3

Otimização

Neste capítulo são descritos conceitos básicos de um problema de otimização, sua classificação com relação ao comportamento das funções objetivos e das respectivas restrições impostas. Em seguida, outros conceitos importantes como otimização multiobjetivo, dominância e fronteira de Pareto são apresentados. Finalmente, é apresentado em detalhes o método de otimização multiobjetivo, utilizado neste trabalho para a otimização da configuração RCHA.

Martins (2011) define o problema de otimização como o processo de busca dos argumentos (ou variáveis de projeto) que resultam no valor extremo de uma ou mais funções que representem o problema em questão. Carbono (2005) apresenta a otimização como sendo o conjunto de procedimentos por meio dos quais se busca minimizar ou maximizar uma determinada função, denominada função objetivo, sujeita ou não a restrições de igualdade e desigualdade, obtendo assim, um melhor aproveitamento dos recursos disponíveis. Assim, é possível encontrar na literatura diversas definições para a otimização. De maneira global, a otimização é o processo de pesquisa da melhor opção em termos do aproveitamento dos recursos dentro de uma categoria de possíveis soluções a partir das variáveis do projeto.

No contexto da engenharia *offshore* a otimização pode ser aplicada para se escolher a melhor configuração geométrica de um *riser*, ou seja, selecionar a configuração que melhor se adequa aos objetivos desejados (por exemplo: minimizar as tensões na região de contato entre o *riser* e o solo; maximizar a força de tração no topo do *riser*; minimizar o volume de flutuadores do *riser* etc.) dentre as diversas soluções ou configurações possíveis do modelo estrutural. Essa escolha é determinada pela funcionalidade que se quer otimizar, representada por uma função objetivo, assim como pelas variáveis de entrada do projeto e pelas restrições que se pretende impor ao problema. Dessa maneira, uma forma usual de se formular um problema de otimização é a seguinte:

Encontrar:

$$\arg\min/\max\ f_{obj}\,(x) \tag{3.1}$$

tal que:

$$g_i(x) \leq 0 \text{ para } i = 1, \dots, n_g \tag{3.2}$$

$$h_j(x) = 0 \text{ para } j = 1, \dots, n_h \tag{3.3}$$

$$lb_k \leq x_k \leq ub_k \text{ para } k = 1, \dots, n_x \tag{3.4}$$

PUC-Rio - Certificação Digital Nº 1121493/CA

onde f_{obj} é a função objetivo a ser minimizada (ou maximizada); $g_i(x)$ é a i-ésima restrição de desigualdade; n_g é o número de restrições de desigualdade; h_j é a j-ésima restrição de igualdade; n_h é o número de restrições de igualdade; x_k é o k-ésimo elemento do vetor de variáveis de projeto; n_x é o número de variáveis de projeto; e lb_k e ub_k são os limites inferior e superior da variável de projeto x_k, respectivamente. É importante destacar que tanto as restrições de igualdade quanto as de desigualdade são opcionais, ou seja, podem ou não existir em um determinado problema de otimização. A seguir, os conceitos básicos de otimização são abordados com mais detalhes.

3.1
Conceitos básicos

3.1.1
Função objetivo

A função objetivo (Eq. 3.1.) é a representação matemática do critério de eficiência adotado no problema de otimização. É influenciada pelas variáveis de projeto, conhecidas também como variáveis de controle do problema. Esse critério pode ser representado por uma única função objetivo, no caso de problemas mono-objetivos, ou por várias funções objetivo, no caso de problemas multiobjetivos. Em um problema de engenharia estrutural, se o projetista quiser minimizar os esforços produzidos na estrutura, o problema pode ser definido como mono-objetivo, mas se, adicionalmente, ele quiser diminuir o custo econômico da estrutura, o problema passa a ser multiobjetivo (minimizar os esforços e o custo da estrutura). Da mesma forma, quando o critério de eficiência é influenciado por uma única variável, a função objetivo é dita unidimensional; e no caso em que várias variáveis tenham influência sobre o critério, a função objetivo é denominada multidimensional. Além disso, se a função tem um único mínimo (ou máximo), ela é chamada de função unimodal; e quando a função tem vários mínimos (ou máximos), ela é denominada multimodal.

3.1.2
Variáveis de projeto

As variáveis de projeto (Eq. 3.4.) são os parâmetros variáveis no processo de otimização que definem as características do modelo analisado. Em outras palavras,

PUC-Rio - Certificação Digital Nº 1121493/CA

são as variáveis que se deseja determinar dentro de um intervalo prédefinido e para as quais o problema é resolvido. De forma geral, as variáveis podem adotar qualquer valor, mas isso poderia produzir soluções inviáveis no processo de otimização. Por isso, é necessaário introduzir as variáveis do projeto em um intervalo cujo limite inferior lb_k e cujo limite superior ub_k delimitem a variação das mesmas. Igualmente importante é definir o tipo de variável, seja ela discreta ou contínua, podendo existir problemas onde possam ser utilizados ambos os tipos de variáveis. Uma variável contínua é aquela que pode assumir qualquer valor dentro de um intervalo de variação, enquanto que uma variável discreta é aquela que pode assumir valores específicos. Um exemplo de variáveis discretas pode ser o diâmetro externo do *riser*, quando o projeto está limitado aos valores de diâmetros encontrados comercialmente.

No caso da engenharia estrutural, as variáveis de projeto podem representar dentre outras, a geometria da estrutura, as propriedades de seus materiais e as forças impostas.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.1.3
Restrições

As restrições (Eqs. 3.2 e 3.3) são equações de igualdade ou de desigualdade que determinam os limites de viabilidade do projeto, expressando uma condição desejável do comportamento do sistema. Geralmente, as restrições estão relacionadas com a geometria, os esforços admissíveis, os recursos disponíveis, os custos envolvidos, dentre outros.

3.1.4
Espaço de busca

O espaço de busca é a região do domínio que satisfaz às restrições do problema. É também conhecido como domínio viável do problema. É delimitado pelas restrições impostas ao sistema e pelo intervalo de variação das variáveis do projeto. Podem existir “melhores soluções” fora do espaço de busca do problema, porém, tecnicamente, elas são inviáveis pelo fato de violarem as restrições.

3.1.5
Ponto ótimo

O ponto ótimo é o vetor das variáveis de projeto que minimiza (ou maximiza) a função objetivo e satisfaz as restrições do problema. O valor da função objetivo correspondente a essas variáveis é denominado valor ótimo. Consequentemente, o

par formado pelo ponto ótimo e o valor ótimo é chamado de solução ótima, podendo ser local, se for um ponto de mínimo (ou máximo) em relação a uma vizinhança desse ponto, ou global, se for um valor mínimo (ou máximo) em relação a todo o espaço de busca do problema.

3.1.6
Programação matemática

É a área da matemática que estuda os problemas de otimização.

3.2
Classificação dos problemas de otimização

Geralmente, os problemas de otimização são classificados em relação a dois critérios principais: quanto ao comportamento das funções envolvidas e quanto à existência de restrições.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.2.1
Com relação ao comportamento das funções

De acordo com as características da função objetivo e das restrições, os problemas de otimização podem ser classificados como:

Programação linear: Quando a função objetivo e as restrições são lineares com relação às variáveis de projeto. Segundo Carbono (2005), a representação padrão de um problema de programação linear é da seguinte forma:

Minimizar:

$$f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots\dots\dots + c_nx_n \quad (3.5)$$

Sujeita a:

$$a_{11}x_1 + a_{12}x_2 + \dots\dots\dots + a_{1n}x_n = b_1 \quad (3.6)$$

$$a_{21}x_1 + a_{22}x_2 + \dots\dots\dots + a_{2n}x_n = b_2 \quad (3.7)$$

$$\dots\dots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots\dots\dots + a_{mn}x_n = b_m \quad (3.8)$$

e

$$x_1 \geq 0, x_2 \geq 0, \dots\dots\dots, x_n \geq 0 \quad (3.9)$$

onde f é a função objetivo, x_i são as variáveis de projeto, e a_i, b_i, c_i são os coeficientes da função objetivo e das restrições do problema.

Programação não linear: Quando a função objetivo é não linear e pelo menos uma das restrições é uma função não linear das variáveis do projeto. Em geral, os métodos para solução de problemas de programação não linear são classificados em dois subgrupos: determinísticos e não determinísticos.

Nos métodos determínisticos, também conhecidos como métodos clássicos, a busca do ponto ótimo é realizada minimizando-se a função objetivo em sucessivas direções, a partir da seguinte fórmula de recorrência:

$$x_{k+1} \approx x_k + \lambda_k d_k \tag{3.10}$$

onde x_k é a posição corrente, x_{k+1} é a próxima posição que minimiza a função objetivo, λ_k é o passo de cálculo (busca linear), d_k é a direção de busca e k é o índice que identifica a iteração corrente (Figura 3.1).

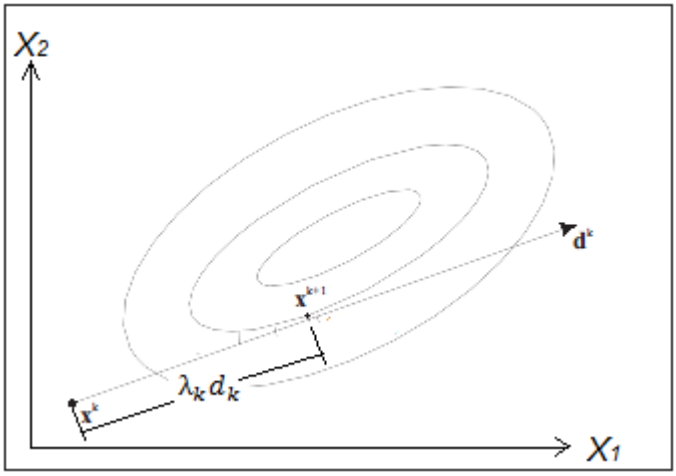


Figura 3.1 – Busca do ponto ótimo baseada em métodos clássicos
(Vaz, L.E. et al., 2009)

Em geral, a obtenção das direções de busca envolve o cálculo de derivadas (vetores gradientes e matrizes *Hessianas*) ou suas respectivas aproximações numéricas. A diferença entre os métodos de programação não linear consiste na estratégia adotada para a determinação do vetor d_k, caracterizando-os como sendo de ordem zero (exemplos: Univariante, *Powell*), nos quais nenhuma informação de

PUC-Rio - Certificação Digital Nº 1121493/CA

derivada é utilizada na determinação dessas direções; de ordem um (exemplos: *Steepest Decent*, *Fletcher-Reeves*, *BFGS*), nos quais apenas as informações de primeira ordem (vetores gradientes) são utilizadas; e de ordem dois (exemplo: *Newton-Raphson*), nos quais são utilizadas tanto as informações de primeira ordem (vetores gradientes) quanto as de segunda ordem (matrizes *Hessianas*).

Os métodos não determinísticos são aqueles que procuram "imitar" fenômenos ou processos encontrados na natureza e pertencem a uma área denominada de computação natural. Dentre esses métodos, estão os algoritmos genéticos, que são inspirados na teoria da genética e no princípio Darwiniano da evolução das espécies. Os algoritmos genéticos são muito utilizados em problemas multiobjetivos (problemas de otimização com mais de uma função objetivo) e foram adotados neste trabalho como a ferramenta básica de otimização. Detalhes teóricos sobre os algoritmos genéticos e a otimização multiobjetivo serão apresentados nas seções subsequentes.

Programação quadrática: É um caso particular da programação não linear no qual a função objetivo é quadrática e as restrições são funções lineares das variáveis do projeto.

A Tabela 3.1 apresenta um resumo da classificação dos problemas de otimização com relação ao comportamento das funções envolvidas.

Tabela 3.1 – Classificação dos problemas de otimização quanto ao comportamento das funções objetivo e das restrições.

$\boldsymbol{f}_{obj}(\boldsymbol{x})$	$\boldsymbol{g_i}(\boldsymbol{x}), \boldsymbol{h_j}(\boldsymbol{x})$	***Problema***
Linear	Lineares	Programação linear
Quadrática	Lineares	Programação quadrática
Não linear	Não lineares	Programação não linear

3.2.2
Com relação à existência de restrições

Os problemas de otimização se classificam também quanto à existência de restrições, podendo ser sem restrições (OSR, "otimização sem restrições") ou com restrições (OCR, "otimização com restrições").

Neves (1997) apresenta uma classificação geral dos métodos clássicos de programação matemática para resolver problemas de otimização, conforme ilustrado na Figura 3.2.

PUC-Rio - Certificação Digital Nº 1121493/CA

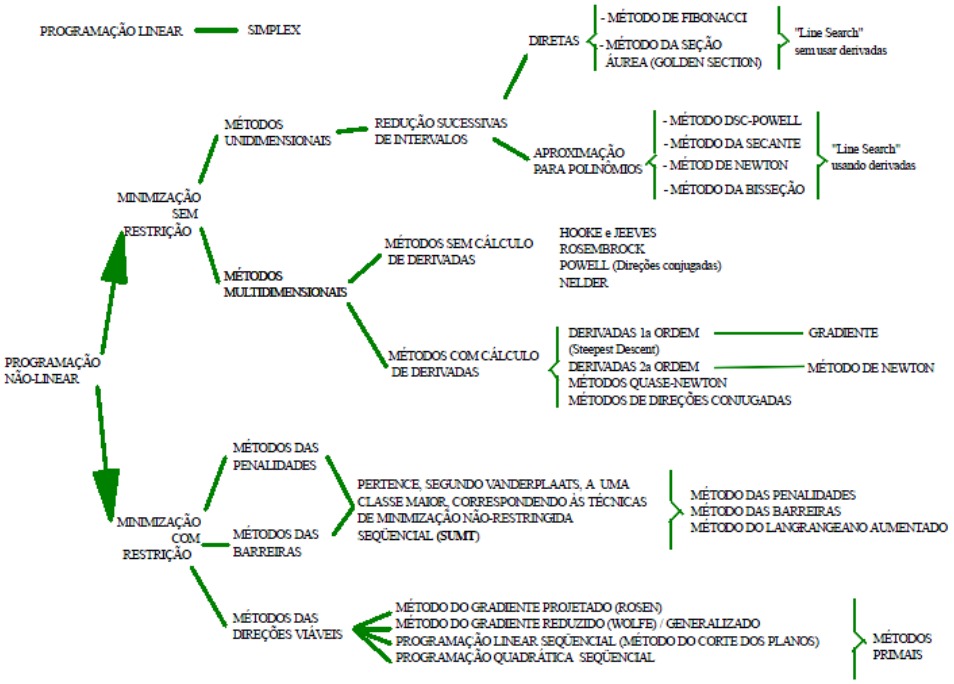


Figura 3.2 – Classificação geral dos métodos de otimização (Senra, 2004)

3.3
Otimização multiobjetivo

Quando um problema de otimização é representado por uma única função objetivo ele é definido como “mono-objetivo” ou *SOOP* (*single-objective otimization problem*). Nessa classe de problemas, os métodos de otimização tentam encontrar uma solução “viável” que represente o mínimo (ou máximo) da função objetivo. Entretanto, há casos em que se procura obter a “melhor resposta” em mais de um aspecto do sistema e, consequentemente, atender a vários objetivos simultaneamente. Esse tipo de problema é definido como “multiobjetivo” ou *MOOP* (*multi-objective otimization problem*). Nesses casos, é comum que esses objetivos sejam conflitantes, ou seja, buscar a melhor resposta do problema em relação a um dado objetivo pode representar uma resposta não tão boa em relação a outros objetivos do problema.

Um exemplo de problema com objetivos conflitantes pode ser o de se buscar o preço mais baixo de um carro levando-se em consideração o seu nível de conforto interno: quanto mais confortável, mais elevado será o seu preço; ou seja, esses objetivos são conflitantes. O ideal é que o carro seja o mais barato e o mais confortável possível, mas isso não acontece na realidade. Portanto, não existe uma única solução “ótima”, e sim um conjunto de soluções. Tais soluções são ditas “ótimas”

PUC-Rio - Certificação Digital Nº 1121493/CA

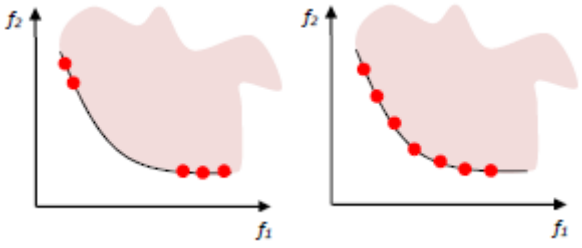
porque não existem outras, melhores do que elas, no espaço de busca, considerando-se todos os objetivos simultaneamente.

Um outro exemplo de objetivos conflitantes é encontrado com certa frequência nas implementações computacionais, onde se buscam algoritmos que forneçam resultados de melhor qualidade com o menor custo computacional. À medida em que um algoritmo procura resultados com uma melhor qualidade, normalmente, ele vai se tornando mais lento. Dessa forma, nenhuma solução de melhor qualidade e que seja mais lenta pode ser considerada superior a outra de pior qualidade porém mais rápida. Contudo, existem soluções consideradas superiores a outras, ou seja, aquelas que são rápidas e que fornecem resultados de qualidade. Tais soluções são conhecidas como *não dominadas*, enquanto que as soluções que podem ser superadas por outras são conhecidas como *dominadas*.

Portanto, o grande desafio da otimização multiobjetivo é encontrar um conjunto de soluções não dominadas para que o projetista seja capaz de escolher, dentre elas, a que melhor possa atender às necessidades do projeto.

Deb (2001) apresenta as principais diferenças entre um problema *SOOP* e *MOOP*:

Metas diferentes: enquanto a otimização mono-objetivo procura encontrar a "solução ótima", de preferência global, de um dado problema, a otimização multiobjetivo apresenta duas metas principais: encontrar o conjunto de soluções não dominadas em equilíbrio de compromisso com todos os objetivos envolvidos, e manter a diversidade entre essas soluções, ou seja, garantir que elas estejam bem distribuídas no espaço de soluções, favorecendo assim o aparecimento de "soluções ótimas" mais variadas em termos de valores de funções objetivos. A Figura 3.3 apresenta a distribuição de soluções ótimas no espaço de soluções: a parte (a) apresenta soluções mal distribuídas (concentradas em determinadas regiões), enquanto que a parte (b) apresenta uma boa distribuição das soluções.

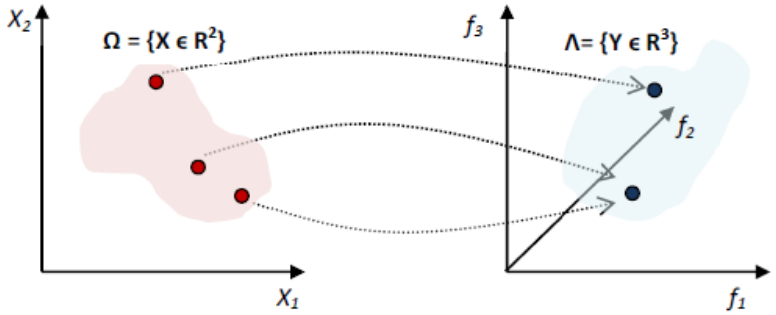


(a) (b)

Figura 3.3 – Distribuição de "soluções ótimas" no espaço de soluções (Martins, 2011)

PUC-Rio - Certificação Digital Nº 1121493/CA

Quantidade de espaços de busca envolvidos. O espaço de busca em um problema *SOOP* é delimitado pelos intervalos de suas variáveis de projeto e de suas restrições. Entretanto, em um problema *MOOP*, além do espaço das variáveis surge um novo espaço de busca associado às funções objetivo e conhecido como espaço objetivo. Dessa maneira, é possível mapear as variáveis de projeto com o seu correspondente espaço objetivo. Este processo pode não ser unívoco, ou seja, diferentes vetores de decisão podem resultar em uma mesma resposta ou solução no espaço objetivo. A Figura 3.4 ilustra essa relação de mapeamento entre duas variáveis de projeto e o espaço objetivo delimitado por três objetivos.



Espaço das variáveis do projeto Espaço das funções objetivo

Figura 3.4 – Mapeamento das variáveis de projeto no espaço objetivo (Martins, 2011)

Por outro lado, o problema de otimização multiobjetivo pode ser abordado de duas formas:

i. Definindo pesos ou prioridades aos objetivos de interesse, segundo informações fornecidas *a priori*, transformando o problema multiobjetivo em um problema mono-objetivo equivalente e, dessa maneira, poder aplicar métodos clássicos de otimização ou algum outro método específico para problemas com um único objetivo.

ii. Sem informações adicionais que permitam definir pesos, a segunda abordagem consiste em se encontrar as soluções de Pareto para que, dentre elas, o projetista possa escolher a “melhor”, de acordo com os seus critérios, utilizando métodos não determinísticos.

O grande desafio da primeira abordagem é a definição dos pesos para cada objetivo, devido ao fato de que uma escolha errada desses pesos pode conduzir a “soluções ótimas” de baixa qualidade. Portanto, na falta de informações qualitativas e

PUC-Rio - Certificação Digital Nº 1121493/CA

quantitativas dos problemas multiobjetivos, que permitam uma definição apropriada dos respectivos pesos, e considerando-se que os métodos clássicos não foram projetados para trabalhar com múltiplos objetivos, a segunda abordagem (que adota

métodos não determinísticos) torna-se mais adequada, além de mais interessante, para o desenvolvimento dos problemas multiobjetivos. Esta foi a abordagem adotada neste trabalho.

3.3.1
Definição de um problema multiobjetivo

Segundo Castro (2001), um problema multiobjetivo pode ser descrito por um vetor y de *n* funções objetivo f_i que dependem de um vetor $\boldsymbol{x}$ de *m* variáveis de projeto. Dessa maneira, uma vez que um problema *MOOP* busca satisfazer simultaneamente essas funções objetivo, a Equação 3.1. pode ser reescrita como:

$$\text{achar}\arg\min \; y = f_{obj}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})] \tag{3.11}$$

sujeito a

$$\boldsymbol{x} = (x_1, x_2, \dots\dots\dots, x_m) \in \mathrm{X} \tag{3.12}$$

$$\boldsymbol{y} = (y_1, y_2, \dots\dots\dots, y_m) \in Y \tag{3.13}$$

onde:

$\boldsymbol{y}$: vetor objetivo;
f_i : funções objetivos;
Y : espaço objetivo;
$\boldsymbol{x}$: vetor de decisão;
x_i : variáveis de decisão;
X : espaço das variáveis de decisão;

Em outras palavras, o problema de otimização consiste em se encontrar um vetor $\boldsymbol{x} = (x_1, x_2, \dots\dots\dots, x_m)$, das variáveis de projeto, que minimiza (otimiza) a função vetorial $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})]$.

Caso o problema apresente restrições, o vetor das variáveis de projeto também deve satisfazê-las, ou seja:

PUC-Rio - Certificação Digital Nº 1121493/CA

$$g_i(\boldsymbol{x}) \geq 0 \qquad i = 1,2,\ldots\ldots\ldots,m \tag{3.14}$$

$$h_j(\boldsymbol{x}) = 0 \qquad j = 1,2,\ldots\ldots\ldots,p \tag{3.15}$$

onde m e p representam, respectivamente, o número de restrições de desigualdade e igualdade.

O conjunto de soluções desse problema consiste em todos os vetores de decisão que não podem melhorar algum objetivo sem degradar pelo menos um dos demais. Esse conjunto de soluções é conhecido como “Ótimo de Pareto”.

3.3.2
Definições de Pareto

Sem perda de generalidade, são apresentadas, a seguir, as definições formais de Pareto para um problema de minimização:

Dominância de Pareto

Dados dois pontos $U = (u_1, u_2, \ldots\ldots, u_m)$ e $V = (v_1, v_2, \ldots\ldots, v_m)$, diz-se que U domina V se:

$$\forall i \in \{1,2,\ldots\ldots,n\} : f_i(U) < f_i(V) \text{ e } \exists\ j \in \{1,2,\ldots\ldots n\} : f_j(U) < f_j(V) \tag{3.16}$$

Essa definição é ilustrada na Figura 3.5.

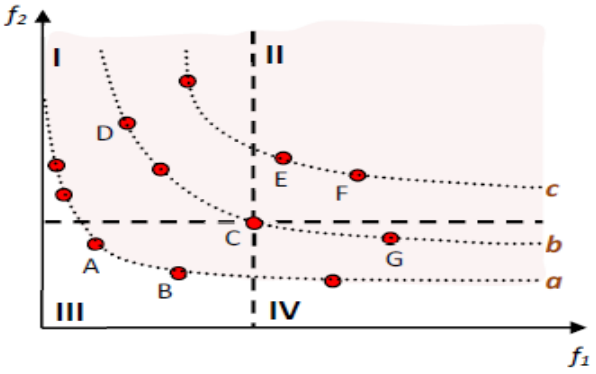


Figura 3.5 -Exemplo do conceito de dominância no espaço objetivo (Martins, 2011)

PUC-Rio - Certificação Digital Nº 1121493/CA

Baseando-se na definação de dominância, é possivel concluir que os pontos do quadrante II, *E* e *F,* são dominados pelo ponto *C*, assim como os pontos do quadrante III, *A* e *B,* dominam *C*. Por outro lado, os pontos do quadrante I e IV, D e G, respectivamente, e o ponto *C* são indiferentes entre eles. A Tabela 3.2 apresenta um resumo da dominância em relação ao ponto *C*.

Tabela 3.2 – Relação de dominância entre pontos no espaço objetivo.

II. Dominados por *C*	III. Dominam *C*	I e IV. Indiferente a *C*
$f_1(C) < f_1(E)$	$f_1(A) < f_1(C)$	$f_1(D) < f_1(C)\ mas\ f_2(D) > f_2(C)$ $f_1(G) > f_1(C)\ mas\ f_2(G) < f_2(C)$
$f_1(C) < f_1(F)$	$f_1(B) < f_1(C)$	
$f_2(C) < f_2(E)$	$f_2(A) < f_2(C)$	
$f_2(C) < f_2(F)$	$f_2(B) < f_2(C)$	

Solução não dominada ou Ótimo de Pareto

O ponto U é uma solução não dominada ou um Ótimo de Pareto, se não existe outro ponto V , tal que:

$$V = f(x) = (v_1, v_2, ..., v_k) \text{ domine } U = f(x^*) = (u_1, u_2, ..., u_k) \qquad (3.17.)$$

onde V e U pertencem ao espaço *S*, que representa o espaço de busca das soluções.

Dessa maneira, estabelece-se que uma solução U é um Ótimo de Pareto quando não existe outra solução V que melhore algum objetivo sem degradar pelo menos um dos demais.

Fronteira de Pareto

De acordo com Deb (2001), existe uma pequena diferença entre um conjunto de soluções não dominadas e o conjunto de soluções de Pareto denominada “Fronteira de Pareto”. As soluções não dominadas são definidas em apenas uma amostra do espaço *S*, enquanto que a Fronteira de Pareto são aquelas soluções não dominadas definidas na totalidade P^*, do espaço de busca das soluções. Assim, a Fronteira de Pareto pode ser definida como:

PUC-Rio - Certificação Digital Nº 1121493/CA

$$FP^* = \{U = f_{obj}(x^*) | x^* \in P^*\} \quad (3.18)$$

3.3.3

Exemplo ilustrativo da Fronteira de Pareto

Com a finalidade de ilustrar o conceito de Fronteira de Pareto, o problema de otimização multiobjetivo, proposto a seguir e testado por Schaffer (1984), é definido como:

Minimizar a função f que é composta por duas funções objetivo g e h (conforme ilustrado na Figura 3.6):

$f(g(x), h(x))$, sendo: $g(x) = x^2$ e $h(x) = (x-2)^2$

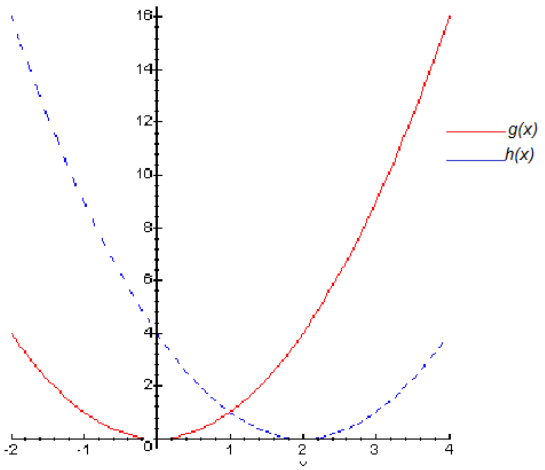


Figura 3.6 – Funções g e h de um problema multiobjetivo (Schaffer, 1984)

De acordo com Figura 3.6, ambas as funções diminuem no intervalo $(-\infty, 0]$ e crescem em $[2, \infty)$. Torna-se relativamente fácil identificar que, no intervalo $[0,2]$ devem estar compreendidas as soluções ótimas de Pareto. No entanto, observa-se também que, nesse intervalo, quando a função h diminui, a função g cresce. Portanto, para identificar o conjunto de soluções ótimas de Pareto, recorre-se usualmente a um gráfico paramétrico das funções (Figura 3.7).

PUC-Rio - Certificação Digital Nº 1121493/CA

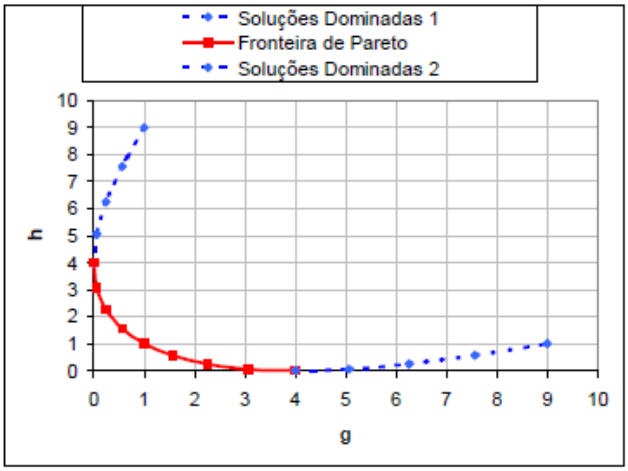


Figura 3.7 – Identificação de uma Fronteira de Pareto (Schaffer, 1984)

O gráfico paramétrico da Figura 3.7 mostra a Fronteira de Pareto, ou o conjunto de soluções ótimas de Pareto. Os pontos vermelhos correspondem às soluções não dominadas devido ao fato de que não existem outros pontos que melhorem um objetivo sem degradar o outro.

3.4
Algoritmos genéticos

Nesta seção são apresentados conceitos relacionados com algoritmos genéticos, tais como a estrutura de um algoritmo genético (AG); os principais operadores genéticos e os parâmetros de controle levados em consideração no processo de evolução do algoritmo. É realizada ainda, uma analogia da terminologia do método de algoritmos genéticos com a teoria da genética para um melhor entendimento do tema. Finalmente, é abordado o método de otimização NSGA-II (*Elitist Non-Dominated Sorting Genetic Algorithm*) (Deb, K. et al., 2002), disponível no programa comercial modeFRONTIER (Mode Frontier, 2011) e adotado neste trabalho durante o processo de otimização da configuração do *riser* RCHA, conforme explicado nas subseções e capítulos subsequentes.

3.4.1
Generalidades

Os algoritmos genéticos foram propostos inicialmente por John Holland, na Universidade de Michigan (Goldberg, 1989). A finalidade de sua pesquisa foi a de explicar rigorosamente o processo adaptativo dos sistemas naturais e o desenvolvimento de sistemas artificiais que conservam os mecanismos importantes

PUC-Rio - Certificação Digital Nº 1121493/CA

dos sistemas naturais. Muitos artigos e teses têm estabelecido a validade da técnica para a otimização de funções. Embora sejam computacionalmente simples, os algoritmos genéticos são capazes de encontrar a solução ótima de problemas complexos para os quais os algoritmos clássicos apresentam limitações. Eles não estão limitados por restrições relacionadas com seu espaço de busca, tais como continuidade ou existência de derivadas.

Goldberg (1989) descreve a natureza dos algoritmos genéticos de seleção por meio da combinação de uma sobrevivência Darwiniana do mais forte com um processo estruturado, porém aleatório, em que as informações são combinadas para formar um procedimento de busca canônica que é capaz de lidar com um amplo espectro de problemas. Os algoritmos genéticos são procedimentos de busca baseados nos mecanismos da genética natural e da seleção natural. Eles combinam o

conceito da sobrevivência artificial dos mais aptos com operadores genéticos abstraídos da natureza para formar um mecanismo de busca robusto.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.4.2
Aplicações dos algoritmos genéticos

Os algoritmos genéticos têm sido mais comumente utilizados para solucionar problemas de otimização. Contudo, como nem todos os problemas podem ser resolvidos por essa técnica, é recomendável considerar as seguintes características do problema antes de utilizá-la (Arranz de la Peña e Truyol, 2007):

- O espaço de busca deve estar delimitado dentro de um determinado intervalo;
- Deve ser possível definir uma função de *fitness* ou *ranking* que indique quão boa ou ruim é uma determinada resposta;
- As soluções deverão ser codificadas de uma maneira que seja relativamente simples de serem implementadas no computador.

Como exemplos de diferentes áreas de aplicação dos algoritmos genéticos é possível citar:

- Desenvolvimento de novos materiais que satisfaçam múltiplos objetivos;
- Otimização do carregamento de contêineres;
- Atribuição de processos em topologias de rede com processamento distribuído;
- Posição de arquivos em sistemas de armazenamento distribuído;
- Projeto de circuitos integrados;

- Otimização da infraestrutura de telefonia celular;
- Engenharia aeroespacial;
- Jogos;
- Robótica.

3.4.3
Diferença entre algoritmos genéticos e clássicos de otimização

Os algoritmos genéticos diferem dos clássicos em muitos fatores, dentre os quais podem se descatar (Rajeev e Krishnamoorthy, 1992):

- Os algoritmos genéticos não exigem conhecimento do problema específico para realizar uma pesquisa. Por exemplo, muitos dos métodos clássicos de otimização precisam de informações de derivadas para realizar uma busca, enquanto que os algoritmos genéticos são indiferentes à essa informação do problema;
- Os algoritmos genéticos trabalham com variáveis de projeto codificadas, que são cadeias de comprimento finito. Essas cadeias representam cromossomos artificiais. Cada caracter na cadeia é um gene artificial. Os algoritmos genéticos processam sucessivas populações desses cromossomos artificiais em sucessivas gerações;
- Os algoritmos genéticos usam simultaneamente uma população de pontos em contraste com a abordagem de um único ponto pelos métodos clássicos de otimização. Isso significa que, em um determinado momento, os algoritmos genéticos podem processar uma série de projetos;
- Os algoritmos genéticos usam operadores aleatórios ao invés dos operadores determinísticos usuais. Os operadores aleatórios melhoram o processo de busca de uma forma de adaptação.

Essas quatro propriedades são de grande importância na aplicação de algoritmos genéticos. Como mencionado anteriormente, eles se baseiam em operadores genéticos, os quais são responsáveis pelas mais diferentes operações, tais como: reprodução, cruzamento, mutação, elitismo, dominância, inversão, duplicação intracromossômica, eliminação, translocação, separação, especiação, migração e compartilhamento (Rajeev e Krishnamoorthy, 1992). Dependendo da natureza do problema e das exigências de desempenho, as soluções fornecidas por meio de algoritmos genéticos podem ser melhoradas combinando-se e aplicando-se alguns desses operadores. A seguir, alguns dos principais operadores são explicados em mais detalhes.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.4.4
Analogia do método de algoritmos genéticos com a teoria da genética

Como os algoritmos genéticos são procedimentos de busca baseados nos mecanismos da genética natural e da seleção natural, é necessário fazer uma analogia entre as terminologias usadas na biologia e no método dos algoritmos genéticos aplicados ao problema de otimização:

- Cromossomo: É uma cadeia de caracteres que armazenam e representam informação relativa às variáveis do problema. Por exemplo, a representação binária de um cromossomo caracterizado pela definição de três variáveis de projeto (x_1, x_2, x_3), cujos valores são $x_1 = 2$, $x_2 = 5$, $x_3 = 3$, é dada por:

$$\{x_1, x_2, x_3\} = \{2,5,3\} = \{1001,0011,0111\}$$

- Gene: É a unidade básica do cromossomo. Cada cromossomo tem um determinado número de genes que descrevem uma certa variável do problema;
- Alelos: Representam os valores atribuídos aos genes;
- Fenótipo: Corresponde à decodificação de um cromossomo;
- População: Conjunto de cromossomos ou soluções;
- Geração: É um número da iteração (passo) que o algoritmo genético executa;
- Operações genéticas: Correspondem às operações que o algoritmo genético realiza sobre os cromossomos (cruzamento, mutação, elitismo etc.);
- Aptidão ou função de *fitness*: Representa uma medida da capacidade de sobrevivência de um cromossomo no processo de evolução e, consequentemente, a probabilidade de se reproduzir. Em um problema de minimização, o indivíduo que tenha o menor valor da função objetivo terá maior aptidão. Isso significa que essa solução é melhor do que outra que tenha menor aptidão. A aptidão pode ser calculada de diversas maneiras, mas sempre em função do valor da função objetivo.

A Figura 3.8 ilustra o esquema da analogia descrita acima.

PUC-Rio - Certificação Digital Nº 1121493/CA

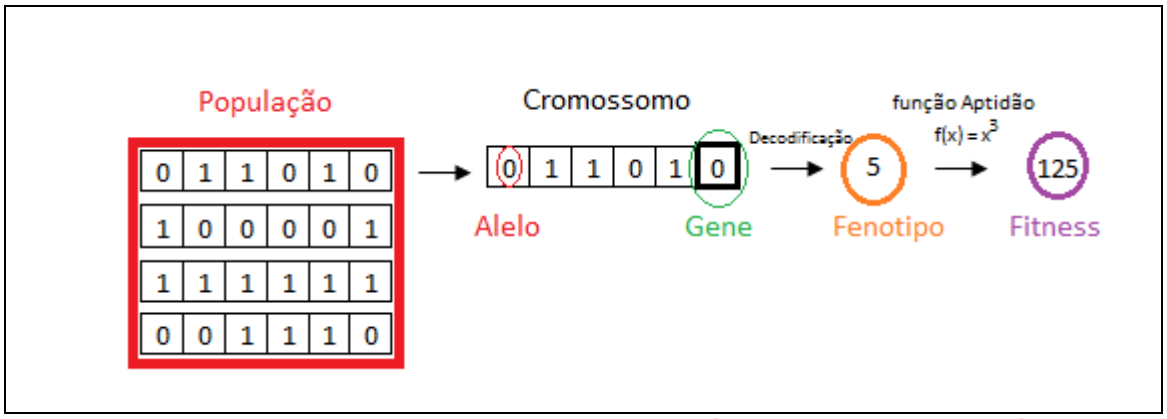


Figura 3.8 – Analogia entre o os algoritmos genéticos e a teoria da genética

3.4.5
Esquema básico de um algoritmo genético simples

Na natureza, os processos de evolução biológica são realizados de forma natural e automática, como por exemplo, a evolução dos seres vivos (nascimento, crescimento, reprodução e morte) mas, para se aplicar o algoritmo genético ao campo da resolução de problemas, deve-se seguir uma série de passos que podem variar de acordo com o problema em questão. Uma premissa básica é definir um tamanho da população que seja suficientemente grande para garantir a diversidade de soluções. Recomenda-se que a população seja gerada de maneira aleatória para se obter essa diversidade. Os passos básicos de um algoritmo genético simples são:

- Avaliar a pontuação de cada um dos cromossomos gerados na população atual;
- Permitir a reprodução dos cromossomos, sendo os mais aptos os que têm maior probabilidade de se reproduzir aplicando-se os operadores genéticos;
- Selecionar os indivíduos que vão sobreviver na geração seguinte;
- Organizar a nova população.

Esses passos são repetidos até que se possa atingir uma determinada condição de convergência, que obedeça a alguns critérios, tais como:

- Não exceder o número máximo de iterações ou número máximo de populações (gerações);
- Quando não houver mais mudanças nos valores da população atual em relação aos da população anterior. Por exemplo, o critério de convergência pode ser uma similaridade percentual (entre 80% e 85%) do valor da função objetivo de um

PUC-Rio - Certificação Digital Nº 1121493/CA

indivíduo da população atual e o valor da função objetivo do mesmo indivíduo na população anterior;

- Quando o valor da média do *fitness* for muito semelhante ao valor da função objetivo de algum indivíduo da população.

A seguir, a Figura 3.9 apresenta a estrutura de um algoritmo genético simples:

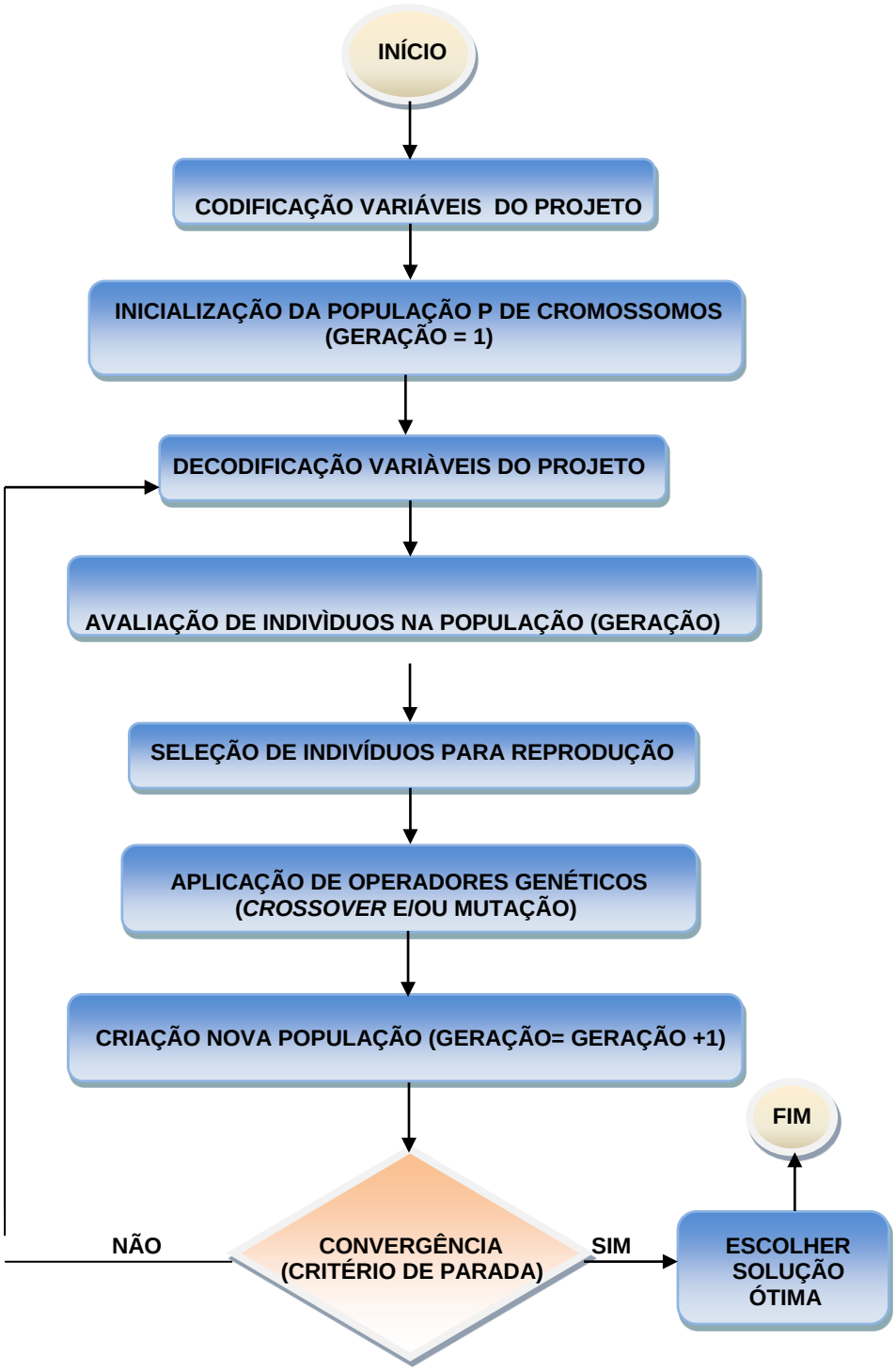


Figura 3.9 – Esquema básico (fluxograma) de um algoritmo genético simples

PUC-Rio - Certificação Digital Nº 1121493/CA

É importante notar que a estrutura de um algoritmo genético pode ser caracterizada de acordo com os seguintes componentes:

- Problema a ser otimizado;
- Forma de inicialização de uma população;
- Tipo de seleção dos indivíduos;
- Codificação e decodificação do problema;
- Tipo de operadores genéticos aplicados;
- Representação das soluções do problema.

Dessa maneira, no desenvolvimento dos algoritmos genéticos, novas estratégias foram desenvolvidas, diferenciando-se uns algoritmos de outros, dependendo do tipo de problema de otimização. Dentre os algoritmos genéticos mais conhecidos na literatura, destacam-se: *VEGA* (*vector evaluated genetic algorithm*) (Schaffer, 1984); *MOGA* (*multi-objective genetic algorithm*) (Fonseca e Fleming, 1993); *NPGA* (*niched Pareto genetic algorithm*) (Horn e Nafpliotis, 1993); NSGA (*non-dominated sorting genetic algorithm*) (Srinivas e Deb, 1994); e *NSGA-II* (*elitist non-dominated sorting genetic algorithm*) (Deb, K. et al., 2002).

A seguir, são mencionados alguns aspectos importantes que devem ser levados em consideração no processo de otimização por meio de algoritmos genéticos.

3.4.6
Parâmetros de controle dos algoritmos genéticos

O desempenho dos algoritmos genéticos pode ser bastante influenciado pelos parâmetros que os caracterizam. A escolha desses parâmetros geralmente é feita com base na experiência do projetista. Os principais parâmetros a serem considerados, são:

Tamanho da população

Indica o número de cromossomos que representa a população de uma determinada geração. Se esse parâmetro for insuficiente, o algoritmo genético tem poucas possibilidades de realizar reproduções e, portanto, as soluções encontradas podem ser de baixa qualidade. Por outro lado, se a população for excessiva, o algoritmo genético será demasiadamente lento. Algumas pesquisas mostram que há um limite para o tamanho da população, acima do qual os algoritmos se tornam ineficientes.

PUC-Rio - Certificação Digital Nº 1121493/CA

Probabilidade de cruzamento

Indica a frequência com que se produzem cruzamentos entre os cromossomos pais, ou seja, a probabilidade de reprodução entre eles. Se não houver probabilidade de reprodução, os filhos serão cópias exatas dos pais. Caso contrário, os filhos terão partes dos cromossomos dos pais. Se a probabilidade de cruzamento é de 100%, os filhos são gerados totalmente por cruzamento e não por partes.

Probabilidade de mutação:

Indica a frequência com que os genes de um cromossomo são alterados. Se não ocorrer mutação, os descendentes serão os mesmos gerados após a reprodução. Caso contrário, parte do cromossomo descendente é modificada e, se a probabilidade de mutação é de 100%, o cromossomo é totalmente modificado.

3.4.7
Codificação de variáveis

Os cromossomos devem conter informações sobre as soluções que representam. Tais informações são codificadas de diferentes maneiras, como por exemplo usando números binários, números inteiros ou cadeia de palavras. A forma mais adotada pelos algoritmos genéticos é aquela que utiliza cadeias de números binários (1 ou 0).

A escolha da codificação depende do problema a ser resolvido e, portanto, merece uma atenção especial no processo de otimização via algoritmos genéticos.

3.4.8
Seleção

A seleção é um processo que permite preservar os indivíduos mais qualificados, de acordo com a teoria de Darwin, para que possam sobreviver e criar uma nova população mais dotada. Cada cromossomo é avaliado e, em função da pontuação obtida, deve-se criar uma nova população, considerando-se as boas características que os melhores indivíduos transmitirão à nova geração. Os tipos de seleção mais conhecidos são: seleção por roleta, por classificação, elistista e hierárquica.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.4.9
Operadores genéticos

Durante o processo de otimização são utilizados operadores genéticos que determinam diretrizes de busca da solução ótima. Assim, são geradas novas gerações, geneticamente melhoradas, com uma melhor aptidão ou *fitness*. Tais operadores são usados em cada geração até que a convergência do problema seja atingida. Dentre os principais operadores, destacam-se:

3.4.9.1
Cruzamento

Depois de realizada a seleção dos cromossomos, é feita a reprodução ou o cruzamento entre dois desses cromossomos. Especificamente, o cruzamento consiste na troca de material genético entre os cromossomos. Portanto, o objetivo principal do cruzamento é fazer com que o descendente obtenha uma melhor aptidão ou *fitness* em relação aos seus pais.

Existem diferentes maneiras de se realizar o cruzamento, como por exemplo: (a) o cruzamento de um único ponto: onde as informações genéticas dos pais são cortadas em um único ponto e copiadas do início até este ponto de um dos pais e deste ponto até o final as informações são copiadas do outro pai, tal como mostrado na Figura 3.10; (b) cruzamento multipontos: este tipo de cruzamento é semelhante ao anterior, porém com a diferença de que os cromossomos pais são cortados em dois ou mais pontos. As informações dos pais são copiadas de forma alternada nesses intervalos criados, conforme ilustrado na Figura 3.11; (c) cruzamento uniforme: nesse caso, os genes de cada descendente são obtidos de qualquer um dos pais de forma aleatória, por meio da geração de um número randômico. Se esse número superar um determinado limite, um dos pais é escolhido para ter a informação genética copiada. Caso contrário, o outro pai será escolhido. A Figura 3.12 ilustra este tipo de cruzamento.

PUC-Rio - Certificação Digital Nº 1121493/CA

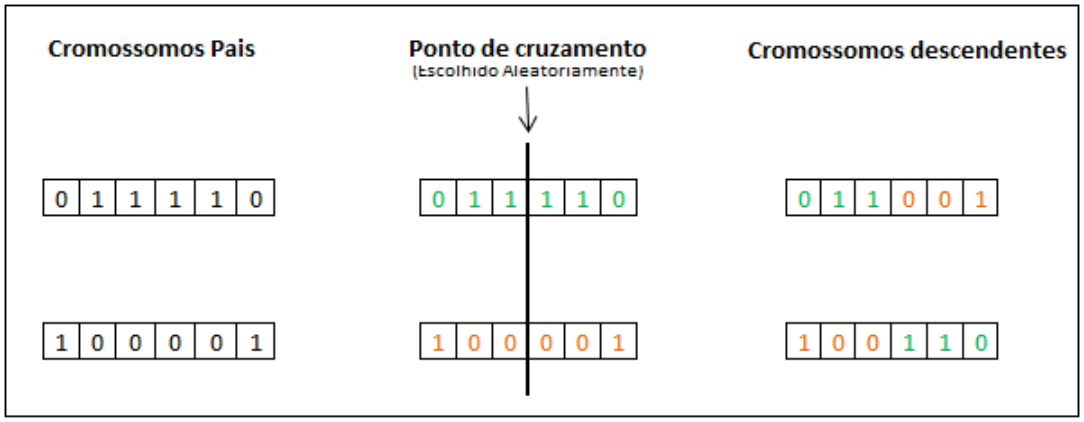


Figura 3.10 – Exemplo de cruzamento de um único ponto

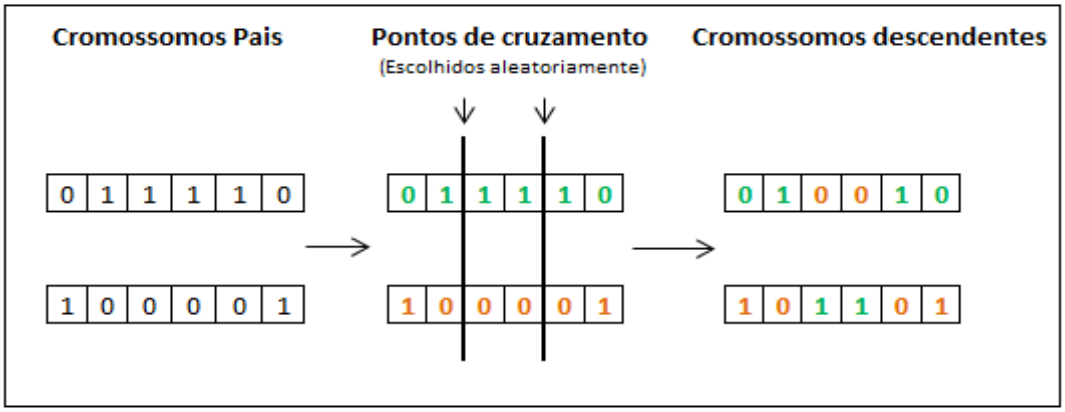


Figura 3.11 – Exemplo de cruzamento de dois pontos

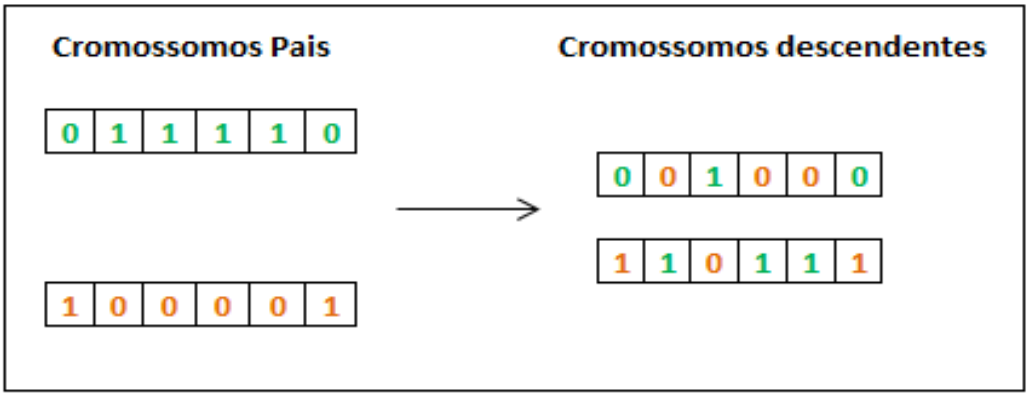


Figura 3.12 – Exemplo de cruzamento uniforme

3.4.9.2

Mutação

Este operador é responsável pela manutenção da diversidade genética, possibilitando uma maior varredura do espaço de busca e evitando que o algoritmo

PUC-Rio - Certificação Digital Nº 1121493/CA

genético convirja prematuramente para mínimos locais. Trata-se de uma alteração no valor de algum alelo (que são as formas alternativas de um mesmo gene) de uma cadeia sorteada com uma determinada probabilidade, conhecida como probabilidade de mutação. A Figura 3.13. apresenta um exemplo esquemático do operador mutação para uma cadeia em codificação binária:

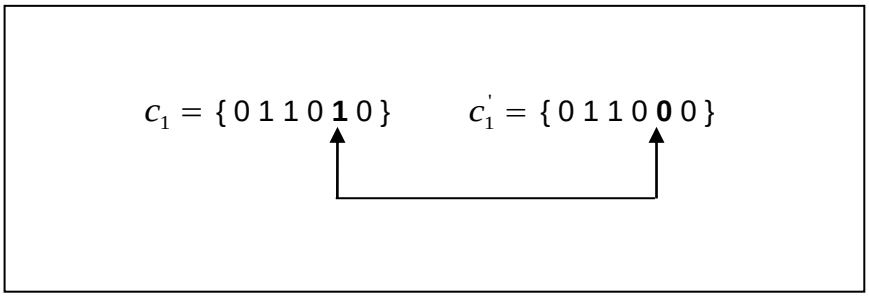


Figura 3.13 – Exemplo de mutação de um cromossomo:
$c_1 =$ antes da mutação; $c_1' =$ depois da mutação

PUC-Rio - Certificação Digital Nº 1121493/CA

3.4.9.3

Elitismo

A finalidade do elitismo é preservar as melhores soluções da população atual e incorporá-las, sem modificações, à população posterior. Assim, evita-se que as soluções de bom desempenho da população atual sejam perdidas no processo de evolução.

3.4.10

Vantagens e desvantagens dos algoritmos genéticos na otimização

Segundo Arranz de la Peña e Truyol (2007), as principais vantagens dos algoritmos genéticos são:

(i) operam de forma simultânea com várias soluções, enquanto que as técnicas tradicionais exploram o espaço de soluções na busca de uma única solução em cada passo; (ii) são capazes de “fugir” dos chamados mínimos locais na busca de uma melhor solução, enquanto que os métodos clássicos normalmente ficam “presos” nesses pontos sub-ótimos; (iii) possuem a capacidade de manipular vários parâmetros simultaneamente, o que os tornam interessantes quando se trabalha com problemas com vários objetivos; (iv) não necessitam de conhecimentos específicos sobre o problema que está sendo estudado: eles podem realizar alterações aleatórias em suas soluções candidatas utilizando a função *fitness* para determinar se essas alterações

produzem uma melhoria na solução; (v) usam operadores probabilísticos, em contrapartida aos típicos operadores determinísticos de outras técnicas.

As principais desvantagens dos algoritmos genéticos, são:
(i) elevado custo computacional para obtenção da convergência numérica, podendo não convergir em muitos casos, a depender dos parâmetros escolhidos (como: tamanho da população, número de gerações, probabilidades de cruzamento e de mutação, dentre outros); (ii) podem convergir prematuramente devido a uma série de fatores, como por exemplo: se um indivíduo que é mais apto do que a maioria de seus competidores emerge rapidamente no curso da execução, ele pode se reproduzir abundantemente diminuindo assim a diversidade da população e fazendo com que o algoritmo convirja para soluções sub-ótimas (de baixa qualidade) e impedindo que o algoritmo possa encontrar a solução global do problema. Este é um problema comum quando as populações são pequenas.

PUC-Rio - Certificação Digital Nº 1121493/CA

3.4.11
Escolha do algoritmo genético NSGA- II no presente trabalho

Com base em aplicações bem-sucedidas de problemas multiobjetivos encontradas na literatura técnica (Andrade. et al., 2010; Martins, 2011; Oliveira, 2008), o algoritmo genético *NSGA II* (Deb, K. et al., 2002) foi escolhido neste trabalho para o processo de otimização de configurações RCHA. É importante salientar que, embora existam diferentes algoritmos genéticos conhecidos na literatura, o *NGSA II* implementa uma metodologia em que se aplicam simultaneamente conceitos sobre dominância e diversidade (ver seção 3.3 sobre otimização multiobjetivo), o que faz deste algoritmo uma ferramenta bastante poderosa e eficiente na otimização de problemas multiobjetivos.

O *NSGA-II* é uma variação do algoritmo *NSGA*, que busca aumentar a eficiência por meio da incorporação do conceito de elitismo, proposto por Deb, K. et al. (2002). Mais detalhes sobre esses algoritmos são apresentados na seção seguinte.

3.4.11.1
NSGA (*non-dominated sorting genetic algorithm*)

Esse método foi proposto por Srinivas e Deb (1994) e aplica simultaneamente critérios de dominância na avaliação e classificação de soluções, assim como o conceito de diversidade da população. A estratégia utilizada no *NSGA* para a seleção dos indivíduos é baseada em um ordenamento para privilegiar as soluções correntes

não dominadas. Assim, antes de realizar a seleção, os indivíduos são classificados com base na avaliação de seu desempenho em termos da função objetivo. Os indivíduos são ordenados considerando-se o nível de não dominância entre eles, formando fronteiras (Figura 3.14). Dessa maneira, as soluções não dominadas da população corrente recebem valores elevados de *fitness*, sendo os mesmos valores para todos os indivíduos não dominados, ou seja, assegurando-se o mesmo potencial reprodutivo.

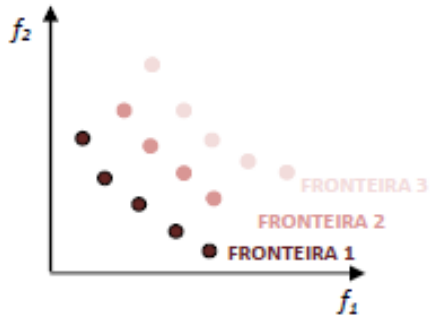


Figura 3.14 – Representação da classificação dos indivíduos no *NSGA* (Martins, 2011)

Com esta metodologia, os indivíduos da primeira fronteira terão maiores valores de *fitness* e, assim, maior possibilidade de reprodução, o que prioriza a busca de indivíduos não dominados.

Para garantir a diversidade da população, o conceito de nichos (Mahfoud, 1995) foi introduzido. Os nichos são grupos de indivíduos com caraterísticas comuns localizados em uma vizinhança próxima e com valores similares de *fitness.* A estratégia consiste em atribuir a cada indivíduo pertencente ao mesmo nicho um valor denominado *fitness* compartilhado. Dessa maneira, as soluções de um nicho menos ocupado terão um *fitness* compartilhado mais alto, permitindo que as soluções menos representadas tenham um lugar de privilégio na sua classificação e, assim, possam ser selecionadas com maior probabilidade. Na realidade, esse valor denominado *fitness* compartilhado é o resultado de um ajuste do valor de *fitness* calculado inicialmente.

3.4.11.2

NSGA-II (*elitist non-dominated sorting genetic algorithm*)

Para realizar a classificação dos indivíduos de uma determinada população, utiliza-se o conceito de soluções dominadas e não dominadas.

PUC-Rio - Certificação Digital Nº 1121493/CA

Outro conceito importante é a comparação da aglomeração (*crowding*) que uma técnica utiliza para garantir a diversidade das soluções e evitar a existência de nichos localizados. O *crowding* é um operador de comparação de aglomeração que dá prioridade aos indivíduos menos aglomerados. Os algoritmos genéticos buscam encontrar a maior quantidade de soluções pertencentes à fronteira de Pareto, porém, com este operador, o algoritmo prioriza soluções menos aglomeradas, garantindo a diversidade da população. Dessa forma, as soluções que tiverem uma distância menor de aglomeração serão eliminadas.

A seguir, apresenta-se um sumário do método *NGSA II*:

1. Geração da população inicial de tamanho *n*;
2. Identificação das frentes de dominância e as distâncias de aglomeração;
3. Aplicação dos operadores genéticos de cruzamento e mutação para a geração da população de descendentes de tamanho *n*;
4. A população inicial (pais) e a população de descendentes (filhos) são reunidas em um mesmo conjunto de tamanho 2*n*;
5. Os indivíduos desse conjunto são classificados de acordo com as frentes de dominância, selecionando-se aqueles que têm maior grau de dominância;
6. A nova população é criada. Caso os indivíduos selecionados superem o tamanho *n* da população, são eliminados aqueles que tenham a menor distância de aglomeração;
7. O critério de convergência é avaliado. Caso seja atingido, o processo é finalizado; caso contrário, deve-se retornar ao passo um e continuar até que o critério de convergência seja atingido.

A seguir, a Figura 3.15, apresenta, esquematicamente, o procedimento acima descrito:

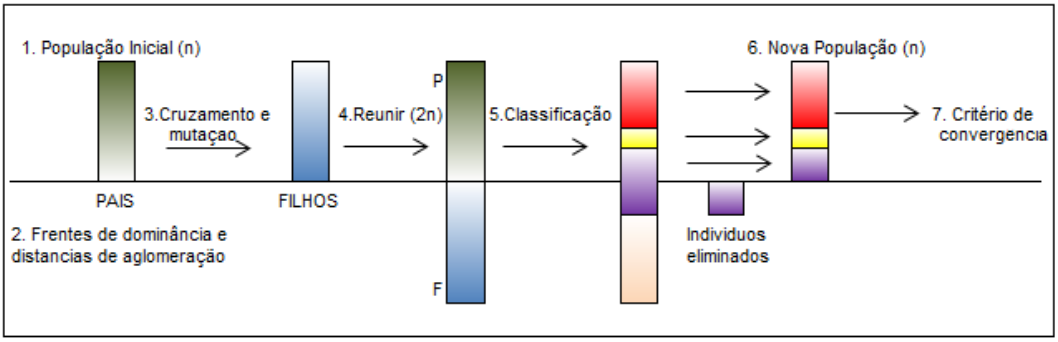


Figura 3.15 – Esquema do *NGSA II* (Martins, 2011)

PUC-Rio - Certificação Digital Nº 1121493/CA