

Volume 55, 2011

Editores

**Célia A. Zorzo Barcelos**

Universidade Federal de Uberlândia - UFU

Uberlândia, MG, Brasil

**Eliana X.L. de Andrade**

Universidade Estadual Paulista - UNESP

São José do Rio Preto, SP, Brasil

**Maurílio Boaventura**

Universidade Estadual Paulista - UNESP

São José do Rio Preto, SP, Brasil

A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa, também, a publicar livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7)**, as figuras em **eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja outros títulos publicados em formato e-book na página

<http://www.sbmac.org.br/notas.php>



Sociedade Brasileira de Matemática Aplicada e Computacional

2011



# MÉTODOS ESTOCÁSTICOS DE OTIMIZAÇÃO: ALGORITMOS GENÉTICOS E EVOLUÇÃO DIFERENCIAL

Milena Almeida Leite Brandão

milabrand@yahoo.com.br

Sezimária de Fátima Pereira Saramago

saramago@ufu.br

Faculdade de Matemática

Universidade Federal de Uberlândia



Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil

2011

Coordenação Editorial: Elbert Einstein Nehrer Macau

Coordenação Editorial da Série: Eliana Xavier Linhares de Andrade

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Copyright ©2011 by Milena Almeida Leite Brandão e Sezimária de Fátima Pereira Saramago.

Direitos reservados, 2011 pela SBMAC. A publicação nesta série não impede o autor de publicar parte ou a totalidade da obra por outra editora, em qualquer meio, desde que faça citação à edição original.

**Catálogo elaborado pela Biblioteca do IBILCE/UNESP**  
**Bibliotecária: Maria Luiza Fernandes Jardim Froner**

Brandão, Milena A. L.

Métodos Estocásticos de Otimização: Algoritmos Genéticos e Evolução Diferencial - São Carlos, SP : SBMAC, 2011, 69 p., 20.5 cm - (Notas em Matemática Aplicada; v. 55)

e-ISBN 978-85-86883-57-6

1. Otimização Estocástica
2. Algoritmos Genéticos
3. Evolução Diferencial

I. Brandão, Milena A. L. II. Saramago, Sezimária de F. P.

IV. Título. V. Série

CDD - 51

# Agradecimentos

Às pessoas que contribuíram de diferentes formas para a realização deste trabalho, principalmente aos alunos que participam do grupo de pesquisa em otimização.

À CAPES e à FAPEMIG pelo auxílio/bolsa concedidos durante a realização deste trabalho e das diversas pesquisas do grupo.



# Lista de Figuras

1.1	Representação de mínimos locais e global . . . . .	18
3.1	Esquemática de um cromossomo . . . . .	30
3.2	Exemplo de cruzamento por um único ponto . . . . .	35
3.3	Exemplo de mutação pontual . . . . .	36
3.4	Otimização utilizando Algoritmo Genético . . . . .	36
4.1	Processo de gerar o vetor doador $V^{(q+1)}$ para uma função objetivo bidimensional . . . . .	39
4.2	Cruzamento binomial . . . . .	41
4.3	Cruzamento exponencial . . . . .	42
5.1	Fluxograma do Método Evolução com Conjuntos Embaralhados(fonte: Duan <i>et al.</i> (1993), adaptado pelas autoras). . . . .	48
5.2	Arquitetura do Método Evolução com Conjuntos Embaralhados (fonte: Duan <i>et al.</i> (1993), adaptado pelas autoras). . . . .	50
5.3	Fluxograma do Método Evolução com Conjuntos Competitivos (fonte: Duan <i>et al.</i> (1993), adaptado pelas autoras). . . . .	52
7.1	Projeto de um recipiente de pressão . . . . .	55
7.2	Projeto de uma viga engastada . . . . .	57
7.3	Equilíbrio estático de um pendulo submetido a uma força $F$ . . . . .	62
7.4	Equilíbrio estático de um sistema de 2 molas. . . . .	63
7.5	Projeto de um mastro de bandeira. . . . .	64





# Lista de Tabelas

4.1	Estratégias do método Evolução Diferencial . . . . .	43
7.1	Comparação de alguns resultados do problema do recipiente de pressão	56
7.2	Resultados do problema de um recipiente de pressão para ECE. . .	57
7.3	Comparação de alguns resultados para o projeto de um viga engastada	59
7.4	Resultados do problema de uma viga engastada para ECE. . . . .	60
7.5	Funções propostas para o exercício 1. . . . .	61



# Conteúdo

<b>Prefácio</b>	<b>13</b>
<b>1 Introdução</b>	<b>15</b>
<b>2 Estratégias Evolutivas</b>	<b>21</b>
2.1 Estratégia (1+1) . . . . .	22
2.1.1 Propriedade de Convergência . . . . .	25
<b>3 Algoritmos Genéticos</b>	<b>27</b>
3.1 Codificação das variáveis do problema . . . . .	31
3.2 Seleção . . . . .	32
3.3 Cruzamento . . . . .	34
3.4 Mutação . . . . .	35
<b>4 Evolução Diferencial</b>	<b>37</b>
4.1 Operadores da Evolução Diferencial . . . . .	38
4.1.1 Mutação . . . . .	38
4.1.2 Cruzamento . . . . .	40
4.1.3 Seleção . . . . .	40
4.2 Parâmetros da evolução diferencial . . . . .	41
4.3 Estratégias da Evolução Diferencial . . . . .	42
<b>5 Evolução com Conjuntos Embaralhados</b>	<b>45</b>
5.1 O Método Evolução com Conjuntos Embaralhados - ECE . . . . .	46
5.2 O Método Evolução com Conjuntos Competitivos - ECC . . . . .	49
<b>6 Método de Penalidade Aplicado às Técnicas Evolutivas</b>	<b>53</b>

<b>7 Simulações Numéricas</b>	<b>55</b>
7.1 Projeto de um recipiente de pressão . . . . .	55
7.2 Projeto de uma viga engastada . . . . .	57
7.3 Exercícios Propostos . . . . .	60
<b>Bibliografia</b>	<b>65</b>

# Prefácio

A otimização representa uma ferramenta importante para tomada de decisão na análise e projeto de sistemas físicos, sendo aplicada em situações em que se deseja maximizar ou minimizar uma função numérica de várias variáveis, num contexto em que podem existir restrições.

Pode-se utilizar otimização em várias áreas, como, por exemplo, no projeto de sistemas ou componentes, planejamento e análise de operações, problemas de otimização de estruturas, otimização de forma, controle de sistemas dinâmicos. A grande vantagem é determinar a melhor configuração de projeto sem ter que testar todas as possibilidades envolvidas. Além disso, diminui o tempo dedicado ao projeto, possibilitando o tratamento simultâneo de uma grande quantidade de variáveis e restrições de difícil visualização gráfica ou tabular, possibilitando a obtenção de soluções não tradicionais com menor custo.

Técnicas clássicas de otimização são confiáveis e possuem aplicações nos mais diferentes campos de engenharia e de outras ciências. Porém, estas técnicas podem apresentar algumas dificuldades numéricas e problemas de robustez relacionados com: a falta de continuidade das funções a serem otimizadas ou de suas restrições, funções não convexas, multimodalidade, existência de ruídos nas funções, necessidade de se trabalhar com valores discretos para as variáveis, existência de mínimos ou máximos locais, etc. Assim, os estudos de métodos heurísticos, com busca randômica controlada por critérios probabilísticos, reaparecem como uma forte tendência nos últimos anos, principalmente devido ao avanço dos recursos computacionais, pois um fator limitante destes métodos é a necessidade de um número elevado de avaliações da função objetivo.

Os Algoritmos Evolucionários (AEs), que representam uma classe dos métodos heurísticos, imitam os princípios de evolução natural para fornecer procedimentos de otimização e busca e se baseiam em população de indivíduos, onde cada indivíduo representa um ponto de busca no espaço de soluções potenciais de um dado problema. As bases teóricas para os AEs foram propostas por John Holland na década de 1960 e desde então diversas técnicas evolucionárias vêm sendo desenvolvidas e aprimoradas. Os AEs tem como vantagens sua grande capacidade

de encontrar ótimos globais de funções com alto grau de complexidade e o fato de não exigir o cálculo de derivadas.

O objetivo deste trabalho é apresentar um estudo dos métodos de otimização natural denominados Algoritmos Genéticos e Evolução Diferencial. Para verificar a eficiência das técnicas estudadas, são utilizadas funções matemáticas clássicas e alguns problemas de engenharia. Estes métodos podem ser aplicados com eficiência a problemas de otimização multi-objetivo e na presença de restrições.

Uberlândia, 25 de janeiro de 2011.

Milena Almeida Leite Brandão  
Sezimária de Fátima Pereira Saramago

# Capítulo 1

## Introdução

Segundo Pirsig (1974) a “Otimização matemática é o título formal dada ao ramo da ciência computacional, que procura responder à pergunta Qual é o melhor? para os problemas em que a qualidade de qualquer resposta pode ser expressa como um valor numérico. Tais problemas surgem em todas as áreas da matemática, das ciências físicas, químicas e biológicas, engenharia, arquitetura, economia e gestão, e a gama de técnicas disponíveis para resolvê-los é quase tão grande quanto”.

Mas não é só nas ciências que se encontram problemas de otimização. Otimização é uma realidade que se aplica todos os dias, em quase todos os assuntos. Até mesmo os animais de uma forma instintiva aplicam a otimização no seu dia-a-dia. Um exemplo clássico disto são as abelhas. Esses insetos usam cera para construir os alvéolos das colméias, os quais são usados como depósitos para o mel que fabricam. Os alvéolos que compõem o favo de mel das abelhas europeias possuem um formato poliédrico. Mas por que não circular, já que o círculo é a forma geométrica que tem maior volume com menor área? Porque as abelhas constroem os alvéolos procurando a forma mais econômica, ou seja, que apresente maior volume para a menor porção de material gasto. Como os alvéolos não poderiam ser, por exemplo, cilíndricos, pois a falta de paredes comuns entre eles deixaria uma grande quantidade de espaços inaproveitáveis, a solução encontrada pelas abelhas é que os alvéolos devem ter a forma de um prisma, pois assim, as paredes de um alvéolo servem também aos alvéolos vizinhos, desta forma conseguem guardar um maior volume de mel com a mínima quantidade de material gasto. Este é apenas um dos vários exemplos encontrados na natureza.

A otimização é uma ferramenta importante para tomada de decisão durante a análise e o projeto de sistemas físicos. Para utilizar esta ferramenta torna-se necessário identificar qual o objetivo, ou seja, a quantidade que irá medir a performance do sistema (ex: energia, custo, potência, tempo, temperatura, etc.). O ob-

jetivo depende das características do sistema, que são as *variáveis* ou *parâmetros*, denominados variáveis de projeto ou variáveis de decisão. Assim, a otimização visa encontrar o valor das variáveis para otimizar o objetivo. Normalmente, estas variáveis devem obedecer a restrições, que são limites impostos ao sistema estudado.

O processo de identificação dos objetivos, variáveis e restrições é denominado *modelagem* do problema. Escrever um modelo adequado é muitas vezes o passo mais importante do processo de otimização. Se o modelo é muito simples, ele pode não representar o problema real. Por outro lado, se o modelo é muito complexo, ele pode trazer muitas dificuldades para a obtenção da solução.

Após a formulação do modelo, um algoritmo de otimização é utilizado para obter a solução do problema, usualmente com a ajuda de códigos computacionais. Não existe um algoritmo universal, mas um conjunto de métodos, entre os quais alguns são mais apropriados para determinadas aplicações específicas. A escolha do método de otimização depende do usuário e pode determinar a eficácia ou falha para a solução do problema.

Após a aplicação do algoritmo o projetista deve ser capaz de analisar a solução encontrada procurando identificar se o processo de otimização obteve sucesso ou se falhou. Em uma análise teórica, existem elegantes expressões matemáticas conhecidas como condições necessárias para existência do ponto ótimo (optimality condition) para verificar se a solução atual representa a solução ótima local do problema, mas nos problemas reais elas são difíceis de serem verificadas.

O modelo pode ser aperfeiçoado usando “análise de sensibilidade” que permite avaliar quais variáveis de projeto são mais relevantes no modelo que representa o problema.

Um problema de otimização é freqüentemente chamado de programação matemática. Os algoritmos de otimização são *iterativos*. Iniciam com uma estimativa inicial  $x^0$  e geram uma sequência de aproximações até encontrar o ponto mínimo. As estratégias utilizadas para se mover de uma iteração para outra é que distingue os diversos algoritmos. Muitas estratégias (métodos clássicos) usam o valor da função objetivo e suas derivadas. Algumas estratégias armazenam informações de iterações anteriores e outras consideram a informação apenas da iteração atual. Bons algoritmos de otimização devem ter as seguintes propriedades:

*Robustez*: devem ter uma boa performance para uma grande variedade de problemas e obter resultados razoáveis para quaisquer pontos de partida.

*Eficiência*: não devem exigir tempo computacional excessivo ou grande capacidade de armazenamento de dados.

*Precisão*: devem ser capazes de identificar a solução com precisão, sendo pouco sensíveis a erros nos dados ou arredondamentos numéricos quando o algoritmo é implementado no computador.



O problema de otimização pode ser escrito como:

$$\begin{array}{ll} \min f(x) & \text{sujeito a} \\ x \in \mathbb{R}^n & \begin{array}{l} c_i(x) = 0, i \in I \\ c_i(x) \geq 0, i \in D \end{array} \end{array} \quad (1.1)$$

sendo que,

- $x$  é o vetor das variáveis de projeto (ou variáveis de decisão, ou parâmetros do projeto);
- $f$  é a função objetivo (ou índice de performance, ou função custo);
- $c_i$  são funções de restrição (funções escalares de  $x$  que definem certas equações ou inequações que as variáveis devem obedecer);
- $I$  e  $D$  representam os conjuntos de índices das restrições de igualdade e desigualdade, respectivamente.

Existem diversos métodos de otimização e desta forma vários autores já apresentaram diferentes propostas para classificá-los, não sendo possível escolher uma proposta universalmente aceita. Os métodos dependem do tipo do problema em estudo (programação linear ou não-linear, problemas restritos ou irrestritos), das variáveis de projeto (contínuas ou discretas), do tipo de resposta desejada (otimização local ou global; projeto ótimo ou controle ótimo), das técnicas empregadas (determinísticas, estocástica ou híbrida). Além disso, os problemas podem ser classificados segundo o número de variáveis (pequenos ou grandes) ou segundo a suavidade das funções (diferenciáveis ou não-diferenciáveis). A seguir serão apresentadas as denominações mais comuns nesta área:

- *Problemas de programação linear ou não-lineares:* quando a função objetivo e todas as restrições são funções lineares de  $x$  (este tipo de problema é bastante comum em aplicações econômicas e nas áreas de transporte) denominam-se os problemas de programação linear.

Os problemas de programação não-lineares são aqueles onde ao menos uma das restrições ou a função objetivo são não-lineares, eles aparecem naturalmente nas ciências físicas e nas engenharias.

- *Otimização irrestrita ou restrita:* problemas de otimização irrestritos são aqueles onde  $I = D = \emptyset$ . Este caso pode acontecer quando: os problemas práticos não dependem de restrições; as restrições foram desprezadas porque não afetam diretamente a solução ou o problema com restrição foi reescrito de forma que as restrições foram substituídas por funções de penalidade.

Os problemas de otimização restritos, que aparecem devido às imposições do projeto, são aqueles onde  $I \neq \emptyset$  e  $D \neq \emptyset$ . As restrições podem ser lineares (ex:  $\sum x_i \leq 1$ ), laterais (ex:  $0 \leq x_i \leq 20$ ), não-lineares (ex:  $\sum x_i^2 - \text{sen} x_i \leq 0,5$ ), ou, ainda, combinações destas.

- *Otimização discreta ou contínua:* em alguns problemas de otimização as variáveis de projeto só têm sentido se forem considerados seus valores inteiros (ex: a variável  $x_i$  representa o número de caminhões em uma frota de transporte de cargas). Nestes tipos de problemas, deve-se incluir as “restrições de integralidade”, ou seja  $x_i \in \mathbb{Z}$  ( $\mathbb{Z}$  é o conjunto dos números inteiros). Pode-se também incluir “restrições binárias”, neste caso,  $x_i \in \{0, 1\}$ . Estes tipos de problemas é denominado problemas de otimização inteira ou *problemas de otimização discreta*. Neste caso, os problemas podem considerar além das variáveis inteiras ou binárias, variáveis abstratas, as quais são representadas por conjuntos finitos (que podem ser bastante grandes). Na *otimização contínua* as variáveis  $x_i \in \mathbb{R}$ .
- *Otimização global ou local:* a solução *global* é aquele ponto onde a função objetivo atinge o *menor* valor entre todos os pontos da região viável. A solução ou *ótimo local* é o ponto cuja função objetivo é menor que todos os pontos vizinhos da região viável, conforme representado na Figura 1.1. Em muitas aplicações a solução global é desejada, mas difícil de ser reconhecida. Para problemas *convexos* e na *programação linear* pode-se garantir que uma solução local é também solução global.

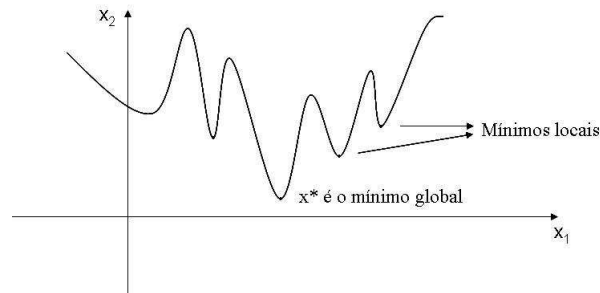


Figura 1.1: Representação de mínimos locais e global

- *Projeto ótimo ou controle ótimo:* segundo Arora (1987) o projeto ótimo e o controle ótimo de sistemas são duas atividades distintas. Existem numerosas aplicações onde a obtenção de projeto ótimo é útil para a concepção dos sistemas. Existem outras aplicações onde os conceitos de controle ótimo são necessários. Além disso, existem algumas aplicações onde tanto os conceitos de projeto ótimo quanto os de controle ótimo são usados. Uma amostra de onde isto ocorre inclui a robótica e estruturas aeroespaciais. Acontece que problemas de controle ótimo podem ser transformados em problemas de

projeto ótimo e tratados pelos métodos de otimização. Um problema de controle ótimo visa encontrar uma entrada controlada para o sistema produzir a saída desejada. Neste caso, o sistema tem componentes ativos que percebem as flutuações na saída. Os sistemas de controles são automaticamente ajustados para corrigir a saída e otimizar alguma medida de desempenho. Normalmente, o controle é aplicado em problemas de natureza dinâmica. Por outro lado, no projeto ótimo escreve-se o sistema e os componentes para otimizar a função objetivo. O sistema então permanece fixo durante toda a aplicação. Esta é a maior diferença entre as duas aplicações. Como um exemplo, considere o mecanismo de controle de viagem de um carro de passageiros. A idéia da resposta do sistema é controlar a injeção de combustível para manter a velocidade do carro constante. Assim, a saída do sistema é conhecida, ou seja, a velocidade da viagem. O trabalho do mecanismo de controle é perceber as flutuações na velocidade e ajustar adequadamente a injeção de combustível. A quantidade de injeção de combustível depende das condições da estrada. Quando o carro está numa subida, a injeção de combustível é maior do que em uma descida.

- *Otimização determinística ou estocástica*: nos problemas de *otimização determinística* o modelo é completamente conhecido. Os *algoritmos clássicos* de otimização, que dependem do conhecimento das derivadas da função objetivo, são exemplos de algoritmos determinísticos. Os melhores resultados destes métodos são para funções contínuas, convexas e semi-modais. Os métodos determinísticos possuem uma grande vantagem que é o baixo número de avaliações da função objetivo, fazendo com que tenham convergência rápida e baixo custo computacional.

Na *otimização estocástica* pode-se trabalhar com “incertezas” de algumas variáveis de projeto e produzir soluções que otimizam a “performance esperada” do modelo. Pode-se, também, garantir que as variáveis  $x$  satisfaçam às restrições para alguma probabilidade especificada anteriormente (modelos probabilísticos). Estes métodos também são conhecidos como métodos heurísticos. O termo heurístico tem sua origem na palavra grega “heuriskein” que significa “encontrar/descobrir”. Estes métodos utilizam apenas informações da função a ser otimizada, que pode ser de difícil representação, não-diferenciável, descontínua, não-linear, multimodal. Alguns exemplos dos métodos heurísticos são os métodos naturais, que tentam simular os processos usados na natureza para resolver problemas complexos. Entre estes métodos pode-se citar: Busca Tabu, Simulated Annealing, métodos baseados em população (Algoritmos Genéticos, Evolução Diferencial, Otimização por Bando de Pássaros, Otimização por Colônia de Formigas, etc). Estas técnicas trabalham com um conjunto de pontos, necessitam de muitas avaliações da

função objetivo e sua maior desvantagem é o alto custo computacional.

- *A otimização híbrida:* consiste na utilização conjunta de métodos determinísticos e heurísticos. Tem sido largamente utilizada nos dias atuais, visa unir as boas características de convergência dos métodos determinísticos às vantagens dos métodos heurísticos, por exemplo, a independência do tipo de função e o fato de não ficarem “presos” pelos mínimos locais.

## Capítulo 2

# Estratégias Evolutivas

A tarefa de imitar as estruturas e processos biológicos com o objetivo de resolver problemas técnicos é tão antiga quanto a engenharia. O mito de Dédalo e Ícaro comprova este empreendedorismo humano.

A estratégia evolutiva (EE) teve origem em 1964 na Universidade Técnica de Berlim, Alemanha. O problema original em estudo era o da otimização de forma para objetos inseridos em túneis de vento. Estratégias usando o método do gradiente não foram bem sucedidas. Dois estudantes, Ingo Rechenberg e Hans-Paul Schwefel tiveram a idéia de efetuar alterações randômicas nos parâmetros que definiam a forma dos objetos, baseados na idéia da seleção natural, este estudo resultou no mecanismo denominado (1+1), um esquema simples de seleção-mutação, que trabalha em um único indivíduo, que gera um único descendente por geração, através da mutação Gaussiana (Bäck *et al.* 1991). Mais tarde, esta teoria evoluiu para o chamado mecanismo  $(\mu + 1)$ , neste caso, uma população de  $\mu$  indivíduos se recombina de maneira randômica para formar um descendente, o qual, após sofrer mutação, substitui (se for o caso) o pior elemento da população. Ainda que este mecanismo nunca tenha sido muito usado, ele permitiu a transição para os mecanismos chamados  $(\mu + \lambda)$  e  $(\mu, \lambda)$ , já no final dos anos 70. Na primeira, os  $\mu$  ancestrais e os  $\lambda$  descendentes convivem, enquanto na segunda, os  $\mu$  ancestrais “morrem”, deixando apenas os  $\lambda$  descendentes “vivos”.

A regra de cada iteração  $k$  para minimizar uma função por meio de uma busca aleatória é:

$$x^{k+1} = \begin{cases} x^k + z^k & \text{se } f(x^k + z^k) \leq f(x^k) & \text{sucesso} \\ x^k & \text{caso contrário} & \text{falha} \end{cases} \quad (2.1)$$

O vetor aleatório  $z^k$ , que nesta notação efetua mudanças no vetor  $x^k$ , pertence à distribuição normal  $n$ -dimensional  $(0, \sigma^2)$  com esperança  $\xi = 0$  e variância  $\sigma^2$ ,

que no caso mais simples é a mesma para todas as componentes. Pode-se, assim, considerar  $\sigma$ , ou melhor  $\sigma\sqrt{n}$ , como um tipo de comprimento do passo médio. A direção de  $z^k$  é uniformemente distribuída em  $\mathbb{R}^n$ , isto é, sua escolha é puramente aleatória. Distribuições Gaussianas para os incrementos também são utilizados por Bekey *et al.* (1966), Stewart, *et al.* (1967) e DeGraag (1970). Gonzalez (1970) e White (1970) adotam, ao invés da distribuição normal, a distribuição uniforme, que considera uma pequena região na forma de um cubo  $n$ -dimensional centrado no ponto inicial.

A abordagem EE é adequada para uma vasta gama de problemas de otimização, porque não necessita de muitas informações sobre o problema. Ela é capaz de resolver problemas multidimensionais, multimodais e não lineares sujeitos a restrições lineares ou não lineares (Heitkoetter *et al.* 1994).

Na estratégia evolucionária, um indivíduo corresponde a um ponto no espaço de soluções, e possui um genótipo formado por variáveis objetivo e variáveis estratégicas. As variáveis objetivo são aquelas que, sofrendo recombinação e mutação, permitem incrementar a aptidão dos indivíduos em direção ao mínimo ou máximo global de otimização. As variáveis estratégicas representam variâncias e co-variâncias, que devem ser operadas junto às variáveis de controle para produzir mutações.

Existem vários tipos de estratégias evolutivas, podendo ser citadas:

- Dois-membros: (1 + 1)-EE
- Multimembros: ( $\mu$  + 1)-EE
- Multimembros: ( $\mu$  +  $\lambda$ )-EE
- Multimembros: ( $\mu$ ,  $\lambda$ )-EE

Por exemplo, na estratégia ( $\mu$  +  $\lambda$ )-EE tem-se que de ( $\mu$  +  $\lambda$ ) indivíduos,  $\mu$  serão selecionados, já na estratégia ( $\mu$ ,  $\lambda$ )-EE de  $\lambda$  indivíduos,  $\mu$  serão selecionados ( $\lambda \geq \mu$ ). Nas próximas seções serão discutidas algumas estratégias evolutivas.

## 2.1 Estratégia (1+1)

Na estratégia (1+1) uma solução gera outra a cada geração. Nessa operação é aplicada uma mutação normal (ou seja, pequenas alterações têm maior probabilidade de ocorrer do que grandes alterações, seguindo a distribuição normal) até que o descendente tenha um desempenho melhor que seu ascendente, quando então ele lhe toma o lugar. Assim, o filho (indivíduo mutado) é aceito na nova geração se e somente se ele possuir um custo melhor do que o do pai (se for viável).

Nos termos da biologia, as regras, segundo Schwefel (1995), são as seguintes:

1. **Inicializaçãõ:** Uma dada população é constituída por dois indivíduos, um pai e um descendente (filho). Cada um é identificado por seu genótipo de acordo com um conjunto de  $n$  genes. Apenas o genótipo parental tem de ser especificado como ponto de partida.
2. **Mutaçãõ:** O pai  $p^q$  da geração  $q$  produz um descendente  $d^q$ , cujo genótipo é ligeiramente diferente do seu. As variações se referem ao genes individuais, sendo aleatórios e independentes uns dos outros.
3. **Seleçãõ:** Por causa da diferença dos genótipos, os dois indivíduos têm uma capacidade diferente para a sobrevivência (no mesmo ambiente). Apenas um dos dois (pai ou descendente) pode produzir descendentes na próxima geração. O indivíduo que tiver a maior capacidade de sobrevivência se tornará o pai  $p^{q+1}$  da próxima geração  $q + 1$ .

Assim, algumas suposições podem ser feitas:

- O tamanho da população permanece constante;
- Um indivíduo tem, em princípio, um tempo infinitamente longo de vida e capacidade para produzir descendentes (assexuadamente);
- Não existem diferenças entre genótipo (codificação) e fenótipo (aparência), ou esta é inequívoca e reproduzível associada com o outro;
- Somente pontos de mutação ocorrem, independentemente de cada um de todos os parâmetros locais;
- O ambiente e, portanto, o critério de sobrevivência é constante ao longo do tempo.

No algoritmo, dado a seguir, os índices  $p$  representa pai,  $d$  representa descendente (filho) e  $q$  representa a  $q$ -ésima geração:

$$\begin{cases} \min f(X), & X \in \mathbb{R}^n \\ \text{sujeito a } C_j(X) \geq 0, & j = 1, \dots, m \end{cases} \quad (2.2)$$

**Passo 1:** (Inicialização) Defina  $X_p^0$  tal que  $C_j(X^0) \geq 0$  para todo  $j = 1, \dots, m$ .

**Passo 2:** (Mutaçãõ) Faça  $X_d^q = X_p^q + z^q$  com componentes  $X_{d,i}^q = X_{p,i}^q + z_i^q$  para todo  $i = 1, \dots, n$ .

**Passo 3:** (Seleçãõ) Escolha

$$X_p^{q+1} = \begin{cases} X_d^q & \text{se } f(X_d^q) < f(X_p^q) \text{ e } C_j(X_d^q) \geq 0 \quad \forall j = 1, \dots, m \\ X_p^q & \text{caso contrário} \end{cases}$$

Faça  $q = q + 1$  e vá para o passo 1 até que algum critério de parada seja satisfeito.

Apesar de existir um único indivíduo na população, este procedimento é denominado de estratégia evolutiva de dois membros pois o filho compete com o pai. Sendo assim, durante o processo de seleção (competição) existem, temporariamente, dois indivíduos na população.

A questão que permanece é como escolher os vetores aleatórios  $z^g$ , denominados fatores de mutação. Esta escolha tem a regra de mutação. As mutações, como são entendidas hoje em dia, devem ser aleatórias, eventos sem finalidade (purposeless events), que, além disso, só ocorrem muito raramente. Se interpretado como uma soma de muitos eventos individuais (como é feito aqui), é natural escolher uma distribuição de probabilidade de acordo com a observação biológica de que pequenas variações ocorrem com maior frequência do que grandes variações, e de que os filhos herdam características dos pais, ou seja, são parecidos com os pais (teorema do limite central das estatísticas).

Para variações discretas, pode-se usar a *distribuição binomial*, por exemplo, e para variações contínuas a *distribuição Gaussiana* ou *distribuição normal*.

Dois exigências surgem por analogia com a evolução natural:

- Que o valor da esperança  $\xi_i$  para cada componente  $z_i$  vale zero;
- Que a variância  $\sigma^2$ , o desvio médio quadrado para a média, é pequena.

A função densidade de probabilidade para eventos aleatórios distribuídos normalmente vale:

$$w(z_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(z_i - \xi_i)^2}{2\sigma_i^2}\right) \quad (2.3)$$

Se  $\xi_i = 0$ , obtém-se a chamada distribuição normal  $(0, \sigma_i^2)$ . Há ainda, contudo, um total de  $n$  parâmetros livres  $\{\sigma_i, i = 1, \dots, n\}$  para especificar os desvios-padrão dos componentes aleatórios individuais. Por analogia, nas estratégias de busca determinísticas,  $\sigma_i$  pode ser chamado tamanho do passo, no sentido de que eles representam valores médios dos comprimentos dos passos aleatórios.

Para a ocorrência de um vetor aleatório particular  $z = \{z_i, i = 1, \dots, n\}$ , cujas componentes são distribuídas independentemente  $(0, \sigma_i^2)$ , a função densidade de probabilidade é dada por:

$$w(z) = w(z_1, \dots, z_n) = \prod_{i=1}^n w(z_i) = \frac{1}{(2\pi)^{n/2} \prod_{i=1}^n \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^n \left(\frac{z_i}{\sigma_i}\right)^2\right) \quad (2.4)$$



ou mais compactamente, se  $\sigma_i = \sigma$  para todo  $i = 1, \dots, n$ ,

$$w(z) = \left( \frac{1}{\sigma\sqrt{2\pi}} \right)^n \exp\left( \frac{-zz^T}{2\sigma^2} \right) \quad (2.5)$$

O próprio Darwin (1874) salientou que a evolução dos seres vivos não é um processo puramente aleatório. No entanto, contra a sua teoria da descendência, a polêmica ainda está travada na impossibilidade de se demonstrar que a vida poderia surgir de um processo puramente aleatório. Mesmo ao nível da simples imitação da evolução orgânica, uma escolha adequada do tamanho do passo ou das variâncias são de fundamental significância.

### 2.1.1 Propriedade de Convergência

Sob o ponto de vista puramente teórico, se todos os componentes do vetor de desvios padrões  $\sigma$  forem idênticos, ou seja,  $\sigma_i = \sigma, \forall i$ , e o problema de otimização for regular, então é possível provar o teorema da convergência e calcular a taxa de convergência. Para isto, será dado a seguir a definição de um problema de otimização regular.

**Definição:** Um problema de otimização é dito regular, se e somente se, as seguintes condições forem satisfeitas (Born, 1978):

- a função  $f$  é contínua;
- o domínio  $D$  da função  $f$  é um conjunto fechado;
- $\forall x_0 \in D, D_0 := \{x \in D : f(x) \leq f(x_0)\}$  é um conjunto fechado;
- $\forall \epsilon > 0, L_{f^*+\epsilon}^0 := \text{int}(L_{f^*+\epsilon})$ , onde  $\text{int}$  denota o conjunto de todos pontos interiores, e  $L_{f^*+\epsilon} := \{x \in D : f(x) \leq f^* + \epsilon\}$  é um conjunto nível de  $f$ .

Observe que na definição acima  $L_a := \{x \in D : f(x) = a\}$ .

**Teorema 2.1.** Para  $\sigma > 0$  e dado um problema regular de otimização com  $f^* > -\infty$  (minimização) ou  $f^* < \infty$  (maximização), vale a seguinte propriedade:

$$p \{ \lim_{k \rightarrow \infty} f(x^k) = f^* \} = 1$$

A demonstração deste teorema é dada em Born (1978).

O Teorema 2.1 garante que o ótimo global será determinado com probabilidade 1 caso o tempo de busca (quantidade de iterações) seja suficientemente grande. Entretanto, devido aos recursos computacionais, e portanto temporais, limitados, a determinação da taxa de convergência do algoritmo é mais importante sob o ponto de vista prático do que a garantia de convergência num tempo infinito.

A taxa de convergência é definida como sendo o quociente entre a distância percorrida até o ótimo e a quantidade de gerações (iterações) necessárias para percorrer esta distância.

Para otimizar a taxa de convergência, Rechenberg (1973) propôs uma regra denominada *regra do 1/5 para o sucesso*. A razão  $\varphi$  entre as mutações que geram um indivíduo melhor do que o pai (mutações positivas) em relação a todas as mutações deve ser 1/5, isto é,

$$\varphi = \frac{\text{número de mutações positivas}}{\text{número total de mutações}} = 1/5.$$

Devido à relativa simplicidade da estratégia (1 + 1) é que esta regra pode ser aplicada (Heitkoetter *et al.* (1994)). Pode-se enunciá-la como segue:

**Regra do 1/5 para o sucesso:** *quando a taxa de sucesso (o descendente tem aptidão maior do que o ascendente) for maior do que 1/5, após a mutação, então a variância da distribuição normal deve ser aumentada. Caso esta taxa seja menor do que 1/5, então a variância deve ser diminuída (Schwefel (1995)).*

É importante mencionar que esta regra foi proposta baseado em um estudo feito para calcular as taxas de convergência de duas funções modelos: a função corredor, e a função da esfera.

Embora, em geral, os problemas de interesse podem ter características distintas das funções corredor e esfera, a seguinte heurística pode ser utilizada para fazer um ajuste dinâmico de  $\sigma$  a cada  $q$  gerações:

$$\sigma^{k+1} = \begin{cases} c_d \sigma^k & \text{se } \varphi(q) < 1/5 \\ c_i \sigma^k & \text{se } \varphi(q) > 1/5 \\ \sigma^k & \text{se } \varphi(q) = 1/5 \end{cases}$$

Em seus experimentos, Schwefel (1981) utiliza os seguintes valores:  $c_d = 0,82$  e  $c_i = 1,22$ .

Note que cada indivíduo detém a sua própria variância e eventualmente a covariância (que são os parâmetros básicos da mutação), o que caracteriza o conceito de auto-adaptação.

Observe que, nas estratégias de busca determinísticas, tanto a direção como o comprimento do passo são determinados no processo de uma forma fixa, ou com base em informações anteriormente recolhidas e suposições plausíveis sobre a topologia da função objetivo. Já na estratégia evolutiva, a direção é puramente aleatória e o comprimento do passo, exceto para um número pequeno de variáveis, é praticamente fixo.

## Capítulo 3

# Algoritmos Genéticos

Charles Darwin, estudando as espécies e suas evoluções, coletou durante anos uma grande quantidade de material que comprovaram, principalmente, a existência de inúmeras variações em cada espécie. Seus estudos, associados às pesquisas de outros cientistas do assunto, tornaram evidentes que as espécies animais se modificam. Um dos principais pontos dos estudos de Darwin foi o aspecto das variações apresentadas entre indivíduos da mesma espécie. Através de estudos em pombos, por exemplo, ele observou a enorme variedade de indivíduos que se obtém, cruzando uma mesma espécie. Segundo esse estudioso, todas as novas espécies são produzidas por meio de uma seleção natural.

Há quase cinco bilhões de anos, a natureza vem resolvendo problemas com sucesso. Cada organismo possui cromossomos, genes, exons, íntrons e códons, constituindo um sistema genético. Um determinado grupo de indivíduos que vivem junto, constituem uma população. Nesta população existem organismos mais hábeis, que são os que têm maior chance de gerar bons descendentes. Estes descendentes são mais aptos do que a média da população, pois receberam genes melhores. Darwin mostrou que apenas os mais adaptados ao nicho ecológico da população irão sobreviver. Estes transmitirão suas características às gerações subsequentes, melhorando a cada geração a aptidão dos seus descendentes.

Por volta de 1900, o trabalho de Gregor Mendel, desenvolvido em 1865, sobre os princípios básicos de herança genética, foi redescoberto pelos cientistas e teve grande influência sobre os futuros trabalhos relacionados à evolução. A moderna teoria da evolução combina a genética e as idéias de Darwin e Wallace sobre a seleção natural, criando o princípio básico de Genética Populacional: a variabilidade entre indivíduos de uma população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética.

Este princípio foi desenvolvido durante os anos 30 e 40, por biólogos e mate-

máticos de importantes centros de pesquisa. Nos anos 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi John Holland quem começou, seriamente, a desenvolver as primeiras pesquisas no tema. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro *Adaptation in Natural and Artificial Systems*, hoje considerado a Bíblia dos Algoritmos Genéticos. O método foi desenvolvido por Holland durante cursos nas décadas de 60 e 70, mas foi um dos seus alunos, David Goldberg, quem popularizou o método ao resolver um difícil problema envolvendo o controle de transmissão de gás em uma tubulação, em sua dissertação em 1989. Desde então, estes algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização e aprendizado de máquina.

Segundo Goldberg (1989), os Algoritmos Genéticos (AGs) diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos:

1. AGs trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros.
2. AGs trabalham com uma população e não com um único ponto.
3. AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
4. AGs utilizam regras de transição probabilísticas e não determinísticas.

Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, mas aleatória, que é voltada em direção ao reforço da busca de pontos de “alta aptidão”, ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos).

Apesar de aleatórios, eles caminham de forma direcionada, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada iteração é chamada de geração.

Durante cada iteração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Através da seleção, se determina quais indivíduos conseguirão se reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pela seu índice de aptidão. Em outras palavras, os indivíduos com maior adaptação relativa têm maiores chances de se reproduzir.

Inicialmente, é gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. Durante o processo evolutivo, esta população é avaliada: para cada indivíduo é dada uma

pontuação, ou índice, refletindo sua habilidade de adaptação a determinado ambiente. Uma porcentagem dos mais adaptados é mantida, enquanto os outros indivíduos são descartados (darwinismo). Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamento (crossover) ou recombinação genética gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada.

Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são suficientemente complexos para fornecer mecanismos adaptativo de busca poderosos e robustos.

Para realizar o processo de otimização via Algoritmo Genético é necessário seguir seis operações básicas que compreendem:

1. codificação das variáveis do problema;
2. criação dos indivíduos iniciais, ou seja, formar a população inicial;
3. avaliar os indivíduos da população;
4. selecionar tais indivíduos, de acordo com critérios estabelecidos;
5. realizar o cruzamento entre os indivíduos e;
6. aplicar mutação nestes indivíduos.

As operações seleção, cruzamento e mutação são repetidas até atingir algum critério de parada. Finalmente, o melhor indivíduo é apresentado como a solução do problema.

Na terminologia natural, cromossomos são compostos de genes, que podem tomar alguns números de valores chamados alelos. A posição de um gene (locus) é identificada separadamente a partir da função que o gene assume, desta forma, pode-se falar de um gene particular. Por exemplo, no gene que determina a cor dos olhos de um animal, o locus é a posição 10 e o valor do alelo, olhos azuis. Assim, gene é a característica e o alelo é o valor da característica.

Como os algoritmos genéticos usam um vocabulário emprestado da genética natural, em seguida é dada uma breve descrição dos termos mais comuns:

- Cromossomo (genótipo): Estrutura que representa as variáveis do problema e é chamado de indivíduo, o qual forma a população do Algoritmo Genético. Cada cromossomo corresponde a uma solução do problema dentro do espaço de busca (um ponto do espaço).
- Gene: São os constituintes dos cromossomos, ou seja, cada cromossomo é formado por um conjunto de genes. Cada gene representa uma variável de

projeto  $x_i$ . A quantidade de genes em um cromossomo é sempre um número par.

- Alelos: São as unidades presentes nos genes, cada gene é composto por  $m$  alelos, donde  $m$  é o comprimento do gene.
- Fenótipo: Representa um cromossomo decifrado.
- Operações Genéticas: Operações realizadas sobre os cromossomos através de sucessivas iterações. Estas incluem seleção, cruzamento e mutação.
- Geração: Corresponde à iteração atual do Algoritmo Genético.

A Figura 3.1 exemplifica estas definições.

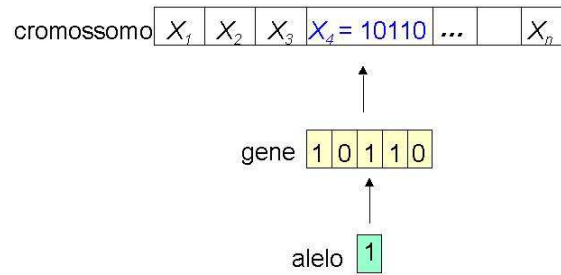


Figura 3.1: Esquematização de um cromossomo

O Algoritmo Genético para otimização de parâmetros tem sido apresentado da seguinte forma (Schwefel, 1995):

**Passo 1:** (Inicialização)

Uma dada população é composta por  $\lambda$  indivíduos. Cada um é caracterizado pela seu genótipo composto por  $n$  genes, que determinam a vitalidade, ou adaptação (fitness) para sobrevivência. O genótipo de cada indivíduo é representado por uma seqüência de bits (binário), representando os valores dos parâmetros objetivo diretamente ou por meio de um esquema de codificação.

**Passo 2:** (Seleção)

Dois pais são escolhidos com probabilidade proporcional à sua posição relativa na população atual, ou medidos por sua contribuição para o valor médio da função objetivo na geração (seleção proporcional) ou por sua posição (seleção ranking linear).

**Passo 3:** (Cruzamento)

Dois descendentes diferentes são produzidos pela recombinação de dois genótipos parentais por meio de cruzamento de uma dada probabilidade de recombinação  $p_c$ . Apenas um dos filhos (aleatoriamente) é tomado em consideração nesta etapa. Os passos 1 e 2 são repetidos até que  $\lambda$  indivíduos representem a próxima geração.

**Passo 4:** (Mutação)

Os descendentes (com uma dada probabilidade  $p_m$  fixa e pequena) sofrem outras alterações subjacentes por meio de mutações pontuais de trabalho em bits individuais, quer invertendo 1 por 0, ou vice-versa, ou lançando um dado para a escolha de 0 ou 1, independentemente do valor original.

Observe que à primeira vista, o esquema parece muito similar a estratégia evolutiva multimembros. Para revelar as diferenças é preciso analisar os operadores seleção (S), mutação (M) e recombinação (R). A sequência dos eventos no Algoritmo Genético (S-R-M) é oposta da sequência (M-R-S) utilizada na estratégia evolutiva multimembros, o que não causa uma diferença muito significativa já que todo o processo é circular. Inverter a ordem de mutação e recombinação é uma questão de evitar operações desnecessárias ou não. Em aplicações, a avaliação dos indivíduos com relação aos seus correspondentes valores da função objetivo normalmente domina todas as outras operações. Os valores canônicos adotados são:  $p_c = 0,6$  para a probabilidade de cruzamento,  $n_c = 2$  para o número de pontos de cruzamento e  $p_m = 0,001$  para a probabilidade de mutação.

Em seguida será estudado detalhadamente cada operador.

### 3.1 Codificação das variáveis do problema

Para trabalhar com Algoritmo Genético é preciso codificar as informações do problema em uma estrutura denominada cromossomo, de modo que o método consiga trabalhar com os dados que lhe são fornecidos.

O processo de codificação das variáveis pode ser realizado por meio de uma representação binária em que cada valor associado representa a presença (valor 1) ou não (valor 0) de certa característica no indivíduo.

Na representação binária, uma variável de projeto  $x_i$  é descrita por um vetor de  $m_i$  bits (*zero ou um*). Desse modo, para  $m_i = 5, \forall i = 1, \dots, n$ , um cromossomo binário é descrito como:

$$\text{cromossomo} = [x_1 x_2 \dots x_n] = [1101100110 \dots 11001] \quad (3.1)$$

sendo que

$$\begin{aligned} x_1 &= 11011 \\ x_2 &= 00110 \\ &\vdots \\ x_n &= 11001 \end{aligned}$$

Na equação 3.1, cada  $x_i$  equivale a um gene e os alelos são representados por cada um de seus bits, por exemplo, os alelos de  $x_1$  são 1, 1, 0, 1 e 1. O processo de decodificação dos cromossomos, isto é, conversão da sequência binária (base 2) em base 10, é realizado pela seguinte função matemática (Oliveira, 2006):

$$x_i = x_i^{inf} + \left( \sum_{j=1}^{m_i-1} x_i(j)2^j \right) \left( \frac{x_i^{sup} - x_i^{inf}}{2^{m_i} - 1} \right) \quad (3.2)$$

onde  $x_i^{inf}$  e  $x_i^{sup}$  são os limites inferior e superior da variável  $x_i$  e

$$m_i = \log_2 \left( \frac{x_i^{sup} - x_i^{inf}}{prec_s} \right) \quad (3.3)$$

sendo  $prec_s$  a precisão requerida para a solução do problema e  $m_i$  o tamanho do  $i$ -ésimo gene.

## 3.2 Seleção

Definido o processo de codificação das variáveis do problema, torna-se necessário criar os indivíduos que irão formar a população inicial. Geralmente os indivíduos iniciais da população são criados de maneira totalmente aleatória, ou através de uma distribuição uniforme pelo espaço de busca, ou ainda pode ser definida pelo operador.

O tamanho da população é importante para o método, pois populações pequenas reduzem o espaço de busca e podem antecipar a solução para um ótimo local, por outro lado, se a população é muito grande pode acarretar em lentidão operacional (Haupt & Haupt, 1998). Assim, a escolha do tamanho da população deve ser feito com cautela e baseada em pré-simulações.

Com a população definida, precisa-se avaliar o desempenho de cada indivíduo desta população, ou seja, é necessário selecionar os indivíduos mais adaptados. Este processo é feito simplesmente pelo cálculo da função objetivo, também chamada função custo ou função de avaliação.



Assim, a seleção é um processo que será atribuído às cadeias que possuem o maior valor objetivo e, portanto uma probabilidade mais elevada de contribuir à geração seguinte, criando pelo menos um descendente. Quanto maior o valor da função objetivo, maiores são as chances do indivíduo sobreviver no ambiente e reproduzir-se passando parte de seu material genético a gerações posteriores (Braga, 1998).

Existem algumas técnicas para realizar a escolha destes indivíduos. A *seleção Elitista* consiste em substituir os piores indivíduos da próxima geração pelos melhores indivíduos da geração atual. Esta estratégia possui como vantagem principal manter o melhor indivíduo na população, desta forma, fornece uma seqüência em que o resultado permanece o mesmo ou é melhorado. A desvantagem está em cair prematuramente em um ótimo local.

A *seleção Aleatória* consiste em selecionar os indivíduos de maneira totalmente aleatória sem levar em consideração sua aptidão na população. Segundo Soares (1997) fazer a escolha desta maneira não trará resultados satisfatórios para boa parte das aplicações.

Outra estratégia é a *seleção por Torneio*, na qual grupos, contendo dois indivíduos cada, são formados pela escolha aleatória de indivíduos da população. O melhor indivíduo de cada grupo é selecionado baseando-se em uma probabilidade previamente definida. Então, gera-se um número aleatório no intervalo real  $[0, 1]$ , se este número for menor que a probabilidade, significa que o melhor indivíduo será o escolhido, caso contrário, o outro será selecionado.

O processo de seleção utilizado neste trabalho será a *seleção pela Roleta*, no qual a chance de um indivíduo ser selecionado está proporcionalmente ligada a sua aptidão fornecida através da função objetivo. Em linhas gerais, cada indivíduo da população é representado em uma roleta de acordo com a relação entre o seu valor de aptidão e a soma total das aptidões dos indivíduos, gerando, então, uma porcentagem da roleta que o indivíduo ocupa. Assim, define-se

$$p_j = \frac{f(X_j)}{\sum_{j=1}^{N_p} f(X_j)}, \quad i = 1, \dots, N_p \quad \text{e} \quad q_i = \sum_{j=1}^{N_p} p_j \quad (3.4)$$

onde  $X_j = (x_1 x_2 \dots x_n)$ ,  $j = 1, \dots, N_p$  sendo que  $N_p$  é o número de indivíduos na população.

Em seguida, para selecionar um indivíduo, geram-se  $N_p$  números aleatórios pertencentes ao intervalo  $[0, 1]$ . Para cada indivíduo na roleta, faz-se a comparação entre o valor de sua porcentagem  $q_i$  com o número aleatório gerado. Caso a porcentagem na roleta seja maior que o número gerado, este indivíduo será selecionado, senão passa para o próximo indivíduo até que todos os números aleatórios tenham sido comparados. Note que a seleção por Roleta privilegia os indivíduos mais aptos, dado que estes podem ser selecionados mais de uma vez, ao passo que os piores

indivíduos são descartados. Depois de selecionados, os cromossomos dão origem a uma nova população que irá participar da próxima etapa do Algoritmo Genético, denominada cruzamento.

### 3.3 Cruzamento

O cruzamento é um procedimento no qual indivíduos trocam material genético formando seus descendentes, ou seja, novos indivíduos. Geralmente, este processo ocasiona a propagação das características dos indivíduos mais aptos da população, além de fazer o método buscar soluções em regiões não conhecidas do espaço de projeto. Além disso, o processo de cruzamento é uma condição necessária para que a convergência do método seja alcançada ao longo das gerações (Rodrigues, 2000).

Assim como na seleção, existem várias maneiras de se fazer o cruzamento. Neste trabalho será utilizado a seguinte técnica para se fazer o cruzamento:

Dois indivíduos, caracterizados como pais, são escolhidos através de uma probabilidade de cruzamento  $p_c$ , mencionada anteriormente. Desta forma, estes indivíduos através de alguma estratégia de cruzamento trocam seu material genético. Uma das estratégias de cruzamento mais utilizadas é a de estabelecer um ponto de cruzamento (single-point crossover). A partir deste ponto é realizada a troca de alelos entre os cromossomos pais.

O ponto  $k$ , que define a posição de cruzamento na cadeia de bits de cada cromossomo, será escolhido aleatoriamente. A quantidade de cromossomos a ser submetida ao processo de cruzamento é definida através da probabilidade de cruzamento  $p_c$ , especificada pelo usuário. Cada cadeia é partida neste ponto  $k$  e todas as informações do cromossomo Pai 1, a partir do ponto escolhido, são copiadas para o cromossomo Pai 2 e vice-versa, conforme pode ser observado na Figura 3.2.

O processo de escolha de quem será cruzado deve ser feito em pares, sorteando números randômicos  $r_i$  entre 0 e 1. Quando não for possível formar os pares um novo sorteio deverá ser feito até obter os pares necessários para o cruzamento. Se  $r_i$  for menor que a probabilidade  $p_c = 0,6$ , então o cromossomo  $C_i$  será selecionado.

Um novo número randômico deve ser gerado para determinar a posição  $k$ , onde duas novas cadeias são formadas pela troca de todos os caracteres compreendidos entre as posições  $k + 1$  e  $\sum_{i=1}^{N_p} m_i$ . Esta posição  $k$  é determinada pela seguinte fórmula:

$$k = 1 + r \left( \sum_{i=1}^{N_p} (m_i - 1) - 1 \right) \quad (3.5)$$

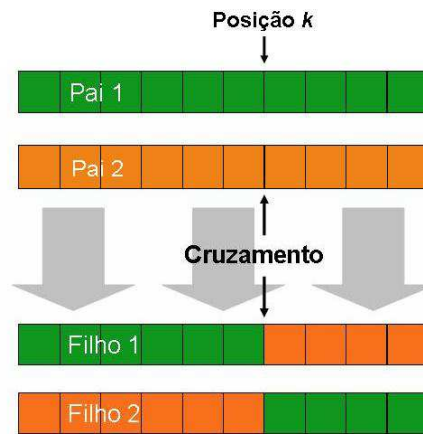


Figura 3.2: Exemplo de cruzamento por um  nico ponto

sendo  $r$  um n mero aleat rio entre  $[0; 1]$  e  $m_i$  representa o tamanho do  $i$ - simo alelo do cromossomo. Outra possibilidade para este tipo de cruzamento  , ao inv s de fazer o cruzamento escolhendo um  nico ponto, efetuar a escolha de mais pontos e, ent o, seguir a mesma metodologia.

### 3.4 Muta o

Se alguns cuidados n o forem tomados, o Algoritmo Gen tico pode convergir muito rapidamente para uma regi o da superf cie de custo. Se esta  rea est  na regi o do m nimo global  timo. No entanto, uma vasta quantidade de fun es tem muitos m nimos locais. Se nada for feito para resolver esta tend ncia de convergir rapidamente, pode acontecer do algoritmo encontrar um m nimo local ao inv s de um m nimo global. Para evitar esse problema de converg ncia excessivamente r pida, for a-se o algoritmo a explorar outras  reas da regi o de busca por introduzir mudan as aleat rias, ou muta es, em algumas vari veis. Para o AG bin rio, esta muta o   realizada mudando o bit de 0 para 1, ou vice-versa.

O operador muta o   necess rio para a introdu o e manuten o da diversidade gen tica na popula o. Alterando arbitrariamente um ou mais alelos de um indiv duo escolhido, a muta o fornece meios para introdu o de diferentes indiv duos na popula o.

Uma, das v rias estrat gias utilizadas pelo operador muta o, considerando a representa o bin ria, corresponde a uma muta o pontual, ou muta o de ponto  nico. Na muta o pontual seleciona-se randomicamente uma posi o em um

cromossomo, obedecendo a probabilidade de mutação  $p_m$ , e muda o valor deste bit. Se o número randômico gerado for menor que a taxa de mutação,  $p_m = 0,01$ , o bit será invertido, isto é, se for zero vira um e vice-versa. Observe que é necessário gerar  $N_p \cdot \sum_{i=1}^{N_p} m_i$  números randômicos  $r$  entre 0 e 1. Se  $r$  for menor que a probabilidade  $p_m = 0,01$  será feita a mutação no bit correspondente. A Figura 3.3 mostra a operação de mutação sendo realizada.

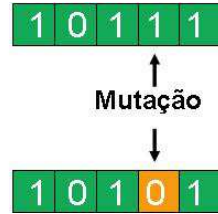


Figura 3.3: Exemplo de mutação pontual

Ao realizar todos estes passos, codificação, seleção, cruzamento e mutação, completa-se uma iteração, ou geração, do Algoritmo Genético. As operações descritas são repetidas até que algum critério de parada seja obedecido. O indivíduo com melhor valor de função objetivo será selecionado como sendo a solução do problema. A Figura 3.4 mostra o esquema de funcionamento do Algoritmo Genético.

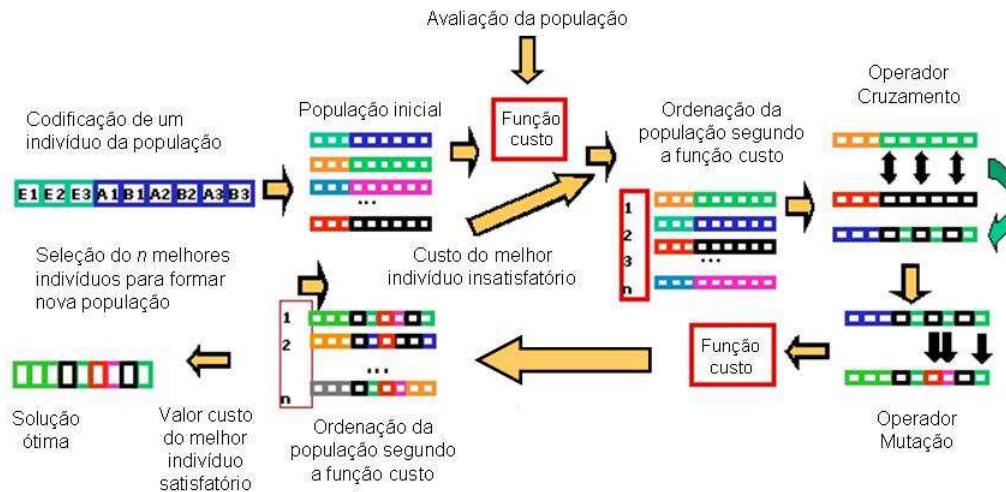


Figura 3.4: Otimização utilizando Algoritmo Genético

## Capítulo 4

# Evolução Diferencial

A Evolução Diferencial (ED) foi criada após Ken Price tentar usá-la para resolver o problema do polinômio de Chebychev o qual foi apresentado por Rainer Storn. A descoberta aconteceu quando Price teve a idéia de utilizar diferentes vetores para recriar o vetor população criando uma abordagem diferente das encontradas nas estratégias de evolução que tratam de problemas de otimização. Desde então foram feitos vários testes e aperfeiçoamentos os quais tornaram o algoritmo ED versátil e robusto.

O algoritmo é iniciado criando uma população inicial de  $N_p$  indivíduos, escolhida aleatoriamente e devendo cobrir todo o espaço de busca. Geralmente, é criada por uma distribuição de probabilidade uniforme, quando não há nenhum conhecimento sobre o problema.

Cada indivíduo, chamado de *vetor*, possui  $n$  componentes representadas por valores reais, sendo  $n$  o número de variáveis de projeto. Assim, a população segue uma evolução natural, em que o número de indivíduos é constante durante todas as gerações.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada *mutação*.

As componentes do indivíduo doador são misturadas com as componentes de um indivíduo escolhido aleatoriamente (denotado vetor alvo), para resultar o chamado vetor tentativa, ou vetor experimental. O processo de misturar os parâmetros é referido como *cruzamento*. Se o vetor experimental resultar um valor da função objetivo menor que o vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte. Esta última operação é chamada *seleção*. O procedimento é finalizado quando algum critério de parada é obedecido.

## 4.1 Operadores da Evolução Diferencial

Assim como o Algoritmo Genético, a Evolução Diferencial apresenta algumas operações a serem seguidas durante cada geração para manter a convergência não-prematura do método, garantindo, então, a diversidade da população e a obtenção da solução ótima, ou próxima desta.

Considere o seguinte problema:

$$\begin{cases} \min f(X), & X \in \mathbb{R}^n \\ \text{sujeito a } C_j(X) \geq 0, & j = 1, \dots, m \end{cases} \quad (4.1)$$

onde,  $f(X)$  representa a função objetivo, também chamada de função custo ou função adaptação,  $C_j$  as restrições de desigualdade, assumindo valores reais, e sejam  $X^L = (x_1^L, x_2^L, \dots, x_n^L)^T$  e  $X^U = (x_1^U, x_2^U, \dots, x_n^U)^T$  as restrições laterais inferior e superior respectivamente.

A população de indivíduos durante a  $q$ -ésima geração é definida como:

$$X_d^{(q)} = (x_{d,1}, x_{d,2}, \dots, x_{d,n})^T, \quad i = 1, \dots, n \quad e \quad d = 1, \dots, N_p \quad (4.2)$$

A população inicial é escolhida aleatoriamente, dentro do espaço de busca, definido pelas restrições laterais, fornecidas pelo usuário:

$$x_{d,i}^{(0)} = x_i^L + r(x_i^U - x_i^L), \quad i = 1, \dots, n \quad e \quad d = 1, \dots, N_p \quad (4.3)$$

sendo que  $r$  é um número gerado randomicamente entre 0 e 1.

Agora com a população estabelecida, é necessário a criação de novos indivíduos, isto se dará através dos operadores que serão apresentados em seguida.

### 4.1.1 Mutaçãõ

Sejam os vetores  $X_\alpha, X_\beta$  e  $X_\gamma$  escolhidos aleatoriamente e distintos entre si. Na geração  $q$  um par de vetores  $(X_\beta, X_\gamma)$  define uma diferença  $X_\beta - X_\gamma$ . Esta diferença é multiplicada por  $F > 0$ , sendo denotada por diferença ponderada, e é usada para perturbar o terceiro vetor  $X_\alpha$  ou o melhor vetor  $X_{best}$  da população. Este processo, que resulta no vetor doador  $V^{(q+1)}$ , pode ser escrito matematicamente como:

$$\begin{aligned} V^{(q+1)} &= X_\alpha^{(q)} + F \left( X_\beta^{(q)} - X_\gamma^{(q)} \right) \\ &\text{ou} \\ V^{(q+1)} &= X_{best}^{(q)} + F \left( X_\beta^{(q)} - X_\gamma^{(q)} \right) \end{aligned} \quad (4.4)$$

sendo que os índices aleatórios  $\alpha, \beta, \gamma \in \{1, \dots, N_p\}$  são inteiros distintos entre si e diferentes do índice  $d$ . O número de indivíduos da população,  $N_p$ , deve ser

maior ou igual a 4. O fator de escala, ou taxa de perturbação,  $F$ , é um fator real, positivo e constante variando entre 0 e 2, cujo objetivo é controlar a amplitude da diferença ponderada. A Figura 4.1 mostra um exemplo bidimensional que ilustra os diferentes vetores que participam da geração do vetor doador  $V^{(q+1)}$ .

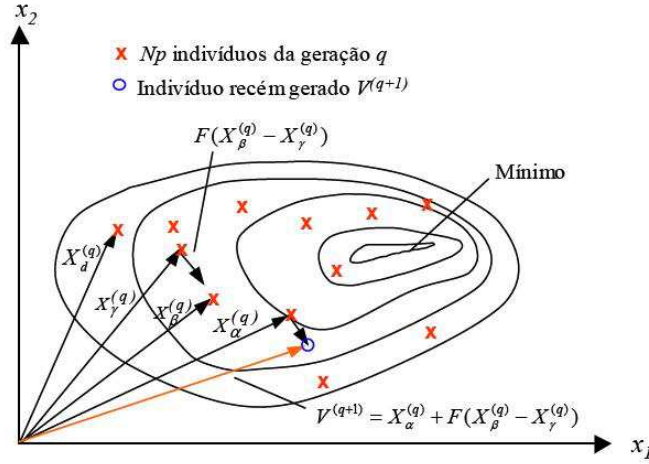


Figura 4.1: Processo de gerar o vetor doador  $V^{(q+1)}$  para uma função objetivo bidimensional

Se o número de indivíduos da população é grande o suficiente, a diversidade da população pode ser melhorada usando duas diferenças ponderadas para perturbar um vetor existente, ou seja, cinco vetores distintos são escolhidos aleatoriamente na população atual. O vetor diferença ponderada usa dois pares de diferenças ponderadas e é usado para perturbar o quinto vetor ou o melhor vetor da população atual. Este processo pode ser dado por:

$$\begin{aligned}
 V^{(q+1)} &= X_{\alpha}^{(q)} + F \left( X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)} \right) \\
 \text{ou} \\
 V^{(q+1)} &= X_{best}^{(q)} + F \left( X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)} \right)
 \end{aligned} \tag{4.5}$$

onde os índices aleatórios  $\alpha, \beta, \delta, \gamma \in \{1, \dots, N_p\}$ , são inteiros mutuamente distintos e diferentes do índice  $d$ , tais que  $N_p \geq 6$ .

Existem outras maneiras, que serão apresentadas adiante, de realizar a operação de mutação, o que diferencia as diversas estratégias que podem ser utilizadas pelo método da Evolução Diferencial.

### 4.1.2 Cruzamento

Na operação cruzamento, cujo objetivo é aumentar a diversidade dos indivíduos que sofreram a mutação, as componentes do vetor doador,  $V^{(q+1)}$ , são misturadas com as componentes de outro indivíduo denominado vetor alvo. A escolha do vetor alvo, que deve ser diferente dos vetores já usados anteriormente, é aleatório, segundo uma probabilidade de cruzamento  $CR$ . Desta forma, obtém-se o vetor tentativa ou experimental,  $U^{(q+1)}$ , definido como:

$$u_i^{(q)} = \begin{cases} v_i^{(q+1)}, & \text{se } r_i \leq CR \\ x_{d,i}^{(q+1)}, & \text{se } r_i > CR, \quad i = 1, \dots, n. \end{cases} \quad (4.6)$$

onde  $r_i$  é um número randômico entre 0 e 1 e  $x_{d,i}$  são as componentes do vetor alvo  $X_d^{(q)}$ . A probabilidade do cruzamento ocorrer,  $CR$ , representa a probabilidade do vetor experimental herdar os valores das variáveis do vetor doador, e está compreendida entre 0 e 1, sendo fornecida pelo usuário. Quando  $CR = 1$ , por exemplo, todas as componentes do vetor experimental virão do vetor doador  $V^{(q+1)}$ . Por outro lado, se  $CR = 0$ , todas as componentes do vetor experimental virão do vetor alvo  $X_d^{(q)}$ .

Se após o cruzamento uma ou mais componentes do vetor experimental estiver fora da região de busca, fazem-se as correções:

$$\begin{cases} \text{Se } u_i < x_i^L \text{ então } u_i = x_i^L \\ \text{Se } u_i > x_i^U \text{ então } u_i = x_i^U \end{cases}$$

O cruzamento dado pela Equação 4.6 é denominado de *cruzamento binomial* e pode ser observado na Figura 4.2.

Outra forma de realizar o cruzamento, denominado *cruzamento exponencial*, é definida segundo a Figura 4.3. As componentes do vetor experimental são dadas pelas componentes do vetor doador enquanto o número randômico for menor ou igual à probabilidade de cruzamento  $CR$ .

Após determinado o vetor experimental  $U^{(q)}$ , torna-se necessário selecionar os melhores descendentes. Esta operação será vista na próxima seção.

### 4.1.3 Seleção

A seleção é o processo de produzir melhores filhos. Diferentemente de outros algoritmos evolutivos, a evolução diferencial não usa hierarquia (elitismo) nem seleção proporcional. Em vez disso, o custo do vetor experimental  $U^{(q+1)}$  é calculado e comparado com o custo do vetor alvo  $X_d^{(q)}$ . Se o custo do vetor alvo for menor que



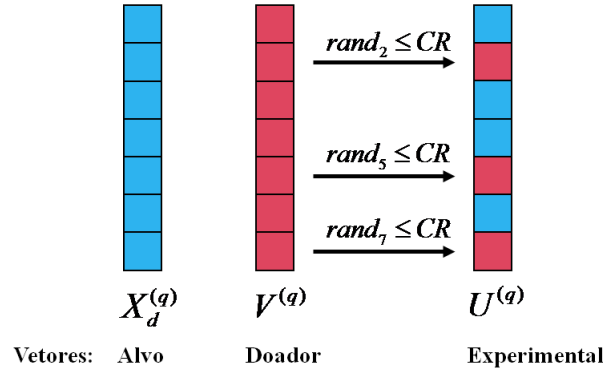


Figura 4.2: Cruzamento binomial

o custo do vetor experimental, o vetor alvo é permitido avançar para a próxima geração. Caso contrário, o vetor experimental substitui o vetor alvo na geração seguinte. Em outras palavras, este processo pode ser escrito como:

$$\begin{cases} \text{Se } f(U^{(q+1)}) \leq f(X_d^{(q)}), \text{ então } X_d^{(q+1)} = U^{(q+1)} \\ \text{Se } f(U^{(q+1)}) > f(X_d^{(q)}), \text{ então } X_d^{(q+1)} = X_d^{(q)} \end{cases} \quad (4.7)$$

O processo iterativo continua até que seja alcançado algum critério de parada, sendo que um número máximo de gerações deve ser estabelecido. Para problemas com restrição, um critério pode ser a não violação das restrições ou o melhor indivíduo ter encontrado um valor dentro de uma precisão pré-estabelecida. Assim, os seguintes critérios de parada podem se estabelecidos:

$$\begin{cases} q < max \\ |f_{min}^{(q+1)} - f_{min}^{(q)}| < \epsilon \end{cases} \quad (4.8)$$

## 4.2 Parâmetros da evolução diferencial

O seguinte conjunto de regras pode ajudar na escolha das variáveis de controle  $N_p$ ,  $CR$  e  $F$ , conforme Storn (1996):

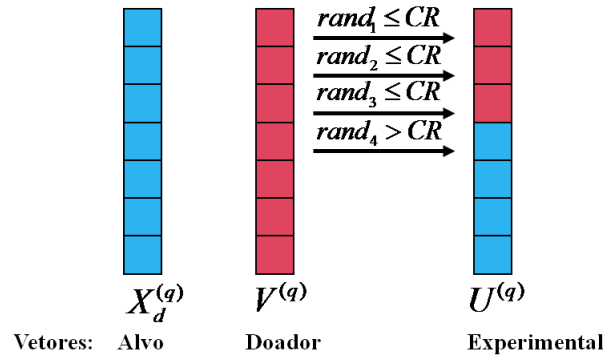


Figura 4.3: Cruzamento exponencial

- A população inicial deve ser gerada o mais próximo possível da superfície da função objetivo;
- Frequentemente a probabilidade de cruzamento  $CR$  deve ser considerada menor do que um, por exemplo  $CR = 0,3$ . Caso não ocorra convergência, adotar  $CR \in [0,8;1]$  pode ajudar;
- Para muitas aplicações  $N_p = 10D$ , onde  $D$  é igual a dimensão ou ao número de variáveis, é uma boa escolha. Normalmente,  $F$  pode ser escolhido no intervalo de  $[0,5;1]$ .
- Quanto maior for o tamanho da população escolhida, menor deve ser o valor de  $F$ .
- Tem-se um bom sinal de convergência quando os parâmetros do melhor componente da população variam muito de geração para geração, especialmente durante o início do processo de minimização, mesmo se seu valor da função objetivo decrescer lentamente;
- O valor da função objetivo do melhor indivíduo não pode cair de forma brusca, caso isto aconteça, a otimização está em um mínimo local;

### 4.3 Estratégias da Evolução Diferencial

Vale ressaltar que a Evolução Diferencial apresenta diferentes estratégias obtidas a partir da forma com que os operadores de mutação e cruzamento trabalham, ou

Estratégia	Operador Mutaçãõ	Notaçãõ	
1	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/bin	
2	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/bin	
3	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/bin	
4	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/bin	
ht	5	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/bin
6	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/exp	
7	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/exp	
8	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/exp	
9	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/exp	
10	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/exp	

Tabela 4.1: Estratégias do método Evolução Diferencial

seja, as estratégias da evolução diferencial podem variar de acordo com o tipo de indivíduo a ser modificado na formação do vetor doador, o número de indivíduos considerados para a perturbação e o tipo de cruzamento a ser utilizado, podendo ser escritas como: ED/a/b/c, sendo que:

a → especifica o vetor a ser perturbado, podendo ser *rand* (um vetor da população escolhido aleatoriamente) ou *best* (o vetor de menor custo da população);

b → determina o número de diferenças ponderadas usadas para a perturbação de a;

c → denota o tipo de cruzamento (exp: exponencial; bin: binomial).

Em 1995, Storn and Price deram o princípio de trabalho da estratégia básica usando apenas o operador cruzamento binomial (devido aos experimentos binomiais independentes), onde o cruzamento é executado em cada variável sempre que um número  $r \in [0; 1]$  aleatório for menor que a probabilidade de cruzamento  $CR$ .

Alguns anos mais tarde, Storn and Price (1997) desenvolveram mais estratégias usando o operador cruzamento exponencial, em que o cruzamento é executado nas variáveis em um laço até que esteja dentro do limite de  $CR$ . A primeira vez que um número  $r \in [0; 1]$  aleatório ultrapassa o valor de  $CR$ , nenhum cruzamento é executado e as variáveis restantes são deixadas intactas. Resumidamente, as dez estratégias podem ser descritas de acordo com a Tabela 4.1.

No entanto, uma estratégia que funciona bem para um dado problema pode não funcionar bem quando aplicada a outro problema. A estratégia a ser adotada para um problema é determinada por tentativa e erro.



## Capítulo 5

# Evolução com Conjuntos Embaralhados

A idéia principal da abordagem da Evolução com Conjuntos Embaralhados (ECE) é tratar a procura global como um processo de evolução natural. Neste estudo será aplicado esse conceito para o algoritmo da ED. Os  $N_p$  pontos de amostragem, constituem uma população que é dividida em diversos conjuntos (sub-populações), tais que cada conjunto tenha o mesmo número de indivíduos. A cada um dos conjuntos é permitido evoluir de forma independente (ou seja, explorar o espaço de busca em direções diferentes). Depois de um certo número de gerações, os conjuntos são forçados a se misturar e novos conjuntos são formados através de um processo de embaralhamento. Este procedimento aumenta a capacidade de sobrevivência por um intercâmbio de informações sobre o espaço de busca adquirida de forma independente por cada conjunto.

Cada membro de um conjunto é um pai com a capacidade de participar de um processo de reprodução. Por fim, cada novo descendente substitui o seu ponto de destino correspondente no complexo atual ao invés do ponto de toda a população. Isto garante que cada pai receba pelo menos uma chance de contribuir para o processo de reprodução antes de serem substituídos ou eliminados. Assim, nenhuma das informações contidas na amostra é ignorada. O método ECE é projetado para melhorar características do método da ED, incorporando em si o poderoso conceito de conjuntos embaralhados.

Conjuntos embaralhados ajudam a garantir que as informações contidas na amostra são eficientes e completamente exploradas. Também ajudam a garantir que o conjunto de informações não se degenera. Estas vantagens do método ECE que motivam a aplicá-lo sobre o algoritmo da ED, com a esperança de se obter melhores propriedades de convergência global sobre uma ampla gama de

problemas. Em outras palavras, dado um número pré-especificado de avaliações da função objetivo (nível fixo de eficiência), espera-se que o método ECE tenha uma maior probabilidade de sucesso em seu objetivo de encontrar o ótimo global em comparação à ED.

O método ECE é baseado em uma síntese de quatro conceitos que se revelaram bem sucedidas para a otimização global (Holland, 1975): (a) combinação de abordagens aleatórias e determinísticas, (b) o conceito de computação paralela, (c) o conceito de uma evolução sistemática de um conjunto de pontos que exploram o espaço, no sentido da melhoria global, (d) o conceito de evolução competitiva.

Uma breve discussão destes conceitos é apresentada aqui. O uso de estratégias determinísticas permite que o algoritmo ECE faça uso efetivo das informações da superfície de resposta para orientar a pesquisa, enquanto a inclusão de elementos aleatórios ajuda a tornar o algoritmo robusto e flexível. A pesquisa começa com um conjunto de pontos selecionado ao acaso abrangendo todo o espaço de busca  $\Omega$ . Um grande, porém suficiente, número de pontos ajuda a garantir que o conjunto contenha informações quanto ao número, localização e tamanho das principais regiões viáveis.

A implementação de uma estratégia de agrupamento implícito ajuda a concentrar a busca na mais promissora das regiões identificadas pelo conjunto inicial. O uso de uma estratégia de evolução sistemática ajuda a garantir que a busca seja relativamente robusta e guiada pela estrutura da função objetivo. A robustez é resultado do fato de que a estrutura de conjuntos é capaz de lidar muito bem com superfícies da função objetivo rudes, insensíveis e altamente não convexas e é relativamente pouca afetada pelos mínimos locais pequenos que são encontrados na busca pela solução global. Além disso, não são necessárias informações de derivadas. A implementação de uma estratégia de evolução competitiva, conforme descrita por Holland (1975), tem sido feita por ser útil na melhoria da eficiência da convergência global.

## 5.1 O Método Evolução com Conjuntos Embaralhados - ECE

Em seguida será apresentado um algoritmo para o método Evolução com Conjuntos Embaralhados conforme descrito na trabalho de Duan *et al.* (1993). Um fluxograma para este método é apresentado na Figura 5.1.

**Passo 1: Inicialização.** Selecione  $p \geq 1$  e  $m \geq n + 1$ , onde  $p$  é o número de conjuntos e  $m$  é o número de pontos em cada conjunto. Calcule o tamanho da amostra  $s = p \times m$ .

**Passo 2:** *Geração da amostra.* Obtenha a amostra de  $s$  pontos  $x_1, \dots, x_s$  pertence à região viável  $\Omega \subset \mathbb{R}^n$ . Calcule o valor da função  $f_i$  em cada ponto  $x_i$ . Na ausência de informação prévia, use uma distribuição uniforme de amostragem.

**Passo 3:** *Rank points.* Organize os  $s$  pontos na ordem crescente do valor da função. Guarde-os em uma matriz  $D = \{x_i, f_i; i = 1, \dots, s\}$ , de modo que  $i = 1$  represente o ponto com o menor valor da função.

**Passo 4:** *Particionando em conjuntos.* Particione  $D$  em  $p$  conjuntos  $A^1, \dots, A^p$ , cada um contendo  $m$  pontos, tais que:

$$A^k = \{x_j^k, f_j^k; x_j^k = x_{k+p(j-1)}, f_j^k = f_{k+p(j-1)}, j = 1, \dots, m\}.$$

**Passo 5:** *Evoluir cada conjunto.* Evoluir cada conjunto  $A^k, k = 1, \dots, p$  de acordo com o algoritmo da evolução com conjuntos competitivos (ECC) descrito separadamente.

**Passo 6:** *Conjuntos embaralhados.* Substitua  $A^1, \dots, A^p$  em  $D$ , tal que  $D = \{A^k, k = 1, \dots, p\}$ . Ordene  $D$  em ordem de crescimento do valor da função.

**Passo 7:** *Testando a convergência.* Se os critérios de convergência foram satisfeitos, pare. Caso contrário, retorne ao passo 4.

A filosofia por trás da abordagem ECE é tratar a busca global como um processo de evolução natural. Os  $s$  pontos amostrados, constituem uma população. A população é dividida em várias comunidades (conjuntos), sendo que cada uma deles é permitido evoluir de forma independente, ou seja, busca o espaço em diferentes direções. Depois de um certo número de gerações, as comunidades são forçadas a se misturar e as novas comunidades são formadas por um processo de embaralhamento. Este procedimento aumenta a capacidade de sobrevivência por um intercâmbio de informações (sobre o espaço de busca) obtido de forma independente por cada sub-população.

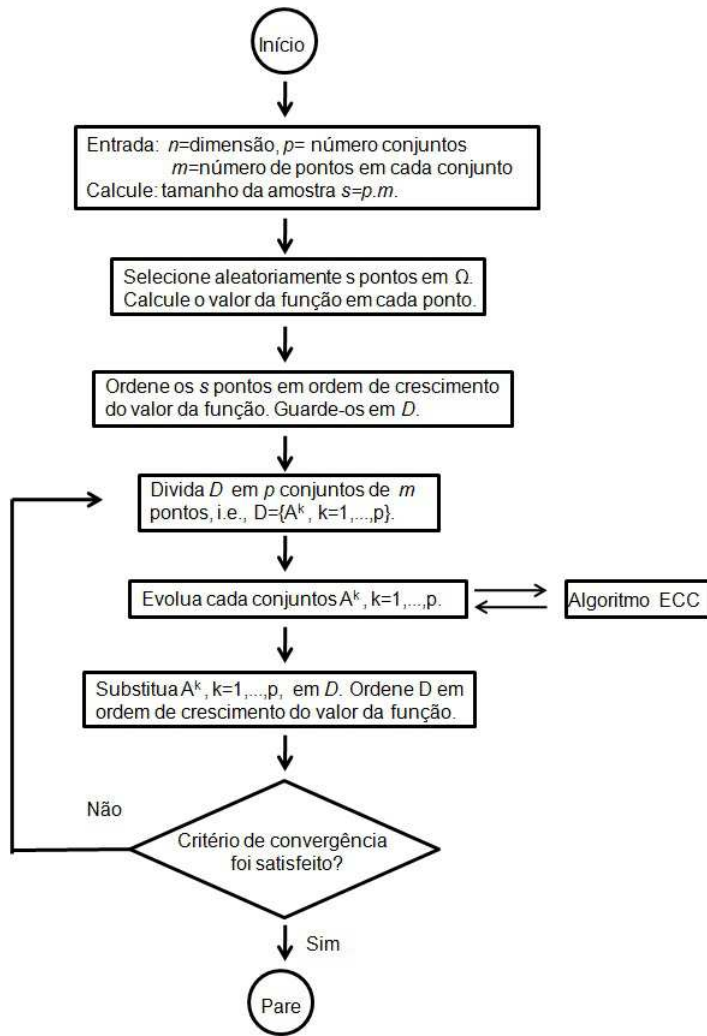


Figura 5.1: Fluxograma do Método Evolução com Conjuntos Embaralhados (fonte: Duan *et al.* (1993), adaptado pelas autoras).



Cada membro de um conjunto é um pai em potencial com a capacidade de participar de um processo de reprodução. Um subconjunto selecionado a partir do conjunto é como um par de pais, salvo que um subconjunto pode ser composto por mais de dois membros. Para garantir que o processo de evolução seja competitivo, é necessário que a probabilidade de que os melhores pais contribuam para a geração de descendentes seja maior que a dos piores pais. O uso de uma distribuição de probabilidade triangular assegura esta competitividade. O procedimento de Nelder e Mead (1965) é aplicada a cada subconjunto para gerar a maior parte dos descendentes. Esta estratégia utiliza as informações contidas no subconjunto para conduzir a evolução para uma direção de melhora. Além disso, os descendentes são introduzidos em locais aleatórios do espaço viável sob determinadas condições, a fim de garantir que o processo de evolução não fique preso em regiões pouco promissoras. Isso é um pouco semelhante ao processo da mutação em resposta à convergência prematura que pode ocorrer na evolução biológica. Cada mutação também ajuda a aumentar a quantidade de informação armazenada na amostra. Por fim, cada descendente novo substitui o pior ponto do subconjunto atual, ao invés de o pior ponto da população. Isso garante que cada pai recebe pelo menos uma chance de contribuir para o processo de reprodução, antes de ser substituído ou descartado. Assim, nenhuma das informações contidas na amostra é ignorada. A arquitetura do método ECE é ilustrado na Figura 5.2.

## 5.2 O Método Evolução com Conjuntos Competitivos - ECC

O algoritmo da Evolução com Conjuntos Competitivos (ECC) necessário para a atualização de cada conjunto no Passo 5 do método ECE é apresentado a seguir e é ilustrado na Figura 5.3.

**Passo 1:** *Inicialização.* Selecione  $q, \alpha, e \beta$ , onde  $2 \leq q \leq m, \alpha \geq 1$ , e  $\beta \geq 1$ .

**Passo 2:** *Atribuir pesos.* Atribuir uma distribuição de probabilidade triangular para  $A^k$ , isto é,

$$p_i = 2(m + 1 - i)/m(m + 1), \quad i = 1, \dots, m. \quad (5.1)$$

Observando a Equação (5.1), verifica-se que o ponto  $x_1^k$  ( $i = 1$ ) tem a maior probabilidade,  $p_1 = 2/(m + 1)$ . O ponto  $x_m^k$  ( $i = m$ ) tem a menor probabilidade,  $p_m = 2/m(m + 1)$ .

**Passo 3:** *Seleção dos pais.* Escolher aleatoriamente  $q$  pontos distintos  $u_1, \dots, u_q$  de  $A^k$  de acordo com a distribuição de probabilidade acima especificada (os  $q$  pon-

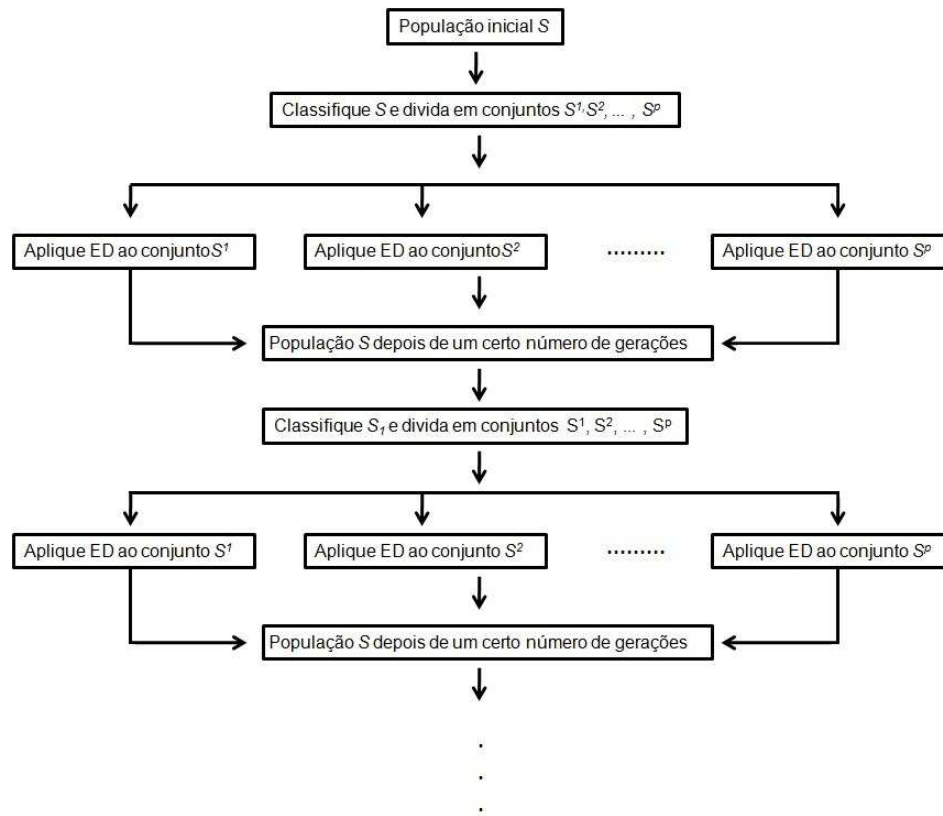


Figura 5.2: Arquitetura do Método Evolução com Conjuntos Embaralhados (fonte: Duan *et al.* (1993), adaptado pelas autoras).

tos definem um sub-conjunto). Armazene-os na matriz  $B = \{u_i, v_i, i = 1, \dots, q\}$  onde  $v_j$  é o valor da função associado ao ponto  $u_j$ . Armazene em  $L$  as localizações de  $A^k$  que são usadas para construir  $B$ .

**Passo 4:** *Geração de descendentes.*

(a) Ordenar  $B$  e  $L$  de modo que os  $q$  pontos estejam dispostos em ordem crescente do valor da função. Calcule o centróide  $g$ , utilizando a expressão:

$$g = [1/(q-1)] \sum_{j=1}^{q-1} u_j. \quad (5.2)$$

(b) Calcule o novo ponto (etapa de reflexão)

$$r = 2g - u_q. \quad (5.3)$$

(c) Se  $r \in \Omega$ , calcule o valor da função  $f$  e vá para a etapa (d), caso contrário, calcule o menor hipercubo  $H \subset \mathbb{R}^n$  que contenha  $A^k$ , gere aleatoriamente um ponto  $z \in H$ , calcule  $f_z$ , defina  $r = z$  e  $f_r = f_z$  (etapa mutação).

(d) Se  $f_r < f_q$ , substitua  $u_q$  por  $r$  e vá para a etapa (f); caso contrário, (etapa contração) calcule

$$c = (q + u_q)/2 \quad e \quad f_c. \quad (5.4)$$

(e) Se  $f_c < f_q$ , substitua  $u_q$  por  $c$  e vá para a etapa (f); caso contrário, gere aleatoriamente um ponto  $z \in H$  e calcule  $f_z$  (etapa mutação). Substitua  $u_q$  por  $z$ .

(f) Repita  $\alpha$  vezes as etapas de (a) até (e), sendo que  $\alpha \geq 1$  é um parâmetro especificado pelo usuário.

**Passo 5:** *Substituição dos pais pelos descendentes.* Substitua  $B$  por  $A^k$  usando a localização original armazenada em  $L$ . Ordene  $A^k$  segundo o crescimento do valor da função.

**Passo 6:** *Iteração.* Repita  $\beta$  vezes os passos de (1) até (4), onde  $\beta \geq 1$  é um parâmetro especificado pelo usuário que determina quantos filhos deverão ser gerados (o quanto cada conjunto deverá evoluir).

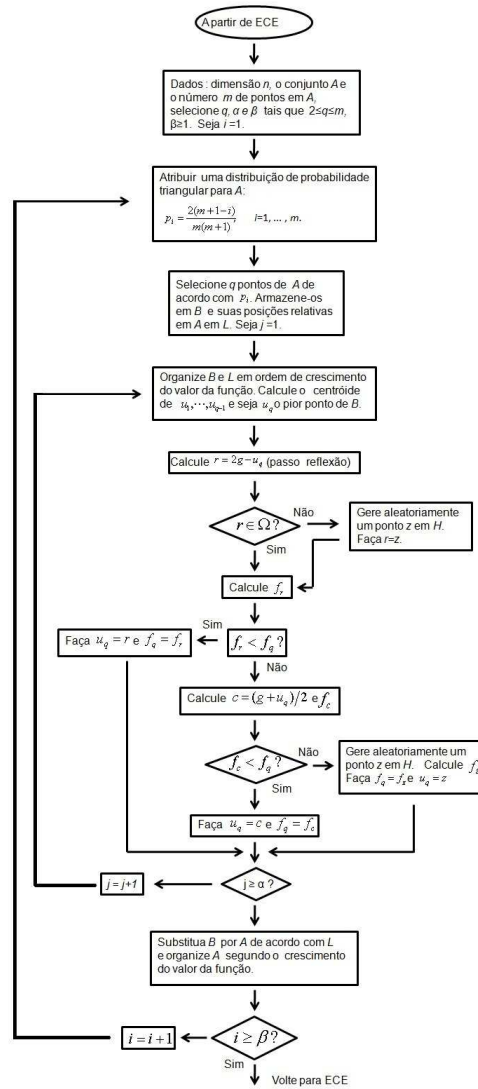


Figura 5.3: Fluxograma do Método Evolução com Conjuntos Competitivos (fonte: Duan *et al.* (1993), adaptado pelas autoras).

## Capítulo 6

# Método de Penalidade Aplicado às Técnicas Evolutivas

Muitos algoritmos de otimização foram desenvolvidos para resolver problemas ir-restritos, considerando apenas os limites laterais das variáveis, como no caso do Algoritmo Genético e da Evolução Diferencial, enquanto apenas alguns algoritmos processam as restrições de desigualdade e igualdade do problema para limitar a região viável. Assim, torna-se necessário utilizar algum artifício para que os problemas com restrições também se apliquem aos métodos de otimização irrestrita.

Uma das abordagens fundamentais para a otimização restrita é substituir o problema original por uma função de penalidade que é composta da função objetivo original do problema de otimização com restrições somada a um termo adicional para cada restrição, o qual é positivo quando o ponto atual  $X$  viola essa restrição e zero caso contrário.

A maioria das abordagens definem uma seqüência de funções de penalidade, na qual os termos penalizados por violarem as restrições são multiplicados por um coeficiente positivo. Ao fazer este coeficiente crescer, penalizam-se as violações das restrições mais severamente, forçando o ponto mínimo da função de penalidade se aproximar cada vez mais da região viável do problema restrito.

Essas abordagens são conhecidas como Método da Penalidade Exterior, porque o termo penalizado por cada restrição é diferente de zero somente quando  $X$  é inviável em relação a tal restrição. Muitas vezes, o ponto de mínimo das funções de penalidade são inviáveis com relação ao problema original, e aproximam-se da

viabilidade apenas no limite em que os parâmetros de penalidade se tornam cada vez maiores.

Considere o seguinte problema:

$$\begin{array}{ll} \min f(X) & \text{sujeito a } g_j(X) \geq 0, \text{ restrições de desigualdade} \\ X \in \mathbb{R}^n & h_l(X) = 0, \text{ restrições de igualdade} \end{array} \quad (6.1)$$

Afim de que os problemas restritos 6.1 sejam transformados em problemas irrestritos será utilizado, neste estudo, o Método da Penalidade Exterior. Se o objetivo da otimização é minimizar a função objetivo  $f(X)$ , a função de penalidade  $P(X)$  é somada a função principal  $f(X)$  de modo a aumentar o valor da função nos pontos que estão fora da região viável, por outro lado, se o objetivo é maximizar, a função de penalidade é subtraída da função principal se o ponto não obedece às restrições.

Esta nova função objetivo, chamada *pseudo objetivo*, é penalizada de acordo com um *fator de penalidade*  $r_p$  toda vez que encontrar uma restrição ativa. Assim, este escalar amplia a penalidade, e quanto maior for seu valor, maior será a eficiência do método para obedecer às restrições. O modelo matemático da função pseudo objetivo  $\Phi(X)$  utilizada para minimizar uma função  $f(X)$  é dada por:

$$\Phi(X) = f(X) + r_p P(X), \quad (6.2)$$

e a função de penalidade  $P(X)$  dada por:

$$P(X) = \sum_{j=1}^m \left( \max [0, g_j(X)]^2 \right) + \sum_{k=1}^l h_l(X)^2, \quad (6.3)$$

sendo  $g_j(X)$  as restrições de desigualdade,  $h_l(X)$  as restrições de igualdade e  $m$  e  $l$  o número de restrições de desigualdade e igualdade respectivamente.

Deste modo, os pontos fora do espaço viável apresentarão valores muito ruins para a otimização e serão facilmente desconsiderados pelos algoritmos de otimização irrestrita.

## Capítulo 7

# Simulações Numéricas

Para testar e avaliar os algoritmos Genético, Evolução Diferencial e Evolução com Conjuntos Embaralhados, serão propostos, a seguir, alguns problemas-teste, necessários para comparar as técnicas apresentadas neste estudo. O código foi escrito na forma de um toolbox para *MATLAB*®.

### 7.1 Projeto de um recipiente de pressão

O objetivo deste problema é o de minimizar o custo total do desenho de um recipiente de pressão conforme a Figura 7.1.

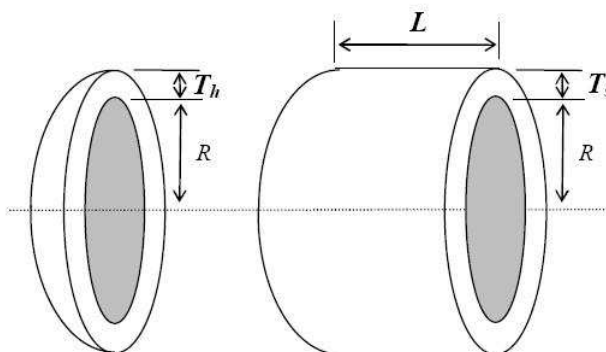


Figura 7.1: Projeto de um recipiente de pressão

Existem quatro variáveis no projeto:  $x_1$  ( $T_s$ , espessura do invólucro),  $x_2$  ( $T_h$ ,

Variáveis de projeto	Hu <i>et al.</i> (2003)	Coello (2000)	Deb (1997)	Algoritmo Genético	Evolução Diferencial
$x_1$	0,8125	0,8125	0,9375	0,9613	0,2233
$x_2$	0,4375	0,4375	0,5000	0,4755	0,9611
$x_3$	42,09845	40,3239	48,3290	49,7928	53,7741
$x_4$	176,6366	200,0000	112,6790	100,0000	175,9351
$g_1(x)$	0,0	-0,034324	-0,004750	-0,0003	0,8146
$g_2(x)$	-0,03588	-0,052847	-0,038941	-0,0005	-0,4481
$g_3(x)$	-5,820e-11	-27,10584	-3652,876	-21,0985	-953602,7787
$g_4(x)$	-63,3624	-40,0000	-127,3210	-140,0000	-64,0649
$f(x)$	6059,1312	6288,7445	6410,3811	6281,3716	6343,9254

Tabela 7.1: Comparação de alguns resultados do problema do recipiente de pressão

espessura da tampa),  $x_3$  ( $R$ , raio interno) e  $x_4$  ( $L$ , comprimento da secção cilíndrica do recipiente), sendo que  $R$  e  $L$  são contínuos. O problema pode ser descrito conforme segue:

$$\min f(x) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (7.1)$$

Sujeito a:

$$g_1(x) = -x_1 + 0,0193x_3 \leq 0 \quad (7.2)$$

$$g_2(x) = -x_2 + 0,00954x_3 \leq 0 \quad (7.3)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (7.4)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (7.5)$$

Os seguintes limites laterais das variáveis  $x = (x_1, x_2, x_3, x_4)$  serão utilizados:  
 $0 \leq x_1 \leq 10$ ,  $0 \leq x_2 \leq 10$ ,  $10 \leq x_3 \leq 100$ ,  $100 \leq x_4 \leq 200$ .

Na Tabela 7.1 são apresentados alguns resultados encontrados na literatura e os obtidos utilizando os algoritmos Genético e Evolução Diferencial para a resolução do problema de recipiente de pressão. Na Tabela 7.2 são apresentados os resultados obtidos com o método Evolução com Conjuntos Embaralhados em quantidades diferentes de conjuntos, sendo que  $t$  representa o tempo de processamento em segundos.



Variáveis de projeto	2 Conjuntos $t=5,2570s$	4 Conjuntos $t=5,7250s$	10 Conjuntos $t=5,5380s$	20 Conjuntos $t=7,4560s$
$x_1$	0,8631	0,8555	0,7787	0,7784
$x_2$	0,4267	0,4229	0,3850	0,3842
$x_3$	44,6980	44,3239	40,3499	40,3275
$x_4$	146,9913	150,8823	199,5795	199,8933
$g_1(x)$	-0,0004	-9,15e-6	-6,72e-7	-4,0e-5
$g_2(x)$	-0,0003	-6,14e-5	-8,26e-5	-1,0e-4
$g_3(x)$	-679,6647	-3,3706	-7,3757	-13,1134
$g_4(x)$	-93,0087	-89,1177	-40,4205	-40,1067
$f(x)$	6052,4298	6031,2781	5886,6091	5886,2353

Tabela 7.2: Resultados do problema de um recipiente de pressão para ECE.

## 7.2 Projeto de uma viga engastada

O objetivo é minimizar o custo de uma viga com as limitações de tensão de cisalhamento, tensão de dobramento na viga, esforço de carga na barra, reflexão final da viga e restrições laterais, conforme expresso na Figura 7.2.

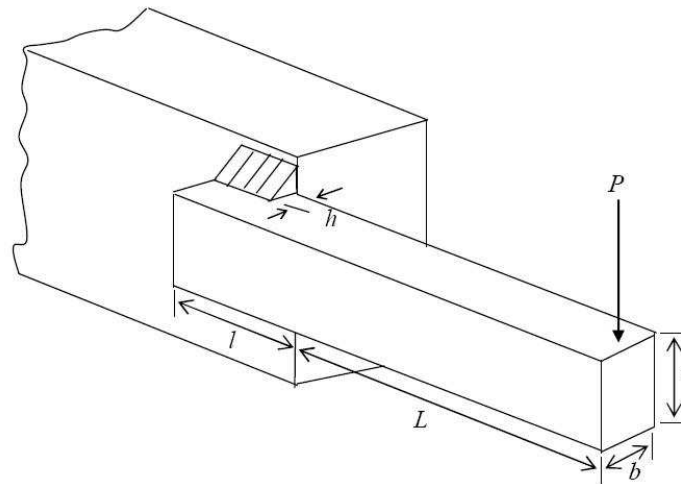


Figura 7.2: Projeto de uma viga engastada

O problema pode ser expresso como:

$$\min f(x) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14,0 + x_2) \quad (7.6)$$

sujeito a:

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (7.7)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (7.8)$$

$$g_3(x) = x_1 - x_4 \leq 0 \quad (7.9)$$

$$g_4(x) = 0,10471x_1^2 + 0,04811x_3x_4(14,0 + x_2) - 5 \leq 0 \quad (7.10)$$

$$g_5(x) = 0,125 - x_1 \leq 0 \quad (7.11)$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0 \quad (7.12)$$

$$g_7(x) = P - P_c(x) \leq 0 \quad (7.13)$$

sendo que,

$$\tau(x) = \sqrt{\tau'^2 + 2\tau'\tau'' \frac{x_2}{2R} + \tau''^2} \quad (7.14)$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2} \quad (7.15)$$

$$\tau'' = \frac{MR}{J} \quad (7.16)$$

$$M = P(L + \frac{x_2}{2}) \quad (7.17)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (7.18)$$

$$J = 2 \left[ \sqrt{2}x_1x_2 \left( \frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right) \right] \quad (7.19)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad (7.20)$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4} \quad (7.21)$$

$$P_c(x) = \frac{4,013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right) \quad (7.22)$$

onde:

$P = 6000 \text{ lb}$ ,  $L = 14 \text{ in}$ ,  $E = 30 \cdot 10^6 \text{ psi}$ ,  $G = 12 \cdot 10^6 \text{ psi}$ ,  $\tau_{max} = 13600 \text{ psi}$ ,  
 $\sigma_{max} = 30000 \text{ psi}$  e  $\delta_{max} = 0,25 \text{ in}$ .

Os seguintes limites laterais das variáveis  $x = (x_1, x_2, x_3, x_4)$  serão utilizados:  $0, 1 \leq x_1 \leq 2$ ,  $0, 1 \leq x_2 \leq 10$ ,  $0, 1 \leq x_3 \leq 10$ ,  $0, 1 \leq x_4 \leq 2$ .

Na Tabela 7.3 são apresentados os resultados encontrados na literatura e obtidos com os algoritmos Genético e Evolução Diferencial. Na Tabela 7.4 são apresentados os resultados para o método Evolução com Conjuntos Embaralhados, na qual  $t$  representa o tempo de processamento em segundos.

Variáveis de projeto	Hu <i>et al.</i> (2003)	Coello (2000)	Deb (1997)	Algoritmo Genético	Evolução Diferencial
$x_1$	0,20573	0,20880	0,2489	0,409904	0,190675
$x_2$	3,47049	3,42050	6,1730	2,015463	4,219093
$x_3$	0,03662	8,99750	8,1739	6,533373	8,466207
$x_4$	0,20573	0,21000	0,2533	0,393606	0,240833
$g_1(x)$	0,0	-0,337812	-5758,603	-0,049774	-299,421047
$g_2(x)$	0,0	-353,9026	-255,5769	-1,930750	-803,114581
$g_3(x)$	-5,551e-17	-0,001200	-0,004400	0,016298	-0,050158
$g_4(x)$	-3,432983	-3,411865	-2,982866	-3,000997	-3,209018
$g_5(x)$	-0,080729	-0,083800	-0,123900	-0,284904	-0,065675
$g_6(x)$	-0,235540	-0,235649	-0234160	-0,230001	-0,234979
$g_7(x)$	-9,094e-13	-363,2323	-4465,270	-27262,031799	-3212,606384
$f(x)$	1,7248508	1,7483094	2,433116	2,358168	1,956630

Tabela 7.3: Comparação de alguns resultados para o projeto de um viga engastada

Os resultados obtidos mostram que os algoritmos utilizados, Genético, Evolução Diferencial e Evolução com Conjuntos Embaralhados, foram eficientes em encontrar um valor ótimo para os problemas apresentados. Embora os problemas estudados sejam relativamente simples de serem resolvidos, enquanto os Algoritmos Genéticos e Evolução Diferencial levaram cerca de 20 segundos para obter o ótimo, o método Evolução com Conjuntos Embaralhados precisou, em média, de um tempo computacional três vezes menor.

Variáveis de projeto	2 Conjuntos t=7,41s	4 Conjuntos t=4,8830s	10 Conjuntos t=6,7230s	20 Conjuntos t=7,9880s
$x_1$	0,2794	0,2055	0,2057	0,2058
$x_2$	2,8417	3,4753	3,4724	3,4703
$x_3$	7,4943	9,0393	9,0366	9,0358
$x_4$	0,2991	0,2057	0,2058	0,2058
$g_1(x)$	-0,5225	-2,1166	-3,9511	-1,1344
$g_2(x)$	-0,8868	-17,8126	-4,2678	-0,4141
$g_3(x)$	-0,0197	-0,0002	-0,0001	0
$g_4(x)$	-3,1754	-3,4321	-3,4326	-3,4328
$g_5(x)$	-0,1544	-0,0805	-0,0807	-0,0808
$g_5(x)$	-0,2326	-0,2355	-0,2355	-0,2355
$g_7(x)$	-1089,7967	-1,1995	-2,6463	-3,1720
$f(x)$	2,0614	1,7256	1,7253	1,7251

Tabela 7.4: Resultados do problema de uma viga engastada para ECE.

### 7.3 Exercícios Propostos

Para os problemas apresentados abaixo, obter as soluções ótimas aplicando Algoritmos Genéticos (utilizando o código computacional GAOT, toolbox do *MATLAB*<sup>®</sup>) e a Evolução Diferencial (utilizando o código computacional ED).

1) Obter a raiz, para as funções e intervalos apresentados na Tabela 7.5, escrevendo a função na forma de um problema de otimização.

2) Obter o mínimo das seguintes funções relacionadas abaixo, adotando as restrições laterais  $X_i \in [-10, 10]$ .

a)  $F(X) = 10X_1^4 - 20X_1^2X_2 + 10X_2^2 + X_1^2 - 2X_1 + 5$

b)  $F(X) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2$

c)  $F(X) = X_1^4 + X_2^4 + 32X_1 - 4X_2 + 52$

d)  $F(X) = (X_1 + 10X_2)^2 + 5(X_3 - X_4)^2 + (X_2 - 2X_3)^4 + 10(X_1 - X_4)^4$

e)  $F(X) = X_1^2 + 2X_1X_2 + 2X_2^2 + X_3^2 - X_2X_3 + X_1 + 3X_2 - X_3$

Exemplo	f(x)	Intervalo
a	$(x - 1)e^{-nx} + x^n; \quad n = 15$	[0;1]
b	$(\cos(x) - xe^{-x})^3$	[0;3]
c	$(x - 10)x + 23 - x^{0.4}$	[0;7]
d	$x^2 \left( \frac{1}{3}x^2 + \sqrt{2}\text{sen}(x) \right) - \frac{\sqrt{3}}{19}$	[-1;2]
e	$2x + \ln(x)$	[0.1;1]
f	$x^3 + 7 - e^x$	[0;5]
g	$e^{1-x}(x - 1)10 - 1$	[0;2]
h	$x^2 - \ln(x) - 2$	[1;3]

Tabela 7.5: Funções propostas para o exercício 1.

$$f) F(X) = (X_1 + 2X_2 - 7)^2 + (2X_1 + X_2 - 5)$$

3) Sob a ação de uma força  $F$ , o pêndulo move de  $A$  até a posição de equilíbrio  $B$ , conforme Figura 7.3. Obtenha esta posição de equilíbrio, que corresponde à mínima energia potencial:

$$\min Ep = Wy - Fx,$$

onde,  $x = L\text{sen}\theta$ ,  $y = L(1 - \cos\theta)$ .

Dados:  $W = 500N$ ;  $F = 100N$ ;  $L = 2,5m$ , restrições laterais:  $\theta \in [0, \pi/2]$ .

4) Determinar a posição de equilíbrio estático de um sistema constituído de 2 molas ( $K_1$  e  $K_2$ ) solicitado por duas forças constantes ( $P_1$  e  $P_2$ ), conforme Figura 7.4, de forma a minimizar sua energia potencial:

$$\begin{aligned} \min Ep = & 0,5k_1 \left( \sqrt{X_1^2 + (l_1 - X_2)^2} - l_1 \right)^2 + \\ & + 0,5k_2 \left( \sqrt{X_1^2 + (l_2 - X_2)^2} - l_2 \right)^2 - P_1X_1 - P_2X_2 \end{aligned}$$

Dados:  $P_1 = P_2 = 5 N$ ;  $K_1 = 8 N/cm$ ;  $K_2 = 1 N/cm$ ;  $l_1 = l_2 = 10 cm$ ; restrições laterais:  $X_i \in [0, 10]$ .

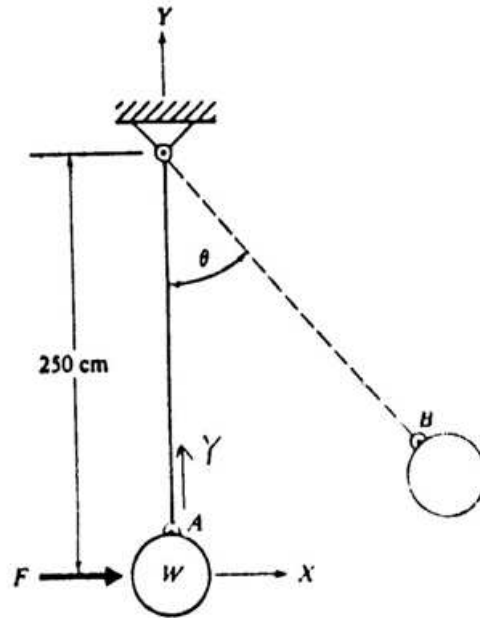


Figura 7.3: Equilíbrio estático de um pêndulo submetido a uma força  $F$ .

5) Deseja-se projetar um mastro de bandeira de altura  $H$  com a menor massa possível obedecendo certas restrições impostas ao problema. O mastro é composto por um tubo circular, oco e uniforme, sendo  $d_o$  e  $d_i$  seus diâmetros externo e interno, respectivamente. O projeto deve considerar a ação de fortes ventos que eventualmente pode atuar sobre o mastro da bandeira, que deve resistir sem apresentar falhas.

Por razões de projeto, o mastro será considerado com uma viga engastada sujeita a uma carga uniformemente distribuída na lateral  $w$  (kN/m), representando a atuação do vento. Além disso, o vento também induz uma carga concentrada  $P$  (kN) no topo do mastro, como mostrado na Figura 7.5.

O problema se resume a minimização da área da seção transversal do mastro, definida pela função  $A$ , aliada a algumas restrições: a deflexão no topo  $\delta$  não deve exceder 10 cm; a razão do diâmetro médio pela espessura não deve exceder 60; as espessuras mínima e máxima são 0,8 e 2 cm, respectivamente; a tensão de flexão admissível é  $\sigma_b=165$  MPa e a tensão de cisalhamento admissível é  $\tau_s=50$  MPa.

$$A = \frac{\pi}{4} (d_o^2 - d_i^2).$$

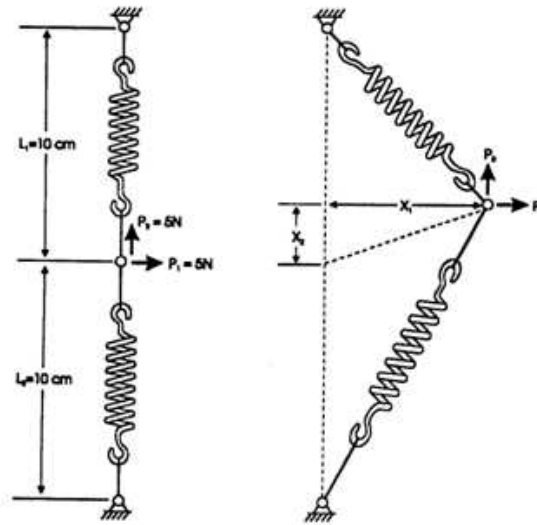


Figura 7.4: Equilíbrio estático de um sistema de 2 molas.

As funções de restrição são dadas por:

$$\delta = \frac{PH^3}{3EI} + \frac{wH^4}{8EI} \leq 10 \text{ cm}$$

$$\frac{d_o + d_i}{d_o - d_i} \leq 60$$

$$0,8 \leq \frac{d_o - d_i}{2} \leq 2 \text{ cm}$$

$$\sigma = \frac{M}{2I}d_o \leq 165 \text{ MPa}$$

$$\tau = \frac{S}{12I} (d_o^2 + d_o d_i + d_i^2) \text{ KPa}$$

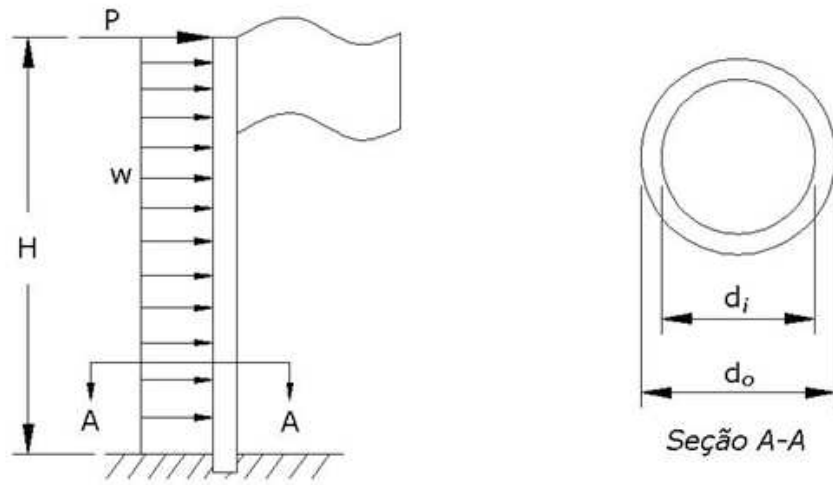


Figura 7.5: Projeto de um mastro de bandeira.

Os valores dos parâmetros pertinentes ao problema são dados a seguir:

$$\begin{aligned}
 H &= 5 \text{ m} \\
 E &= 210 \text{ GPa} \\
 \rho &= 7800 \text{ Kg/m}^3 \\
 w &= 2 \text{ KN/m} \\
 P &= 4 \text{ KN} \\
 I &= \frac{\pi}{64}(d_o^4 - d_i^4)
 \end{aligned}$$

$$\begin{aligned}
 M &= (PH + 0,5wH^2) \text{ KN} \cdot \text{m} \\
 S &= (P + wH) \text{ KN}
 \end{aligned}$$

Os limites laterais das variáveis de decisão são:

$$\begin{aligned}
 5 \leq d_o &\leq 50 \text{ cm} \\
 4 \leq d_i &\leq 45 \text{ cm}
 \end{aligned}$$



# Bibliografia

- [1] Bäck, T., Hoffmeister, F. Schwefel, H.-P (1991), *A Survey of Evolution Strategies*, In L. B. Belew and Booker, R. K. (eds.), *Proc. of the 4th Int. Conf. on GAs*, pp. 2-9, San Diego, CA, Julho.
- [2] Bekey, G.A., M.H. Gran, A.E. Sabroff, A. Wong (1966), Parameter optimization by random search using hybrid computer techniques, *AFIPS Conf. Proc.* 29, 191-200.
- [3] Braga, C. G.; *O uso de Algoritmos Genéticos para aplicação em Problemas de Otimização de Sistemas Mecânicos*; dissertação de mestrado, Universidade Federal de Uberlândia, 1998.
- [4] Born, J., *Evolutionstrategie zur Numerischen Lösung von Adaptations Aufgaben*, Dissertação, Humboldt-Universität, Berlin (1978).
- [5] Darwin, C. (1874), *Die Abstammung des Menschen*, translation of the 2nd rev. ed. of *The descent of man*, Kröner, Stuttgart, 1966.
- [6] Deb, K. *GenaAS: a robust optimal design technique for mechanical component design*, Dasgupta, D. and Michalewicz 2 (eds). *Evolutionary Algorithms in Engineering Applications* Berlin, pp. 497-514, 1997.
- [7] DeGraag, D.P., *Parameter optimization techniques for hybrid computers*, *Proceedings of the VIth International Analogue Computation Meeting*, Munich, Aug-Sept. 1970, pp. 136-139, 1970.
- [8] Duan, Q. A., Gupta, V. K. and Sorooshian, S., *Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization*, *Journal of Optimization Theory and Applications*: Vol. 76, N° 3, March 1993.
- [9] Goldberg, David E., (1989), *Genetic algorithms in search, optimization, and machine learning*, The University of Alabama, Addison-Wesley Publishing Company, INC

- [10] Gonzalez, R.S. (1970), An optimization study on a hybrid computer, Ann. Assoc. Int'I Calcul Analog. 12, 138-148.
- [11] Haupt, R. L. e Haupt, S. E.; 1998, *Practical Genetic Algorithm*, John Wiley G. Sons Inc; New York, p.p.25-48.
- [12] Heitkoetter J. and Beasley D., editors. *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions*. USENET:comp.ai.genetic. Available via anonymous EMAIL from rtfm.mit.edu:/pub/usenet/news.answers/ai-faq/genetic/, 1994.
- [13] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.
- [14] Hu, X.; Eberhart, R.; Shi, Y. (2003) *Engineering optimization with particle swarm*, IEEE Conference on Swarm Intelligence, Indianapolis, Indiana.
- [15] Nelder, J. A., and Mead, R., *A Simplex Method for Function Minimization*, Computer Journal, Vol. 7, pp. 308-313, 1965.
- [16] Oliveira, G. T. S., *Estudo e aplicações da evolução diferencial*, 2006.
- [17] Pirsig, R.M., *Zen and the Art of Motorcycle Maintenance: An Inquiry into Values*, New York: Quill. 25th Anniversary Edition, 1974.
- [18] Rechenberg, I. (1973), *Evolutionsstrategie-Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart.
- [19] Rodrigues, L. F. *Meta-Heurísticas Evolutivas para Dimensionamento de Lotes com Restrições de Capacidade em Sistemas Multiestágios*. 2000. Monografia de Curso de Especialização, Universidade de São Paulo, São Carlos, Brasil.
- [20] Schwefel, H.-P. (1981), *Optimum Seeking Methods-User's Guides*, Internal report of the Programme Group of Systems Analysis and Technological Development, KFASTE-IB-7/81, Oct. 1981, Nuclear Research Center (KFA) Jülich, Germany.
- [21] Schwefel, H.-P. (1995), *Evolution and optimum seeking*, A Wiley-Interscience publication, New York, USA.
- [22] Soares, G. L. *Algoritmos Genéticos: Estudo, Novas Técnicas e Aplicações*. 1997. Monografia de Curso de Especialização, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil.

- [23] Stewart, E.C., W.P. Kavanaugh, D.H. Brocker (1967), Study of a global search algorithm for optimal control, Proceedings of the Vth International Analogue Computation Meeting, Lausanne, Aug.-Sept. 1967, pp. 207-230.
- [24] Storn, R., Price, K., 1995, *Differential Evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report TR-95-012, *International Computer Science Institute*, Berkeley.
- [25] Storn, R., On the Usage of Differential Evolution for Function Optimization, 1996. Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS,1996), Berkeley, p. 519-523, IEEE.
- [26] Storn, R. and Price, K., 1997, *Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, *Journal of Global Optimization* vol. 11, pp. 341-359.
- [27] White, R.C., Jr. (1970), Hybrid-computer optimization of systems with random parameters, Ph.D. thesis, University of Arizona, Tucson AZ, June 1970.

# Índice

- Alelos, 29, 30, 32, 34, 35
- Algoritmo Genético, 28–31, 34–36, 38, 53, 59
- Análise de sensibilidade, 16
  
- Bits, 30–32, 34
- Busca Aleatória, 21
  
- Charles Darwin, 27
- Computação paralela, 46
- Controle ótimo, 17, 18
- Critério de parada, 29, 36, 37, 41
- Cromossomo, 27, 29, 31, 32, 34, 35
- Crossover, 28, 34
- Cruzamento, 28–31, 34–37, 40, 42, 43
- Cruzamento binomial, 40, 43
- Cruzamento exponencial, 40, 43
- Cruzamento por um único ponto, 34
  
- David Goldberg, 27
- Diversidade da população, 38, 39
  
- Eficiência, 16
- Estocástica, 17, 19
- Estratégia Evolutiva, 21, 22
- Evolução com Conjuntos Competitivos, 47, 49
- Evolução com Conjuntos Embaralhados, 45, 56, 59
- Evolução Competitiva, 46
- Evolução Diferencial, 20, 37–43, 53, 55, 56, 59, 60
  
- Fator de escala, 39
- Fator de penalidade, 54
- Fenótipo, 23, 30
- Função Objetivo, 26
- Função objetivo, 16–20, 30–33, 36–38, 42, 46, 53, 54
- Função pseudo objetivo, 54
- Funções de penalidade, 53, 54
- Funções de restrição, 17, 54, 63
  
- GAOT, 60
- Genética Populacional, 27
- Genótipo, 22, 23, 29–31
- Gene, 27, 29, 30, 32
- Geração, 21, 27, 28, 30–33, 36–40, 42, 45
- Goldberg, 28
- Gregor Mendel, 27
  
- Indivíduos, 27–35, 37–40, 42
  
- John Holland, 27
  
- Ken Price, 37
  
- Método da Penalidade Exterior, 53
- Métodos clássicos, 16
- Modelagem, 16
- Mutação, 21–23, 27–31, 35–40, 42, 49, 51
- mutação, 24
- Mutação pontual, 36
- Mutações, 35

- Objetivo, 16–20, 30, 32, 36–38, 42
- Operações genéticas, 30
- Otimização, 21, 22
- Otimização contínua, 18
- Otimização determinística, 19
- Otimização discreta, 17
- Otimização global, 18, 28, 46
- Otimização híbrida, 20
- Otimização irrestrita, 17, 53, 54
- Otimização local, 18
- Otimização restrita, 17, 53
  
- Parâmetros, 16, 17, 21, 28, 30
- Performance, 16, 17, 19
- Pirsig, 15
- Ponto de cruzamento, 34
- População, 20, 27–30, 32–35, 37–39, 43, 45
- População inicial, 29, 32, 37, 38, 42
- Precisão, 16, 32, 41
- Probabilidade de cruzamento CR, 31, 34, 40, 42, 43
- Progamação matemática, 16
- Programação linear, 17
- Projeto ótimo, 17, 18
  
- Rainer Storn, 37
- Recipiente de pressão, 55, 56
- Recombinação, 27, 28, 31
- Restrição, 54
- Restrições, 16, 17, 20, 22, 38, 41, 53, 57, 60–62
- Robustez, 16
  
- Seleção, 21, 23, 24, 28–34, 36, 37, 40
- Seleção Aleatória, 33
- Seleção Elitista, 33
- Seleção Natural, 21, 27, 28
- Seleção por Torneio, 33
- Seleção proporcional, 30, 40
- Seleção ranking linear, 30
- Single-point crossover, 34
  
- Storn and Price, 43
- Taxa de mutação, 35
- Taxa de perturbação, 39
  
- Variáveis, 16, 17, 19, 29, 31, 32, 35, 37, 40, 41, 43, 55, 56, 59, 64
- Variáveis de projeto, 16, 17, 37
- Vetor alvo, 37, 40
- Vetor diferença ponderada, 39
- Vetor doador, 38–40, 42
- Vetor experimental, 37, 40
- Vetor tentativa, 37, 40
- Viga engastada, 57, 62