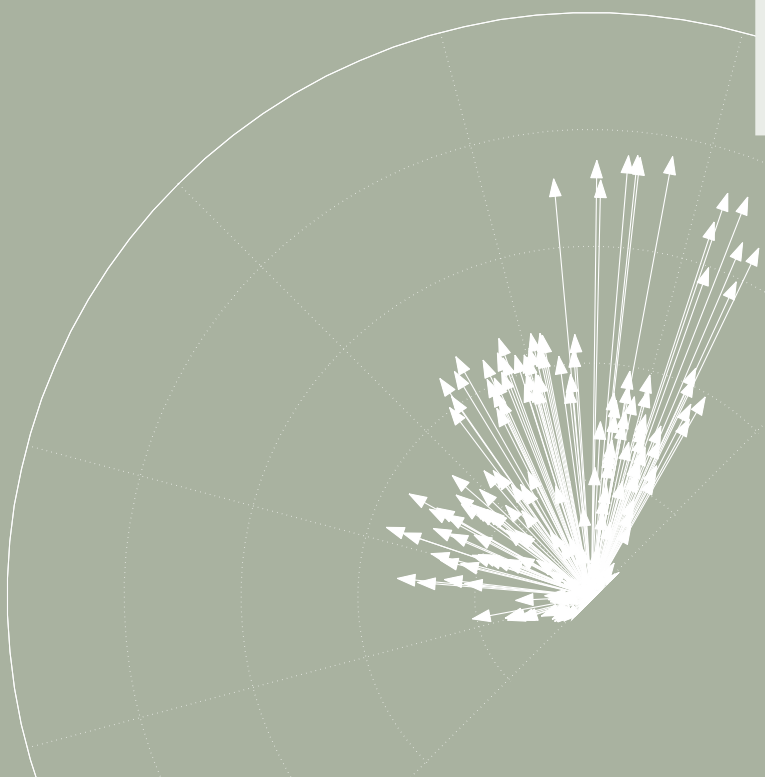
MANUAL DE COMPUTAÇÃO EVOLUTIVA E METAHEURÍSTICA

ANTÓNIO GASPAR-CUNHA
RICARDO TAKAHASHI
CARLOS HENGGELER ANTUNES
COORDENADORES



IMPRENSA DA UNIVERSIDADE DE COIMBRA
COIMBRA UNIVERSITY PRESS

EDITORA ufmg

CAPÍTULO 11

Algoritmos de Estimação de Distribuição

Pablo A. D. de Castro [*] *Fernando J. Von Zuben* [**]

[*] *Instituto Federal de Educação, Ciência e Tecnologia de São Paulo*
São Carlos - SP

[**] *Departamento de Engenharia de Computação e Automação Industrial*
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas

Nos últimos anos, tem havido um aumento no interesse pelo desenvolvimento e aplicação de um novo paradigma de computação evolutiva, denominado Algoritmo de Estimação de Distribuição. No lugar de usar os operadores tradicionais de mutação e recombinação de código genético, essa classe de algoritmos explora o espaço de busca construindo, a cada iteração, um modelo probabilístico que representa a distribuição de probabilidade das melhores soluções. Em seguida, novas soluções candidatas são amostradas a partir do modelo probabilístico obtido. A estimativa da distribuição de probabilidade das melhores soluções permite capturar correlações entre variáveis do problema por meio das regularidades encontradas nessas soluções. Consequentemente, o algoritmo é capaz de manipular eficientemente blocos construtivos, que correspondem a soluções parciais de boa qualidade contidas em trechos do vetor de soluções, evitando seu rompimento. Se, ao longo das gerações de um algoritmo evolutivo, é possível identificar e preservar blocos construtivos, o número de soluções candidatas que precisam ser avaliadas até se atingir um certo nível de desempenho tende a ser bem menor que o número

de avaliações requerida por algoritmos evolutivos tradicionais, para obter o mesmo desempenho. No entanto, o custo computacional por geração cresce quando se emprega um algoritmo de estimação de distribuição. Este capítulo descreve as principais propostas existentes na literatura e formaliza os modelos gráficos probabilísticos geralmente adotados junto a problemas de otimização de interesse prático, tanto em espaços contínuos quanto discretos.

1. Introdução

Algoritmos genéticos são meta-heurísticas baseadas em população, inspirados na teoria da seleção natural e nos princípios da genética. Apesar de terem sido aplicados com sucesso na resolução de problemas de busca e otimização nas mais diversas áreas, existem duas principais limitações associadas a estes algoritmos que podem produzir um impacto negativo em problemas de otimização mais complexos: (*i*) a definição adequada dos operadores de mutação e recombinação de código genético para cada tipo de problema, bem como os valores para seus parâmetros; e (*ii*) a ausência de um mecanismo capaz de extrair e utilizar conhecimento sobre as regularidades do problema por meio de soluções promissoras. Em muitos casos, as variáveis do problema se relacionam, formando subconjuntos que representam soluções parciais de boa qualidade para o problema. Essas soluções parciais são denominadas blocos construtivos (do inglês *building blocks*)(Goldberg, 1989).

Os algoritmos genéticos, na sua forma mais simples, com codificações fixas e operadores generalizados, independentes do problema, frequentemente rompem os blocos construtivos e, desse modo, tendem a tornar o processo de busca por boas soluções menos eficaz.

Várias propostas surgiram para eliminar ou amenizar estas limitações, desde a mudança na representação das soluções candidatas até a criação de operadores de mutação e recombinação mais específicos. Entretanto, esta abordagem é altamente dependente do problema. Diante deste cenário, na última década uma nova classe de algoritmos evolutivos, chamada Algoritmo de Estimação de Distribuição (AED) (do inglês *Estimation of Distribution Algorithm*), foi concebida como uma extensão dos algoritmos genéticos (Baluja, 1994; Mühlenbein e Paass, 1996). Os AEDs combinam os conceitos de computação evolutiva e aprendizado de máquina para evoluir a população de soluções. Nesses algoritmos, os operadores de mutação e recombinação são substituídos por um modelo probabilístico que representa a distribuição de probabilidade conjunta das melhores soluções. Em seguida, utiliza-se este modelo gerado para amostrar novas soluções. As escolhas do modelo mais apropriado e da forma de construí-lo variam conforme a aplicação.

A habilidade de identificar e preservar os blocos construtivos implicitamente confere a estes algoritmos um potencial diferenciado como ferramentas de otimização. A busca torna-se mais robusta, no sentido de que tende a ocorrer uma redução acentuada na variação de desempenho entre re-execuções do algoritmo para um mesmo problema. Além disso, tende a ocorrer também uma redução acentuada no número de avaliações de soluções candidatas até que se atinja um certo nível de desempenho.

Os objetivos deste capítulo são apresentar as motivações que levaram ao desenvolvimento dos AEDs, explicar detalhadamente seu princípio de funcionamento, elencar algumas aplicações práticas em que os AEDs estão sendo empregados com sucesso e apresentar perspectivas para o desenvolvimento de novos AEDs.

2. Blocos Construtivos

O algoritmo genético, assim como diversos outros algoritmos de busca e otimização, representa as soluções candidatas para um determinado problema como um vetor de atributos. É sabido que, em muitos problemas de otimização, as variáveis podem se relacionar umas com as outras, formando soluções parciais de boa qualidade para o problema. A essas soluções parciais dá-se o nome de blocos

construtivos. Os blocos construtivos podem ser vistos, então, como pequenos conjuntos de posições do vetor de soluções que se relacionam e, juntos, representam soluções parciais para o problema. Esses blocos construtivos contribuem de forma significativa para garantir a qualidade das boas soluções e devem ser sempre preservados.

Para ilustrar melhor o conceito de blocos construtivos, considere o problema artificial Trap-5 (Deb e Goldberg, 1992). Este problema consiste em dividir o vetor de soluções (vetor binário n-dimensional) em partições disjuntas de 5 bits cada. Este particionamento permanece fixo durante todo o processo de otimização e o algoritmo não possui nenhum conhecimento sobre como foi feito tal particionamento. Em seguida, uma função (Equação 11.1) é aplicada para cada grupo de 5 bits.

$$trap_5(u) = \begin{cases} 5, & \text{se } u = 5 \\ 4 - u, & \text{se } u < 5 \end{cases} \tag{11.1}$$

em que u denota a quantidade de vezes que aparece o símbolo 1 no bloco de 5 bits. Na Figura 11.1 é possível visualizar esta função para um bloco de 5 bits.

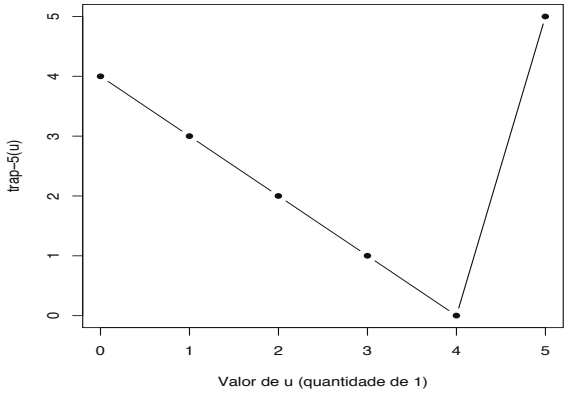


Figura 11.1: Exemplo do problema Trap-5 para um bloco de 5 bits

O objetivo é maximizar a função. Para um vetor de soluções n-dimensional, esta função possui seu ótimo global quando todos os seus bits são iguais a "1", e possui $2^{n/5} - 1$ ótimos locais. A contribuição de todas as funções são combinadas para formar o valor de adaptação ou *fitness* do indivíduo (solução candidata). Portanto, o *fitness* de uma solução candidata n-dimensional X é dado por:

$$fitness(X) = \sum_{i=1}^{n/5} trap_5(S_i) \tag{11.2}$$

em que S_i representa o i-ésimo bloco de 5 bits.

Este exemplo sugere que existem problemas em que o tratamento de variáveis de forma isolada não funciona, pois cada grupo de 5 bits contendo a sequência "11111" é um bloco construtivo. Um algoritmo genético tradicional apresenta dificuldades ao tentar resolver problemas desse tipo (Trap-k, com $k > 2$), pois os operadores de mutação e recombinação não conseguem preservar os blocos construtivos, levando-os constantemente ao rompimento. Esta situação é conhecida como problema da ligação (do inglês, *linkage problem*) (Goldberg et al., 1989).

Diversas tentativas para evitar o rompimento dessas soluções parciais foram realizadas, incluindo a mudança na representação das soluções e o desenvolvimento de operadores evolutivos específicos. Infelizmente, abordagens deste tipo são altamente dependentes do problema, levando ao desenvolvimento de algoritmos muito específicos. Além disso, elas requerem que o usuário possua um conhecimento prévio no domínio do problema. Entretanto, em muitos problemas do mundo real, este conhecimento não está disponível previamente.

Nesse sentido, uma maneira mais eficiente de lidar com os blocos construtivos é deixar que o próprio algoritmo descubra as relações existentes entre as variáveis, usando para tanto informações estatísticas extraídas a partir do conjunto das melhores soluções. Esta abordagem é denominada aprendizado da ligação (do inglês, *linkage learning*).

As melhores soluções podem ser encaradas como amostras geradas a partir de uma distribuição de probabilidade desconhecida. Em estatística, o termo distribuição de probabilidade para um conjunto de dados refere-se à probabilidade desses dados serem observados. O algoritmo poderia, então, estimar a distribuição de probabilidade das melhores soluções e, posteriormente, utilizar esta distribuição para gerar novas soluções candidatas.

Dessa maneira, um algoritmo com esse princípio de funcionamento consegue identificar e manter blocos construtivos do problema, durante sua própria execução, evitando o rompimento de parte das soluções parciais. Cabe ressaltar que nem todos os blocos construtivos são detectados e geralmente a informação disponível para se gerar um modelo de distribuição de probabilidade é limitada, conduzindo assim a modelos aproximados e possivelmente tendenciosos.

Na Figura 11.2, está ilustrado o esquema de evolução de soluções candidatas para um problema fictício, usando um modelo probabilístico em lugar dos tradicionais operadores de mutação e recombinação. Algoritmos baseados neste princípio de funcionamento para explorar o espaço de busca são chamados de Algoritmos de Estimação de Distribuição (AEDs) e serão descritos na seção seguinte.

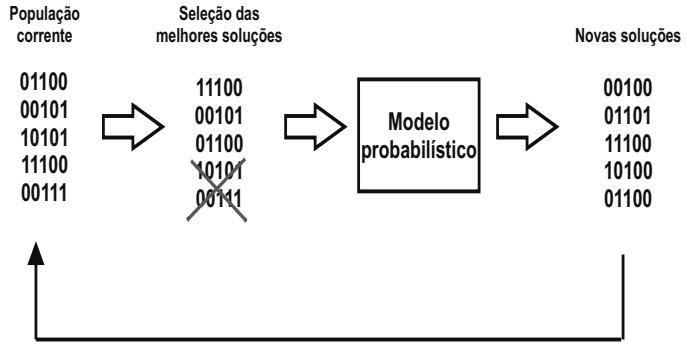


Figura 11.2: Exemplo do uso de um modelo probabilístico para gerar novas soluções candidatas.

3. Algoritmos de Estimação de Distribuição

Algoritmos de Estimação de Distribuição (AEDs) compreendem um conjunto de algoritmos evolutivos que substituem os operadores de mutação e recombinação por um modelo probabilístico que representa a distribuição de probabilidade conjunta para as melhores soluções encontradas até então. Esses algoritmos também são chamados de Algoritmos Genéticos Baseados em Modelos Probabilísticos (Baluja, 1994; Mühlenbein e Paass, 1996).

O funcionamento de um AED é similar ao de um algoritmo evolutivo. A população inicial de soluções é gerada aleatoriamente. A cada iteração, todas as soluções são avaliadas usando a função de

fitness e as melhores são selecionadas para a construção de um modelo probabilístico que represente a distribuição de probabilidade dessas melhores soluções. Após ter construído o modelo, ele é utilizado para amostrar novas soluções candidatas. Este processo é repetido até que o critério de parada seja satisfeito. A diferença, portanto, entre um Algoritmo de Estimação de Distribuição e um algoritmo evolutivo tradicional é a maneira como eles processam as melhores soluções. O pseudocódigo de um AED é apresentado no Algoritmo 1.

Algoritmo 1 Algoritmo de Estimação de Distribuição (AED)

Início
  Iniciar a população aleatoriamente;
enquanto condição de parada não for satisfeita **faça**
  Avaliar a população;
  Selecionar as melhores soluções;
  Construir o modelo probabilístico;
  Amostrar novas soluções usando o modelo probabilístico;
fim enquanto
Fim

Além da manipulação eficiente de blocos construtivos, pode-se citar mais duas motivações para se utilizar essa nova forma de evoluir a população de soluções. A primeira está relacionada com a possibilidade de analisar melhor e entender o processo de evolução do algoritmo. Já a outra reside no fato de não ser mais preciso se preocupar com a escolha dos operadores de mutação e recombinação mais adequados para cada tipo de problema, bem como os valores para seus parâmetros. É evidente que a construção do modelo probabilístico requer a determinação de alguns parâmetros, mas estes são de mais fácil definição, no sentido de que sua variação não altera drasticamente o desempenho do algoritmo.

Ao projetar um AED, surgem três questões fundamentais:

1. Que modelo probabilístico adotar para representar a distribuição de probabilidade das melhores soluções?

2. Como construir este modelo probabilístico?

3. Como usar o modelo probabilístico construído?

A resposta à primeira questão depende da complexidade do problema que se está tentando resolver e do grau de relacionamento entre as variáveis que se deseja capturar. Neste sentido, pode-se classificar os Algoritmos de Estimação de Distribuição em 3 categorias: (*i*) sem dependência entre as variáveis; (*ii*) com dependência entre pares de variáveis; e (*iii*) com dependência entre múltiplas variáveis. Ainda nesta seção, serão apresentados os algoritmos mais relevantes de cada categoria.

Para responder às outras duas questões, o projetista precisa ter em mente que é preciso existir um compromisso entre acuidade do modelo e custo computacional para gerá-lo. Quanto mais abrangente e geral for o modelo, maior será o tempo para construí-lo e também para amostrar novas soluções. Na Seção 5, serão descritas abordagens simples e eficientes para construir redes bayesianas e redes gaussianas, dois modelos gráficos probabilísticos amplamente utilizados por Algoritmos de Estimação de Distribuição e dedicados a capturar relacionamentos entre múltiplas variáveis.

Sem dependência entre as variáveis

Os algoritmos que não consideram nenhuma relação entre as variáveis, ou seja, supõem que as variáveis são independentes, são os mais simples. Devido à simplicidade do modelo probabilístico utilizado, os algoritmos nesta categoria não requerem custo computacional elevado e apresentam bom desempenho quando aplicados a problemas junto aos quais o relacionamento entre as variáveis não é tão significativo.

O primeiro algoritmo proposto nesta categoria foi o aprendizado incremental baseado em população (PBIL, do inglês *Population Based Incremental Learning*) (Baluja, 1994). Nesse algoritmo, as soluções candidatas são representadas por vetores binários, denominados cromossomos. Existe um vetor de probabilidades, da mesma dimensão do cromossomo, denotando a probabilidade de cada gene do cromossomo receber o valor "1". Inicialmente, o vetor de probabilidades contém o mesmo valor para todas as posições, 50%. A cada iteração, as melhores soluções são selecionadas e o vetor de probabilidades é atualizado seguindo uma regra baseada no aprendizado hebbiano. Esse vetor é utilizado, então, para amostrar novas soluções a cada iteração. Sebag e Ducoulombier (1998) desenvolveram uma versão desse algoritmo para lidar com problemas de otimização no espaço contínuo, denominada PBIL_c, na qual cada variável é modelada por um função densidade de probabilidade gaussiana. A cada iteração, a média e a variância de cada gaussiana são estimadas usando as melhores soluções.

Posteriormente, Mühlenbein e Paass (1996) propuseram o algoritmo de distribuição marginal univariada (UMDA, do inglês *Univariate Marginal Distribution Algorithm*). Seu funcionamento é similar ao do PBIL, com a exceção de que o vetor de probabilidades é atualizado calculando-se apenas a frequência com que aparece o número "1" nas melhores soluções selecionadas.

O algoritmo genético compacto (cGA, do inglês *Compact Genetic Algorithm*), proposto por Harik et al. (1999), também utiliza um vetor de probabilidades, o qual é iniciado contendo o valor 50% para todas as posições. Em seguida, duas soluções candidatas são amostradas a partir desse vetor e avaliadas pela função de *fitness*. O vetor de probabilidades é atualizado utilizando o valor dos genes do cromossomo com melhor *fitness*. Se os cromossomos possuírem o mesmo valor num mesmo gene, o vetor de probabilidades não é atualizado para aquela posição. Esse procedimento continua até que o vetor de probabilidades tenha convergido.

O algoritmo de estimação de distribuição com aprendizado por reforço (RELEDA, do inglês *Reinforcement Learning Estimation of Distribution Algorithm*), proposto por Paul e Iba (2003), é similar aos outros três algoritmos já apresentados. A diferença é que o vetor de probabilidades é atualizado de acordo com um método de aprendizado por reforço. Os autores compararam RELEDA com PBIL e UMDA e mostraram que seu algoritmo converge mais rapidamente para a solução ótima que os concorrentes. Os autores também desenvolveram o algoritmo de estimação de distribuição com código real (RECEDA, do inglês *Real-Coded Estimation of Distribution Algorithm*) para otimização de funções com variáveis contínuas. A partir das melhores soluções, a média e a matriz de covariância são calculadas.

O algoritmo de estimação de distribuição usando campo aleatório de Markov (DEUM, do inglês *Estimation of Distribution Algorithm Using MRF*) (Shakya et al., 2004) pode ser visto como uma adaptação dos outros quatro algoritmos. Em vez de calcular a frequência para cada gene do cromossomo, ele atualiza o vetor de probabilidades utilizando um campo aleatório de Markov (MRF, do inglês *Markov Random Field*). Os autores mostraram empiricamente que DEUM apresenta bons resultados para uma série de problemas de otimização (Shakya et al., 2005).

Dependência entre pares de variáveis

Existem algoritmos que consideram dependência entre pares de variáveis. Dessa forma, é preciso definir como representar a estrutura do modelo probabilístico, a fim de expressar o condicionamento das variáveis.

O algoritmo de maximização da informação mútua para agrupamento de entradas (MIMIC, do inglês *Mutual Information Maximization for Input Clustering*) (De Bonet et al., 1997) utiliza uma estrutura em cadeia entre as variáveis para poder representar o relacionamento entre elas, conforme ilustrado na Figura 11.3(a). Para encontrar a melhor estrutura de cadeia de variáveis, o algoritmo utiliza um mecanismo de busca que visa maximizar a informação mútua.

Baluja e Davies (1997) propuseram a combinação de otimizadores com informação mútua hierárquica (COMIT, do inglês *Combining Optimizers with Mutual Information Trees*), um algoritmo que utiliza uma estrutura em árvore para modelar a dependência entre as variáveis. Nesse tipo de estrutura, todas as variáveis precisam ter um pai, com exceção da variável-raiz. Este algoritmo é mais geral que o MIMIC, pois duas ou mais variáveis podem ser condicionadas a uma outra variável em comum. Na Figura 11.3(b) está um exemplo gráfico do modelo probabilístico utilizado pelo COMIT. Para obter a estrutura em árvore, os autores empregaram o algoritmo de árvore geradora máxima ponderada (MWST, do inglês *Maximum Weight Spanning Tree*) (Chow e Liu, 1968).

Já o algoritmo de distribuição marginal bivariada (BMDA, do inglês *Bivariate Marginal Distribution Algorithm*) (Pelikan e Mühlenbein, 1999) pode ser considerado como uma extensão do COMIT. A diferença é que o modelo em árvore utilizado por ele permite que existam mais de um nó-raiz, como mostrado na Figura 11.3(c). Para criar as sub-árvores e decidir quais variáveis devem ser conectadas ou não, o algoritmo emprega o teste do Qui-quadrado (Greenwood e Nikulin, 1996).

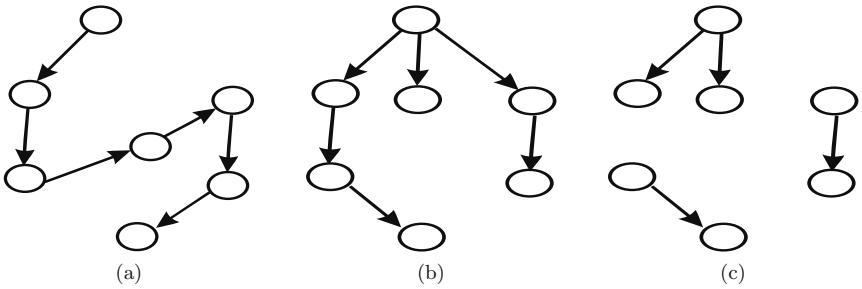


Figura 11.3: Modelos gráficos que consideram dependência entre pares de variáveis: (a) MIMIC, (b) COMIT e (c) BMDA.

Dependências entre múltiplas variáveis

Pelikan et al. (1999) mostraram que, em problemas complexos que possuem dependências entre múltiplas variáveis, os algoritmos pertencentes às duas categorias anteriores não são capazes de solucioná-los de forma satisfatória. Nesse sentido, várias propostas surgiram recentemente utilizando modelos capazes de expressar o relacionamento entre diversas variáveis. Como resultado, o modelo probabilístico é mais poderoso, embora sua obtenção se torne mais complexa.

O algoritmo genético compacto estendido (ECGA, do inglês *Extended Compact Genetic Algorithm*) (Harik et al., 2006) é uma extensão do cGA. A idéia básica é agrupar as variáveis em grupos independentes e, então, aplicar o cGA nesses grupos (veja Figura11.4(a)). O algoritmo de agrupamento primeiramente considera a existência de n grupos, sendo uma variável em cada grupo. Em seguida, tenta-se unificar pares de grupos usando medidas de entropia e a métrica Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*). Problemas que podem ser decompostos em sub-problemas sem sobreposição são tratados de forma satisfatória pelo ECGA. Por outro lado, se existir sobreposição, o algoritmo falha, pois o problema não pode ser modelado de forma acurada pela simples divisão das variáveis em classes distintas.

Mühlenbein e Mahnig (1999) propuseram o algoritmo de distribuição fatorizada (FDA, do inglês

Factorized Distribution Algorithm). Este algoritmo utiliza uma estrutura em grafo definida previamente e que se mantém fixa durante todo o processo de otimização, sendo atualizados apenas os parâmetros numéricos (probabilidades). Portanto, é preciso ter conhecimento sobre o problema a ser tratado para se construir um grafo adequado. O FDA utiliza a distribuição de Boltzmann para amostrar novos indivíduos. Um exemplo de grafo utilizado pelo FDA pode ser visto na Figura 11.4(b).

O algoritmo de otimização bayesiano (BOA, do inglês *Bayesian Optimization Algorithm*) (Pelikan et al., 1999) utiliza uma rede bayesiana para modelar as dependências entre as variáveis, conforme ilustrado na Figura 11.4(c). A cada iteração, uma rede bayesiana é gerada a partir das melhores soluções, tentando refletir da melhor forma as relações entre as variáveis. Após gerada, a rede é utilizada para amostrar novas soluções candidatas. Para construir a rede bayesiana, BOA começa com um estrutura sem arcos e os vai adicionando até que não ocorra melhora na métrica utilizada para avaliar a qualidade do modelo. Usualmente, BOA emprega a métrica *Bayesian Dirichlet* (BD). O algoritmo de otimização bayesiano com código real (rBOA, do inglês *real-coded Bayesian Optimization Algorithm*) (Ahn et al., 2006) é uma extensão do BOA para otimização em espaços contínuos, empregando modelo de mistura gaussiana. Já o algoritmo de otimização bayesiano hierárquico (hBOA, do inglês *hierarchical Bayesian Optimization Algorithm*) (Pelikan, 2005a) resolve problemas de otimização decompondo-os hierarquicamente, em vez de decompô-los em um único nível.

Outro algoritmo que também utiliza redes bayesianas foi proposto posteriormente por Larrañaga et al. (2000a), sendo denominado algoritmo de estimação de redes bayesianas (EBNA, do inglês *Estimation of Bayesian Networks Algorithm*). Seu funcionamento é similar ao BOA e diversas versões do algoritmo foram criadas com diferentes métricas para avaliar os modelos. Posteriormente, os autores desenvolveram a versão do algoritmo para lidar com otimização em espaços contínuos, denominada algoritmo de estimação de redes gaussianas (EGNA, do inglês *Estimation of Gaussian Networks Algorithm*) (Larrañaga et al., 2000b). EGNA utiliza redes gaussianas para modelar a dependência entre as variáveis.

Bosman e Thierens (2000) propuseram o algoritmo evolutivo com densidade iterativa (IDEA, do inglês *Iterated Density Evolutionary Algorithm*), que é uma arquitetura geral de algoritmos para aplicação em problemas de otimização tanto no domínio discreto quanto contínuo. No caso contínuo, os algoritmos utilizam funções densidade de probabilidade gaussianas multivariadas.

Uma proposta mais recente é o algoritmo de estimação de distribuição com rede de Markov (MN-EDA, do inglês *Markov Network Estimation of Distribution Algorithm*) (Santana, 2005), que utiliza uma rede de Markov como modelo probabilístico, como mostrado na Figura 11.4(d). Essa rede é gerada usando um procedimento estatístico conhecido como aproximação de Kikuchi (Kikuchi, 1951). Novas soluções são produzidas usando amostragem de Gibbs.

Desempenho do BOA no Trap-5

Nesta seção, é apresentado o desempenho do BOA junto ao problema Trap-5, já descrito na seção 2, e comparado com um algoritmo genético (AG). Os algoritmos foram comparados em termos de número de avaliações de função-objetivo necessárias até a população convergir. A escalabilidade dos algoritmos também foi testada, variando o valor de n no Trap-5 (tamanho do vetor de soluções).

Os parâmetros do BOA foram estabelecidos da seguinte forma: o tamanho da população variou de 200 (para n=20) até 600 (para n=80). Foi imposta uma restrição no número de pais que um nó pode ter na rede bayesiana, definido como 2. Para o algoritmo genético, o tamanho da população foi definido igual ao do BOA. Foi utilizado o operador de recombinação de um ponto com taxa de 90%. A taxa de mutação foi definida como 1%. Ambos os algoritmos utilizaram seleção por truncamento com 50%, ou seja, metade da população era selecionada. Estes parâmetros foram definidos empiricamente.

Os resultados médios em 30 execuções podem ser vistos na Tabela 11.1. Todos os algoritmos encontraram o ótimo global em todos os casos. Entretanto, BOA precisou de um número menor de

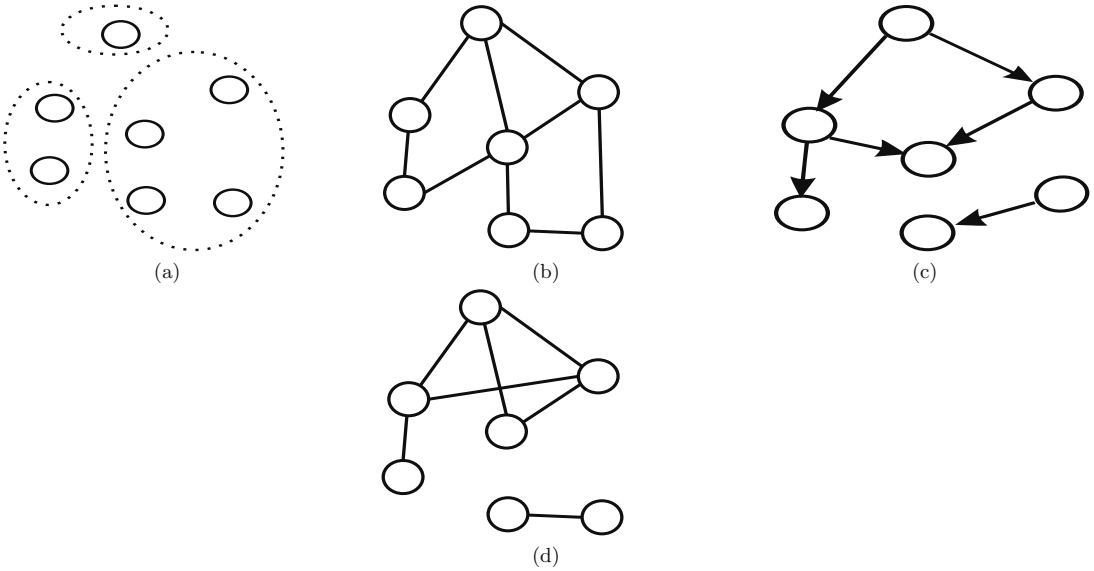


Figura 11.4: Modelos gráficos que consideram múltiplas dependências entre as variáveis: (a) ECGA, (b) FDA, (c) BOA e EBNA e (d) MN-EDA.

avaliações de função que o AG. Isto porque os operadores de mutação e recombinação do AG rompiam os blocos construtivos.

Tabela 11.1: Média dos resultados comparativos entre BOA e AG para o problema Trap-5, em termos de avaliações de *fitness*

n	BOA	AG
20	8100	14300
30	9200	22400
50	40000	70600
80	65400	114800

4. Limitações dos AEDs

Embora os algoritmos de estimação de distribuição (AEDs) possuam diversas características vantajosas e os experimentos nas mais diversas áreas comprovem isto, eles apresentam algumas limitações. A primeira está relacionada ao custo computacional para implementá-los. Já que a cada iteração é necessário construir o modelo probabilístico, os algoritmos não são indicados para problemas de otimização muito simples, pois existem técnicas mais eficientes para resolvê-los.

Outra limitação está relacionada com a perda de diversidade na população. Dado que tais algoritmos geram novas soluções com base na distribuição de probabilidade das melhores soluções selecionadas previamente, eles tendem a explorar o espaço de busca de uma forma polarizada. O algoritmo tende a visitar uma região do espaço de busca já visitada anteriormente, levando-o a mínimos locais facilmente. Dessa forma, qualquer algoritmo de estimação de distribuição básico apresenta desempenho

aquém do desejado quando é aplicado junto a problemas de otimização multimodal.

Por fim, a qualidade do modelo probabilístico influencia diretamente no comportamento do algoritmo. A construção de redes bayesianas ou redes gaussianas torna-se uma tarefa mais difícil frente a problemas de dimensão elevada, com muitas variáveis. Outro fator que pode comprometer a qualidade do modelo é a quantidade de dados disponíveis para construí-lo. Por essa razão, sugere-se que os AEDs trabalhem com populações contendo muitos indivíduos.

Na Seção 7 são apresentadas algumas possíveis abordagens para amenizar estas limitações.

5. Modelos Gráficos Probabilísticos

Esta seção fornece uma definição formal dos dois modelos gráficos probabilísticos mais utilizados em Algoritmos de Estimação de Distribuição: a rede bayesiana, utilizada quando o problema é de otimização com variáveis discretas, e a rede gaussiana, no caso das variáveis serem contínuas. Será descrita também uma possível abordagem para construir os modelos gráficos probabilísticos a partir de um conjunto de dados e, em seguida, utilizá-los para amostrar novos dados. Repare que estas duas tarefas são fundamentais para um Algoritmo de Estimação de Distribuição. A etapa de aprendizado do modelo corresponde à estimação da probabilidade conjunta das melhores soluções encontradas, enquanto a amostragem de novos dados a partir do modelo obtido corresponde à geração de novas soluções candidatas.

Formalmente, um modelo gráfico probabilístico para um conjunto $X=\{X_1, X_2, ..., X_n\}$ de variáveis aleatórias é uma fatoração gráfica da distribuição de probabilidade conjunta de X. Se as variáveis forem discretas, os modelos recebem o nome de rede bayesiana. Se as variáveis forem contínuas, o modelo é chamado de rede gaussiana. O modelo consiste de uma estrutura em grafo que representa as dependências condicionais das variáveis em X e de um conjunto de probabilidades para cada variável (Cowell et al., 1999; Jensen, 1996, 2001; Pearl, 1988).

Tanto as redes bayesianas quanto as redes gaussianas são modelos gráficos probabilísticos cuja estrutura é um grafo acíclico direcionado, no qual:

- cada nó corresponde a uma variável aleatória;
- os arcos ligando os nós indicam as relações de dependência entre as variáveis. Um arco saindo de uma variável X_i e chegando numa variável X_j significa que "X_i é pai de X_j" e que "X_j é filho de X_i";
- cada variável possui uma distribuição de probabilidade.

Na Figura 11.5, é apresentado um exemplo de modelo gráfico probabilístico que ilustra os conceitos definidos previamente. O conjunto de variáveis $X=\{X_1, X_2, X_3, X_4, X_5\}$ retrata as variáveis do modelo, que são representadas pelos nós do grafo.

Além da dependência condicional explícita entre as variáveis, os modelos gráficos probabilísticos representam implicitamente as independências condicionais entre elas. Uma variável é dita ser condicionalmente independente de outras que não são suas filhas, dados os seus pais. Na rede da Figura 11.5, a variável X_5 é condicionalmente independente de X_1, X_2 e X_4, dado seu pai X_3.

Portanto, no caso em que o modelo gráfico probabilístico da Figura 11.5 é uma rede bayesiana, a distribuição conjunta das variáveis pode ser expressa por:

$$p(x_1x_2x_3x_4x_5) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2x_3)p(x_5|x_3), \tag{11.3}$$

em que $p(.)$ representa a função massa de probabilidade.

De uma forma mais geral, a distribuição conjunta de X é dada por:

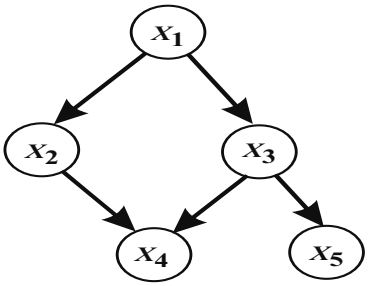


Figura 11.5: Exemplo de rede bayesiana.

$$p(x_1x_2x_3x_4x_5) = \prod_{i=1}^{5} p(x_i|\boldsymbol{pa_i}). \tag{11.4}$$

em que x_i é um possível valor da variável aleatória X e $\boldsymbol{pa_i}$ é o conjunto de pais da variável X_i. Por exemplo, na Figura 11.5, $\boldsymbol{pa_4}$=$\{X_2, X_3\}$. O termo $p(x_i|\boldsymbol{pa_i})$ é a probabilidade condicional de x_i, dados os valores das variáveis contidas no conjunto $\boldsymbol{pa_i}$.

Por outro lado, se for uma rede gaussiana, a distribuição conjunta é representada por uma distribuição gaussiana com vetor de médias $\boldsymbol{\mu}$ e matriz de covariância $\boldsymbol{\Sigma}$, da forma:

$$f(x_1x_2x_3x_4x_5) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}e^{-1/2(x-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})}, \tag{11.5}$$

em que $|\Sigma|$ denota o determinante da matriz de covariância e Σ^{-1} é a inversa da matriz de covariância, também chamada de matriz de precisão e algumas vezes denotada por W.

A decomposição em produtos da distribuição de probabilidade conjunta para X adquire a seguinte forma, quando o modelo utilizado é uma rede gaussiana:

$$f(x_1x_2x_3x_4x_5) = \prod_{i=1}^{5} f(x_i|\boldsymbol{pa_i}), \tag{11.6}$$

em que

$$f(x_i|\boldsymbol{pa_i}) = \mathcal{N}(\mu_i + \sum_{x_k \in \boldsymbol{pa_i}} b_{ki}(x_k - \mu_k), \sigma_i^2), \tag{11.7}$$

sendo que μ_i denota a média de X_i, σ_i^2 é a variância de X_i, condicionada aos pais de X_i e b_{ki} é o coeficiente de regressão linear refletindo o relacionamento entre as variáveis X_k e X_i. Se $b_{ki} = 0$, então não existe relacionamento entre as variáveis X_k e X_i.

Para tornar evidente a relação entre redes gaussianas e distribuições gaussianas multivariadas, Shachter e Kenley (1989) mostraram a transformação de b_{ki} e σ_i^2 da rede gaussiana em matriz de precisão W da distribuição gaussiana equivalente. A transformação é realizada por meio da seguinte fórmula recursiva:

$$W(i+1) = \begin{pmatrix} W(i) + \frac{b_{i+1}b_{i+1}^t}{\sigma_{i+1}^2} & \frac{-b_{i+1}}{\sigma_{i+1}^2} \\ \frac{-b_{i+1}^t}{\sigma_{i+1}^2} & \frac{1}{\sigma_{i+1}^2}, \end{pmatrix} \tag{11.8}$$

em que $W(i)$ denota a submatriz do canto superior esquerdo, $W(1) = \frac{1}{\sigma_i^2}$, b_i é o vetor coluna $(b_{1i}, ..., b_{(i-1)i})^t$ e b_i^t é o seu vetor transposto.

É a independência condicional que permite graficamente decompor a distribuição de probabilidade conjunta de X em produtos, levando a uma redução no número de parâmetros necessários para especificar o modelo.

Aprendizado de modelos gráficos probabilísticos

O aprendizado de modelos gráficos probabilísticos a partir de um conjunto de dados (no caso, as melhores soluções selecionadas) pode ser dividido em duas etapas. A primeira é a definição das relações de interdependência entre as variáveis, ou seja, a estrutura em grafo. Uma vez que a estrutura foi definida, entra em cena a segunda etapa, que é o cálculo das distribuições de probabilidade para as variáveis. Esta segunda tarefa é mais simples que a primeira. Basta maximizar a verossimilhança do modelo, considerando os dados.

Já o aprendizado da estrutura da rede é mais complexo e consiste em determinar relações entre as variáveis, adicionar ou excluir arcos, estabelecer direções, enfim definir um grafo acíclico direcionado para o qual serão calculadas distribuições de probabilidade. Geralmente, o aprendizado da estrutura pode ser considerado como um problema de otimização, em que um mecanismo de busca explora o espaço de busca contendo todos os possíveis grafos acíclicos (todas as possíveis topologias de redes), enquanto uma métrica de qualidade avalia cada solução candidata.

Com relação ao mecanismo de busca, geralmente é utilizado um algoritmo guloso não-populacional, como o de subida da encosta (do inglês *hill climbing*). Embora simples, ele tem sido aplicado com sucesso e apresentado bons resultados, tanto para redes bayesianas quanto para redes gaussianas. A busca pela melhor estrutura da rede bayesiana se inicia considerando uma estrutura de grafo básica (geralmente um grafo sem arcos ou completamente conectado). Em seguida, são realizadas algumas perturbações na estrutura corrente do grafo até que não seja mais possível obter melhoras na qualidade da rede, de acordo com a métrica escolhida. Perturbações típicas são adição, remoção e inversão de arcos. É necessário garantir que, após cada alteração da estrutura, ela ainda seja um grafo acíclico.

Na Figura 11.6, está ilustrado o esquema desta busca para um problema com 4 variáveis. O algoritmo começa sem ligação entre os nós (a) e uma avaliação desta rede não conectada é feita. Em seguida, adiciona-se um arco e a rede resultante é avaliada novamente (b). Se a nova estrutura for melhor, mantém-se o arco. Caso contrário, ele é retirado. Este processo continua até que nenhuma estrutura seja melhor que a anterior. No exemplo, a rede (e) é retornada como solução.

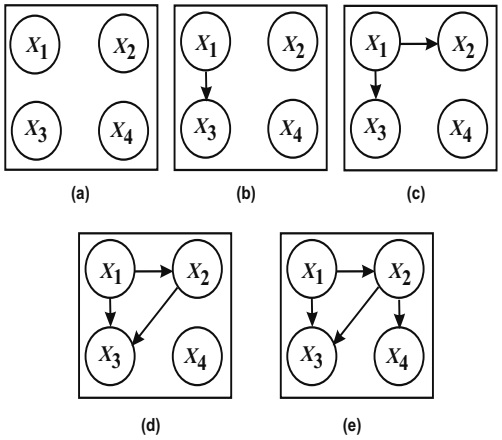


Figura 11.6: Processo iterativo para construção do modelo gráfico probabilístico.

Alguns trabalhos, no intuito de reduzir o espaço de busca, ordenam as variáveis numa lista de

modo que os possíveis pais de uma determinada variável só podem ser aqueles que aparecem antes dele na lista. Outra forma de reduzir a complexidade da busca é limitar a quantidade de pais que uma variável pode ter.

Um ponto que merece destaque é o fato de um AED ser um algoritmo de busca que utiliza outro algoritmo de busca mais simples para obter o modelo gráfico probabilístico. O leitor deve estar se perguntando por que não utilizar uma meta-heurística baseada em população para realizar esta tarefa. A razão para isso (utilizar algoritmo simples de busca local) é que o aprendizado do modelo probabilístico não visa resolver o problema de otimização que o AED está tentando resolver. Os AEDs não precisam da melhor rede bayesiana ou rede gaussiana, somente necessitam que estes modelos expressem algumas relações importantes entre as variáveis. Conforme exposto anteriormente, algoritmos como o de subida da encosta apresentam bons resultados e trocá-lo por um algoritmo baseado em população pode implicar em um aumento de custo computacional não condizente com o possível ganho de desempenho.

Com relação às métricas de avaliação da qualidade da rede bayesiana, as mais utilizadas são as métricas bayesianas. Estas métricas medem a qualidade da rede computando a sua verossimilhança com respeito aos dados, permitindo incluir conhecimento a priori sobre a estrutura e os parâmetros numéricos. Sua forma geral é:

$$P(B|D) = \frac{P(D|B)P(B)}{P(D)}, \tag{11.9}$$

em que B é a rede bayesiana sendo avaliada e D é o conjunto de dados disponível. Como $P(D)$ é constante para todas as redes, pode ser omitido. Portanto, as métricas maximizam $P(D|B)P(B)$.

Métrica para avaliar redes bayesianas

Heckerman et al. (1995) propuseram uma métrica que calcula a verossimilhança marginal e usa conhecimento a priori que segue uma distribuição de *Dirichlet*. A essa métrica, eles deram o nome de *Bayesian Dirichlet* (BD):

$$P(D|B) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(N'_{ij})!}{(N'_{ij} + N_{ij})!} \prod_{k=1}^{r_i} \frac{(N'_{ijk} + N_{ijk})!}{(N'_{ijk})!}, \tag{11.10}$$

em que D representa o conjunto de dados, B é a rede bayesiana sendo avaliada, n é o número de variáveis, q_i denota o número de possíveis valores que os pais de X_i podem assumir, r_i é o número de possíveis valores de X_i, N_{ijk} é o número de casos em que X_i assume o k-ésimo valor com seus pais assumindo o j-ésimo valor, e $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Os termos N'_{ij} e N'_{ijk} representam conhecimento a priori sobre as estatísticas N_{ij} e N_{ijk}.

Outras métricas utilizadas são a métrica baseada na entropia (Acid e Campos, 1996), a métrica do Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*) (Grünwald, 2000; Lam e Bacchus, 1994; Suzuki, 1996) e outros critérios clássicos para seleção de modelos, como BIC (do inglês *Bayesian Information Criterion*) (Schwarz, 1978) e AIC (do inglês *Akaike's Information Criterion*) (Akaike, 1974).

Métrica para avaliar redes gaussianas

A métrica mais utilizada para medir a qualidade de redes gaussianas foi derivada da métrica BIC, a qual utiliza o logaritmo da verossimilhança junto com uma função para penalizar redes muito complexas:

$$P(D|G) = \left[\prod_{l=1}^{N} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi v_i}} e^{-1/2v_i (x_{li} - \mu_i - \sum_{x_k \in \boldsymbol{pa_i}} b_{ki}(x_{lk} - \mu_k))^2} \right] - f(N) * dim(G), \tag{11.11}$$

em que D representa o conjunto de dados, G é a rede gaussiana sendo avaliada, N denota a quantidade de instâncias, n é o número de variáveis, b_{ki} é o coeficiente de regressão linear refletindo o relacionamento entre X_k e X_i, $\boldsymbol{pa_i}$ é o conjunto de pais da variável X_i e v_i é a variância condicional de X_i dados $X_1, ..., X_{i-1}$ $\forall i, k$. A função $f(N) = \frac{1}{2}\ ln\ N$ é responsável por penalizar modelos complexos e $dim(G) = 2n + \sum_{i=1}^{n} \boldsymbol{pa_i}$, que representa a quantidade de parâmetros a serem estimados.

Amostrando novas soluções

Uma vez que o modelo probabilístico foi gerado, novas soluções candidatas podem ser amostradas de acordo com a distribuição de probabilidade expressa pelo modelo. Para amostrar novas soluções, é utilizado o método da Amostragem Probabilística Lógica (PLS, do inglês *Probabilistic Logic Sampling*) (Henrion, 1986). Esse método amostra primeiramente todas as variáveis que não são dependentes de nenhuma outra. Em seguida, amostram-se as variáveis-pais para depois amostrar as variáveis-filhas.

6. Aplicações de AEDs

Por muito tempo, os AEDs permaneceram restritos a problemas de otimização mono-objetivo com variáveis discretas. Nesta seção, são apresentados os principais trabalhos reportados na literatura para outros tipos de problemas, como otimização com variáveis contínuas, otimização multiobjetivo e otimização dinâmica. São apresentadas também propostas em duas áreas de aplicações: bioinformática e aprendizado de máquina. O propósito desta seção é mostrar que os AEDs representam uma alternativa interessante em relação a outros algoritmos evolutivos e estão recebendo ótima aceitação da comunidade científica. Mais detalhes destas e outras aplicações podem ser encontradas em (Larrañaga e Lozano, 2002), (Pelikan, 2005a) e (Pelikan, 2005b).

Otimização com variáveis contínuas

Em se tratando de problemas de otimização com variáveis contínuas, existem os AEDs que utilizam uma função densidade de probabilidade gaussiana para modelar cada variável do problema, como no caso do RECEDA (Paul e Iba, 2003) e do PBIL$_c$ (Sebag e Ducoulombier, 1998). A cada iteração, a média e a variância de cada variável são atualizadas. Repare, portanto, que estes algoritmos não são capazes de capturar o relacionamento entre as variáveis.

Outras abordagens, como o EGNA (Larrañaga et al., 2000b) e IDEA (Bosman e Thierens, 2000), utilizam redes gaussianas para tentar capturar relacionamento entre múltiplas variáveis. A cada iteração, uma rede gaussiana é construída utilizando algum algoritmo de busca junto com uma métrica de avaliação das redes candidatas. Assim como nas versões discretas, novas soluções são amostradas a partir do modelo obtido. Estes algoritmos consideram que os dados disponíveis para construir o modelo probabilístico foram gerados por uma única distribuição gaussiana multivariada. Para problemas com funções unimodais, o algoritmo consegue estimar corretamente a distribuição de probabilidade para os dados. Entretanto, para funções multimodais, em que os ótimos locais não estão concentrados em uma única região, mas sim espalhados, uma única distribuição gaussiana multivariada não é capaz de modelar os dados de forma satisfatória.

Para sanar esta deficiência, alguns algoritmos passaram a trabalhar com modelo de mistura usando funções densidade de probabilidade gaussiana , como o *real-coded Bayesian Optimization Algorithm* (rBOA) (Ahn et al., 2006, 2004). Com esta modificação, a função densidade de probabilidade conjunta das melhores soluções é expressa por:

$$f(x) = \sum_{i=1}^{k} \alpha_i f_i(x), \qquad \sum_{i=1}^{k} \alpha_i = 1, \quad \alpha_i \geq 0, \tag{11.12}$$

em que $f_i(x)$ é uma única função densidade de probabilidade gaussiana multivariada, k é a quantidade de *clusters* e, consequentemente, a quantidade de componentes do modelo de mistura, e α_i é o coeficiente do i-ésimo componente da mistura. Os parâmetros da mistura podem ser obtidos via algoritmo EM (*Expectation Maximization*) ou por meio de algoritmos de agrupamento.

Os AEDs estão sendo aplicados com sucesso a várias funções comumente utilizadas na literatura, como a Griewank, Rosembrock, Rastrigin, Schwefel, Michalewicz e Ackley, superando outros algoritmos de otimização. Apenas para fornecer uma impressão da complexidade destes problemas, são apresentadas as definições e os gráficos de cada uma destas funções para duas dimensões. O emprego de duas dimensões aqui é para efeito de visualização gráfica, pois são consideradas na prática dezenas e até centenas de dimensões junto a essas funções de teste.

- **Griewank**: é uma função que possui muitos ótimos locais. Ela é definida como:

$$F(x) = 1 + \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}). \tag{11.13}$$

O valor do mínimo global desta função, para qualquer número de dimensões (d), é zero, o qual é obtido quando todas as variáveis assumem valor zero. Na Figura 11.7, é ilustrado o gráfico da função Griewank para duas dimensões.

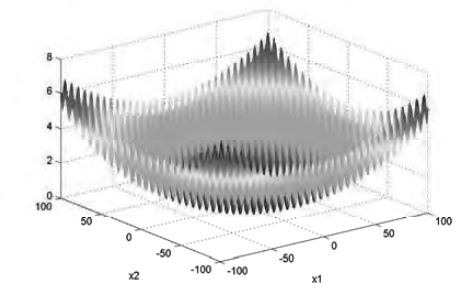


Figura 11.7: Gráfico da função Griewank para duas dimensões.

- **Rosenbrock**: é uma função bastante desafiadora para muitos algoritmos de otimização. Ela possui um vale longo, estreito e curvo, no qual está o ótimo global. A função é definida como:

$$F(x) = \sum_{i=1}^{d-1} 100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2. \tag{11.14}$$

O valor do ótimo global, para qualquer número de dimensões (d), é zero, o qual é obtido quando todas as variáveis assumem o valor um. O gráfico desta função, para duas dimensões, é apresentado na Figura 11.8.

- **Rastrigin**: esta função é baseada na função esfera, a qual recebeu um termo modulador para permitir a existência de muitos ótimos locais. A função Rastrigin é definida por:

$$F(x) = 10d \sum_{i=1}^{d-1} (x_i^2 - 10\cos(2\pi x_i)). \tag{11.15}$$

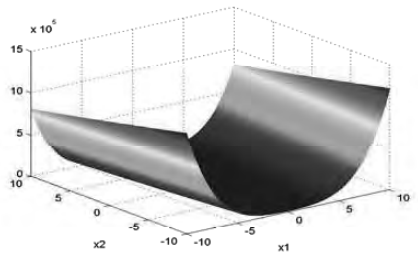


Figura 11.8: Gráfico da função Rosenbrock para duas dimensões.

O valor do ótimo global, para qualquer número de dimensões (d), é zero, o qual é obtido quando todas as variáveis assumem o valor zero. Na Figura 11.9, é ilustrado o gráfico da função Rastrigin, para duas dimensões.

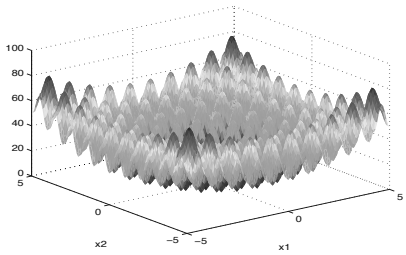


Figura 11.9: Gráfico da função Rastrigin para duas dimensões.

- **Schwefel**: é uma função composta por um grande número de picos e vales. O mínimo global se encontra distante dos mínimos locais, fazendo com que os algoritmos tendam a ficar presos nesses mínimos. Esta função é definida como:

$$F(x) = \sum_{i=1}^{d} -x_i \ \text{sen}(\sqrt{|x_i|}). \tag{11.16}$$

O valor do ótimo global é -418,9829d, o qual é obtido quando todas as variáveis assumem o valor 420,9687. O gráfico desta função, para duas dimensões, é apresentado na Figura 11.10.

- **Michalewicz**: esta função possui muitos ótimos locais e é definida por:

$$F(x) = -\sum_{i=0}^{d-1} \text{sen}(x_i)\text{sen}^{20}\left(\frac{(i+1)x_i^2}{\pi}\right). \tag{11.17}$$

O ótimo global depende do número de dimensões (d). Na Figura 11.11, é ilustrado o gráfico da função Michalewicz para duas dimensões.

- **Ackley**: é uma função considerada de dificuldade moderada, a qual é definida por:

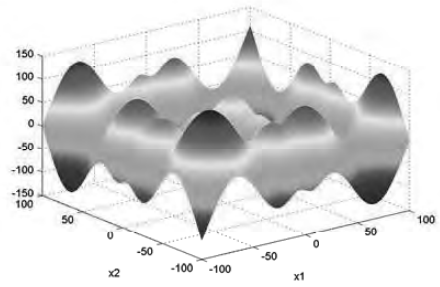


Figura 11.10: Gráfico da função Schwefel para duas dimensões.

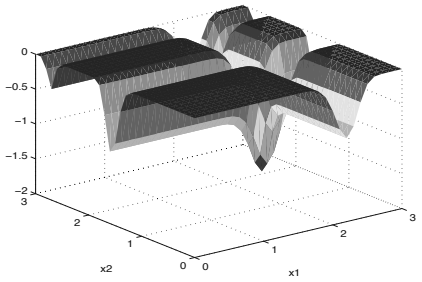


Figura 11.11: Gráfico da função Michalewicz para duas dimensões.

$$F(x) = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i}} - e^{\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)} + a + e^1. \tag{11.18}$$

O criador da função sugere utilizar os seguintes valores para os parâmetros: $a = 20$, $b = 0{,}2$ e $c = 2\pi$. O valor do ótimo global, para qualquer número de dimensões (d), é zero, o qual é obtido quando todas as variáveis assumem o valor zero. Na Figura 11.12, é ilustrado o gráfico da função Ackley para duas dimensões.

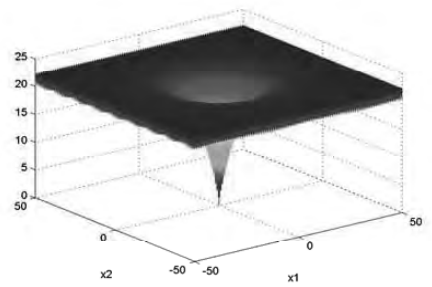


Figura 11.12: Gráfico da função Ackley para duas dimensões.

Maiores detalhes sobre AEDs para problemas com variáveis contínuas podem ser encontrados em (Ahn et al., 2006).

Otimização multiobjetivo

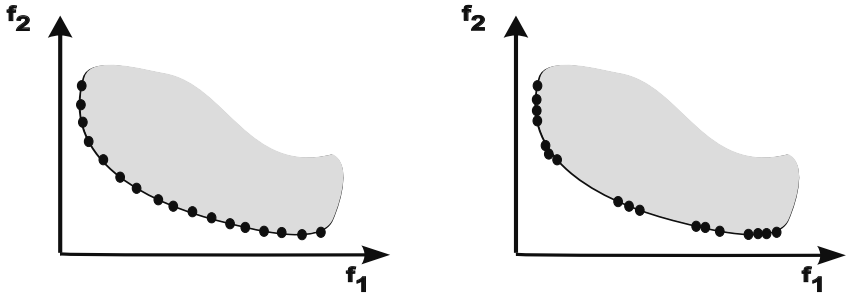
Um problema de otimização multiobjetivo consiste em otimizar simultaneamente duas ou mais funções-objetivo, possivelmente conflitantes, no sentido de que a melhora no valor de um dos objetivos pode degradar o valor dos outros (Deb, 2001; Zitzler et al., 2000). Nesse caso, a noção de otimalidade comumente adotada na literatura é aquela associada com a otimalidade de Pareto, a qual pode ser expressa usando o conceito de dominância. Uma solução domina outra se não é pior que a outra em nenhum objetivo e é estritamente melhor em pelo menos um objetivo. Uma solução é não-dominada se não existem soluções factíveis que a dominem, no sentido de que a melhora em algum objetivo vai implicar a piora em algum outro objetivo. Maiores detalhes podem ser encontrados no capítulo 17 deste livro.

A fronteira de Pareto é formada pelos valores das funções-objetivo correspondentes a cada solução que compõe o conjunto de soluções não-dominadas. Na ausência de informações adicionais sobre a relevância dos objetivos, todas essas soluções são igualmente importantes. Nesse sentido, duas metas práticas devem ser alcançadas por um algoritmo de otimização multiobjetivo:

1. Encontrar um conjunto de soluções que pertença ou que pelo menos esteja o mais próximo possível da fronteira de Pareto;

2. Encontrar um conjunto de soluções com a maior diversidade possível, de preferência uniformemente distribuídas ao longo da fronteira de Pareto, como no exemplo ilustrado na Figura 11.13(a). Já na Figura 11.13(b), é apresentado um exemplo de distribuição não desejável ao longo da fronteira de Pareto, pois embora possua a mesma quantidade de soluções que a proposta mostrada na Figura 11.13(a), as soluções possuem pouca diversidade entre si, ficando concentradas em algumas regiões da fronteira.

A primeira meta é bastante óbvia, já que a fronteira de Pareto satisfaz as condições de não-dominância ou otimalidade de Pareto. Já a segunda meta é necessária para evitar qualquer polarização em favor de uma função-objetivo específica. Somente com um conjunto diverso de soluções cuja distribuição se aproxima de uma distribuição uniforme pela fronteira de Pareto é que se pode promover um "equilíbrio" no que diz respeito a satisfazer os objetivos. A diversidade pode ser promovida nos espaços de decisão e de objetivos, uma vez que a proximidade de duas soluções no espaço de decisão não implica proximidade no espaço de objetivos, e vice-versa.



(a) (b)

Figura 11.13: Exemplos de resultados em otimização multiobjetivo (no espaço dos objetivos): (a) várias soluções distribuídas uniformemente e (b) soluções concentradas em algumas regiões.

A maioria dos AEDs capazes de tratar problemas de otimização multiobjetivo foi concebida tendo como inspiração os algoritmos NSGA-II (Deb et al., 2002) e SPEA2 (Zitzler et al., 2001).

Thierens e Bosman (2001) propuseram o *multiobjective mixture-based iterated density estimation algorithm* (mMIDEA), que utiliza seleção baseada no conceito de dominância. O algoritmo foi aplicado ao problema da mochila (Zitzler e Thiele, 1999) e vários outros com variáveis contínuas, obtendo bons resultados em todos os casos.

Laumanns e Ocenasek (2002) deram origem ao *multiobjective mixed BOA* (mmBOA) incorporando ao algoritmo BOA os operadores do SPEA2 e o conceito de uma segunda população. Quando aplicado ao problema da mochila, mmBOA produziu resultados que dominavam os resultados gerados pelo NSGA-II e SPEA2. Outra extensão do BOA foi proposta por Khan et al. (2002), só que desta vez utilizando o operador de seleção do NSGA-II.

Li e Zhang (2008) apresentaram um AED baseado em decomposição. Simultaneamente, o algoritmo resolve problemas mono-objetivos e depois agrega as soluções. Experimentos em problemas complexos mostraram que o algoritmo supera o NSGA-II significativamente.

Algoritmos capazes de se beneficiar das regularidades contidas na fronteira de Pareto para alguns tipos de problemas foram desenvolvidos por Zhang et al. (2008) e Zhou et al. (2008). Os algoritmos foram aplicados a vários tipos de problemas e comparados com outras propostas encontradas na literatura. Os resultados indicaram que os algoritmos capazes de identificar as regularidades do problema apresentam melhor desempenho.

Martí et al. (2008) e Sastry et al. (2005) apresentam as limitações dos AEDs quando aplicados a problemas de otimização multiobjetivos e apontam novas direções para a pesquisa nesta área.

Otimização em ambientes dinâmicos

No mundo real, muitos problemas são dinâmicos e requerem algoritmos capazes de se adaptarem às mudanças do ótimo global ao longo do tempo. Formalmente, a função objetivo para este tipo de problema é representada como $F(x, t)$, expressando a sua dependência não somente do vetor de variáveis x, mas também do tempo t.

Muitos trabalhos surgiram investigando a eficácia de AEDs junto a esta classe de problemas. Yang e Yao (2005) realizaram uma modificação no algoritmo PBIL, incorporando um mecanismo para manutenção de diversidade. Por meio de vários problemas, eles comparam PBIL com um algoritmo genético e levantaram os pontos positivos e negativos desse algoritmo quando aplicado a problemas de otimização dinâmica.

Liu et al. (2008) apresentaram uma versão do algoritmo UMDA que contém um mecanismo de manutenção de diversidade na população. O algoritmo proposto foi testado em vários *benchmarks* e os resultados mostraram que ele foi capaz de se adaptar às mudanças do ambiente rapidamente.

Fernandes et al. (2008) também adaptaram o UMDA adicionando um mecanismo de memória de soluções, a fim de manter diversidade e mesmo reinicializar o vetor de probabilidade quando ocorre mudança no ambiente. Resultados obtidos nos experimentos mostraram que o algoritmo é eficaz na resolução de problemas de otimização dinâmica, superando outros algoritmos evolutivos.

Já Yuan et al. (2008) propuseram um AED para problemas com variáveis contínuas. Eles estudaram um modelo gaussiano com matriz de covariância diagonal e, portanto, não expressam relacionamentos entre as variáveis. Os autores utilizam a estratégia de manutenção de diversidade na população, amostrando novas soluções aleatoriamente a cada n iterações.

Problemas de bioinformática

Algoritmos de Estimação de Distribuição têm se mostrado eficazes na resolução de uma variedade de problemas de bioinformática. Os pesquisadores nesta área alegam que o uso desta classe de algoritmos oferece vantagens sobre algoritmo evolutivos mais simples, como por exemplo, a possibilidade de estudar os modelos probabilísticos obtidos ao longo do processo de busca a fim de encontrar informações

adicionais. Devido à elevada dimensão do espaço de busca nesses tipos de problemas, a maioria dos trabalhos utiliza algoritmos mais simples, os quais consideram que as variáveis são independentes.

Peña, Lozano e Larrañaga (2004) apresentam a aplicação do UMDA para agrupar dados de expressão gênica. O algoritmo foi avaliado em conjuntos de dados artificiais e reais. Em ambos os casos, grupos de genes relevantes foram obtidos. Outra proposta de agrupamento utilizando o UMDA pode ser encontrada em (Cano et al., 2006).

Posteriormente, Palacios et al. (2006) aprimoraram o UMDA para trabalhar com *biclusterização* de dados de expressão gênica. Quando comparado com algoritmos genéticos, o algoritmo proposto apresentou melhores resultados.

Aplicações junto à predição de estrutura de proteínas usando modelos probabilísticos markovianos podem ser encontradas em (Santana et al., 2004) e (Santana et al., 2008b). Posteriormente, Santana et al. (2008a) combinaram um Algoritmo de Estimação de Distribuição com um processo de busca local para a mesma tarefa.

Para uma extensa lista de aplicações em bioinformática, consulte (Armañanzas et al., 2008).

Aprendizado de máquina

Os Algoritmos de Estimação de Distribuição estão sendo aplicados com sucesso também a problemas de aprendizado de máquina, especialmente em seleção de atributos, aprendizado de redes neurais artificiais e de sistemas nebulosos.

A seleção de atributos é um problema de busca e otimização multimodal em que os atributos possuem relacionamento. Por exemplo, considere um conjunto de dados com 10 atributos. Suponha também que foi verificado que qualquer subconjunto que possuir os atributos 3, 4 e 7, dentre outros, é tido como boa solução. Neste caso, a presença dos atributos 3, 4 e 7 no subconjunto forma um bloco construtivo (solução parcial). Inza et al. (2000) e Cantú-Paz (2002) trataram desse tipo de problema usando a abordagem *wrapper* (Kohavi e John, 1997). Os autores utilizaram AEDs para explorar o espaço de busca e o classificador naïve Bayes (Theodoridis e Koutroumbas, 2006) para avaliar cada solução candidata. Em seus algoritmos, o modelo probabilístico empregado foi a rede bayesiana.

Cotta et al. (2001) avaliaram as versões dos algoritmos UMDA e MIMIC para problemas com variáveis contínuas na tarefa de evoluir pesos de redes neurais artificiais MLP. Durante os experimentos, os autores compararam os algoritmos de estimação de distribuição com algoritmos genéticos e estratégias evolutivas. Já Galic e Höhfeld (1996) utilizaram o algoritmo PBIL para evoluir a arquitetura e os pesos de redes neurais.

O uso do UMDA e do MIMIC para a geração de regras SE-ENTÃO de sistemas nebulosos foi investigado por delaOssa et al. (2009). Os termos linguísticos para cada variável foram definidos previamente por meio de funções de pertinência triangulares distribuídas uniformemente ao longo do universo de discurso.

7. Novas Perspectivas em AEDs

Nesta seção, são apresentados alguns tópicos relevantes relacionados ao desenvolvimento de AEDs capazes de resolver problemas mais complexos, de explorar o espaço de busca da melhor forma e com um custo computacional associado reduzido. Os trabalhos citados a seguir representam as primeiras tentativas de se obter AEDs com estas características e, evidentemente, investigações futuras devem ser realizadas.

Informações adicionais sobre direções a serem estudadas no desenvolvimento de AEDs podem ser encontradas em (Santana et al., 2009) e (Sastry et al., 2006).

Algoritmos híbridos

No intuito de melhorar a capacidade de exploração do espaço de busca dos AEDs, surgiram algumas propostas combinando esta classe de algoritmos com técnicas simples de busca local (Li, Zhang, Tsang e Ford, 2004; Tang e Lau, 2005; Zhang et al., 2007, 2003). Mais recentemente, os AEDs estão sendo combinados com algoritmos que apresentam propriedades complementares, como evolução diferencial (Chen et al., 2008; Sun et al., 2005), sistemas imunológicos artificiais (Castro e Von Zuben, 2009a,b, 2008; Chang et al., 2009), enxame de partículas (Wang et al., 2009; Wang, 2007; Zhou et al., 2007) e algoritmos genéticos (Peña, Robles, Larrañaga, Herves, Rosales e Pérez, 2004; Robles et al., 2006, 2005).

Algoritmos paralelos

Visando distribuir o custo computacional entre vários processadores e, consequentemente, acelerar o tempo de execução do algoritmo, alguns trabalhos desenvolveram AEDs paralelos (Jaros e Schwarz, 2007; Ocenasek et al., 2006; Schwarz e Jaros, 2008). Geralmente, a arquitetura adotada é a mestre-escravo (do inglês *master-slave*), na qual existe um processador central incumbido de coordenar a divisão de tarefa entre os processadores escravos. Se o gargalo do algoritmo está na etapa de avaliação das soluções candidatas, esta tarefa é dividida entre p processadores escravos, enquanto o restante do algoritmo é executado no processador mestre. Por outro lado, se o gargalo está na construção do modelo probabilístico, o processador mestre realiza todas as outras etapas, deixando a construção do modelo sob responsabilidade dos processadores escravos.

Aprendizado incremental e esporádico do modelo probabilístico

Nos AEDs, a tarefa de construção do modelo probabilístico a cada iteração é, geralmente, a que consome mais tempo. Visando aliviar este custo computacional, alguns trabalhos têm se dedicado a projetar algoritmos em que o modelo probabilístico é construído esporadicamente, e não em todas as iterações (Pelikan et al., 2008b). O que permanece em todas as iterações é a atualização dos parâmetros numéricos do modelo probabilístico. Pelikan et al. (2008a) investigaram o aprendizado incremental do modelo probabilístico. A cada iteração, o modelo probabilístico obtido na iteração anterior é utilizado como ponto de partida.

8. Material Adicional

Em nenhum dos itens desta seção é apresentada uma lista exaustiva, sendo que podem existir outros bons *softwares*, congressos e periódicos que tratam de AEDs, além daqueles mencionados na sequência.

Softwares

- *Bayesian Optimization Algorithm* (BOA): `http://medal-lab.org/`.
- *Hierarchical Bayesian Optimization Algorithm* (hBOA): `http://medal-lab.org/`.
- *Mixed Bayesian Optimization Algorithm* (mBOA): `http://jiri.ocenasek.com/`.
- *Real-coded Bayesian Optimization Algorithm* (rBOA): `http://www.evolution.re.kr/`.
- *Multiobjective Real-coded Bayesian Optimization Algorithm* (mrBOA): `http://www.evolution.re.kr/`.

- *Regularity Model Based Multiobjective Estimation of Distribution Algorithm* (RM-MEDA): `http://dces.essex.ac.uk/staff/zhang/`.

Congressos

Existem duas principais conferências em que os pesquisadores na área de AEDs apresentam seus trabalhos:

- *IEEE Congress on Evolutionary Computation* (CEC).
- *Genetic and Evolutionary Computation Conference* (GECCO).

Periódicos

Os seguintes periódicos publicam com frequência trabalhos sobre AEDs:

- *Evolutionary Computation. MIT Press.*
- *IEEE Transactions on Evolutionary Computation. IEEE Press.*
- *Informations Sciences. Elsevier.*
- *International Journal of Approximate Reasoning. Elsevier.*

9. Conclusões

Neste capítulo, foram abordados os conceitos básicos de uma nova classe de algoritmos evolutivos, denominada Algoritmos de Estimação de Distribuição (AEDs). Em vez de explorar o espaço de busca evoluindo a população de soluções por meio de operadores de cruzamento e mutação, os AEDs utilizam um modelo gráfico probabilístico que representa a distribuição de probabilidade conjunta para as melhores soluções encontradas até então. A cada iteração, este modelo probabilístico é construído e, posteriormente, utilizado para amostrar novas soluções. Operando desta maneira, estes algoritmos são capazes de identificar as regularidades do problema e utilizar este conhecimento ao longo do processo de busca. Consequentemente, os AEDs podem manipular blocos construtivos (soluções parciais para o problema) de forma eficiente.

O capítulo apresentou as motivações que levaram ao desenvolvimento dos AEDs, seguido de um pseudocódigo para um AED canônico. Em seguida, foram apresentados os algoritmos mais importantes, conforme o modelo probabilístico utilizado para expressar o relacionamento entre as variáveis. Foram elencados os principais trabalhos reportados na literatura para problemas de otimização com variáveis contínuas, otimização multiobjetivo e otimização dinâmica. Algumas propostas na área de bioinformática e aprendizado de máquina também foram citadas. O capítulo forneceu também novas tendências para o desenvolvimento de AEDs com melhor capacidade de exploração do espaço de busca e com reduzido custo computacional. Por fim, fontes para informações adicionais foram incluídas, caso o leitor deseje se aprofundar no assunto.