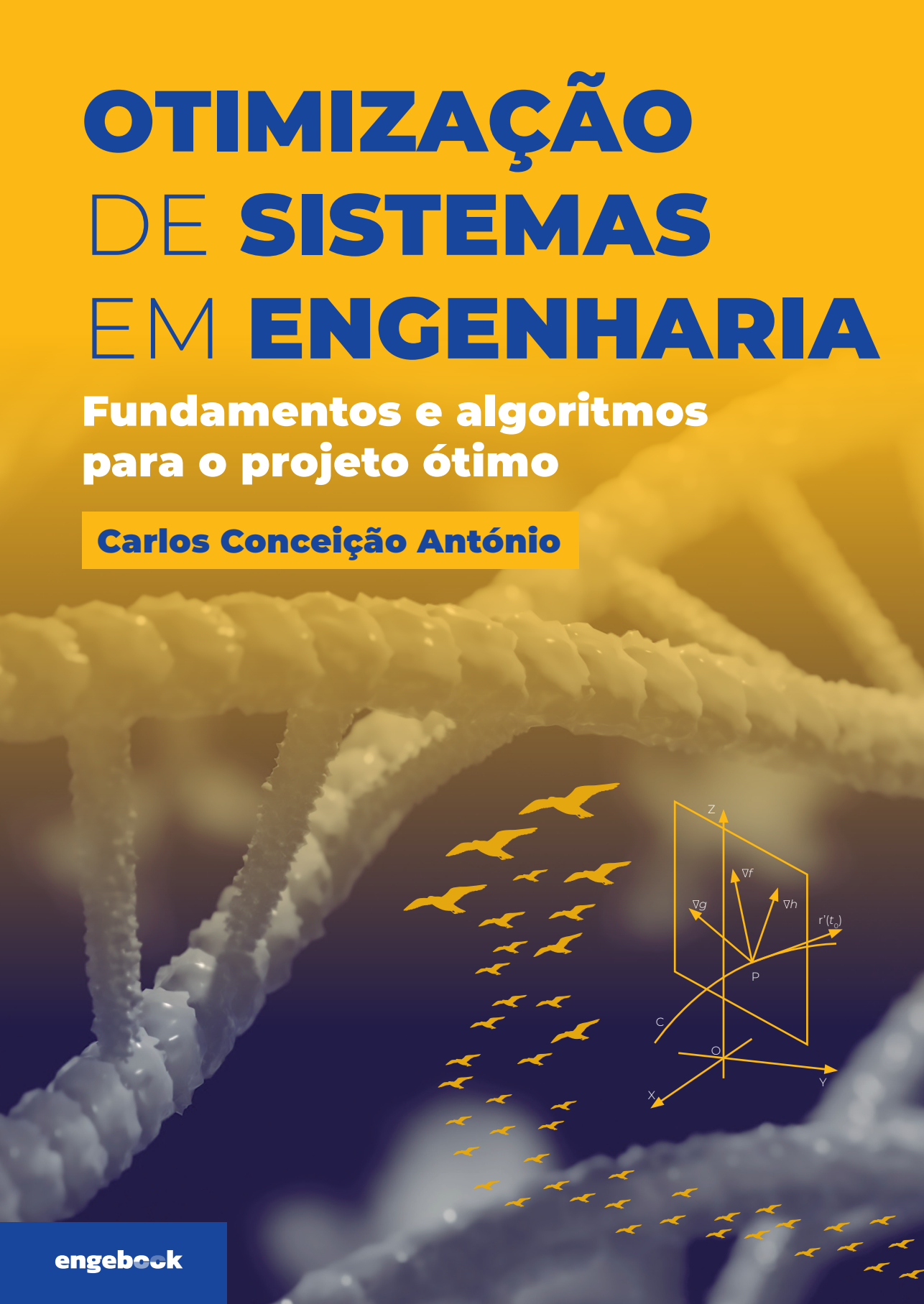
OTIMIZAÇÃO DE SISTEMAS EM ENGENHARIA

Fundamentos e algoritmos para o projeto ótimo

Carlos Conceição António



engebook

ÍNDICE

1.1. Motivação e enquadramento

A Otimização é um ramo da área da Matemática e dos Métodos Numéricos que tem sido objeto de investigação teórica e aplicada intensiva ao longo das últimas décadas. Em particular, a Engenharia é uma das áreas onde a otimização tem encontrado terreno propício para a sua aplicação e desenvolvimento teórico e prático. As técnicas de otimização alcançaram uma maturidade sem precedentes e são usadas num largo espectro de aplicações industriais em diferentes áreas nomeadamente engenharia estrutural, aeroespacial, automóvel, química, eletrónica, fabricação assistida por computador e produção industrial. Com o rápido avanço das tecnologias computacionais, a dimensão e a complexidade dos problemas resolvidos com o recurso a técnicas de otimização têm aumentado. A integração com poderosas ferramentas de projeto e fabrico assistido por computador (CAD/CAM) é já uma realidade.

A otimização pode ser definida como o estabelecimento racional de um projeto que é o melhor dentro de todos os projetos possíveis de acordo com um ou mais objetivos predefinidos e obedecendo a um conjunto prescrito de restrições geométricas e/ou restrições relacionadas com o comportamento e a integridade dos sistemas em engenharia, restrições tecnológicas, etc.

A otimização sugere a seleção da melhor opção dentro de uma gama de possíveis escolhas. É natural considerar este aspeto quando se produz um novo produto. Como diz o velho ditado: "um trabalho que vale a pena fazer deve ser bem feito". Porque não elaborar um projeto que é o melhor na sua classe, se existem meios para o fazer? Pode não custar mais tempo e dinheiro para projetar e produzir e, em última análise, até pode custar menos. Os clientes do produto ficarão mais satisfeitos com ele.

Mas como fazer a seleção, eis a questão. Talvez, na prática, haja satisfação com algo que está um pouco abaixo do "ótimo". Talvez se pense que é suficiente uma melhoria

dos detalhes. Isso incluirá a configuração dos principais itens estruturais e a instalação de sistemas e equipamentos, além de decisões sobre os principais aspetos de engenharia.

Por exemplo, na *Otimização Estrutural* abordam-se diferentes tópicos como dimensionamento ótimo de estruturas e componentes, otimização de forma e de topologia. Na área da *Otimização de Processos Tecnológicos*, as técnicas de otimização apoiadas em simulações numéricas permitem evitar não só experiências de fabrico reais com gastos elevados de tempo e material como também reduzir os desperdícios de material devido à necessidade de operações tecnológicas posteriores. A figura 1.1. aponta alguns dos benefícios da integração da otimização em áreas da engenharia mecânica.

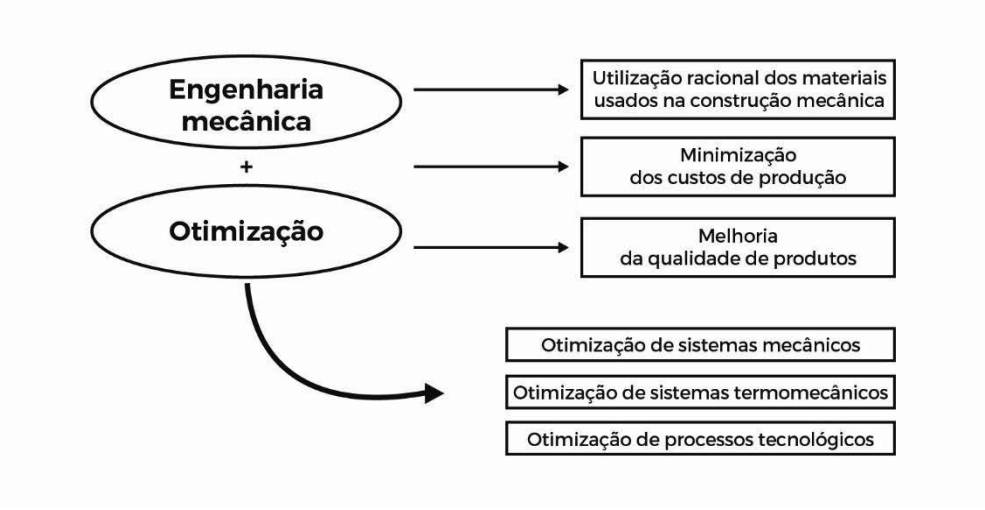


Figura 1.1. Consequências da otimização de sistemas em engenharia mecânica.

Este texto descreve os conceitos básicos de otimização e os algoritmos numéricos de solução para o projeto ótimo de sistemas em engenharia em geral e na engenharia mecânica em particular.

e os recursos de projeto subsequentes podem ser gerados apenas posteriormente ao estudo e podem envolver muita investigação e discussão.

Muitas possibilidades e fatores devem ser considerados durante a fase de definição do projeto. As considerações económicas desempenham um papel importante na criação de sistemas. Em geral, para concluir o projeto de um sistema de engenharia, os projetistas de diferentes áreas da engenharia devem cooperar. Por exemplo, o projeto de um autocarro de passageiros requer a cooperação entre especialistas em estruturas, mecânica, automóvel, engenheiros eletrotécnicos, informáticos, químicos, hidráulicos e especialistas de comportamento humano. Assim, num ambiente interdisciplinar, é necessária uma interação considerável entre as diferentes equipas de projetistas para concluir o projeto. Para a maioria das aplicações, o projeto global deve ser dividido em vários problemas elementares, que são tratados de maneira um pouco independente. Cada um dos problemas elementares pode ser equacionado como um problema de projeto ótimo.

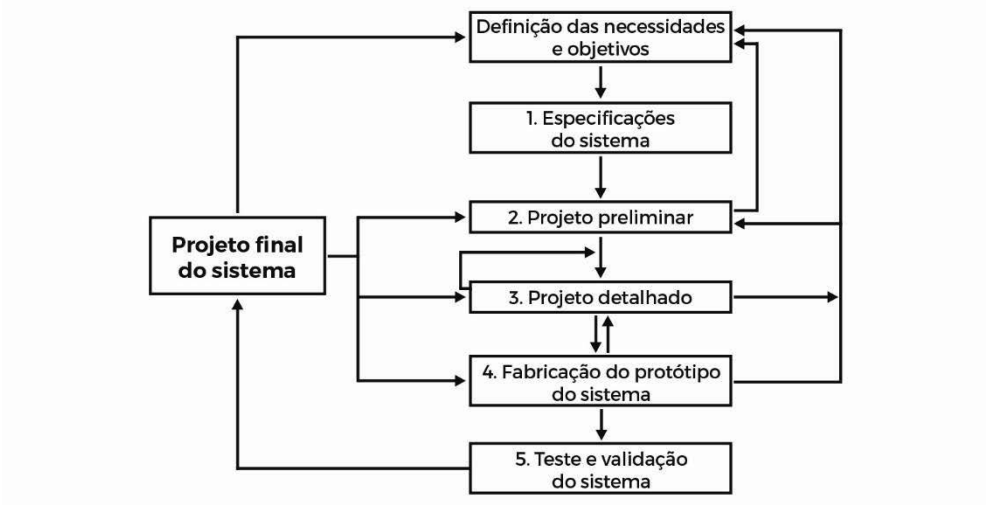


Figura 1.2. Visão geral de um ciclo de evolução do projeto e a interação das diferentes etapas de desenvolvimento.

para otimização com um objetivo e multiobjetivo no projeto de engenharia foram apresentadas por Rao em 1986 [1]. Estes métodos não serão apresentados neste livro devido à sua especificidade.

1.4.1. Métodos de otimização clássicos

A existência de métodos de otimização está referenciada desde os tempos de Newton, Lagrange e Cauchy. O desenvolvimento de métodos de otimização baseados no cálculo diferencial foi possível devido às contribuições de Newton e Leibniz. Por outro lado, Bernoulli, Euler, Lagrange e Weierstrass estabeleceram os fundamentos do cálculo de variações que está associado à minimização de funcionais. Cauchy fez a primeira aplicação do *método da descida mais* rápida (em inglês, *steepest descent method*) ou *método do gradiente puro* para resolver problemas de minimização sem restrições. O método de otimização para problemas com restrições, que envolve a introdução de multiplicadores como incógnitas, tornaram-se conhecidos através de Lagrange.

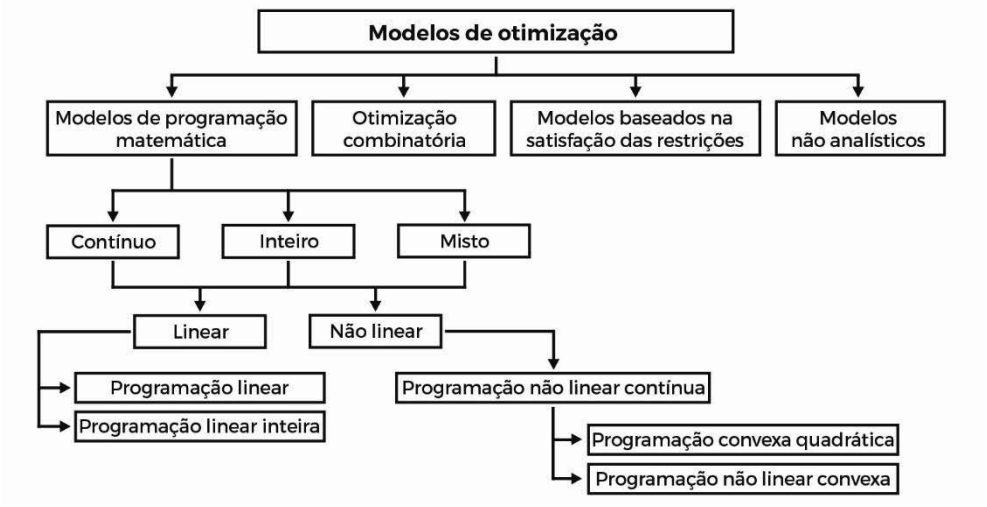


Figura 1.4. Métodos clássicos de otimização. Classes com possível sobreposição.

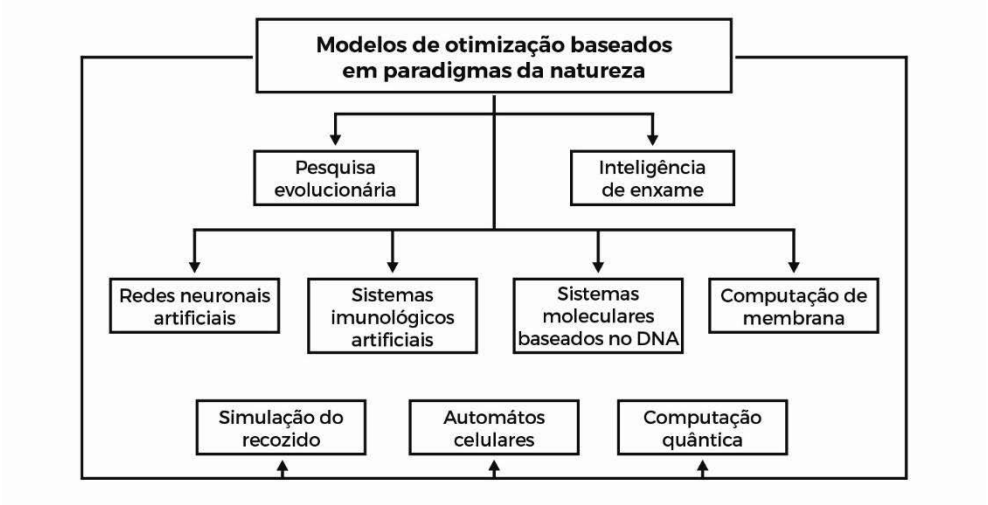


Figura 1.5. Modelos de otimização inspirados na natureza.

A computação biomolecular estuda o potencial do uso de moléculas biológicas para realizar a computação. A computação de ADN (ácido desoxirribonucleico) e a computação de membrana são duas técnicas de computação naturais no nível biomolecular. Em particular a computação de membrana é um modelo de computação paralela e distribuída que abstrai os modelos formais de computação da estrutura e funcionamento das células vivas, bem como da cooperação de células em tecidos, órgãos e outras estruturas de ordem superior [20].

A computação quântica baseia-se no princípio quântico de sobreposição de estados. Com efeito, em mecânica quântica, é possível que uma partícula esteja em dois ou mais estados ao mesmo tempo. A esta capacidade de estar simultaneamente em vários estados chama-se sobreposição. Para qualquer variável física o que é mensurável é a probabilidade de o sistema ter um certo valor dessa quantidade e que todos os estados possam ser processados em paralelo, a fim de otimizar uma função objetivo.

A computação quântica usa operadores unitários que atuam em vetores de estado discretos. O processamento quântico permite que um problema de otimização seja

2.1. Aspetos gerais da definição do problema de projeto ótimo

A formulação correta do problema de otimização de projeto é fundamental, porque a qualidade da solução ótima está dependente da formulação do problema. Por exemplo, se for omitida uma restrição crítica na formulação, a solução ótima provavelmente não irá satisfazer a restrição em causa. Além disso, se forem consideradas muitas restrições ou se estas forem inconsistentes, pode não haver solução para o problema. No entanto, uma vez que o problema é formulado adequadamente, geralmente existe um algoritmo para resolvê-lo.

O procedimento para formular um problema pode ser definido em cinco passos:

- Descrição do projeto ou do problema;
- Recolha da informação ou dos dados;
- Definição das variáveis de projeto;
- Critério de otimização/definição do(s) objetivo(s);
- Identificação/formulação das restrições.

O processo de formulação começa com o desenvolvimento de uma memória descritiva do projeto/problema. A memória descritiva descreve os objetivos gerais do projeto e os requisitos a serem atendidos.

Para desenvolver uma formulação matemática para o problema, é necessário recolher informações sobre as propriedades dos materiais, requisitos de desempenho, limites dos recursos, custo das matérias-primas e assim por diante. Além disso, a maioria dos problemas requer a capacidade de analisar e de testar o projeto. Portanto, os procedimentos de análise e as ferramentas de análise devem ser identificados nesta fase. Por exemplo, o método dos elementos finitos é geralmente usado para análise de estrutu-

A matriz Hessiana de f é uma matriz $n \times n$ de derivadas parciais de segunda ordem, definida da seguinte forma:

$$\mathbf{H}_f(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}^*) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}^*) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}^*) \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}^*) & \frac{\partial^2 f}{\partial x_n \partial x_2}(\mathbf{x}^*) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}^*) \end{bmatrix} \tag{2.8}$$

A condição (ii) da proposição 2.3. é equivalente a estabelecer que a matriz $\mathbf{H}_f(\mathbf{x}^*)$ é semidefinida positiva. A matriz Hessiana desempenha um papel fundamental na análise dos métodos iterativos de solução de problemas de otimização sem restrições. A estrutura desta matriz é determinante na convergência dos algoritmos de minimização da função f.

2.4.2. Condições suficientes para um mínimo relativo

Reforçando a segunda condição da proposição 2.3. obtém-se um conjunto de condições que implicam que o ponto $\mathbf{x}^*$ é um mínimo relativo. Estas condições são aplicáveis apenas a problemas sem restrições, ou a problemas onde o ponto de mínimo é interior à região admissível. As condições correspondentes para problemas onde o mínimo é alcançado num ponto da fronteira da região admissível é um bom desafio, mas de valor prático ou teórico marginal.

Proposição 2.4. Condições suficientes de segunda ordem para problemas sem restrições.

Seja $f \in C^2$ uma função definida numa região na qual o ponto $\boldsymbol{x}^*$ é um ponto interior do conjunto $\boldsymbol{X}$. Suponha-se também que:

i. $\nabla f(\mathbf{x}^*) = \mathbf{0}$;
ii. $\mathbf{H}_f(\mathbf{x}^*)$ *é definida positiva.*

3.1. Introdução

Construir o modelo matemático é pelo menos metade do trabalho para alcançar um projeto ótimo. Nunca é demais enfatizar a importância de um bom modelo. Mas o que constitui um modelo "bom"? As ideias apresentadas no capítulo 1 indicam uma importante característica de um bom modelo de projeto ótimo*: o modelo deve representar a realidade da maneira mais simples e significativa*. Um modelo de otimização é *significativo* se captura os equilíbrios que fornecem perceções rigorosas para quem toma decisões num contexto particular. Deve-se começar com o modelo mais simples e adicionar complexidade (mais funções, variáveis, parâmetros) apenas como a necessidade de estudar balanços mais complicados ou extensos. Essa necessidade é gerada por um estudo anterior de otimização bem-sucedido (e mais simples), novos modelos de análise ou alteração dos requisitos de projeto. Claramente, o processo é subjetivo e beneficia da experiência e intuição do projetista.

Em geral, um estudo de otimização é realizado depois de uma sofisticada análise ou modelo de simulação construído e validado. Os conceitos da otimização são introduzidos para converter um recurso de análise num recurso de projeto. Nessas circunstâncias, ainda se deve começar com o modelo mais simples possível. Uma maneira de reduzir a complexidade é usar metamodelos: modelos de análise mais simples extraídos dos mais sofisticados, usando uma variedade de técnicas de manipulação de dados. Os primeiros estudos de otimização são então conduzidos usando estes metamodelos.

3.2. Abordagem do problema de otimização

3.2.1. Conceito dos três pilares no projeto ótimo

Um problema de otimização pode geralmente ser tratado de acordo com o *Conceito dos Três Pilares* (em inglês, *Three-Columns-Concept*) integrado no processo de otimização de projeto. Este conceito estabelece três entidades:

1. O modelo de análise do sistema;
2. O algoritmo de otimização;
3. O modelo de otimização.

Na figura 3.1. descreve-se a integração destes três conceitos durante a construção do modelo a usar no processo de otimização tendo como objetivo o projeto ótimo de sistemas mecânicos.

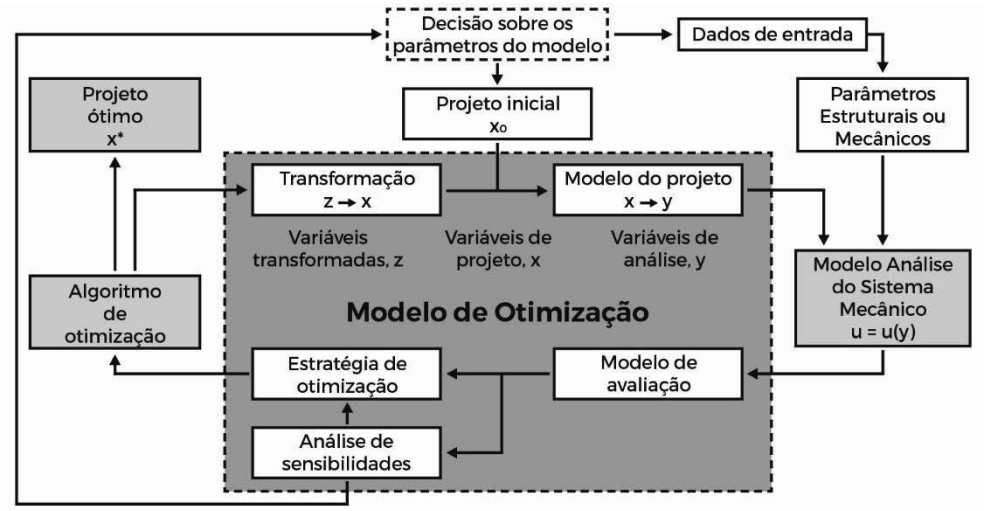


Figura 3.1. Abordagem do problema de otimização baseada no Conceito dos Três Pilares.

4.1. Introdução

A programação não linear é uma área da matemática aplicada que aborda os problemas de otimização quando as funções envolvidas são não lineares. Neste capítulo consideram-se as classes principais de algoritmos para a solução de problemas com funções objetivo e restrições não lineares. Quando, num problema de otimização, algumas ou todas as funções (função objetivo e/ou funções das restrições) são não lineares este é designado como um problema de *Programação Não Linear* (PNL) (em inglês, *nonlinear programming*, NLP). Este capítulo concentra-se nos conceitos e na descrição de métodos para problemas de otimização não linear. Os tópicos abordados no capítulo 2, nomeadamente as condições de otimalidade devem ser consideradas nas secções seguintes.

4.2. Programação não linear não restringida

Considere-se o problema de programação não linear sem restrições:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{Min}}\ f(\mathbf{x}) \tag{4.1}$$

onde $\mathbf{x} \in \mathbb{R}^n$ é o vetor das variáveis de projeto e $f : \mathbb{R}^n \to \mathbb{R}$ é a função objetivo. Os algoritmos aqui descritos podem ser facilmente adaptados para resolver problemas $\underset{\mathbf{x}\in\mathbf{X}}{\text{Min}}\ f(x)$ quando $\mathbf{X}$ é um conjunto aberto, desde que seja conhecida uma solução admissível $\mathbf{x^0} \in \mathbf{X}$.

Em muitos casos pode ser necessário impor condições mais fortes na amplitude do passo α^k. Em particular, as condições de Wolfe são largamente usadas. Neste caso, a condição de Armijo:

$$f\left(\mathbf{x}^k + \alpha\, \mathbf{d}^k\right) \leq f\left(\mathbf{x}^k\right) + \gamma\, \alpha\, \nabla f\left(\mathbf{x}^k\right)^T \mathbf{d}^k \tag{4.8}$$

é acoplada à condição:

$$\nabla f\left(\mathbf{x}^k + \alpha^k\, \mathbf{d}^k\right)^T \mathbf{d}^k \geq \beta\, \nabla f\left(\mathbf{x}^k\right)^T \mathbf{d}^k \tag{4.9}$$

ou à condição mais forte:

$$\left|\nabla f\left(\mathbf{x}^k + \alpha^k\, \mathbf{d}^k\right)^T \mathbf{d}^k\right| \leq \beta\, \left|\nabla f\left(\mathbf{x}^k\right)^T \mathbf{d}^k\right| \tag{4.10}$$

Onde $\beta \in (\gamma, 1)$, sendo γ o valor usado em (4.8). É possível provar que existe um intervalo finito de valores de α onde as condições (4.8) e (4.10) são satisfeitas (também (4.8) e (4.9)). Claro que um algoritmo de pesquisa linear verificando (4.8) e (4.9) impõem a satisfação das condições (4.3) e (4.7). Algoritmos de pesquisa linear para encontrar um passo de amplitude α^k que satisfaçam as condições de Wolfe são mais complicados que o algoritmo de pesquisa linear de Armijo e não serão detalhados aqui.

Em alguns casos, como nos métodos sem cálculo de derivadas (em inglês, *derivative-free methods*), é necessário considerar algoritmos de pesquisa linear para cálculo do passo α^k que não requeiram o uso de derivadas. Neste caso, a condição (4.6) não pode ser verificada e não pode ser assegurada uma direção descendente $\mathbf{d}^k$. Portanto, as técnicas de pesquisa linear também devem levar em consideração a possibilidade de usar um valor negativo para o passo, ou seja, o movimento na direção oposta $-\mathbf{d}^k$. Além disso, devem ser impostas condições adequadas para finalizar a pesquisa linear com $\alpha^k = 0$ quando é provável que $f\left(\mathbf{x}^k + \alpha\, \mathbf{d}^k\right)$ tenha um minimizador em

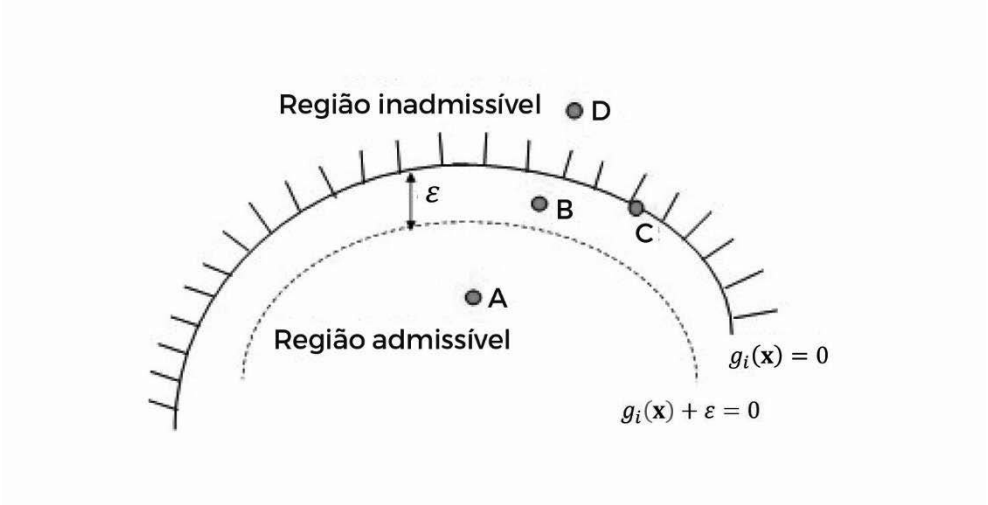


Figura 4.1. Estado de uma restrição nos pontos (soluções) A, B, C e D.

Para entender o conceito do estado de uma restrição, tome-se como referência a figura 4.1. Considere a restrição de desigualdade $g_i(\mathbf{x}) \leq 0$. É traçada a fronteira da restrição (superfície no espaço *n*-dimensional), $g_i(\mathbf{x}) = 0$, e são identificadas as regiões de valores admissíveis e inadmissíveis associadas à restrição. Também é traçada uma fronteira artificial a uma distância de ε da fronteira $g_i(\mathbf{x}) = 0$ e dentro da região admissível. Considerem-se quatro soluções de projeto A, B, C e D, como mostrado na figura 4.1. Para a solução de projeto correspondente ao ponto A, a restrição $g_i(\mathbf{x})$ é negativa e até $g_i(\mathbf{x}) + \varepsilon < 0$. Portanto, a restrição é *inativa* para a solução de projeto associada ao ponto A. Para o ponto de projeto B, $g_i(\mathbf{x})$ é estritamente menor que zero, logo é inativa. No entanto, $g_i(\mathbf{x}) + \varepsilon > 0$ portanto, a restrição é *ε-ativa* para a solução de projeto correspondente ao ponto B. Para o ponto de design C, $g_i(\mathbf{x}) = 0$, conforme se mostra na figura 4.1. Portanto, a restrição está *ativa* nesse ponto. Para o ponto de design D, $g_i(\mathbf{x})$ é maior que zero, portanto, a restrição é *violada*.

5.1. Introdução

Os métodos de otimização baseados em *Critérios de Otimalidade* (CO) (em inglês, *Optimality Criteria*, OC) e os métodos de *Programação Matemática* (PM) (em inglês, *Mathematical Programming*, MP) têm o mesmo objetivo no que concerne ao projeto ótimo, mas diferem na forma como é definida a sequência de passos de otimização do projeto. Nos métodos PM a função objetivo é minimizada diretamente através de vários algoritmos numéricos justificados por uma formulação matemática adequada. Nos métodos CO é definido antecipadamente um critério de projeto suportado por uma condição de otimalidade e a premissa é que quando esta condição é satisfeita o ótimo é alcançado. A aplicação dos métodos CO envolve a dedução de um critério apropriado para as condições específicas de projeto e o estabelecimento de um procedimento iterativo para alcançar o projeto ótimo final.

De acordo com a escolha dos critérios os métodos CO podem ser classificados da seguinte forma:

a. Métodos CO baseados em critérios de otimalidade físicos (ou intuitivos), nos quais as fórmulas (relações ou expressões) de recorrência explícitas para a sequência de projeto são deduzidas usando expressões de aproximação das restrições como funções das variáveis de projeto (estas últimas expressões são exatas para certos tipos de problemas de otimização). Em alguns casos, estes métodos apresentam convergência pobre e tendem a ser instáveis do ponto de vista numérico. Todavia, a eficiência (relacionada com o número de análises necessário) é habitualmente boa e independente da dimensão do problema. A técnica de projeto baseado no critério da tensão máxima (CTM) (em inglês, *Fully Stressed Design*, FSD) é provavelmente o método CO que maior sucesso tem obtido. Foi desenvolvido também um critério similar usando os deslocamentos conforme será apresentado neste capítulo [1, 2];

5.3.3. Fórmulas de recorrência para redefinição do projeto

Podem ser estabelecidos diferentes procedimentos numéricos para a redefinição do projeto para critérios de otimalidade similares. Nesta secção serão apresentadas fórmulas para a redefinição do projeto, baseadas em considerações físicas, que satisfazem o critério de otimalidade.

Considere-se primeiro o critério de otimalidade para uma *única restrição de deslocamento* conforme foi dado em (5.24). A última equação pode ser usada como uma fórmula de recorrência para um procedimento iterativo de determinação de x_h. Esta fórmula é similar à do critério da razão de tensões usada para alcançar o projeto ótimo baseado no critério da tensão máxima em (5.24) e pode ser escrita na forma [2]:

$$x_h^{k+1} = \left(\frac{1}{r^*} \sqrt{T_h / l_h} \sum_{i=1}^{I} \sqrt{T_i \, l_i} \right)^k, \; h = 1, \cdots, I \tag{5.29}$$

onde o índice k designa o número da iteração. O processo iterativo consiste na aplicação sucessiva da análise da estrutura e da fórmula de recorrência (5.29). Esta fórmula é baseada na suposição de que os elementos da matriz $\mathbf{T}$ são constantes numa dada iteração, isto é, a redistribuição da força não é sensível às alterações nas dimensões dos elementos (ou componentes estruturais). Esta é uma relação de recorrência intuitiva para a modificação das variáveis de projeto com o objetivo de satisfazer o critério de otimalidade (5.24) sem garantias de convergência. Esta fórmula depende da redistribuição da força durante o processo de solução.

O critério de otimalidade para *múltiplas restrições de deslocamento*, dado por (5.28), pode ser expresso através da seguinte relação de recorrência:

$$x_h^{k+1} = \sqrt{\sum_{j=1}^{J} \left(\lambda_j \frac{T_{jh}}{l_h} \right)^k}, \; h = 1, \cdots, I \tag{5.30}$$

Estas relações são similares às da equação (5.29) e podem ser usadas para obter x_h. No entanto, os multiplicadores λ_j não podem ser eliminados de uma forma simples

(5.28). No entanto, as soluções alcançadas por este método aproximado costumam ser muito próximas das obtidas pelos critérios corretos para múltiplas restrições.

O procedimento de projeto descrito na secção 5.2. para restrições de tensão e geométricas (dimensões dos elementos estruturais) pode ser generalizado de forma a incluir também restrições de deslocamento. O procedimento apresentado aqui é baseado na combinação do critério da razão de tensões, no critério dos deslocamentos e no procedimento de normalização. O processo de resolução envolve os seguintes passos descritos no algoritmo 5.2.:

Algoritmo 5.2. Critérios combinados de otimalidade e normalização

Passo 1: Considere-se uma solução inicial de projeto $\bar{\mathbf{x}}^k$ $(k = 1)$.

Passo 2: A solução de projeto corrente é analisada para obter os deslocamentos $\bar{\mathbf{r}}^k$ e o vetor das tensões $\bar{\boldsymbol{\sigma}}^k$.

Passo 3: O fator de escala μ é determinado por,

$$\mu^L \leq \mu \leq \mu^U \tag{5.31}$$

onde

$$\mu^L = \max_{i,j} \left\{ \begin{matrix} \bar{r}_j^k / r_j^U , \;\; \bar{r}_j^k / r_j^L \\ \bar{\sigma}_i^k / \sigma_i^U \;\; , \;\; \bar{\sigma}_i^k / \sigma_i^L \\ x_i^L / \bar{x}_i^k \end{matrix} \right\} \tag{5.32}$$

$$\mu^U = \min_i \left(x_i^U / \bar{x}_i^k \right) \tag{5.33}$$

$$\mathbf{x}^k = \mu \, \bar{\mathbf{x}}^k \tag{5.34}$$

a solução de projeto é normalizada por (5.34) para obter $\mathbf{x}^k$, e o valor da função objetivo é calculado em $\mathbf{x}^k$. Se alguns critérios de convergência são

6.1. Introdução

Os Algoritmos Evolucionários (AEs) inspiram-se no processo de evolução natural. A implementação computacional destes algoritmos deu origem à designação Computação Evolucionária. Não é de surpreender que os cientistas ligados à teoria da otimização tenham escolhido a evolução natural como fonte de inspiração: o poder da evolução na natureza é evidente nas diversas espécies que compõem o nosso mundo, cada uma adaptada para sobreviver bem no seu próprio meio. A metáfora fundamental associada aos algoritmos evolucionários relaciona essa poderosa *evolução natural a um determinado estilo de resolução* de problemas.

No século XIX, Johann Mendel [1] estabeleceu as bases da hereditariedade de pais para filhos. Mendel iniciou as suas experiências em 1856, tendo publicado os primeiros estudos em 1865. Um pouco antes, Charles Darwin em 1859 apresentou a teoria da evolução no seu livro famoso *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* [2]. Surpreendentemente, a ideia de aplicar os princípios darwinianos à solução automatizada de problemas remonta à década de 1940, muito antes do avanço dos computadores [3]. Já em 1948, Turing [4] propôs a *pesquisa genética ou evolutiva* e, em 1962, Bremermann [5] já havia executado experiências em computador sobre a *otimização através da evolução e recombinação*.

Na década de 60 do século XX, estas teorias da criação de novas espécies e da sua evolução inspiraram os cientistas no desenvolvimento dos a*lgoritmos evolucionários*. Diferentes escolas na área dos algoritmos evolucionários evoluíram independentemente nos últimos 50 anos: nos Estados Unidos apareceram os *Algoritmos Genéticos* (AG) (em inglês, *Genetic Algorithms,* GA), desenvolvidos principalmente por J. H. Holland [6-8] e K. A. De Jong [9] no Michigan e a *Programação Evolucionária* (em inglês, *Evolutionary Programming*) por L. Fogel em San Diego [10,11]; na Alemanha

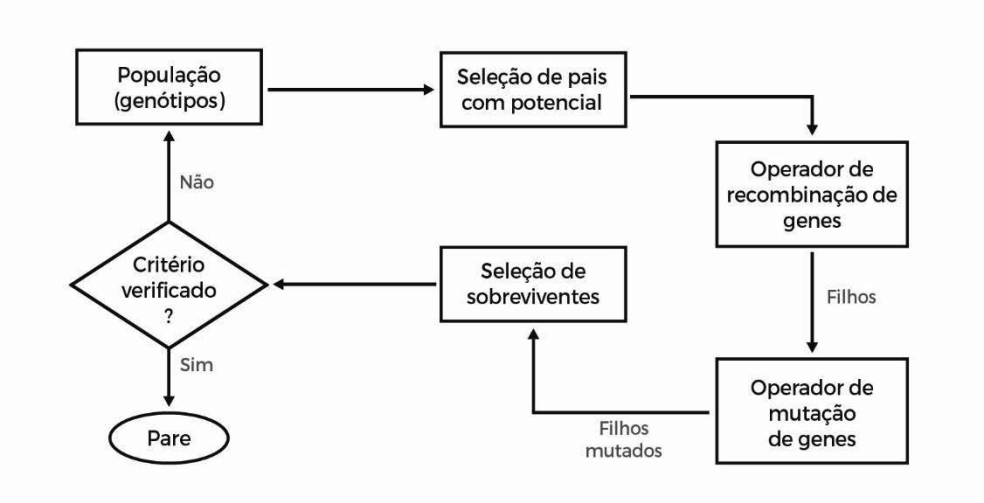


Figura 6.1. Esquema genérico evolutivo dos Algoritmos Evolucionários (AEs).

A figura 6.1. mostra a arquitetura geral dos AEs sob a forma de diagrama de fluxo. Esta arquitetura deve ser considerada como uma estrutura geral para os AEs, não um algoritmo em si. Com efeito, diferentemente da evolução natural, nos AEs existem ainda três outras características importantes: a iniciação, a avaliação e a paragem. Como não estamos preparados para esperar que o modelo evolua por vários milhões de gerações, os AEs geralmente recebem uma informação inicial (semente) de soluções que possuem estruturas e significados fixos, mas com valores aleatórios. Isso significa que se proporciona uma certa quantidade de conhecimento já no início da evolução do algoritmo. Por outro lado, a avaliação nos AEs é responsável por orientar a evolução em direção a melhores soluções. Diferentemente da evolução natural, os algoritmos evolutivos não têm um ambiente real no qual a capacidade de sobrevivência ou a qualidade das suas soluções possam ser testadas; eles devem confiar na simulação, na análise e no cálculo para avaliar as soluções. O processo evolutivo terminará quando um critério de paragem for verificado. A figura 6.2. mostra um modelo de código de um algoritmo evolucionário.

que a população suba a colina errada, deixando todos os indivíduos posicionados em torno de um pico local, mas que não corresponde ao ótimo global.

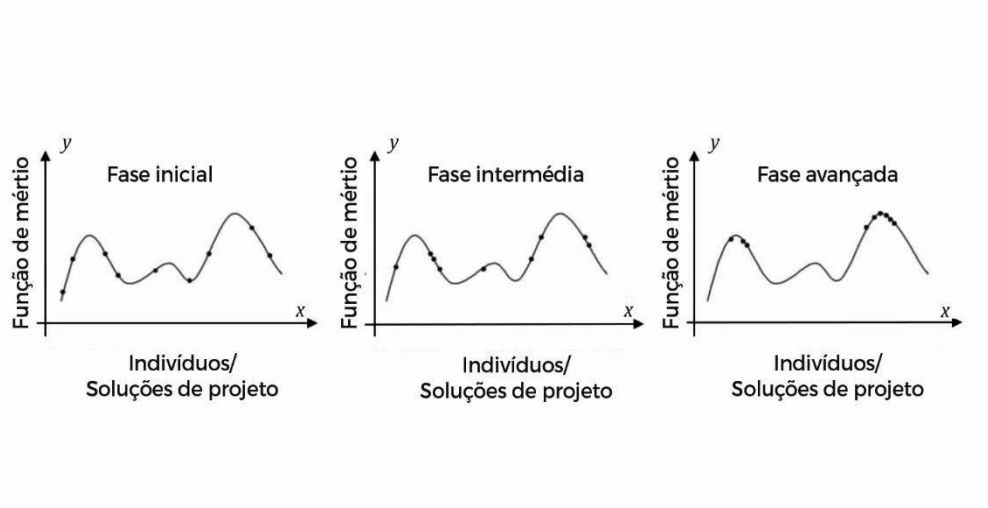


Figura 6.3. Progresso típico associado a um AE, ilustrado em termos de distribuição da população. Para cada ponto x no espaço de pesquisa, y mostra o seu valor de aptidão.

Embora não exista uma definição rigorosa universalmente aceite dos termos *exploração livre abrangente* (em inglês, *exploration*) e *exploração regulada* (em inglês, *exploitation*), estas noções são frequentemente usadas para classificar fases distintas do processo de pesquisa evolucionária. Grosso modo, a *exploração livre abrangente* (diversificação) é a geração de novos indivíduos em regiões ainda não testadas do espaço de pesquisa (ou de projeto), enquanto *exploração regulada* significa a concentração da pesquisa na vizinhança de boas soluções conhecidas tirando partido deste facto (intensificação).

Os processos de pesquisa evolucionária são frequentemente referidos em termos de uma troca equilibrada (em inglês, *trade-off*) entre *exploração livre abrangente* e *exploração regulada*. O uso excessivo da *exploração livre abrangente* embora extensiva, pode levar a uma busca ineficiente. Por outro lado, a *exploração regulada* acentuada,

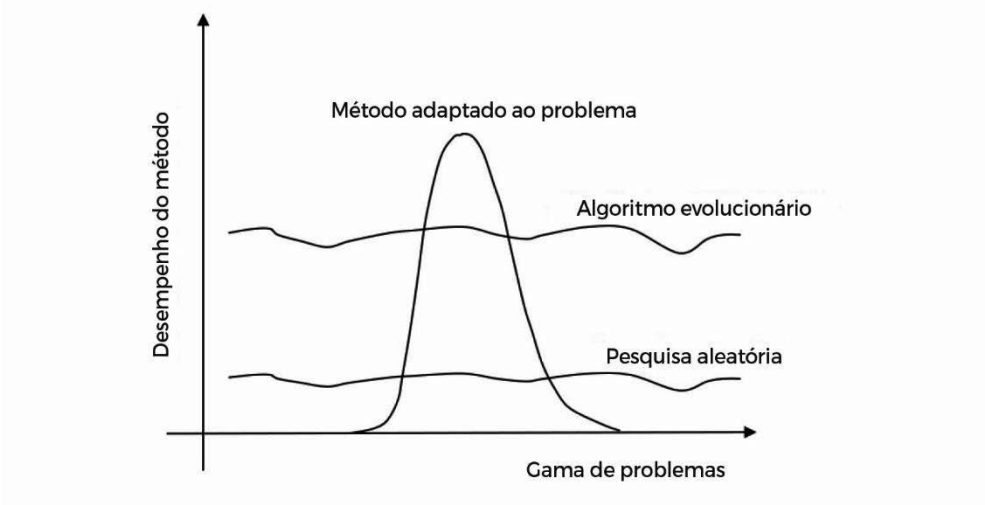


Figura 6.5. Perspetiva do desempenho dos AEs (EAs) após Goldberg (1989) [21].

A figura 6.5. mostra a visão do final da década de 80 do século XX depois de Goldberg [21]. O que a figura indica é que os AEs mostram um desempenho praticamente uniforme numa ampla gama de problemas. Este padrão de desempenho pode ser comparado à pesquisa aleatória e a algoritmos personalizados para um tipo de problema específico. Sugere-se que os AEs claramente superam a pesquisa aleatória.

Por outro lado, um algoritmo adaptado a problemas específicos tem um desempenho muito melhor que um AE, mas apenas no tipo de problemas para o qual foi desenhado. À medida que há um afastamento desse tipo de problema e para problemas diferentes, o desempenho do algoritmo definido para um problema específico diminui rapidamente. Nesse sentido, os AEs e os algoritmos para problemas específicos formam dois extremos opostos.

Esta perceção desempenhou um papel importante no posicionamento de AEs, enfatizando a diferença entre evolução e pesquisa aleatória, mas mudou gradualmente nos anos 90 com base em novas ideias práticas e também teóricas. A visão contemporânea reconhece a possibilidade de combinar os dois extremos num algoritmo híbrido.

7.1. Introdução

7.1.1. Contexto histórico

Após alguns anos de observação e experimentação, Charles Darwin apresentou em 1858 a sua teoria da *evolução através da seleção natural*, simultaneamente com outro naturalista Alfred Russel Wallace. Darwin publica em 1859 o livro *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* onde explica de forma detalhada a sua teoria sustentada por muitas evidências observadas nas suas viagens exploratórias a bordo do navio Beagle [1].

As ideias de Darwin ganharam novo impacto quando por volta de 1900 os cientistas redescobriram os princípios básicos da hereditariedade propostos por Gregor Mendel em 1865 [2]. Da sua conjugação resultou a moderna teoria da evolução que combina a genética e as ideias da seleção natural de Darwin e Wallace, criando o *princípio básico da Genética Populacional*. Este princípio diz que "*a variabilidade entre indivíduos numa população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética*".

Em meados do século XX, biólogos e matemáticos começaram a desenvolver simulações computacionais baseadas no princípio acima referido. O pioneiro foi John Holland que, na sequência do amadurecimento das suas ideias, veio a propor os *algoritmos genéticos* em 1975, no seu livro *Adaptation in Natural and Artificial Systems* [3]. Holland apresentou o *Algoritmo Genético* (AG) (em inglês, *Genetic Algorithm*, GA) como uma abstração da evolução biológica e criou um quadro teórico para a sua adaptação à resolução de problemas de otimização em cenários reais.

O AG de Holland é um método que transforma uma população de cromossomas numa nova população através de um processo de "*seleção natural*" juntamente com

Os algoritmos genéticos, tal como os organismos vivos no seu meio ambiente natural, necessitam de receber informação, $\mathbf{I}(\tau)$ sobre a adaptação da população de indivíduos, $\mathbf{P}(\tau)$, relativamente às condições ambientais em $\boldsymbol{\Omega}$. Na prática $\mathbf{I}(\tau)$ é reportada pela função de avaliação dos indivíduos que compõem a população. A partir desta informação os AGs modificam a população de soluções alcançando uma adaptação progressivamente melhor ao ambiente em que estão inseridas. Em termos gerais, pode-se escrever:

$$\mathbf{AG} : \mathbf{P}(\tau) \times \mathbf{I}(\tau) \rightarrow \mathbf{P}(\tau + 1) \tag{7.3}$$

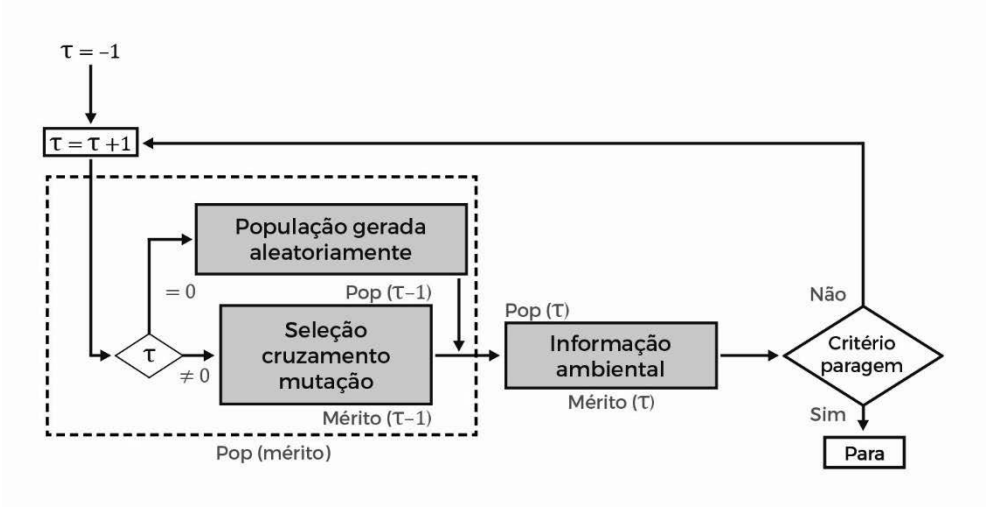


Figura 7.2. Diagrama de fluxo para um algoritmo genético genérico.

A figura 7.2. mostra o diagrama de fluxo de um algoritmo genético genérico. Como os AGs pertencem à família dos Algoritmos Evolucionários (AEs), os conceitos descritos no capítulo 6 sobre as características destes algoritmos permanecem válidos. Atente-se em particular à figura 6.2. (capítulo 6) onde se descreve o esquema evolutivo dos AEs.

mais espaço na roleta. A probabilidade de seleção $P(\mathbf{x}_j^k)$ do indivíduo *j* da população com o fenótipo $\mathbf{x}_j$, numa dada geração *k*, é dada por:

$$P(\mathbf{x}_j^k) = \frac{f(\mathbf{x}_j^k)}{\sum_{j=1}^{N_{pop}} f(\mathbf{x}_j^k)} \tag{7.23}$$

onde $f(\mathbf{x}_j^k)$ é o valor da função de mérito/aptidão e N_{pop} é a dimensão da população. Na figura 7.6. apresenta-se um exemplo da seleção proporcional pelo método da roleta.

Caso a função de mérito possa apresentar valores negativos, pode-se efetuar uma transformação com o intuito de evitar valores negativos (translação para que os valores sejam positivos). Este método tem a desvantagem de fazer com que um bom indivíduo se reproduza muitas vezes (e passe assim a sua carga genética) acabando com a diversidade da população rapidamente ao longo das gerações e ocasionando uma convergência inadequada à solução.

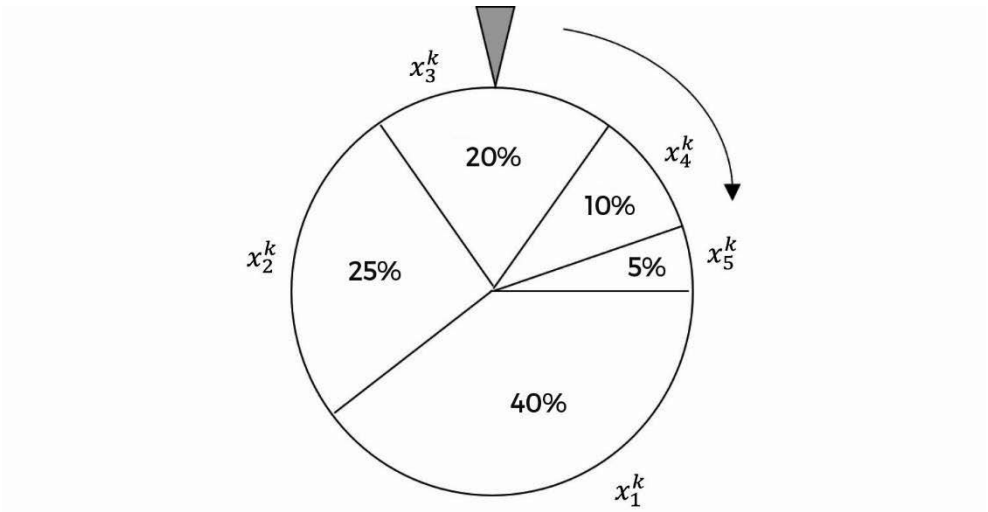


Figura 7.6. Exemplo de seleção pelo *método da roleta*.

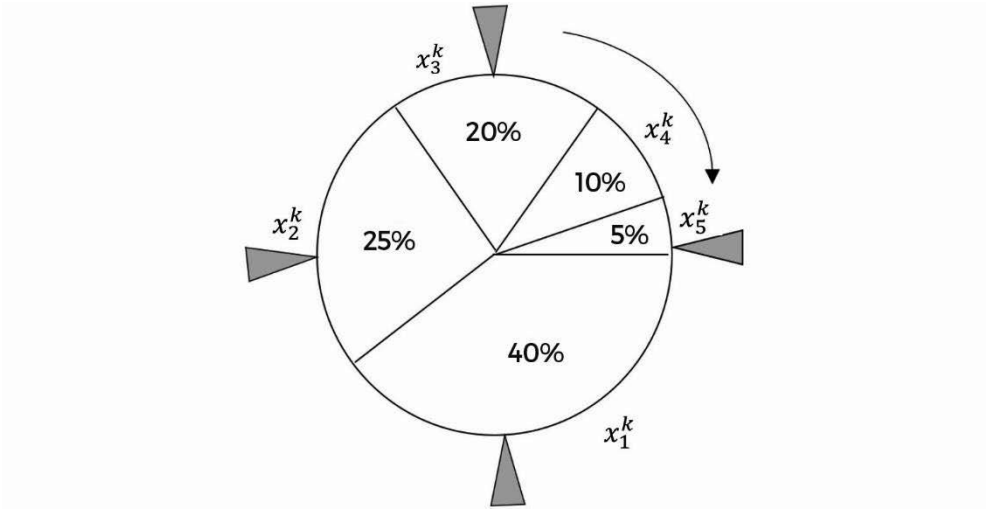


Figura 7.7. Exemplo de seleção pela *amostragem estocástica universal*.

7.6.2. Seleção baseada na ordenação

Os esquemas de seleção baseados na ordenação da população podem ser:

- Seleção por torneio (em inglês, *tournament selection*);
- Seleção por truncagem (em inglês, *truncation selection*);
- Seleção pelo método do ordenamento linear (em inglês, *linear ranking selection*);
- Seleção pelo método do ordenamento exponencial (em inglês, *exponential ranking selection*);
- Seleção elitista (em inglês, *elitist selection*).

Seleção pelo método do torneio

Escolhe-se de forma aleatória um par de indivíduos da população. Uma cópia do melhor dos dois é colocada na população da geração seguinte para recombinação. Os

Cruzamento de um ponto (1PX) (*single-point crossover*)

Este operador está estruturado em dois passos:

1. Considerem-se dois pais com cromossomas (genomas) $\mathbf{p}_1$ e $\mathbf{p}_2$ de comprimento l_c;

2. Sorteia-se aleatoriamente um número k (ponto de corte) qualquer tal que $0 < k < l_c$;

   2.1. O primeiro filho $\mathbf{s}_1$ receberá todos os genes de $\mathbf{p}_1$ de 1 até k e todos os genes de $\mathbf{p}_2$ de k+1 até l_c;

   2.2. O segundo filho $\mathbf{s}_2$ receberá os genes de $\mathbf{p}_1$ e de $\mathbf{p}_2$ de forma inversa ao procedimento seguido para $\mathbf{s}_1$.

Neste caso, a máscara de cruzamento $\mathbf{M}_i$ seria uma sucessão de 0s (zeros) seguida de uma sucessão de 1s (uns) pertencente ao conjunto $\mathbf{M}_i = \{(0n\ 1m) \mid n \geq 0,\ m \geq 0,\ n + m = l_c\}$. Este operador de cruzamento de um ponto tem normalmente fraco desempenho no contexto dos operadores tradicionais.

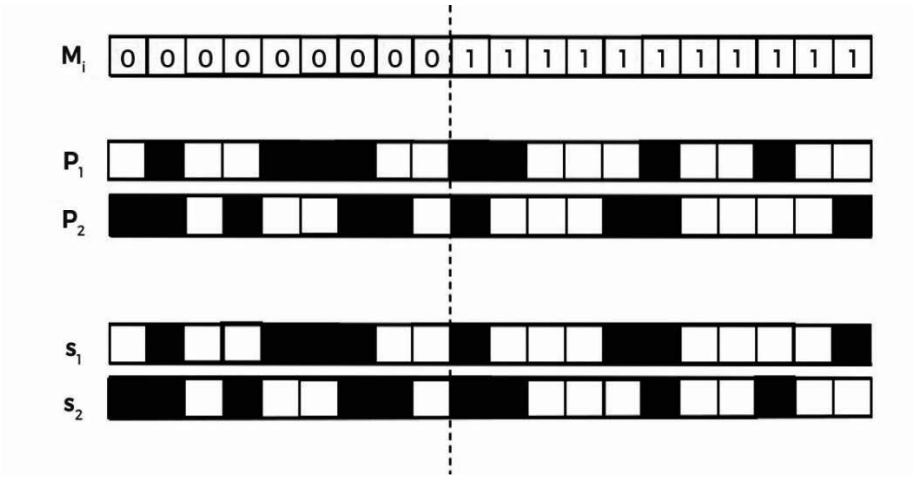


Figura 7.9. O operador de cruzamento de um ponto.

8.1. Introdução

Os *Algoritmos Meméticos* (AMs) (em inglês, *Memetic Algorithms*, MAs) foram desenvolvidos no contexto dos Algoritmos Evolucionários (AEs), introduzindo a aprendizagem individual como um processo independente de refinamento local, com o objetivo de acelerar a pesquisa na otimização. A ideia original era combinar as propriedades exploratórias globais da pesquisa genética com as potencialidades dos procedimentos de busca local o que conduziu ao aparecimento de várias técnicas híbridas. No entanto, esta perspetiva do AM como um procedimento de busca acelerada é uma visão muito simplista deste método. Na verdade, o trabalho de Dawkins [1] introduziu uma interpretação diferente do *darwinismo clássico* com base no gene em vez do cromossoma do indivíduo como a unidade fundamental da evolução. Esta nova interpretação das regras evolutivas foi formalizada pela "*Teoria do Gene Egoísta*" [1]. A ideia de Dawkins de *meme* é usada para analisar a *transmissão cultural* da mesma maneira como o gene é a unidade de *transmissão biológica*. O *meme* como unidade de transmissão cultural pode ser replicado de acordo com a percepção da sua utilidade ou popularidade sendo replicada e propagada através dos meios de comunicação.

Os AMs foram propostos por Moscato [2] em 1989, seguindo os princípios darwinianos da evolução natural e o conceito de *meme* de Dawkins. Originalmente o algoritmo memético foi apresentado como uma fusão da pesquisa global baseada em populações e da aprendizagem heurística (busca local). Assim, o carácter global da pesquisa é dado pela natureza evolutiva da abordagem, enquanto o procedimento de pesquisa local é normalmente realizado por meio de heurísticas inteligentes de busca local ou outros métodos adequados de otimização. Hoje em dia, os AMs são agrupadas segundo várias abordagens como Algoritmo Evolucionário Híbrido (em inglês, *Hybrid Evolutionary Algorithm*), Pesquisa Genética Local (em inglês, *Genetic*

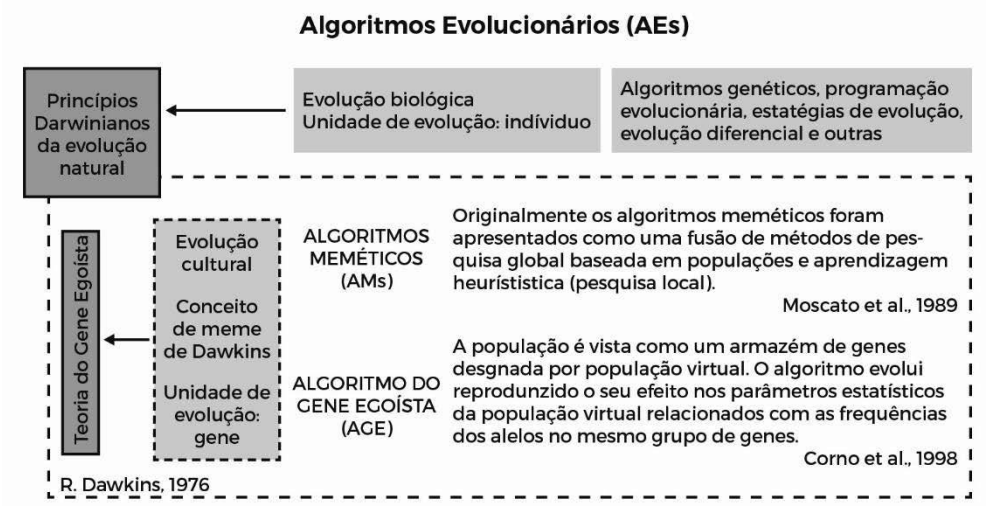


Figura 8.1. Taxonomia dos algoritmos evolucionários segundo dois paradigmas: Darwinismo Universal (Darwin) e a Teoria do Gene Egoísta (Dawkins).

8.2. Teoria do gene egoísta e os algoritmos meméticos

A *Teoria do Gene Egoísta* foi apresentada por Richard Dawkins em 1976 [1]. Dawkins usou pela primeira vez, o termo "*gene egoísta*" expressando uma visão genocêntrica da evolução. Esta teoria sugere que a evolução é melhor interpretada considerando que atua ao nível do gene e defende que o mecanismo de seleção em organismos ou populações é dominado pela escolha baseada nos genes. O termo *meme* também foi introduzido por Dawkins como uma *unidade de evolução cultural* humana análoga ao gene, sugerindo que a replicação egoísta pode também modelar a transmissão cultural humana. Exemplos de *memes* são: toadas, ideias, frases feitas, modas, processos de fabrico de vasos ou pontes, etc. Assim como os genes se propagam a si mesmos reunindo-se a outros genes saltando de corpo para corpo

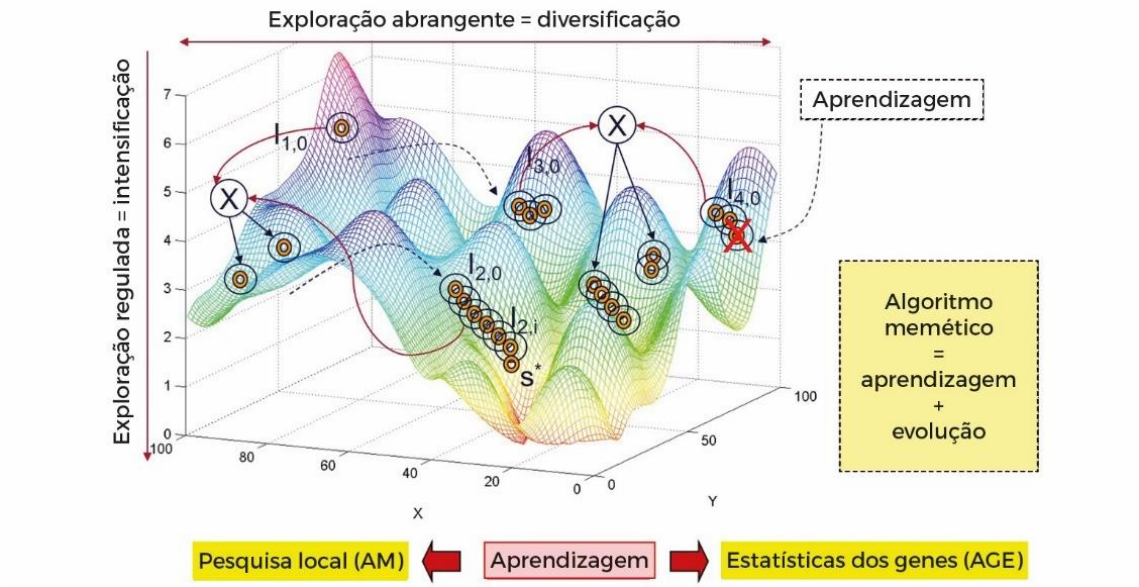


Figura 8.2. Efeito da simbiose evolucionária simultaneamente genética e cultural dos AMs.

O *Algoritmo Memético de Difusão* (AMD) (em inglês, *Diffusion Memetic Algorithm*, DMA) é também um bom exemplo da transferência não genética de *memes* no contexto da otimização evolucionária [23]. Este algoritmo baseia-se no conceito de *difusão dos memes*. Para o efeito a população é organizada numa grelha bidimensional e cada indivíduo é colocado numa célula desta grelha, como se mostra na figura 8.8., sendo associado por aprendizagem a um *meme* que será usado na implementação da pesquisa local.

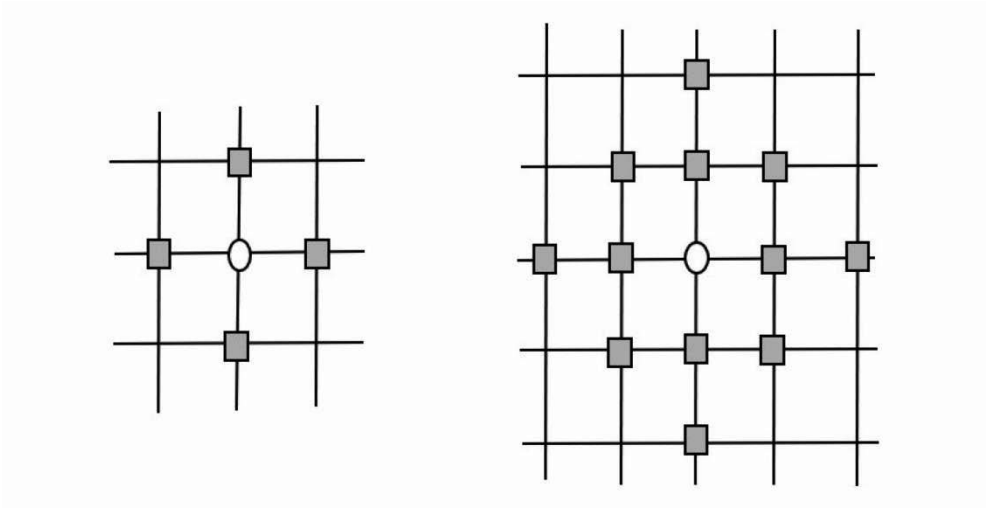


Figura 8.8. Exemplos de estruturas de vizinhança no AMD.

Em cada geração, um indivíduo em cada célula da malha acasala aleatoriamente com um dos seus vizinhos para gerar um novo descendente que toma o lugar do progenitor. Subsequentemente ocorre a aprendizagem dos *memes* com o objetivo de selecionar o *meme* que será atribuído ao descendente. O processo de aprendizagem individual é realizado a cada *n* gerações com cada indivíduo a ser aperfeiçoado pelo *meme* a ele associado. Aqui, *n* refere-se ao intervalo de aprendizagem individual que influenciará

9.1. Introdução

A natureza mostra-nos muitos exemplos impressionantes e intrigantes de comportamento "*de enxame*" de diferentes espécies de animais: bandos de pássaros numa variedade de números e de formações de voo; formigas que caçam e buscam comida coletivamente em grande número; cardumes de peixes que adotam formações unidas e em constante mudança, à medida que confundem predadores e atraem presas. As atividades dos chamados *enxames de organismos* resultam, em muitos casos, em efeitos benéficos para os próprios organismos. Este facto, levou ao conceito de *inteligência de enxame* (em inglês, *swarm intelligence*), que sintetiza o conceito de que o comportamento de um enxame no seu todo pode exibir resultados úteis, funcionais e inteligentes que parecem estar muito para além das capacidades previsíveis, tanto quanto entendemos, de qualquer indivíduo dentro do enxame.

Um indivíduo pertencente a um enxame é muitas vezes designado por "*agente*" especialmente na área da *Inteligência Artificial* (IA). Do ponto de vista técnico, um *agente* é algo capaz de perceber o seu ambiente por meio de sensores e de agir sobre esse ambiente por meio de atuadores.

Alguns exemplos de agentes são:

- Agente humano
    - Sensores: Olhos, ouvidos e outros órgãos sensoriais;
    - Atuadores: Mãos, pernas, boca e outras partes do corpo.
- Agente robótico
    - Sensores: Câmaras e detetores de infravermelho;
    - Atuadores: Vários motores.

Tradicionalmente são usados dois métodos:

- Método do melhor global, ***gbest***: Neste método, a vizinhança é definida como sendo toda a população de partículas;

- Método do melhor local, ***lbest***: Neste método, é associada uma dada topologia ao enxame. Assim, a vizinhança de uma partícula é o conjunto de partículas conectadas diretamente. A vizinhança pode ser um conjunto vazio no qual as partículas estão isoladas [4, 28].

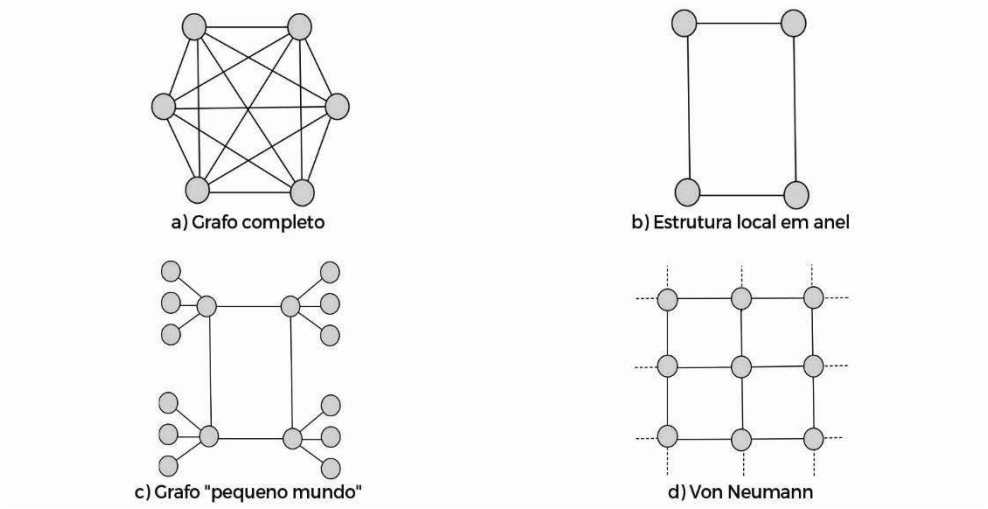


Figura 9.2 Enxames com diferentes redes sociais: a) Método *gbest* no qual a vizinhança é toda a população (grafo completo); b) Método *lbest* onde é usado um grafo incompleto para definir a estrutura da vizinhança (por exemplo, um anel no qual cada partícula tem apenas dois vizinhos); c) Topologia intermédia usando um grafo "pequeno mundo"; d) Vizinhança de von Neumann.

A figura 9.2. mostra quatro topologias diferentes: grafo completo, grafo em anel, grafo tipo "*pequeno mundo*" e grafo de von Neumann. Este modelo é semelhante aos modelos das ciências sociais baseados na imitação mútua dos membros da população,

da quantidade de alimento (processo de reforço). Mais alimento é sinónimo de mais formigas no trilho, logo mais feromonas depositadas. Quanto mais substância química depositada num trilho, mais formigas são atraídas, o que o torna ainda mais químico, reforçando o caminho que novamente atrai ainda mais formigas, numa espiral crescente.

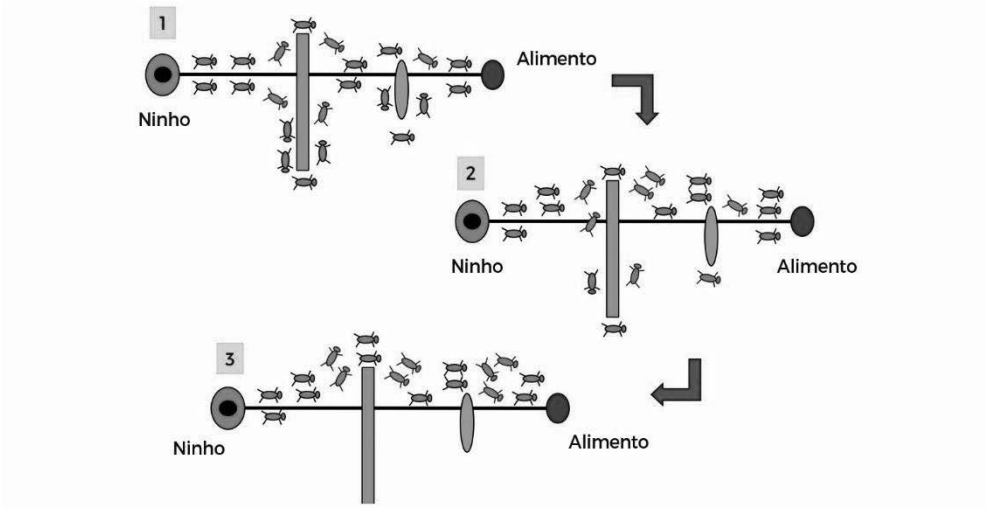


Figura 9.3. Inspiração numa colónia de formigas na procura (em três instantes diferentes, 1-2-3) de um caminho ótimo entre o ninho e uma fonte de alimento.

A figura 9.3. ilustra uma experiência inspirada em Goss et al. [43] com uma colónia de formigas argentinas (*Iridomyrmex humilis*). Note-se que estas formigas têm uma capacidade de visão limitada.

Conforme se mostra na figura 9.3., ao enfrentar um obstáculo, existe uma probabilidade igual para cada formiga de escolher o caminho à esquerda ou à direita. Como o trilho da esquerda é mais curto que o da direita e exige menos tempo de viagem, as

10.1. Introdução

A maioria dos problemas que ocorrem em cenários reais é por natureza de multiobjetivo, devido aos vários critérios de projeto que têm de ser considerados simultaneamente. Os primeiros estudos sobre problemas de otimização multiobjetivo baseavam-se na sua transformação numa sucessão de problemas de otimização de um objetivo. Isto envolveu o uso de abordagens como a ordenação lexicográfica (que otimiza um objetivo de cada vez, considerando primeiro o mais importante, conforme definido pelo projetista) e funções de agregação linear (que usam uma soma ponderada dos objetivos, na qual os pesos indicam a importância de cada um deles, conforme definido pelo projetista).

Ao longo dos tempos, outros tipos de abordagens foram propostos, visando obter soluções de compromisso sem necessidade de incorporar explicitamente as preferências do projetista. Atualmente, muitos algoritmos de multiobjetivo incorporam mecanismos para selecionar e guardar soluções que representam o melhor compromisso possível entre todos os objetivos considerados, sem qualquer necessidade de ordenar ou somar todos os objetivos.

A otimização multiobjetivo tem as suas raízes no século XIX, no trabalho de Edgeworth e Pareto na área da economia [1,2]. Foi usado nos campos das ciências de economia e de gestão durante várias décadas [3,4], e gradualmente nas ciências de engenharia [5]. Atualmente, a otimização multiobjetivo é uma área importante em ciência e engenharia.

A solução ótima para os problemas de otimização multiobjetivo não é uma solução única como no caso de problemas de otimização com um objetivo, mas um conjunto de soluções designadas como soluções ótimas de Pareto. Uma solução é um ótimo de Pareto se não for possível melhorar um dado objetivo sem deteriorar pelo menos um

Não é habitual ter uma solução $\mathbf{x}^*$ associada a um vetor de variáveis de projeto ou de decisão, em que $\mathbf{x}^*$ é ótimo para todos os objetivos:

$$\forall \mathbf{x} \in \mathbf{S},\ \ f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}),\ \ i = 1,2,\ldots,n \tag{10.2}$$

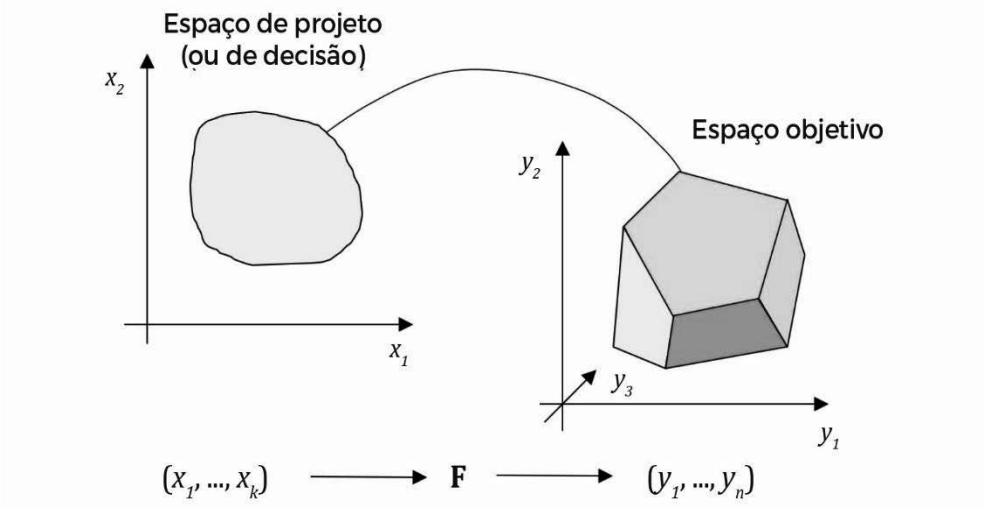


Figura 10.1. Espaço de projeto (ou de decisão) e espaço objetivo no problema de otimização multiobjetivo POM (MOP).

Dado que esta situação não é habitual nos problemas de otimização multiobjetivo em cenário real, onde os critérios estão em conflito, outros conceitos foram estabelecidos para considerar a otimalidade. Uma relação de ordem parcial pode ser definida, conhecida como relação de *dominância*.

Definição 10.2. Dominância de Pareto. *Diz-se que um vetor objetivo* $\boldsymbol{u} = (u_1, \ldots, u_n)$ *domina* $\boldsymbol{v} = (v_1, \ldots, v_n)$ *(designado por* $\boldsymbol{u} \prec \boldsymbol{v}$*) se e somente se nenhum componente de* $\boldsymbol{v}$ *for menor que o componente correspondente de* $\boldsymbol{u}$ *e pelo menos um componente de* $\boldsymbol{u}$ *for estritamente menor,*

Definição 10.7. Ponto de referência. *Um ponto de referência* $\boldsymbol{z}^* = (\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_n)$ *é um vetor que define o nível de aspiração (ou meta)* $\bar{z}_i$ *a ser alcançado para cada objetivo* f_i.

Definição 10.8. Ponto pior *(Nadir point).* *Um ponto* $\boldsymbol{y}^* = (y_1^*, y_2^* \ldots, y_n^*)$ *é o ponto pior (Nadir point) se ele maximiza cada função objetivo* f_i *de* $\boldsymbol{F}$ *sobre o conjunto de Pareto, isto é,* $y_i^* = max\big(f_i(\boldsymbol{x})\big),\ \boldsymbol{x} \in \mathcal{P}^*,\ i \in [1, n]$.

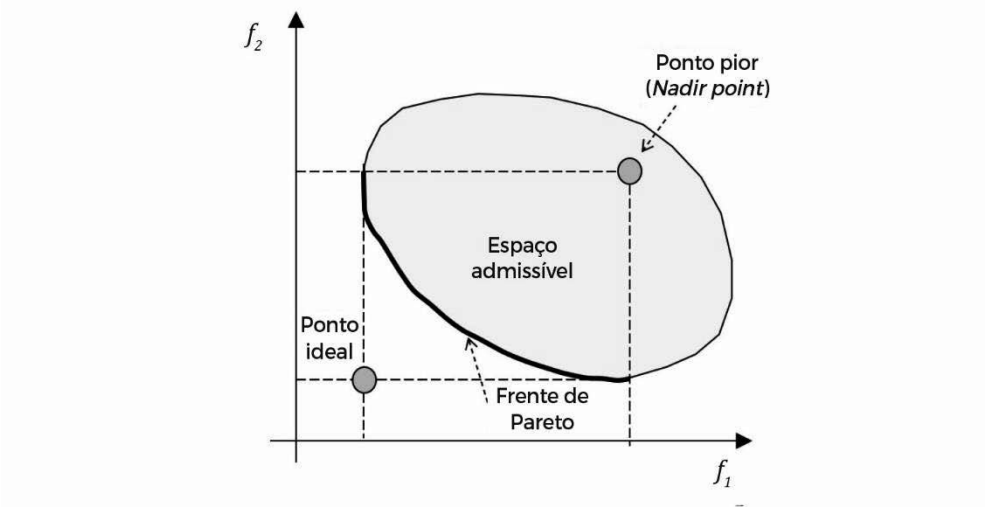


Figura 10.3. Os pontos "ideal" e "pior" em otimização multiobjetivo.

Os pontos "ideal" e "pior" dão alguma informação sobre a gama de valores da frente de Pareto e estão representados na figura 10.3.

Definição 10.9. Função de utilidade. *Uma função de utilidade (ou de valor)* v, *que representa as preferências do decisor (projetista), mapeia o vetor objetivo para uma função com valor escalar:* v: $\mathbb{R}^n \rightarrow \mathbb{R}$.

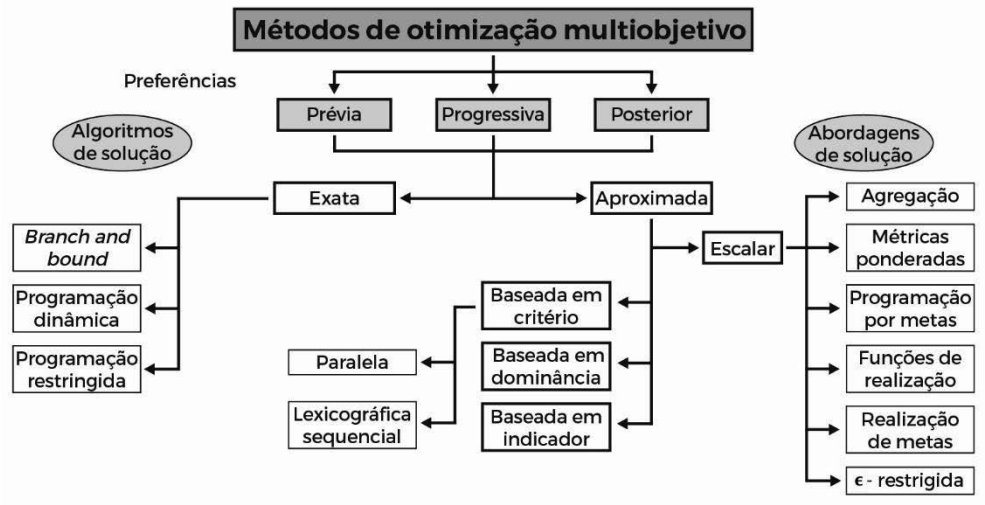


Figura 10.6. Taxonomia dos algoritmos de otimização multiobjetivo.

De acordo com a estratégia de definição do mérito, os algoritmos de otimização podem ser agrupados em quatro categorias, de acordo com a figura 10.6.:

- **Abordagens escalares**: São baseadas na transformação do problema de otimização multiobjetivo (POM) num problema de objetivo único. Esta classe de abordagens inclui, por exemplo, os métodos baseados na agregação que combina os vários objetivos f_i numa única função objetivo F. Estas abordagens requerem que o projetista ou decisor tenha um bom conhecimento do problema;

- **Abordagens baseadas em critério**: neste tipo de abordagem, a pesquisa é efetuada tratando os vários objetivos incomensuráveis (de espécies diferentes) separadamente;

- **Abordagens baseadas na dominância**: estas abordagens usam o conceito de dominância e a otimalidade de Pareto para guiar o processo de pesquisa. O vetor objetivo das soluções é parametrizado usando a relação de dominância;

OTIMIZAÇÃO DE SISTEMAS EM ENGENHARIA

Fundamentos e algoritmos para o projeto ótimo

Carlos Conceição António

Sobre a obra

A otimização é um ramo da área da matemática e dos métodos numéricos que tem sido objeto de investigação teórica e aplicada ao longo das últimas décadas. Em particular, a engenharia é uma das áreas onde a otimização tem encontrado terreno propício para a sua aplicação e desenvolvimento teórico e prático. As técnicas de otimização alcançaram uma maturidade sem precedentes e são usadas num largo espectro de aplicações industriais em diferentes áreas, nomeadamente engenharia estrutural, aeroespacial, automóvel, química, eletrónica, fabricação assistida por computador e produção industrial.

A otimização pode ser definida como o estabelecimento racional de um projeto que é o melhor dentro de todos os possíveis de acordo com um ou mais objetivos predefinidos e obedecendo a um conjunto prescrito de restrições. Em sentido lato, as condições referidas requerem uma integração adequada entre duas áreas bem definidas: a *otimização em engenharia* e a *otimização matemática*. Desta interação resulta a integração da *otimização no ciclo de projeto de sistemas em engenharia*.

Para aplicar os conceitos matemáticos à pesquisa do projeto ótimo de sistemas em engenharia, deve-se formular o problema matemático subjacente e construir o respetivo modelo. Para tal, é necessário definir as variáveis de projeto, os objetivos e as restrições do problema de otimização. Neste livro, é feita uma apresentação detalhada sobre os conceitos matemáticos de otimização, bem como a descrição dos principais algoritmos de acordo com a sua taxonomia.

Considera-se constituir esta obra um importante contributo para a disseminação do conhecimento nesta área e uma referência fundamental para estudantes, investigadores, cientistas e engenheiros.

Sobre o autor

Carlos Conceição António

Nasceu em 1957, em Lobito, Angola. Em 1988, concluiu a licenciatura em Engenharia Mecânica na Faculdade de Engenharia da Universidade do Porto (FEUP), onde também foi bolseiro de investigação em 1988 e 1989. Obteve o grau de Mestre em 1991 em Engenharia Estrutural e o grau de Doutor em Engenharia Mecânica em 1995 pela mesma faculdade, em ambos os graus defendendo teses na área da otimização. Obteve o título de Professor Agregado em Engenharia Mecânica em 2005 com uma lição de síntese em otimização.

Fez toda a sua carreira de docente, de investigador, de gestão e de extensão universitária no Departamento de Engenharia Mecânica (DEMec) da FEUP, desde 1988, nos institutos de interface IDMEC-Pólo FEUP e INEGI. Lecionou principalmente na área da Matemática, em particular nas unidades curriculares de Análise Matemática, tendo também lecionado na área da Otimização. É neste momento Professor Associado Agregado da FEUP.

É membro da ISSMO – *International Society for Structural and Multidisciplinary Optimization* e da Ordem dos Engenheiros de Portugal.

Também disponível em formato *e-book*

ISBN: 978-989-901-734-4

www.**engebook**.pt

engebook