Lowe, T. J., & Tamir, A. (2000). Aggregation Error Bounds for a Class of Location

Models. *Operations Research*, 48(2), 294-307.

Freindenfelds, J. (1981a). *Capacity Expansion, Analysis of Simple Models with Applications*: Elsevier North Holland.

Freindenfelds, J. (1981b). Near-Optimal Solution of a Two-Type Capacity Expansion Problem. *Computers & Operations Research*, 8(3), 221-239.

Galvão, R. D. (1993). The Use of Lagrangean Relaxation in the Solution of Uncapacitated Facility Location Problems. *Location Science*, 1(1), 57-79.

Galvão, R. D., Espejo, L. G. A., & Boffey, B. (2002). A Hierarchical Model for the Location of Perinatal Facilities in the Municipality of Rio de Janeiro. *European Journal of Operational Research*, 138, 495-517.

Galvão, R. D., Espejo, L. G. A., & Boffey, B. (2006). Practical aspects associated with location planning for maternal and perinatal assistance in Brazil. *Annals of Operations Research*, 143(1), 31-44.

Galvão, R. D., Espejo, L. G. A., Boffey, B., & Yates, D. (2006). Load Balancing and Capacity Constraints in a Hierarchical Location Model. *European Journal of Operational Research*, 172, 631-646.

Galvão, R. D., & Santibañez-Gonzalez, E. d. R. (1992). A Lagrangean Heuristic for the P-Median Dynamic Location Problem. *European Journal of Operational Research*, 58, 250-262.

Gao, L.-L., & Robinson, E. P. (1992). A Dual-Based Optimization Procedure for the Two-Echelon Uncapacitated Facility Location Problem. *Naval Research Logistics*, 39, 191-212.

Gao, L.-L., & Robinson, E. P. (1994). Uncapacitated facility location: general solution procedure and computational experience. *European Journal of Operational Research*, 76, 410-427.

Gendron, B., & Crainic, T. G. (1997). A Parallel Branch-and-Bound Algorithm for Multicommodity Location with Balancing Requirements. *Computers & Operations Research*, 24(9), 829-847.

Geoffrion, A. M. (1976). The Purpose of Mathematical Programming is Insight, not Numbers. *Interfaces*, 7(1), 81-92.

Ghiani, G., & Musmanno, F. G. e. R. (2002). The Capacitated Plant Location Problem with Multiple Facilities in the Same Site. *Computers & Operations Research*, 29, 1903-1912.

Ghosh, D. (2003). Neighborhood Search Heuristics for the Uncapacitated Facility Location Problem. *European Journal of Operational Research*, 150, 150-162.

Glover, F., & Kochenberger, G. A. (2003). *Handbook of Metaheuristics*: Kluwer Academic Publishers.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*: Addison-Wesley.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*: Kluwer Academic Publishers.

Goldberg, D. E., Deb, K., Kargupta, H., & Harik, G. (1993). *Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms* (No. 93004).

Goldengorin, B., Tijssen, G. A., Ghosh, D., & Sierksma, G. (2003). Solving the Simple Plant Location Problem Using a Data Correcting Approach. *Journal of Global Optimization*, 25, 377-406.

Gonçalves, M. E. (2000). *Cultura Científica e Participação Pública*: Celta editora.

Gopalan, R., Kolluri, R. S., Batta, R., & Karwan, M. (1990). Modeling Equity of Risk in the Transportation of Hazardous Materials. *Operations Research*, 38(6), 961-973.

Granat, J., & Makowski, M. (1998). *ISAAP - Interactive Specification and Analysis of Aspiration-Based Preferences* (No. IR-98-052).

- Granat, J., & Makowski, M. (2000). Interactive Specification and Analysis of Aspiration-based Preferences. *European Journal of Operational Research*, 122, 469-485.
- Grefenstette, J. J. (1992). *Genetic Algorithms for Changing Environments*. Paper presented at the Parallel Problem Solving from Nature 2, Brussels.
- Gue, K. R. (2003). A Dynamic Distribution Model for Combat Logistics. *Computers & Operations Research*, 30, 367-381.
- Guignard, M., & Spielberg, K. (1979). A Direct Dual Method for the Mixed Plant Location Problem with Some Side Constraints. *Mathematical Programming*, 17, 198-228.
- Hamacher, H., Labbé, M., & Nickel, S. (1999). Multicriteria Network Location Problems with Sum Objectives. *Annals of Operations Research*.
- Hamacher, H. W., Labbé, M., Nickel, S., & Skriver, A. J. V. (2002). Multicriteria semi-obnoxious network location problems (MSNLP) with sum and center objectives. *Annals of Operations Research*, 110, 33-53.
- Hamacher, H. W., & Nickel, S. (1996). Multicriteria Planar Location Problems. *European Journal of Operational Research*, 94, 66-86.
- Hanjoul, P., & Peeters, D. (1985). A Comparison of Two Dual-Based Procedures for Solving the p-Median Problem. *European Journal of Operational Research*, 20, 387-396.
- Hanne, T. (1999). On the Convergence of Multiobjective Evolutionary Algorithms. *European Journal of Operational Research*, 117, 553-564.
- Hansen, M., & Jaszkiewicz, A. (1998). *Evaluating the Quality of Approximations to the Non-Dominated Set* (No. IMM REP 1998-7).
- Hansen, M. P. (1997). *Tabu Search for Multiobjective Optimization: MOTS*. Paper presented at the Presented at MCDM'97, Cape Town, South Africa, January 6-10, 1997.
- Hansen, P., Filho, E. d. L. P., & Ribeiro, C. C. (1992). Location and Sizing of Offshore Platforms for Oil Exploration. *European Journal of Operational Research*, 58, 202-214.
- Hansen, P., & Mladenovic, N. (1997). Variable Neighborhood Search for the p-Median. *Location Science*, 5(4), 207-226.
- Harper, P. R., Shahani, A. K., Gallagher, J. E., & Bowie, C. (2005). Planning Health Services with Explicit Geographical Considerations: a Stochastic Location-allocation approach. *Omega*, 33, 141-152.
- Haug, P. (1985). A Multiple Period, Mixed-Integer-Programming Model for Multinational Facility Location. *Journal of Management*, 11(3), 83-96.
- Herrmann, J. W., Ioannou, G., Minis, I., & Proth, J. M. (1996). A Dual Ascent Approach to the fixed-charge Capacitated Network Design Problem. *European Journal of Operational Research*, 95, 476-490.
- Hertz, A., & Widmer, M. (2003). Guidelines for the Use of Meta-heuristics in Combinatorial Optimization. *European Journal of Operational Research*, 0(151), 247-252.
- Hinojosa, Y., Puerto, J., & Fernández, F. R. (2000). A Multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research*, 123, 271-291.
- Hinomoto, H. (1965). Capacity Expansion with Facilities under Technological Improvement. *Management Science*, 11(5), 581-592.
- Hodgson, M. J. (1981). The Location of Public Facilities Intermediate to the Journey to Work. *European Journal of Operational Research*, 6, 199-204.
- Hodgson, M. J., & Jacobsen, S. K. (2009). A hierarchical location-allocation model with travel based on expected referral distances. *Annals of Operations Research*, 167(1), 271-286.

- Holmberg, K., & Ling, J. (1997). A Lagrangean Heuristic for the Facility Location Problem with Staircase Costs. *European Journal of Operational Research*, 97, 63-74.
- Holmberg, K., Ronnqvist, M., & Yuan, D. (1999). An Exact Algorithm for the Capacitated Facility Location Problems with Single Sourcing. *European Journal of Operational Research*, 113, 544-559.
- Horn, J., Nafpliotis, N., & Goldberg, D. (1994). A Niche Pareto Genetic Algorithm for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*.
- Hosage, C. M., & Goodchild, M. F. (1986). Discrete Space Location-Allocation Solutions from Genetic Algorithms. *Annals of Operations Research*, 6, 35-46.
- Houck, C., Joines, J., & Kay, M. (1996). Comparison of Genetic Algorithms, Random Restart and Two-Opt Switching for Solving Large Location-Allocation Problems. *Computers & Operations Research*, 23(6), 587-596.
- Hu, X., Coello Coello, C. A., & Huang, Z. (2005). A New Multi-Objective Evolutionary Algorithm: Neighbourhood Exploring Evolution Strategy. *Engineering Optimization*, 37(4), 351-379.
- Hultz, J., Klingman, D., Ross, G. T., & Soland, R. (1981). An Interactive Computer System for Multicriteria Facility Location. *Computers & Operations Research*, 8(4), 249-261.
- Huntley, C., & Brown, D. (1996). Parallel Genetic Algorithms with Local Search. *Computers & Operations Research*, 23(6), 559-571.
- Ignacio, A. A. V., Ferreira Filho, V. J. M., & Galvão, R. D. (2008). Lower and upper bounds for a two-level hierarchical location problem in computer networks. *Computers and Operations Research*, 35(6), 1982-1998.
- Ishibuchi, H., & Yoshida, T. (2002). *Hybrid Evolutionary Multi-Objective Optimization Algorithms*. Paper presented at the HIS2002.
- Ishibuchi, H., Yoshida, T., & Murata, T. (2002). *Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms*. Paper presented at the Evolutionary Computation, CSE 848, Fall 2002.
- Ishibuchi, H., Yoshida, T., & Murata, T. (2003). Balance Between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2), 204-223.
- Jacobsen, S. K. (1983). Heuristics for the Capacitated Plant Location Model. *European Journal of Operational Research*, 12, 253-261.
- Jacobsen, S. K. (1990). Multiperiod Capacitated Location Models. In P. B. Mirchandani & R. L. Francis (Eds.), *Discrete Location Theory* (pp. 173-208): Wiley Interscience Series.
- Janssen, R. (1992). *Multiobjective Decision Support for Environmental Management*: Kluwer Academic Publishers.
- Jaramillo, J., Bhadury, J., & Batta, R. (2002). On the Use of Genetic Algorithms to Solve Location Problems. *Computers & Operations Research*, 29, 761-779.
- Jaramillo, J. H. (1998). *Genetic Algorithms for Location Problems*. Unpublished MSc Thesis, Depart. Of Industrial Engineering, State University of New York at Buffalo.
- Jaszkiewicz, A. (2000). *On the Computational Effectiveness of Multiple Objective Metaheuristics*. Paper presented at the Proceedings of the Fourth International Conference on the Multi-objective Programming and Goal Programming MOPGP'00, May 29-June 1, 2000.
- Jaszkiewicz, A. (2002). Genetic Local Search for Multi-objective Combinatorial Optimization. *European Journal of Operational Research*, 137, 50-71.
- Jayaraman, V., Gupta, R., & Pirkul, H. (2003). Selecting hierarchical facilities in a service-operations

- environment. *European Journal of Operational Research*, 147, 613-628.
- Jessup, L., & Valacich, J. (1992). Future Directions and Challenges in the Evolution of Group Support Systems. In L. Jessup & J. Valacich (Eds.), *Group Support Systems, new perspectives*.
- Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-Objective Meta-heuristics: An Overview of the Current State-of-the-art. *European Journal of Operational Research*, 137, 1-9.
- Jungthirapanich, C., & Pratheepthaweephon, T. (1998). *A Geographic Information System-Based Decision Support System (GISDSS) for Facility Location*. Paper presented at the International Conference on Engineering and Technology Management.
- Karacapilidis, N., Papadias, D., Gordon, T., & Voss, H. (1997). Collaborative environmental planning with GeoMed. *European Journal of Operational Research*, 102, 335-346.
- Kim, S.-H., & Ahn, B.-S. (1997). Group Decision Making Procedure Considering Preference Strength Under Incomplete Information. *Computers & Operations Research*, 24(12), 1101-1112.
- King, W. R., & Kugler, J. (2000). The impact of rhetorical strategies on innovation decisions: an experimental study. *Omega*, 0(28), 458-499.
- Klincewicz, J. G., & Luss, H. (1987). A Dual-Based Algorithm for Multiproduct Uncapacitated Facility Location. *Transportation Science*, 21(3), 198-206.
- Klincewicz, J., & Luss, H. (1986). A Lagrangian Relaxation Heuristic for Capacitated facility Location with Single-Source Constraints. *Journal of the Operational Research Society*, 37(5), 495-500.
- Klincewicz, J., & Luss, H. (1987). A Dual-Based Algorithm for Multiproduct Uncapacitated Facility Location. *Transportation Science*, 21(3), 198-206.
- Klose, A. (1995). *A Lagrangean Heuristic to Solve the Two-stage Capacitated Facility Location Problem*.
- Klose, A. (1998). *Obtaining Sharp Lower and Upper Bounds for two-stage Capacitated Facility Location Problems*. Paper presented at the In B.Fleisechmann, J.A.E.Z Van Nunen, M.G.Speranza, P.Stähly (eds) *Advances in Distribution Logistics*.
- Klose, A. (1999). An LP-based heuristic for two-stage capacitated facility location problems. *Journal of the Operational Research Society*, 50(2), 157-166.
- Klose, A. (2000). A Lagrangean Relax-and-Cut Approach for the Two-Stage Capacitated Facility Location Problem. *European Journal of Operational Research*, 126, 408-421.
- Klose, A., & Stähly, P. (1996). Sensitivity Analysis in Facility Location Applied to a Depot Location Problem of a Food Producer. *Operations Research Proceedings*.
- Knowles, J. D., & Corne, D. W. (2002). Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2), 100-116.
- Köksalan, M., & Süral, H. (1999). Efes Beverage Group Makes Location and Distribution Decisions for its Malt Plants. *Interfaces*, 29(2), 89-103.
- Kolen, A. (1983). Solving Covering Problems and the Uncapacitated Plant Location Problem on Trees. *European Journal of Operational Research*, 12, 266-278.
- Kolli, S., & Evans, G. W. (1999). A Multiple Objective Integer Programming Approach for Planning Franchise Expansion. *Computers & Industrial Engineering*, 37, 543-561.
- Köppen, M., & Rudlof, S. (1998). *Multiobjective Optimization by Nussy Algorithm*. Paper presented at the Proceedings of the Third Online Workshop on Soft Computing, Nagoya, Japan, 1998.
- Körkel, M. (1989). On the Exact Solution of Large-Scale Simple Plant Location Problems. *European Journal of Operational Research*, 39, 157-173.
- Krarup, J., & Pruzan, P. (1983). The Simple Plant Location Problem: Survey and Synthesis. *European*

Journal of Operational Research, 12, 36-81.

Krasnogor, N., & Smith, J. (2005). A Tutorial for Competent Memetic Algorithms: Model, Taxonomy and Design Issues. *IEEE Transactions on Evolutionary Computation*, 9(5), 474-488.

Kratka, J. (1999). Improvement of Simple Genetic Algorithm for Solving the Uncapacitated Warehouse Location Problem. *Advances in Soft Computing, Engineering Design and Manufacturing*, 390-402.

Kratka, J., Filipovic, V., Sesum, V., & Tosic, D. (1996). *Solving the Uncapacitated Warehouse Location Problem Using a Simple Genetic Algorithm*. Paper presented at the Proceedings of the XIV International Conference on Material Handling and Warehousing, Belgrade, 1996.

Kratka, J., Tosic, D., Filipovic, V., & Ljubic, I. (2001). Solving the Simple Plant Location Problem by Genetic Algorithm. *RAIRO Operation Research*, 35(1), 127-142.

Kumar, R., & Rockett, P. (2002). Improved Sampling of the Pareto-Front in Multiobjective Genetic Optimizations by Steady-State Evolution: A Pareto Converging Genetic Algorithm. *Evolutionary Computation*, 10(3), 283-314.

Laguna, M., & Marti, R. (2003). *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers.

Landete, M., & Marin, A. (2003). *Using Trees to Strengthen the two-stage Uncapacitated facility Location Polytope* (No. I+2003+12).

Laporte, G., & Dejax, P. (1989). Dynamic Location-Routeing Problems. *Journal of the Operational Research Society*, 40(5), 471-482.

Laumanns, M., Rudolph, G., & Schwefel, H.-P. (1998). *A Spatial Predator-Key Approach to Multi-Objective Optimization: A Preliminary Study*. Paper presented at the PPSN V.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2001). *On the Convergence and Diversity-Preservation Properties of Multi-Objective Evolutionary Algorithms* (No. 108).

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002a). Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization. *Evolutionary Computation*, 10(3), 263-282.

Laumanns, M., Thiele, L., Zitzler, E., & Deb, K. (2002b). *Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization*. Paper presented at the GECCO 2002, Proceedings of the Genetic and Evolutionary Computation Conference, July 2002.

Laumanns, M., Zitzler, E., & Thiele, L. (2000). *A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism*. Paper presented at the Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000).

Laumanns, M., Zitzler, E., & Thiele, L. (2001). *On the Effects of Archiving, Elitism and Density Based Selection in Evolutionary Multi-Objective Optimization*. Paper presented at the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001).

Lee, C. (1991). An Optimal Algorithm for the Multiproduct Capacitated Facility Location Problem with a Choice of Facility Type. *Computers & Operations Research*, 18(2), 167-182.

Lee, C. Y. (1993). A Cross Decomposition Algorithm for a Multiproduct-Multitype Facility Location Problem. *Computers & Operations Research*, 20(5), 527-540.

Lee, S.-B., & Luss, H. (1987). Multifacility-type Capacity Expansion Planning: Algorithms and Complexities. *Operations Research*, 35(2), 249-253.

Lee, S. M., Green, G. I., & Kim, C. S. (1981). A Multiple Criteria Model for the Location-Allocation Problem. *Computers & Operations Research*, 8, 1-8.

Liebman, J. C. (1976). Some Simple-Minded Observations on the Role of Optimization in Public

- Systems Decision-Making. *Interfaces*, 6(4), 102-108.
- Lim, S.-K., & Kim, Y.-D. (1999). An Integrated Approach to Dynamic Plant Location and Capacity Planning. *Journal of the Operational Research Society*, 50, 1205-1216.
- Lis, J., & Eiben, A. E. (1996). *Multi-Sexual Genetic Algorithm for Multiobjective Optimization*. Paper presented at the Proceedings of the 1996 International Conference on Evolutionary Computation.
- Lorena, L., & Lopes, L. D. S. (1997). Genetic Algorithms Applied to Computationally Difficult Set Covering Problems. *Journal of the Operational Research Society*, 48, 440-445.
- Lorena, L. A. N., & Senne, E. L. F. (2004). A Column Generation approach to Capacitated P-Median Problems. *Computers & Operations Research*, 31, 863-876.
- Loughlin, D. H., & Ranjithan, S. (1997). *The Neighborhood Constraint Method: A Genetic Algorithm-Based Multiobjective Optimization Technique*. Paper presented at the Proceedings of the Seventh International Conference on Genetic Algorithms.
- Lu, H., & Yen, G. (2002). Rank-Density Based Multiobjective Genetic Algorithm. *IEEE Transactions on Systems Man and Cybernetics*.
- Luss, H. (1982). Operations Research and Capacity Expansion Problems. *Operations Research*, 30, 907-947.
- Madsen, O. B. G. (1983). Methods for Solving Combined Two Level Location-Routing Problems of Realistic Dimensions. *European Journal of Operational Research*, 12, 295-301.
- Magnanti, T., & Wong, R. (1990). Decomposition Methods for Facility Location Problems. In Mirchandani & R. Francis (Eds.), *Discrete Location Theory* (pp. 209-262): Wiley Interscience.
- Makowski, M., & Wierzbicki, A. (2000). Architecture of Decision Support Systems. In A. Wierzbicki, M. Makowski & J. Wessels (Eds.), *Model-Based Decision Support Methodology with Environmental Applications* (pp. 47-70): Kluwer Academic Publishers.
- Mandell, M. B., & Becker, L. R. (1996). A Model for Locating Automatic External Defibrillators. *Socio-Economic Planning Science*, 30(1), 51-66.
- Maniezzo, V., Mingozzi, A., & Baldacci, R. (1998). A Bionomic Approach to the Capacitated p-Median Problem. *Journal of Heuristics*, 4(3), 263-280.
- Manne, A. S. (1961). Capacity Expansion and Probabilistic Growth. *Econometrics*, 29.
- Marianov, V., & Serra, D. (2001). Hierarchical Location-Allocation Models for Congested Systems. *European Journal of Operational Research*, 135, 195-208.
- Marin, A., & Pelegrin, B. (1997). A Branch-and-Bound Algorithm for the Transportation Problem with Location of p Transshipment Points. *Computers & Operations Research*, 24(7), 659-678.
- Marín, A., & Pelegrín, B. (1998). The Return Plant Location Problem: Modelling and Resolution. *European Journal of Operational Research*, 104, 375-392.
- Marín, A., & Pelegrín, B. (1999). Applying Lagrangian Relaxation to the Resolution of Two-Stage Location Problems. *Annals of Operations Research*, 86, 179-198.
- Marsh, M., & Schilling, D. (1994). Equity Measurement in Facility location Analysis: A Review and Framework. *European Journal of Operational Research*, 74, 1-17.
- Matsatsinis, N. F., & Samaras, A. (2001). MCDA and Preference Disaggregation in Group Decision Support Systems. *European Journal of Operational Research*, 130, 414-429.
- Matwin, S., Szapiro, T., & Haigh, K. (1991). Genetic Algorithms Approach to a Negotiation Support System. *IEEE Transactions on Systems Man and Cybernetics*, 21(1), 102-114.
- Mazzola, J., & Neebe, A. (1999). Lagrangian - Relaxation - Based Solution Procedures for a

- Multiproduct Capacitated Facility Location Problem with Choice of Facility Type. *European Journal of Operational Research*, 115, 285-299.
- Melachrinoudis, E. (1999). Bicriteria Location of a Semi-Obnoxious Facility. *Computers & Industrial Engineering*, 37, 581-593.
- Melachrinoudis, E., & Min, H. (2000). The Dynamic Relocation and Phase-Out of a Hybrid, Two-Echelon Plant/Warehouse Facility: A Multiple Objective Approach. *European Journal of Operational Research*, 123, 1-15.
- Melachrinoudis, E., Min, H., & Wu, X. (1995). A Multiobjective Model for the Dynamic Location of Landfills. *Location Science*, 3(3), 143-166.
- Melachrinoudis, E., & Xanthopoulos, Z. (2003). Semi-obnoxious Single Facility Location in Euclidean Space. *Computers & Operations Research*, 30, 2191-2209.
- Melechovský, J., Prins, C., & Calvo, R. W. (2005). A Metaheuristic to Solve a Location-Routing Problem with Non-Linear Costs. *Journal of Heuristics*, 11, 375-391.
- Melkote, S., & Daskin, M. S. (2001). Capacitated Facility Location/Network Design Problems. *European Journal of Operational Research*, 129, 481-495.
- Mendes, I., Mattarelli, M., Maniezzo, V., Haastrup, P., Paruccini, M., & Rinaldi, F. M. (1998). A Decision Support System for Urban Waste Management. *European Journal of Operational Research*, 109(2), 330-341.
- Menestrel, M. L., & Wassenhove, L. N. V. (2004). Ethics Outside, Within or Beyond OR Models? *European Journal of Operational Research*, 153, 477-484.
- Michalewicz, Z. (1995). *A Survey of Constraint Handling Techniques in Evolutionary Computation Methods*. Paper presented at the Proceedings of the Fourth Annual Conference on Evolutionary Programming.
- Michel, L., & Hentenryck, P. V. (2004). A Simple Tabu Search for Warehouse Location. *European Journal of Operational Research*, 157, 576-591.
- Miettinen, K., & Mäkelä, M. (1999). Comparative Evaluation of Some Interactive Reference Point-Based Methods for Multi-objective Optimisation. *Journal of the Operational Research Society*, 50, 949-959.
- Mika, M., & Hämäläinen, R. P. (1995). Decision analysis interviews in environmental Impact Assessment. *European Journal of Operational Research*, 87, 551-563.
- Min, H. (1988). The Dynamic Expansion and Relocation of Capacitated Public Facilities: a Multi-Objective Approach. *Computers & Operations Research*, 15(3), 243-252.
- Min, H., Jayaraman, V., & Srivastava, R. (1998). Combined Location-Routing Problems: A Synthesis and Future Research Directions. *European Journal of Operational Research*, 108, 1-15.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. The MIT Press.
- Moore, G., & ReVelle, C. (1982). The Hierarchical Service Location Problem. *Management Science*, 28(7), 775-780.
- Moscato, P., & Cotta, C. (2003). A Gentle Introduction to Memetic Algorithms. In F. Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers.
- Moscovici, S., & Doise, W. (1991). *Dissensões e Consenso, uma Teoria Geral das Decisões Coletivas*: Livros Horizonte.
- Mumpower, J. (1991). The Judgement Policies of Negotiators and The Structure of Negotiation Problems. *Management Science*, 37(10), 1304-1324.

- Munda, G., Nijkamp, P., & Rietveld, P. (1995). Qualitative multicriteria methods for fuzzy evaluation problems: An illustration of economic-ecological evaluation. *European Journal of Operational Research*, 82, 79-97.
- Murata, T., Ishibuchi, H., & Gen, M. (1998). Neighborhood Structures for Genetic Local Search Algorithms. *IEEE Transactions on Systems Man and Cybernetics*, 259-263.
- Murty, K. (1983). *Linear Programming*: John Wiley & Sons.
- Narula, S. (1984). Hierarchical Location-Allocation Problems: A Classification Scheme. *European Journal of Operational Research*, 15, 93-99.
- Narula, S., & Ogbu, U. (1985). Lagrangean Relaxation and Decomposition in an Uncapacitated 2-Hierarchical Location-Allocation Problem. *Computers & Operations Research*, 12(2), 169-180.
- Nauss, R. (1978). An Improved Algorithm for the Capacitated Facility Location Problem. *Journal of the Operational Research Society*, 29(12), 1195-1201.
- Neebe, A. W., & Khumawala, B. M. (1981). An Improved Algorithm for the Multi-Commodity Location Problem. *Journal of the Operational Research Society*, 32, 143-149.
- Nijkamp, P., & Bergh, J. v. d. (1997). New Advances in economic modelling and evaluation of environmental issues. *European Journal of Operational Research*, 99, 180-196.
- Nutt, P. C. (2000). Context, Tactics and the examination of alternatives during strategic decision making. *European Journal of Operational Research*, 124, 159-186.
- Odetayo, M. O. (1993). *Optimal Population Size for Genetic Algorithms: an Investigation*.
- Oei, C. K., Goldberg, d. E., & Chang, S.-J. (1991). *Tournament Selection, Niching and the Preservation of Diversity* (No. 91011).
- Ogryczak, W. (1997). On the Lexicographic Minimax Approach to Location Problems. *European Journal of Operational Research*, 100, 566-585.
- Ogryczak, W. (1998). Location Problems from the Multiple Criteria Perspective: Efficient Solutions. *Archives of Control Sciences*, 7(XLIII), 161-180.
- Ogryczak, W. (1999). On the Distribution Approach to Location Problems. *Computers & Industrial Engineering*, 37, 595-612.
- Ogryczak, W. (2000). Inequality Measures and Equitable Approaches to Location Problems. *European Journal of Operational Research*, 122, 374-391.
- Ohsawa, Y., & Tamura, K. (2003). Efficient Location for a Semi-Obnoxious Facility. *Annals of Operations Research*, 123(1-4), 173-188.
- Osman, I. H., & Kelly, J. P. (1996). *Meta-Heuristics: Theory & Applications*: Kluwer Academic Publishers.
- Owen, S., & Daskin, M. (1998a). *A Note on Evolution Programs for Solving Multi-Objective Strategic Facility Location Problems*.
- Owen, S., & Daskin, M. (1998b). *Strategic Facility Location via Evolutionary Programming*.
- Owen, S., & Daskin, M. (1998c). Strategic Facility Location: A Review. *European Journal of Operational Research*, 111, 423-447.
- Pacheco, J. A., & Casado, S. (2005). Solving Two Location Models with Few Facilities by Using a Hybrid Heuristic: a Real Health resources Case. *Computers & Operations Research*, 32, 3075-3091.
- Parmee, I. C., Cvetkovic, D., Watson, A. H., & Bonham, C. R. (2000). Multi-Objective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation*, 8, 197-222.

- Pastor, J. T. (1994). Bicriterion Programs and Managerial Location Decisions: Application to the Banking Sector. *Journal of the Operational Research Society*, 45(12), 1351-1362.
- Pelegrin, B., & Fernandez, F. R. (1988). Determination of Efficient Points in Multiple-Objective Location Problems. *Naval Research Logistics*, 35, 697-705.
- Pereira, A. G., Munda, G., & Paruccini, M. (1994). Generating Alternatives for Siting Retail and Service Facilities using Genetic Algorithms and Multiple Criteria Decision Techniques. *Journal of Retailing and Consumer Services*, 1(2), 40-47.
- Pereira, A. G., Peckham, R. J., & Antunes, M. P. (1993). *GENET: A Method to Generate Alternatives for Facilities Siting using Genetic Algorithms*. Paper presented at the European Conference on Geographical Information Systems EGIS'93, Génova.
- Pereira, Â. G., & Quintana, S. C. (2002). From Technocratic to Participatory Decision Support Systems: Responding to the New Governance Initiatives. *GIDA - Journal of Geographic Information and Decision Analysis*, 6(2), 95-107.
- Pinkus, C., Gross, D., & Soland, R. (1973). Optimal Design of Multiactivity Multifacility Systems by Branch and Bound. *Operations Research*, 21, 270-283.
- Pinsonneault, A., & Kraemer, K. L. (1990). The Effects of Electronic Meetings on Group Processes and Outcomes: An Assessment of the Empirical Research. *European Journal of Operational Research*, 46, 143-161.
- Pirkul, H. (1987). Efficient Algorithms for the Capacitated Concentrator Location Problem. *Computers & Operations Research*, 4(3), 197-208.
- Pirkul, H., & Jayaraman, V. (1998). A Multi-Commodity, Multi-Plant, Capacitated Facility Location Problem: Formulation and Efficient Heuristic Solution. *Computers & Operations Research*, 25(10), 869-878.
- Pirkul, H., & Schilling, D. (1991). The Maximal Covering Location Problem with Capacities on Total Workload. *Management Science*, 37(2), 233-248.
- Pirlot, M. (1996). General Local Search Methods. *European Journal of Operational Research*, 92, 493-511.
- Raiffa, H. (1982). *The Art and Science of Negotiation - how to resolve conflicts and get the best out of bargaining*.
- Rakas, J., Teodorovi, D., & Kim, T. (2004). Multi-objective modeling for determining location of undesirable facilities. *Transportation Research Part D*, 9(2), 125-138.
- Rees, J., & Koehler, G. (2002). Evolution in Groups: a Genetic Algorithm Approach to Group Decision Support Systems. *Information Technology and Management*, 3, 213-227.
- Rees, J., & Koehler, G. J. (2004). Modeling Search in Group Decision Support Systems. *IEEE Transactions on Systems Man and Cybernetics*, 34(3), 237-244.
- Reeves, C., & Höhn, C. (1996). Integrating Local Search into Genetic Algorithms. *Modern Heuristic Search Methods*, 99-115.
- Reeves, C. R. (1993a). *Modern Heuristic Techniques for Combinatorial Problems*: Blakwell Scientific Publications.
- Reeves, C. R. (1993b). *Using Genetic Algorithms With Small Populations*. Paper presented at the Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA.
- Rekiek, B. (1999). *Multiple-Objectives Genetic Algorithm*. Paper presented at the Proceedings of the 1999 Genetic and Evolutionary Computation Conference, Orlando, Florida.

- Resende, M., & Wernick, R. (2004). A Hybrid Heuristic for the p-Median Problem. *Journal of Heuristics*, 10, 59-88.
- Revelle, C., & Laporte, G. (1996). The Plant Location Problem: New Models and Research Prospects. *Operations Research*, 44(6), 864-873.
- ReVelle, C. S., & Eiselt, H. A. (2005). Location Analysis: A Synthesis and Survey. *European Journal of Operational Research*, 165, 1-19.
- Righini, G. (1995). A Double Annealing Algorithm for Discrete Location/Allocation Problems. *European Journal of Operational Research*, 86, 452-468.
- Ro, H.-b., & Tcha, D.-w. (1984). A Branch and Bound Algorithm for the Two-Level Uncapacitated Facility Location Problem with Some Side Constraints. *European Journal of Operational Research*, 18, 349-358.
- Rodríguez-Chía, A. M., & Puerto, J. (2002). Geometrical Description of the Weakly Efficient Solution Set for Multicriteria Location Problems. *Annals of Operations Research*, 111, 181-196.
- Rolland, E., Schilling, D., & Current, J. (1996). An Efficient Tabu Search Procedure for the p-Median Problem. *European Journal of Operational Research*, 96, 329-342.
- Romero-Morales, D., carrizosa, E., & Conde, E. (1997). Semi-obnoxious location models: A global optimization approach. *European Journal of Operational Research*, 102, 295-301.
- Romero, C., Tamiz, M., & Jones, D. (1998). Goal Programming, Compromise Programming and Reference Point Method Formulations: Linkages and Utility Interpretations. *Journal of the Operational Research Society*, 49, 986-991.
- Rönqvist, M., Tragantalemsak, S., & Holt, J. (1999). A Repeated Matching Heuristic for the Single-Source Capacitated Facility Location Problem. *European Journal of Operational Research*, 116, 51-68.
- Rosing, K. E., ReVelle, C. S., & Schilling, D. A. (1999). A Gamma Heuristic for the p-Median Problem. *European Journal of Operational Research*, 117, 522-532.
- Ross, T., & Soland, R. (1980). A Multicriteria Approach to the Location of Public Facilities. *European Journal of Operational Research*, 4, 307-321.
- Rothlauf, F., & Goldberg, D. (2002). *Redundant Representations in Evolutionary Computation* (No. 2002025).
- Rudolph, G. (1998). *On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set*. Paper presented at the Proceedings of the Fifth IEEE Conference on Evolutionary Computation, 1998.
- Rudolph, G., & Agapie, A. (2000). *Convergence Properties of Some Multi-Objective Evolutionary Algorithms*. Paper presented at the Proceedings of the 2000 Conference on Evolutionary Computation.
- Sahin, G., & Süral, H. (2007). A review of hierarchical facility location models. *Computers & Operations Research*, 34, 2310-2331.
- Sakawa, M., & Kato, K. (2002). An Interactive Fuzzy satisficing Method for General Multiobjective 0-1 Programming Problems Through Genetic Algorithms with Double Strings Based on a Reference Solution. *Fuzzy Sets and Systems*, 125, 289-300.
- Sakawa, M., & Kato, K. (2003). Genetic Algorithms with Double Strings for 0-1 Programming Problems. *European Journal of Operational Research*, 144, 581-597.
- Saldanha-da-Gama, F., & Captivo, M. E. (1996). "A Note on "" Dual Based Procedure for Dynamic Facility Location"" (No. 11/96).
- Saldanha-da-Gama, F., & Captivo, M. E. (2002). *A Branch-and-Bound Procedure for the Multi-Period Capacitated Location Problem* (No. 8/2002).

- Saldanha-da-Gama, F., & Silva, M. (2005). *A Decomposition Scheme for a Multi-Period Phase in/Phase-out Location Problem*. Paper presented at the International Network Optimization Conference INOC 2005 20-23 March Lisboa.
- Salhi, S., & Gamal, M. D. H. (2003). A Genetic Algorithm Based Approach for the Uncapacitated Continuous Location-Allocation Problem. *Annals of Operations Research*, 123, 203-222.
- Sarker, R., Liang, K.-H., & Newton, C. (2002). A New Multiobjective Evolutionary Algorithm. *European Journal of Operational Research*, 140, 12-23.
- Sastry, K., Goldberg, D. E., & Pelikan, M. (2001). *Don't Evaluate, Inherit* (No. 2001013).
- Sbalzarini, I. F., Müller, S., & Koumoutsakos, P. (2000). *Multiobjective Optimization Using Evolutionary Algorithms*. Paper presented at the Center for Turbulence Research, Proceedings of the Summer Program, 2000.
- Schaffer, J. D. (1985). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Paper presented at the Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms.
- Schuster, P. (1997). Genotypes with Phenotypes: Adventures in an RNA toy world. *Biophysical Chemistry*, 66, 75-110.
- Serra, D., Marianov, V., & ReVelle, C. (1992). The Maximum-Capture Hierarchical Location Problem. *European Journal of Operational Research*, 62, 363-371.
- Shibuya, M., Kita, H., & Kobayashi, S. (1999). Integration of Multi-objective and Interactive Genetic Algorithms and its Application to Animation Design. *IEEE Transactions on Systems Man and Cybernetics*.
- Shimizu, Y. (1999). Multi-Objective Optimization for Site Location Problems through Hybrid Genetic Algorithm with Neural Networks. *Journal of Chemical Engineering of Japan*, 32(1), 51-58.
- Shimizu, Y., & Wada, T. (2003). *Logistic Optimization for Site Location and Route Selection under Capacity Constraints Using Hybrid Tabu Search*. Paper presented at the 8th International Symposium on Computer-Aided Process Systems Engineering.
- Shulman, A. (1991). An Algorithm for Solving Dynamic Capacitated Plant Location Problems with Discrete Expansion Sizes. *Operations Research*, 39(3), 423-436.
- Sinha, A., & Goldberg, D. E. (2003). *A Survey of Hybrid Genetic and Evolutionary Algorithms* (No. 2003004).
- Skriver, A. J. V., & Andersen, K. A. (2003). The Bicriterion Semi-obnoxious Location (BSL) Problem Solved by an ϵ -approximation. *European Journal of Operational Research*, 146, 517-528.
- Skriver, A. J. V., Andersen, K. A., & Holmberg, K. (2004). Bicriteria Network Location (BNL) Problems with Criteria Dependent Lengths and Minisum Objectives. *European Journal of Operational Research*, 156, 541-549.
- Smith, R. (1981). Planning Horizons for the Deterministic Capacity Problem. *Computers & Operations Research*, 8(3), 209-220.
- Sörensen, K., & Sevaux, M. (2006). MA|PM: Memetic Algorithms with Population Management. *Computers & Operations Research*, 33, 1214-1225.
- Sridharan, R. (1991). A Lagrangean Heuristic for the Capacitated Plant Location with Side Constraints. *Journal of the Operational Research Society*, 42(7), 579-585.
- Sridharan, R. (1993). A Lagrangian Heuristic for the Capacitated Plant Location Problem with Single Source Constraints. *European Journal of Operational Research*, 66, 305-312.
- Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic

- Algorithms. *Evolutionary Computation*, 2(3), 221-248.
- Sun, M., Aronson, J., McKeown, P., & Drinka, D. (1998). A Tabu Search Heuristic Procedure for the Fixed Charge Transportation Problem. *European Journal of Operational Research*, 106, 441-456.
- Swain, R. (1974). A Parametric Decomposition Approach for the Solution of Uncapacitated Location Problems. *Management Science*, 21(2), 189-198.
- Sweeney, D. J., & Tatham, R. (1976). An Improved Long-Run Model for Multiple Warehouse Location. *Management Science*, 22(7), 748-758.
- Syam, S. S. (2002). A Model and Methodologies for the Location Problem with Logistical Components. *Computers & Operations Research*, 29, 1173-1193.
- Sycara, K. (1991). Problem Restructuring in Negotiation. *Management Science*, 37(10), 1248-1268.
- Sycara, K. P. (1990). Negotiation Planning: An AI Approach. *European Journal of Operational Research*, 46, 216-234.
- Szczepanski, M., & Wierzbicki, A. (2002). *Multiple Criteria Evolutionary Algorithms in Relation to Reference Point Approaches and Nadir Estimation*. Paper presented at the International Conference on Multiple Criteria Decision Making, Áustria.
- Taillard, É., Gambardella, L., Gendreau, M., & Potvin, J.-Y. (2001). Adaptive memory Programming: a Unified View of Metaheuristics. *European Journal of Operational Research*, 135, 1-6.
- Tan, K. C., Lee, T. H., & Khor, E. F. (2001). Evolutionary Algorithms with Dynamic Population Size and Local Exploration for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 5(6), 565-588.
- Tcha, D.-w., & Lee, B.-i. (1984). A Branch and Bound Algorithm for the Multilevel Uncapacitated Facility Location Problem. *European Journal of Operational Research*, 18, 35-43.
- Tcha, D.-W., Ro, H.-B., & Yoo, C.-B. (1988). A Dual-Based Add Heuristic for Uncapacitated Facility Location. *Journal of the Operational Research Society*, 39(9), 873-878.
- Teghem, J., Tuytens, D., & Ulungu, E. L. (2000). An Interactive Heuristic Method for Multi-objective combinatorial optimization. *Computers & Operations Research*, 27, 621-634.
- Thanh, P. N., Bostel, N., & Péton, O. (2008). A dynamic model for facility location in the design of complex supply chains. *International Journal of Production Economics*, 113(2), 678-693.
- Thisse, J.-F., & Zoller, H. G. (1983). *Locational Analysis of Public Facilities: Studies in Mathematical and Managerial Economics*, Noth-Holland Publishing Company.
- Tien, J. M., El-Tell, K., & Simons, G. R. (1983). Improved Formulations to the Hierarchical Health Facility Location-Allocation Problem. *IEEE Transactions on Systems Man and Cybernetics*, 13(6), 1128-1132.
- Toth, P. (2000). Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems. *European Journal of Operational Research*, 0(125), 222-238.
- Tragantalerngsak, S., Holt, J., & Rönnqvist, M. (1997). Lagrangian Heuristics for the Two-Echelon, Single-Source, Capacitated Facility Location Problem. *European Journal of Operational Research*, 102, 611-625.
- Tragantalerngsak, S., Holt, J., & Rönnqvist, M. (2000). An Exact Method for the Two-Echelon, Single-Source, Capacitated Facility Location Problem. *European Journal of Operational Research*, 123, 473-489.
- Tsutsui, S., & Ghosh, A. (1998). *A Study on the Effect of Multi-Parent Recombination in Real Coded Genetic Algorithms*. Paper presented at the IEEE 1998 International Conference on Evolutionary Computation.

- Tüshaus, U., & Wittmann, S. (1998). *Strategic Logistic Planning by Means of Simple Plant Location: A Case Study* (Vol. 460).
- Tzeng, G.-H., & Chen, Y.-W. (1999). The Optimal Location of Airport Fire Stations: a Fuzzy Multi-Objective Programming and Revised Genetic Approach. *Transportation Planning and Technology*, 23, 37-55.
- V.J.M.F.Filho, & Galvão, R. D. (1998). A Tabu Search Heuristic for the Concentrator Location Problem. *Location Science*, 6, 189-209.
- Vaithyanathan, S., Burke, L., & Magent, M. (1996). Massively Parallel Analog Tabu Search Using Neural Networks Applied to Simple Plant Location Problems. *European Journal of Operational Research*, 93, 317-330.
- Valacich, J. S., Vogel, D. R., & Jr, J. F. N. (1989). *Integrating Information Across Sessions and Between Groups in Group Decision Support Systems*. Paper presented at the Proceedings of the 22nd Annual Hawaii International Conference on Decision Support and Knowledge Based Systems.
- Valenzuela-Rendón, M., & Uresti-Charre, E. (1997). *A Non-Generational Genetic Algorithm for Multiobjective Optimization*. Paper presented at the Proceedings of the Seventh International Conference on Genetic Algorithms.
- Van Roy, T. (1986). A Cross Decomposition Algorithm for Capacitated Facility Location. *Operations Research*, 34(1), 145-163.
- Van Roy, T., & Erlenkotter, D. (1982). A Dual-Based Procedure for Dynamic Facility Location. *Management Science*, 28(10), 1091-1105.
- Van Veldhuizen, D. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analysis and New Innovations*. Unpublished PhD Thesis, Air Force Institute of Technology, Faculty of the Graduate School of Engineering.
- Van Veldhuizen, D., & Lamont, G. B. (2000). Multiobjective Evolutionary Algorithms: Analysing the State-of-the-Art. *Evolutionary Computation*, 8(2), 125-147.
- Vernekar, A., Anandalingam, G., & Dorny, C. N. (1990). Optimization of Resource Location in Hierarchical Computer Networks. *Computers & Operations Research*, 17(4), 375-388.
- Weber, M., & Borchering, K. (1993). Behavioral Influences on Weight Judgements in Multiattribute Decision Making. *European Journal of Operational Research*, 67, 1-12.
- Weick, K., & Meader, D. (1992). Sensemaking and Group Support Systems. In L. Jessup & J. Valacich (Eds.), *Group Support Systems, new perspectives*.
- Wesolowsky, G., & Truscott, W. (1975). The Multiperiod Location-Allocation Problem with Relocation of Facilities. *Management Science*, 22(1), 57-65.
- Wesolowsky, G. O. (1973). Dynamic Facility Location. *Management Science*, 19(11), 1241-1248.
- Wetmore III, W. R., & Summers, J. (2003). *Group Decision Making: Friend or Foe?* Paper presented at the Engineering Management Conference: Managing Technologically Driven Organizations. The Human Side of Innovation and Change, 2-4 Nov. 2003.
- Wierzbicki, A. (2000a). Multi-Objective and Reference Point Optimization Tools. In M. M. Andrzej Wierzbicki, Jaap Wessels (Ed.), *Model-Based Decision Support Methodology with Environmental Applications* (pp. 215-247): Kluwer Academic Publishers.
- Wierzbicki, A. (2000b). Reference Point Methodology. In M. M. Andrzej Wierzbicki, Jaap Wessels (Ed.), *Model-Based Decision Support Methodology with Environmental Applications* (pp. 71-89): Kluwer Academic Publishers.
- Winterfeldt, D. v., & Edwards, W. (1986). *Decision Analysis and Behavioral Research*: Cambridge University Press.

- Wirasinghe, S. C., & Waters, N. M. (1983). An Approximate Procedure for Determining the Number, Capacities and Locations of Solid Waste Transfer-Stations in an Urban Region. *European Journal of Operational Research*, 12, 105-111.
- Wu, L.-Y., Zhang, X.-S., & Zhang, J.-L. (2006). Capacitated Facility Location Problem with General Setup Cost. *Computers & Operations Research*, 33, 1226-1241.
- Yagiura, M., & Ibaraki, T. (1996). The Use of Dynamic Programming in Genetic Algorithms for Permutation Problems. *European Journal of Operational Research*, 92, 387-401.
- Yen, G. G., & Lu, H. (2003). Dynamic Multiobjective Evolutionary Algorithm: Adaptive Cell-Based Rank and Density Estimation. *IEEE Transactions on Evolutionary Computation*, 253-274.
- Yuhong, Y., Yongtong, H., Ping, L., Danian, Z., Changchao, M., & Shiyuan, Y. (1997). *A Genetic Algorithm for Conflict Resolution in Concurrent Production Development*. Paper presented at the IEEE International Conference on Man System and Cybernetics, Orlando, USA, Oct. 1997.
- Zhang, J., Chen, B., & Ye, Y. (2005). A Multiexchange Local Search Algorithm for the Capacitated Facility Location Problem. *Mathematics of Operations Research*, 30(2), 389-403.
- Zhang, J., & Ye, Y. (2002). A note on the maximization version of the multi-level facility location problem. *Operations Research Letters*, 30, 333-335.
- Zitzler, E. (2002). Evolutionary Algorithms for Multiobjective Optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailou & T. Fogarty (Eds.), *Evolutionary Methods for Design, Optimisation and Control*.
- Zitzler, E., Laumanns, M., & Thiele, L. (2002a). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailou & T. Fogarty (Eds.), *Evolutionary Methods for Design, Optimisation and Control*.
- Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C. M., & Fonseca, V. G. d. (2002b). *Why Quality Assessment of Multi-objective Optimizers is Difficult*. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference, 2002.
- Zitzler, E., & Thiele, L. (1998). *Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study*.
- Zitzler, E., & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257-271.
- Zitzler, E., Thiele, L., & Deb, K. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2), 173-195.

(Página deixada propositadamente em branco)

Série

Investigação

•

Imprensa da Universidade de Coimbra

Coimbra University Press

2010

