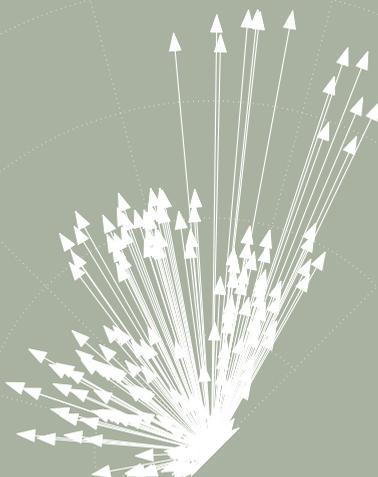


MANUAL DE  
**COMPUTAÇÃO  
EVOLUTIVA  
E META  
HEURÍSTICA**

ANTÓNIO GASPAR-CUNHA  
RICARDO TAKAHASHI  
CARLOS HENGGELER ANTUNES  
COORDENADORES



IMPRESA DA  
UNIVERSIDADE  
DE COIMBRA

COIMBRA  
UNIVERSITY  
PRESS

( EDITORAufmg )

## CAPÍTULO 17

# Algoritmos Evolutivos Multi-Objectivo

*Ricardo H. C. Takahashi* \*    *António Gaspar-Cunha* \*\*    *Carlos M. Fonseca* \*\*\*

*\*Departamento de Matemática  
Universidade Federal de Minas Gerais*

*\*\*Instituto de Polímeros e Compósitos / I3N  
Universidade do Minho*

*\*\*\*Departamento de Engenharia Informática  
Universidade de Coimbra*

Os capítulos anteriores deste livro trataram de diferentes técnicas para resolver problemas de otimização formulados como problemas de escolha dos valores de um conjunto de variáveis de decisão satisfazendo um conjunto de restrições, de forma a minimizar (ou maximizar) uma determinada função (a função-objetivo). Na verdade, grande parte da teoria de otimização, desenvolvida ao longo do século XX, diz respeito a problemas formulados dessa maneira.

No entanto, muitos dos problemas encontrados no mundo prático seriam mais adequadamente formulados em termos de múltiplos objetivos. Por exemplo, em situações de projeto de sistemas de engenharia, normalmente é importante minimizar o custo de construção do sistema, ao mesmo tempo em que também se faz necessário otimizar índices de desempenho diversos. Para o fim de se estabelecer um exemplo concreto, imagine-se uma situação de projeto de um motor elétrico. Deseja-se: minimizar o custo de produção do motor; minimizar o peso final do motor; maximizar a eficiência energética do

motor; maximizar o conjugado eletromecânico fornecido; minimizar a interferência eletromagnética induzida pelo motor em outros equipamentos; maximizar o valor da sobretensão que o motor será capaz de suportar, de curta e de longa duração; etc. É importante notar que o resultado final do projeto, que será o motor efetivamente construído, deverá demonstrar um bom comportamento no que diz respeito a todos esses objetivos, de forma que o procedimento de projeto não poderá se resumir a uma otimização considerando um dos objetivos apenas. Além disso, deve-se notar que em alguma medida, os diferentes objetivos serão tipicamente contraditórios entre si. No exemplo do motor, pode-se perceber que, dentre as diferentes alternativas de projeto final do motor, existirão aquelas que terão menor custo de construção e menor eficiência energética, e outras de maior custo e mais eficientes. Tipicamente, não existirá um único motor que seja o de menor custo construtivo possível e simultaneamente também o de maior eficiência possível. Então, problemas com múltiplos objetivos tipicamente colocarão para o projetista um problema de tomada de decisão que envolve a análise das trocas (ou *trade-offs*) entre objetivos.

O problema de projeto de um motor elétrico, conforme enunciado no parágrafo anterior, se enquadra na classe dos chamados problemas de *otimização multiobjetivo*. Será visto em maior detalhe, nas próximas seções deste capítulo, que a formulação de tais problemas geralmente conduz a que todo um conjunto de soluções, chamadas *soluções eficientes*<sup>1</sup>, que irão satisfazer à condição de otimalidade estabelecida em termos das várias funções-objetivo do problema. De maneira intuitiva, pode-se imaginar que essas seriam aquelas soluções que atingiram o limite de desempenho imposto pela física do problema, no sentido em que, por exemplo, para baixar o custo de produção do motor ainda mais, em relação a uma solução que já seja ótima, seja preciso sacrificar por exemplo um pouco da eficiência energética do motor. Devido à multiplicidade das soluções de problemas formulados dessa maneira, a solução completa de um problema de projeto (que necessariamente deverá indicar um único motor a ser fabricado, no caso do exemplo em questão) usualmente irá envolver duas etapas: a etapa de determinação das soluções eficientes, e a etapa de tomada de decisão, para seleção de uma única solução a ser efetivamente implementada.

Este capítulo irá tratar especificamente da etapa de geração das soluções eficientes. O capítulo 16 irá abordar a questão da tomada de decisão. É importante mencionar, neste ponto, que há três abordagens gerais para a integração da tomada de decisão em problemas de otimização multiobjetivo, que seriam com a decisão sendo tomada *a priori*, ou *progressivamente*, ou ainda *a posteriori*. Na primeira, *a priori*, os critérios a serem utilizados na tomada de decisão seriam explicitados antes da geração de qualquer solução eficiente. Esses critérios já poderiam, portanto, ser considerados no processo de otimização, de forma a conduzir diretamente a uma única solução que, além de ser eficiente, será também a preferida. Na segunda, *progressiva*, um tomador de decisão irá tomar conhecimento das possibilidades de solução existentes para o problema à medida em que estas forem sendo geradas, no decorrer da execução do algoritmo de otimização. Esse tomador de decisão irá então interagir com o algoritmo de otimização, guiando-o para a determinação das soluções eficientes que correspondam à sua preferência, evitando assim a necessidade da geração de todo o conjunto de soluções eficientes. Por fim, a terceira abordagem, *a posteriori*, irá supor que todo um conjunto de amostras do conjunto de soluções eficientes do problema de otimização multiobjetivo deverá ser gerado, para ser apresentado a um tomador de decisão. Esse tomador de decisão, após examinar as diferentes soluções contidas nesse conjunto, irá formar sua opinião a respeito de qual solução deverá ser preferida.

Esta última situação, da tomada de decisão *a posteriori*, foi assumida como contexto-padrão pela grande maioria dos estudos envolvendo a otimização multiobjetivo dentro da abordagem da computação evolutiva<sup>2</sup>. Este capítulo irá apresentar, especificamente, a abordagem da computação evolu-

<sup>1</sup> Na literatura são encontrados os seguintes sinônimos para *soluções eficientes*: soluções Pareto-ótimas, soluções não-dominadas, soluções não-inferiores.

<sup>2</sup> Recentemente, têm aparecido algumas exceções a esse padrão; ver, por exemplo, as referências (Battiti e Passerini, 2010; Deb et al., 2010; Karahan e Koksalan, 2010; Kim et al., 2011).

tiva para tratar os problemas de otimização multiobjetivo, de forma que será pressuposto aqui que o mecanismo de otimização deva cumprir o papel de gerar uma amostragem representativa do conjunto de soluções eficientes, antes de se iniciar qualquer interação com um tomador de decisão.

Deve-se mencionar que, antes do aparecimento da abordagem evolutiva, já existiam outras abordagens para o problema de otimização em contextos envolvendo múltiplos objetivos. Os métodos tradicionais de otimização, construídos para a minimização de um único objetivo, lidam com os problemas em que existem múltiplos objetivos relevantes de duas formas diferentes. Essas formas são: (i) constrói-se uma única função objetivo como uma soma ponderada das diversas funções de interesse; ou (ii) tratam-se todos os objetivos, menos um, como restrições, ficando apenas um objetivo formulado como função-objetivo a ser minimizada. Dessa forma, é possível que uma otimização de uma única função-objetivo conduza a uma solução que seja adequada no que diz respeito a vários objetivos diferentes. Há entretanto diversas possíveis objeções a essas abordagens, relacionadas com dificuldades de aplicação prática e com a perda de soluções possivelmente interessantes. Mas talvez a objeção mais importante a ser feita seja a de que os problemas com múltiplos objetivos normalmente envolvem escolhas dentre soluções que representam diferentes compromissos entre os diferentes objetivos. Portanto, para que a escolha seja feita adequadamente, é importante que o projetista tenha conhecimento das alternativas disponíveis, de forma a poder compará-las – o que significa que métodos que forneçam como saída uma única solução estarão omitindo informação essencial a respeito do problema.

Devido a tais considerações, foi desenvolvida uma teoria para fundamentar a otimização multiobjetivo, inicialmente no âmbito dos métodos determinísticos de otimização, que visava construir métodos capazes de gerar conjuntos de soluções pertencentes a aquele *conjunto de soluções eficientes*. A lógica básica desses métodos consistia em reformular o problema de otimização multiobjetivo, de forma tal a gerar um problema de otimização mono-objetivo (chamado *problema escalarizado*) que gerasse como solução um vetor de variáveis de decisão que garantidamente pertencesse ao conjunto de soluções eficientes. Esse problema seria parametrizado de tal forma que, variando-se os parâmetros do problema escalarizado, seria possível em princípio gerar todas as soluções pertencentes ao conjunto de soluções eficientes. Essa abordagem, por construção, iria requerer a execução completa de um algoritmo de otimização mono-objetivo para gerar cada uma das soluções pertencentes ao conjunto de soluções eficientes do problema multi-objetivo. As referências (Chankong e Haimes, 1983) e (Ehrgott, 2000) descrevem em detalhe essa abordagem.

Com o surgimento dos Algoritmos Evolutivos, capazes de fazer evoluir simultaneamente todo um conjunto de diferentes tentativas de solução – as *populações* – foi possível elaborar técnicas para a obtenção, em uma única execução do algoritmo, de todo um conjunto de soluções que representam diferentes compromissos entre os objetivos do problema. Essas técnicas, denominadas de MOEA, do inglês *Multi-Objective Evolutionary Algorithms*, ou de EMO, do inglês *Evolutionary Multicriteria Optimization*, vieram a ocupar uma posição de destaque dentre as metodologias capazes de tratar problemas de otimização multiobjetivo, constituindo hoje o elenco de ferramentas mais utilizado em problemas práticos dessa natureza (Deb, 2001; Coello Coello et al., 2002).

Diversas das razões que contribuíram para a popularidade dessa abordagem se aplicam tanto ao caso multiobjetivo quanto ao caso em que há uma única função-objetivo:

1. Diversas técnicas de otimização multiobjetivo clássicas requerem que os problemas sejam enquadrados em formulações específicas. Esse é o caso, por exemplo, de formulações que requerem funções-objetivo e restrições lineares com variáveis inteiras, ou funções-objetivo e restrições convexas, ou ainda funções-objetivo convexas com restrições lineares, ou outras. Os MOEA não apresentam esse tipo de limitação, apresentando grande flexibilidade no que diz respeito à estrutura das funções-objetivo e restrições do problema.
2. Os AE's são capazes de tratar problemas que envolvem funções multimodais, com múltiplas bacias de atração de diferentes mínimos locais. A maior parte dos algoritmos de otimização

tradicionais não contêm mecanismos para lidar adequadamente com esse tipo de função.

3. Como outras heurísticas, as técnicas de computação evolutiva admitem que se fixe o tempo de execução, de forma a retornar as melhores soluções encontradas até então. Embora não haja garantia de otimalidade exata dessas soluções, isso torna essa classe de métodos aplicável em problemas nos quais há a limitação do tempo disponível para a entrega da solução – o que inviabilizaria o uso de diversas técnicas existentes.
4. Os AEs são relativamente simples em termos computacionais e fáceis de implementar.
5. É fácil combinar (hibridizar) EAs com outras técnicas de otimização, tais como procura local e outras metaheurísticas;
6. Os AEs são facilmente adaptáveis devido ao facto do mecanismo evolutivo ser separado da representação do problema. Assim, um mesmo AE poderá ser usados em diferentes tipos de problemas, sendo normalmente necessário apenas o tratamento específico da codificação das variáveis do problema.

Uma vantagem adicional dos AE's multiobjetivo, que não se verifica no caso de uma única função objetivo, reside nos ganhos computacionais associados à evolução simultânea de um grande número de soluções, que irão gerar todo um conjunto de amostras do conjunto de soluções eficientes do problema, com uma única execução do algoritmo. A referência (Giel e Lehre, 2010) mostra formalmente essa vantagem, em termos da redução da complexidade computacional do problema.

A Otimização Multi-Objectivo (MOO) constitui hoje um tema de investigação importante para cientistas e engenheiros, não só devido à natureza multi-objectivo da maioria dos problemas do mundo real, mas também devido ao facto de existirem ainda muitos pontos de investigação em aberto nesta área.

Este capítulo está organizado nos seguintes tópicos:

- A seção 1 formaliza o problema da otimização multiobjetivo, estudando sua estrutura matemática e identificando as funções que devem ser executadas por algoritmos de otimização multiobjetivo.
- A seção 2 apresenta os elementos construtivos básicos dos algoritmos evolutivos multiobjetivo, mostrando que alguns desses elementos são idênticos a elementos análogos presentes nos algoritmos evolutivos mono-objectivo, enquanto outros são específicos para o caso multiobjetivo. Essa seção se encerra com a apresentação dos pseudo-códigos de dois AE's multiobjetivo clássicos, o NSGA-II (Non-Dominated Sorting Genetic Algorithm II) e o SPEA2 (Strength Pareto Evolutionary Algorithm 2).
- A seção 3 discute algumas funções que vêm sendo incorporadas a versões recentes de algoritmos evolutivos multiobjetivo, procurando também indicar algumas questões que são tema de pesquisa corrente na área.

## 1. Otimização Multiobjetivo

No capítulo 1, os problemas de otimização foram definidos como:

$$\mathbf{x}^* = \arg \min \mathbf{f}(\mathbf{x})$$

$$\text{sujeito a: } \begin{cases} g_1(\mathbf{x}) \leq 0 \\ g_2(\mathbf{x}) \leq 0 \\ \vdots \\ g_m(\mathbf{x}) \leq 0 \end{cases} \quad (17.1)$$

Essa formulação ainda se aplica aos *problemas de otimização multiobjetivo*. O problema é formulado, nesse caso, como a questão de determinar o vetor  $\mathbf{x}^*$  que minimiza  $\mathbf{f}(\mathbf{x})$ . Novamente, as funções de restrição  $g_1(\mathbf{x})$  a  $g_m(\mathbf{x})$  serão definidas como funções que são satisfeitas quando são menores ou iguais a zero. A diferença em relação ao caso de um único objetivo é que agora  $\mathbf{f}(\mathbf{x})$  passa a ser definido como um vetor de funções-objetivo:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_p(\mathbf{x}) \end{bmatrix}$$

Imediatamente se coloca a questão: o que significa a *minimização* nesta nova situação? Um ponto central da teoria de otimização multiobjetivo é a definição de mínimo a ser adotada.

Cabe em primeiro lugar examinar por quê essa dificuldade não ocorria (ou, pelo menos, não ficava aparente) quando o problema tinha apenas uma função objetivo. Naquele caso, a função  $f(\mathbf{x})$  retornava um único número real, ou seja, um escalar. Um conjunto de  $k$  números escalares  $a_i$  sempre pode ser *ordenado* usando a relação  $\leq$ , ou seja, sempre é possível escrever:  $a_1 \leq a_2 \leq \dots \leq a_k$ . O *mínimo* de um conjunto de números pode ser definido como o elemento que aparece no extremo esquerdo da ordenação  $a_1 \leq a_2 \leq \dots \leq a_k$ , ou seja, como o elemento que é menor ou igual que todos os demais. Note-se que pode haver mais de um elemento nessa situação, se o menor de todos ocorrer mais de uma vez. Assim, por exemplo, o conjunto  $\{3, 7, 5, 3, 4, 2, 9, 2\}$  pode ser ordenado da seguinte forma:  $2 \leq 2 \leq 3 \leq 3 \leq 4 \leq 5 \leq 7 \leq 9$ . O elemento 2, que aparece no extremo esquerdo da ordenação, é chamado de *mínimo* do conjunto. O operador relacional de *menor ou igual* ( $\leq$ ), que fez com que esse conjunto de números fosse ordenado dessa maneira, é chamado de *ordem total*, porque foi capaz de estabelecer uma única ordenação encadeando todos os elementos do conjunto.

A relação de menor ou igual, em essência, é o fundamento para definir o que é um mínimo. Essa relação permite comparar dois elementos, identificando-se o melhor dos dois como aquele que é menor, dentre os dois. Quando essa relação pode ser estabelecida entre todos os pares de elementos de um conjunto, numa ordenação total, passa a existir um elemento que é menor ou igual que todos os demais e, portanto, o melhor de todos do conjunto. Problemas de otimização visam identificar exatamente esse elemento “melhor de todos”, definido dessa forma.

A relação de menor ou igual, quando utilizada para comparar vetores em uma comparação componente a componente, já não é mais capaz de gerar uma ordem total em geral. No caso de conjuntos de vetores, alguns pares de vetores serão relacionados dessa forma, ou seja, em alguns casos, cada componente  $\mathbf{x}_i^1$  de um vetor  $\mathbf{x}^1$  será menor ou igual a cada componente  $\mathbf{x}_i^2$  de um vetor  $\mathbf{x}^2$ . Em outros casos, algumas componentes de  $\mathbf{x}^1$  serão menores, e outras maiores, que as correspondentes componentes de  $\mathbf{x}^2$ . Essa é a origem da dificuldade para se definir o que é um mínimo no caso de vetores.

Para generalizar o conceito de mínimo, de forma a estabelecer um conceito aplicável a vetores, recorre-se à noção de *ordem parcial*. Define-se um operador relacional de *precedência*,  $\prec^P$ , que deverá

possuir as seguintes propriedades:

- (i)  $a \stackrel{p}{\leftarrow} a$  (reflexividade)
- (ii) se  $a \stackrel{p}{\leftarrow} b$  e  $b \stackrel{p}{\leftarrow} a$  então  $a = b$  (anti-simetria)
- (iii) se  $a \stackrel{p}{\leftarrow} b$  e  $b \stackrel{p}{\leftarrow} c$  então  $a \stackrel{p}{\leftarrow} c$  (transitividade)

Uma relação ( $\stackrel{p}{\leftarrow}$ ) que tenha tais propriedades quando aplicada aos elementos de um conjunto é chamada de *ordem parcial*, porque ela faz com que seja possível definir uma precedência entre os elementos desse conjunto. No caso de conjuntos cujos elementos são vetores de números reais, é possível definir uma relação de precedência específica, a relação  $\preceq$ , que será útil para o propósito de reestabelecer uma noção de minimização significativa para o caso de conjuntos de vetores. Sejam os vetores  $\mathbf{a}$  e  $\mathbf{b}$ :

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix}$$

A relação  $\preceq$  entre esses vetores é definida como:

$$\mathbf{a} \preceq \mathbf{b} \quad \text{se:} \quad a_i \leq b_i \text{ para todo } i = 1, \dots, p \quad (17.2)$$

O operador  $\preceq$  definido dessa forma claramente atende às propriedades (i), (ii) e (iii), e portanto é uma *ordem parcial*. Vetores de números reais então podem ser ordenados utilizando a relação  $\preceq$ , de forma que isso se aplica aos vetores de valores das funções-objetivo de problemas de otimização multiobjetivo. A grande diferença da nova situação, quando comparada à otimização de uma única função-objetivo, é que agora a relação de precedência pode não se aplicar a todos os pares de elementos de um conjunto de vetores.

Para exemplificar esse ponto, considere-se o seguinte conjunto:

$$\left\{ \begin{bmatrix} 3 \\ 7 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 9 \\ 8 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix} \right\} \quad (17.3)$$

Pode-se verificar que as seguintes relações são válidas:

$$\begin{bmatrix} 2 \\ 6 \end{bmatrix} \preceq \begin{bmatrix} 3 \\ 7 \end{bmatrix} \preceq \begin{bmatrix} 9 \\ 8 \end{bmatrix} \quad (17.4)$$

Também é válida a relação:

$$\begin{bmatrix} 4 \\ 5 \end{bmatrix} \preceq \begin{bmatrix} 4 \\ 5 \end{bmatrix} \preceq \begin{bmatrix} 9 \\ 8 \end{bmatrix} \quad (17.5)$$

No entanto, neste caso não é possível estabelecer relações utilizando o operador  $\preceq$  entre um elemento  $\begin{bmatrix} 4 \\ 5 \end{bmatrix}$  e o elemento  $\begin{bmatrix} 2 \\ 6 \end{bmatrix}$ , nem entre um elemento  $\begin{bmatrix} 4 \\ 5 \end{bmatrix}$  e o elemento  $\begin{bmatrix} 3 \\ 7 \end{bmatrix}$ . Nesses casos, a primeira componente de um dos vetores é maior que a primeira do outro, enquanto o contrário ocorre na segunda componente. Por esse motivo, o operador  $\preceq$  aplicado a conjuntos de vetores é apenas uma *ordem parcial*, e não uma *ordem total*. A figura 17.1 ilustra o significado da relação  $\preceq$ .

No caso de conjuntos cujos elementos são números escalares, pode-se verificar que o operador  $\leq$ , atende às propriedades (i), (ii) e (iii) e, além disso, para quaisquer dois escalares  $a_i$  e  $a_j$  sempre ocorrerá ou  $a_i \leq a_j$  ou então  $a_j \leq a_i$ . Como vimos, por este motivo a relação  $\leq$  é chamada de *ordem total* para conjuntos de números reais. Um conjunto ordenado com uma ordem total sempre terá um elemento que se encontra à esquerda de todos os demais do conjunto, quando este é ordenado.

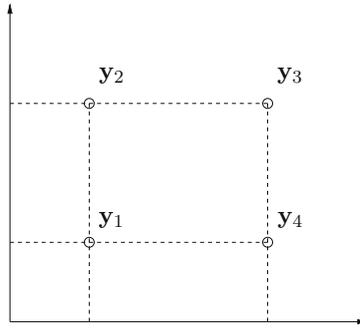


Figura 17.1: Os pontos  $y_1, y_2, y_3$  e  $y_4$  são tais que  $y_1 \preceq y_2, y_1 \preceq y_3$  e  $y_1 \preceq y_4$ , ou seja,  $y_1$  precede todos os demais. Além disso,  $y_2 \preceq y_3$  e  $y_4 \preceq y_3$ , ou seja,  $y_3$  é precedido por todos os demais. Por fim, observa-se ainda que  $y_2 \not\preceq y_4$ , e  $y_4 \not\preceq y_2$ , ou seja, não há relação de precedência entre  $y_2$  e  $y_4$ .

Esse elemento é chamado de *o menor elemento* do conjunto, ou de *mínimo* do conjunto. Já uma ordem parcial pode ou não levar a uma ordenação em que haja um único elemento do conjunto à esquerda de todos os demais. No caso do conjunto mostrado na equação (17.3), que é ordenado em duas sequências diferentes, cada sequência conta com um elemento mínimo diferente,  $\begin{bmatrix} 4 \\ 5 \end{bmatrix}$  para uma sequência e  $\begin{bmatrix} 2 \\ 6 \end{bmatrix}$  para a outra sequência. Se nesse conjunto fosse introduzido por exemplo o elemento  $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ , este se tornaria o elemento mínimo das duas sequências, ou *o menor elemento* do conjunto, pois são válidas as relações  $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \preceq \begin{bmatrix} 4 \\ 5 \end{bmatrix}$  e  $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \preceq \begin{bmatrix} 2 \\ 6 \end{bmatrix}$ , mesmo não havendo uma ordenação total do conjunto.

Como nem sempre existirá um elemento que seja *o menor* de um conjunto cuja ordenação é parcial, como no caso do conjunto em (17.3), o resultado de uma minimização nesse conjunto deve fazer referência a outro tipo de entidade. Considere-se o conjunto representado na equação (17.3). A ordem parcial  $\preceq$  define duas sequências de ordenações diferentes que encadeiam os elementos desse conjunto, indicadas nas equações (17.4) e (17.5). Os elementos que são mínimos dessas duas sequências, no caso os elementos  $\mathbf{a} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$  e  $\mathbf{b} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$ , não são comparáveis entre si, mas compartilham a característica de não terem nenhum outro elemento diferente deles que os preceda (note-se que o elemento  $\mathbf{b}$  é precedido por outro elemento igual a ele). Ou seja, não existe no conjunto nenhum elemento  $\mathbf{c}$  diferente de  $\mathbf{a}$  para o qual seja válida a relação  $\mathbf{c} \preceq \mathbf{a}$ , nem nenhum elemento  $\mathbf{c}$  diferente de  $\mathbf{b}$  que satisfaça a relação  $\mathbf{c} \preceq \mathbf{b}$ . Esses elementos, que são mínimos de alguma sequência definida pela ordem parcial no conjunto, são chamados de *elementos minimais* do conjunto no que diz respeito a tal ordem parcial. Todo conjunto ordenado segundo uma ordem parcial sempre terá pelo menos um elemento minimal, e em geral poderá possuir vários elementos minimais. Na figura 17.1, o elemento  $y_1$  é o elemento minimal, considerando o conjunto constituído pelos elementos  $y_1, y_2, y_3$  e  $y_4$ .

Neste ponto estamos em posição de definir o que seriam as soluções do problema de otimização multiobjetivo (17.1). Considerem-se duas soluções factíveis  $\mathbf{x}_a$  e  $\mathbf{x}_b$  para o problema (ou seja, soluções que satisfazem as restrições). Se se verificar a relação  $\mathbf{f}(\mathbf{x}_a) \preceq \mathbf{f}(\mathbf{x}_b)$  para  $\mathbf{f}(\mathbf{x}_a) \neq \mathbf{f}(\mathbf{x}_b)$ , a solução  $\mathbf{x}_a$  será certamente melhor que a solução  $\mathbf{x}_b$ , pois os valores de todas as funções objetivo de  $\mathbf{x}_a$  serão ou melhores ou no mínimo iguais em relação à outra solução. Então, as soluções minimais do conjunto de soluções factíveis, no que diz respeito à ordem parcial  $\preceq$ , coincidem com o que se espera que sejam “as

melhores soluções” do problema (17.1), e são chamadas de *soluções eficientes* desse problema. Como consequência, um problema de otimização multiobjetivo (17.1) irá possuir, em geral, um conjunto de soluções eficientes, que apenas em casos muito particulares irá se reduzir a uma única solução eficiente.

É importante neste ponto enunciar formalmente as definições de *dominância*, de *solução eficiente* e de *conjunto Pareto-ótimo* e *fronteira Pareto-ótima*:

**Definição (Dominância):** Sejam  $\mathbf{x}_a$  e  $\mathbf{x}_b$  dois vetores de decisão factíveis do problema de otimização multiobjetivo (17.1). Se  $\mathbf{f}(\mathbf{x}_a) \preceq \mathbf{f}(\mathbf{x}_b)$  e  $\mathbf{f}(\mathbf{x}_a) \neq \mathbf{f}(\mathbf{x}_b)$ , então se diz que  $\mathbf{x}_a$  *domina*  $\mathbf{x}_b$ . Essa situação é indicada pela notação:  $\mathbf{x}_a \prec \mathbf{x}_b$ .

Com certo abuso de linguagem também diremos aqui, quando não houver possibilidade de causar ambiguidade, que se  $\mathbf{f}(\mathbf{x}_a) \preceq \mathbf{f}(\mathbf{x}_b)$  e  $\mathbf{f}(\mathbf{x}_a) \neq \mathbf{f}(\mathbf{x}_b)$ , então  $\mathbf{f}(\mathbf{x}_a)$  *domina*  $\mathbf{f}(\mathbf{x}_b)$ .

**Definição (Solução Eficiente):** Um vetor de variáveis de decisão  $\mathbf{x}^*$  factível é uma *solução eficiente* do problema de otimização multiobjetivo (17.1) se não existir qualquer outra solução factível desse problema que domine  $\mathbf{x}^*$ .

Também com abuso de linguagem diremos aqui, quando não houver possibilidade de causar ambiguidade, que se  $\mathbf{f}(\mathbf{x}^*)$  não é dominada pela imagem de nenhuma outra solução, sendo  $\mathbf{x}^*$  factível, então  $\mathbf{f}(\mathbf{x}^*)$  é um *ponto eficiente*.

**Definição (Conjunto Pareto-Ótimo):** O conjunto  $\mathcal{P}$  de todas as soluções eficientes do problema de otimização multiobjetivo (17.1) é denominado *conjunto Pareto-ótimo* desse problema.

**Definição (Fronteira Pareto-Ótima):** A imagem do conjunto  $\mathcal{P}$  de todas as soluções eficientes do problema de otimização multiobjetivo (17.1), constituída de todos os pontos eficientes no espaço de objetivos, é denominada *fronteira Pareto-ótima* desse problema.

A figura 17.2 ilustra os conceitos de dominância, de ponto eficiente, e de fronteira Pareto-ótima.

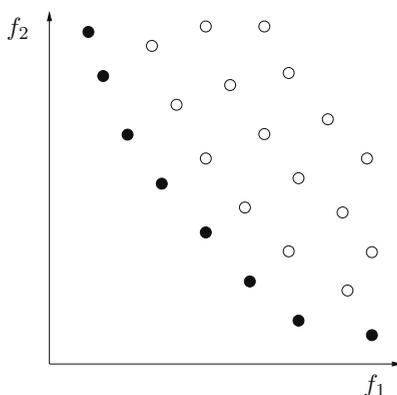


Figura 17.2: Os pontos brancos nesta figura são dominados por algum outro ponto, enquanto os pontos pretos, por sua vez, são não-dominados, no espaço de objetivos  $\mathcal{Y}$ . Desta forma, a fronteira Pareto-ótima relativa ao conjunto de todos os pontos representados na figura é constituída pelo conjunto dos pontos pretos, que são os pontos eficientes.

Finalmente podemos esboçar a tarefa que cabe a um algoritmo de otimização multiobjetivo. Idealmente, o resultado de uma otimização multiobjetivo deveria ter como resultado o conjunto  $\mathcal{P}$ , de todas as soluções eficientes do problema (17.1). No entanto, na prática, a geração numérica de todas as soluções contidas nesse conjunto não é possível em grande parte das vezes, devido à elevada car-

dinalidade desse conjunto. O caso extremo ocorre quando as variáveis de decisão do problema forem contínuas, quando a cardinalidade de  $\mathcal{P}$  usualmente será infinita. Nesse caso, o conjunto Pareto-ótimo será um objeto contínuo ou contínuo por partes, no espaço  $\mathbb{R}^n$ , como mostrado na figura 17.5, adiante. Mas mesmo em problemas de variáveis discretas, muito frequentemente o conjunto Pareto-ótimo poderá conter um número muito elevado de elementos. Assim, o resultado esperado de um procedimento de otimização multiobjetivo normalmente é um conjunto de amostras do conjunto  $\mathcal{P}$ , que deve estar distribuído de forma a representar esse conjunto (ou seja, de forma a dar uma boa idéia do formato geométrico e da localização desse conjunto).

### Interpretação Geométrica

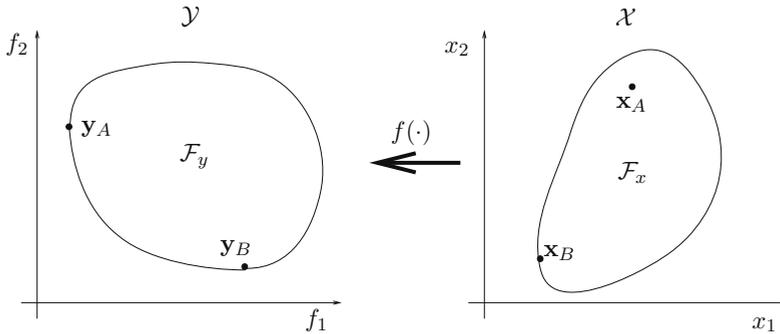


Figura 17.3: Mapeamento do espaço de parâmetros  $\mathcal{X}$  no espaço de objetivos  $\mathcal{Y}$  feito pela função  $f(\mathbf{x})$ . A região factível no espaço de parâmetros é designada por  $\mathcal{F}_x$ , sendo o conjunto-imagem da função  $f(\cdot)$  sobre o conjunto dos pontos factíveis designado por  $\mathcal{F}_y$ . O ponto de mínimo da função  $f_1(\mathbf{x})$  no espaço de parâmetros é designado por  $\mathbf{x}_A$ , e sua imagem no espaço  $\mathcal{Y}$  é designada por  $\mathbf{y}_A$ . O ponto de mínimo da função  $f_2(\mathbf{x})$  no espaço de parâmetros é designado por  $\mathbf{x}_B$ , e sua imagem no espaço  $\mathcal{Y}$  é designada por  $\mathbf{y}_B$ .

Uma interpretação geométrica pode ajudar a constituir um melhor entendimento da estrutura do problema de otimização multiobjetivo. Para o fim de estabelecer tal interpretação, iremos provisoriamente admitir que o vetor de variáveis de decisão  $\mathbf{x}$  do problema de otimização multiobjetivo (17.1) tenha componentes reais, ou seja,  $\mathbf{x} \in \mathbb{R}^n$ . Nesse caso, o conjunto factível  $\mathcal{F}_x$  será uma região do espaço de variáveis de decisão  $\mathcal{X} = \mathbb{R}^n$ , conforme encontra-se representado na figura 17.3. A imagem do vetor de funções-objetivo estará no espaço de objetivos  $\mathcal{Y} = \mathbb{R}^p$ , sendo que o conjunto dos pontos de  $\mathcal{Y}$  que correspondem à imagem de pontos em  $\mathcal{F}_x$  será denotado por  $\mathcal{F}_y$ . Deve-se notar que, nos eixos da representação cartesiana do espaço de objetivos  $\mathcal{Y}$ , estão representados os valores das diversas funções objetivo  $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})$ . Na figura 17.3, o ponto  $\mathbf{y}_A \in \mathcal{Y}$  corresponde ao vetor de objetivos que ocorre como imagem do ponto  $\mathbf{x}_A$  que minimiza a função  $f_1$ . O ponto  $\mathbf{y}_B \in \mathcal{Y}$  corresponde ao vetor de objetivos que ocorre como imagem do ponto  $\mathbf{x}_B$  que minimiza a função  $f_2$ . Observe-se que embora um ponto de mínimo de alguma função  $f_i$  possa ocorrer no interior da região  $\mathcal{F}_x$  (na figura 17.3, esse é o caso do ponto  $\mathbf{x}_A$ ), os pontos correspondentes às imagens de todos os mínimos no espaço  $\mathcal{Y}$  necessariamente se encontrarão na fronteira do conjunto  $\mathcal{F}_y$  (vide o ponto  $\mathbf{y}_A$ ).

A relação de dominância entre soluções factíveis pode ser representada geometricamente no espaço de objetivos  $\mathcal{Y}$ . O conceito de dominância encontra-se ilustrado na figura 17.4. Nessa figura, são mostrados os pontos  $\mathbf{y}_A$  e  $\mathbf{y}_B$  pertencentes ao espaço  $\mathcal{Y}$ . Um cone paralelo aos eixos coordenados do espaço  $\mathcal{Y}$  é colocado com vértice no ponto  $\mathbf{y}_B$ . Esse cone será denominado *cone polar*. Todos os pontos no interior do cone polar dominam  $\mathbf{y}_B$ , de forma que  $\mathbf{y}_B$  é dominado por  $\mathbf{y}_A$ . Mais precisamente, devemos dizer que o ponto  $\mathbf{x}_B$ , cuja imagem é  $\mathbf{y}_B$ , é dominado pelo ponto  $\mathbf{x}_A$ , cuja imagem é  $\mathbf{y}_A$ . A partir desta figura, pode-se inferir que quando se coloca um cone polar sobre a imagem  $\mathbf{y}^*$  de uma

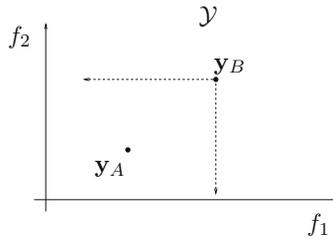


Figura 17.4: Representação da dominância no espaço de objetivos  $\mathcal{Y}$ . O ponto  $y_B$  será dominado por qualquer ponto localizado no cone com vértice em  $y_B$  (cone polar) que se encontra representado na figura. Por exemplo,  $y_A$  domina  $y_B$ .

solução eficiente  $x^*$ , não pode haver, no interior desse cone, a imagem de nenhuma outra solução factível (do contrário, essa solução não seria eficiente).

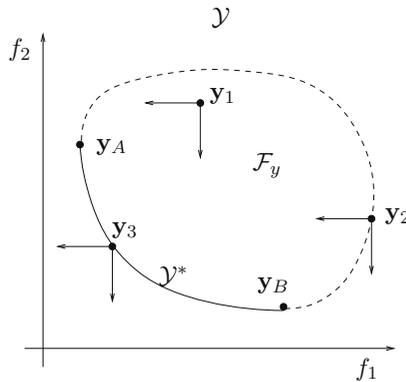


Figura 17.5: A fronteira de Pareto  $\mathcal{Y}^* \subset \mathcal{Y}$ , associada ao conjunto-imagem  $\mathcal{F}_y$  da função  $f(x)$  encontra-se representada em linha contínua. O restante da fronteira do conjunto  $\mathcal{F}_y$  está representado em linha tracejada. O conjunto  $\mathcal{Y}^*$  possui extremos nos pontos  $y_A$  e  $y_B$ , correspondentes aos mínimos individuais das funções  $f_1(x)$  e  $f_2(x)$ . O cone polar colocado sobre o ponto  $y_1$ , situado no interior do conjunto-imagem  $\mathcal{F}_y$ , terá interseção não vazia com esse conjunto. Todos os pontos situados nessa interseção dominam  $y_1$ , de forma que este não constitui uma solução eficiente. O ponto  $y_2$ , situado agora na fronteira de  $\mathcal{F}_y$ , tem seu cone polar ainda apresentando interseção não vazia com  $\mathcal{F}_y$ . Como  $y_2$  é dominado por todos os pontos situados nessa interseção, então  $y_2$  também não pode ser uma solução eficiente. Finalmente, o ponto  $y_3$ , situado em uma parte da fronteira de  $\mathcal{F}_y$  que faz parte da fronteira Pareto-ótima, tem seu cone polar situado de tal forma que sua interseção com o conjunto  $\mathcal{F}_y$  é agora vazia. Como não há nenhuma solução que domina  $y_3$  (se existisse, esta estaria no cone polar), então  $y_3$  tem de ser uma solução eficiente do problema.

Essa observação tem como consequência que todo ponto  $y_A$  pertencente ao interior do conjunto  $\mathcal{F}_y$  será dominado, pois o interior de um cone polar com vértice em  $y_A$  será sempre não-vazio, e haverá portanto algum ponto nesse cone que dominará  $y_A$ . Assim, todo ponto eficiente terá de ter sua imagem não no interior, mas na fronteira do conjunto  $\mathcal{F}_y$ , de forma que um cone polar situado sobre um desses pontos não terá nenhuma interseção com o próprio conjunto  $\mathcal{F}_y$ , de forma que não exista nenhum ponto que possa dominar esses pontos eficientes. Essa estrutura do problema está representada na figura 17.5. Por esse motivo, o conjunto-imagem do conjunto Pareto-ótimo é usualmente denominado *fronteira Pareto-ótima* do problema de otimização multiobjetivo.

Finalmente, pode-se estabelecer o que seria o conjunto de soluções  $\hat{\mathcal{X}}^*$  que se espera que sejam determinadas a partir de um procedimento de otimização multiobjetivo. Espera-se, desse procedimento, que produza: (i) um conjunto de soluções que esteja contido no conjunto Pareto-ótimo ( $\hat{\mathcal{X}}^* \subset \mathcal{X}^*$ ); e (ii) que represente uma amostragem do conjunto Pareto-ótimo capaz de descrever esse conjunto,

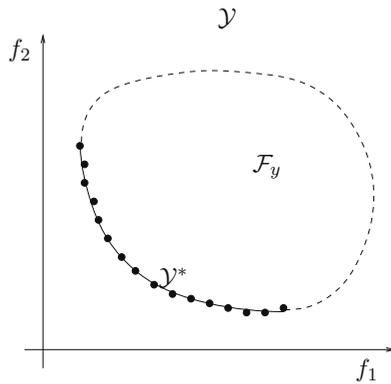


Figura 17.6: A fronteira de Pareto  $\mathcal{Y}^* \subset \mathcal{Y}$ , associada ao conjunto-imagem  $\mathcal{F}_y$  da função  $\mathbf{f}(\mathbf{x})$  encontra-se representada em linha contínua. Essa fronteira Pareto-ótima encontra-se amostrada por um conjunto de pontos  $\hat{\mathcal{Y}}^*$  que representam  $\mathcal{Y}^*$ , no sentido de que todo ponto de  $\mathcal{Y}^*$  está próximo (e portanto representado) por pelo menos um ponto no conjunto  $\hat{\mathcal{Y}}^*$ .

no sentido de que toda solução pertencente ao conjunto  $\mathcal{Y}^*$  tenha pelo menos um ponto de  $\hat{\mathcal{X}}^*$  em sua proximidade, e que portanto a “represente”. Na maioria dos algoritmos de otimização multi-objetivo que foram desenvolvidos até hoje, essa proximidade é medida no conjunto-imagem  $\mathcal{Y}$ , de forma que esses algoritmos são construídos de forma a produzir uma amostragem representativa da fronteira Pareto-ótima  $\mathcal{Y}^*$ . Deve-se notar que uma amostragem representativa da fronteira Pareto-ótima  $\mathcal{Y}^*$  não necessariamente implica uma correspondente amostragem representativa do conjunto Pareto-ótimo  $\mathcal{X}^*$ , e vice-versa. A figura 17.6 mostra um conjunto  $\hat{\mathcal{Y}}^*$  de amostras que seriam, nesse sentido, representativas do conjunto  $\mathcal{Y}^*$ , e que portanto deveriam ser o resultado de um algoritmo de otimização multiobjetivo.

## 2. Algoritmos Evolutivos Multiobjetivo

Algoritmos evolutivos, como foi visto nos capítulos anteriores deste livro, são constituídos essencialmente por operações de:

- **Variação:** a partir de soluções anteriores, devem ser geradas novas soluções;
- **Atribuição de Aptidão:** a cada solução gerada pelo algoritmo, o algoritmo deve atribuir um valor de aptidão, a ser usado para comparar essa solução com outras soluções;
- **Seleção:** dadas as aptidões de várias soluções disponíveis após a aplicação das operações de variação, devem-se escolher algumas para serem mantidas na população, enquanto outras serão eliminadas. Essa escolha é feita
- **Arquivamento:** as melhores soluções até então encontradas pelo algoritmo devem ser armazenadas.

Nos algoritmos genéticos mono-objetivo, por exemplo:

- Os operadores genéticos de mutação e de cruzamento exercem a função de mecanismos de variação.
- Dependendo do mecanismo de seleção empregado, a atribuição de aptidão pode ser feita com a atribuição do próprio valor da função-objetivo, ou utilizando uma transformação não-linear

desse valor, ou ainda pelo uso de um ranqueamento das soluções. Às vezes, o mecanismo de atribuição de aptidão inclui também um operador de nicho, que tem a função de aumentar a aptidão relativa de soluções que se localizam em regiões pouco exploradas do espaço, de forma a gerar uma pressão para o aumento da diversidade da população.

- As seleções baseadas em roleta (com diversas variações) podem utilizar aptidões construídas a partir de transformações não-lineares do valor de função objetivo, ou a partir de ranqueamentos. Já as seleções baseadas em torneio normalmente utilizam a aptidão diretamente expressa como o valor da função-objetivo.
- Os AG's mono-objetivo normalmente contam com um arquivamento baseado em elitismo, que às vezes encontra-se implícito no mecanismo de seleção.

Outros tipos de AE's contam com diferentes operadores, que sempre deverão executar essas funções.

Algoritmos Evolutivos Multiobjetivo também deverão contar com mecanismos que executem essas quatro funções. Normalmente, é possível reaproveitar os mesmos mecanismos de variação utilizados pelo correspondente algoritmo mono-objetivo, para o fim de construção de uma versão multiobjetivo do algoritmo. Assim, as operações de mutação e de cruzamento de um AG mono-objetivo poderão ser utilizadas, *ipsis literis*, em versões multiobjetivo do mesmo algoritmo. Também a função de seleção pode ser realizada, no caso multiobjetivo, com mecanismos idênticos aos empregados no caso mono-objetivo. Já as funções de atribuição de aptidão e de arquivamento irão requerer estruturas específicas para o caso multiobjetivo. A discussão sobre esse ponto é apresentada nas próximas subseções.

Nesta seção, serão estudados em particular os algoritmos *Non-Dominated Sorting Genetic Algorithm II*, ou NSGA-II, proposto por (Deb et al., 2002), e *Strength Pareto Evolutionary Algorithm 2*, ou SPEA2, proposto por (Zitzler et al., 2001). Esses algoritmos recebem tal destaque porque são os dois algoritmos evolutivos multiobjetivo que vieram sendo mais amplamente empregados nesta última década, sendo em grande parte responsáveis pela rápida popularização da otimização multiobjetivo evolutiva. Esses algoritmos são os representantes mais ilustres da chamada 2ª geração de EA's multiobjetivo, caracterizada pelo uso de técnicas de arquivamento de soluções, em combinação com a seleção pela dominância de Pareto. Hoje já existem algoritmos mais avançados que, não obstante, ainda não atingiram tamanha disseminação na comunidade científica e nos campos de aplicação da computação evolutiva. Na próxima seção, serão comentados alguns desses desenvolvimentos posteriores ao NSGA-II e ao SPEA2.

## Aptidão em AE's Multiobjetivo

A atribuição de aptidão em AE's multiobjetivo deve ser feita de forma a cumprir os seguintes requisitos:

- Se existir uma relação de dominância entre duas soluções,  $\mathbf{x}_a \prec \mathbf{x}_b$ , então a aptidão da solução  $\mathbf{x}_a$  deve necessariamente ser maior que a da solução  $\mathbf{x}_b$ .
- Caso não haja relação de dominância entre duas soluções, ou seja,  $\mathbf{x}_a \not\prec \mathbf{x}_b$  e  $\mathbf{x}_b \not\prec \mathbf{x}_a$ , então deve-se fazer a atribuição de aptidão a essas soluções de forma que a solução mais importante para garantir a representatividade da amostragem do conjunto de Pareto (normalmente aquela com menor número de vizinhos sem relação de dominância) tenha maior aptidão.

O primeiro desses requisitos é necessário para que o algoritmo convirja para o conjunto Pareto-ótimo, no qual estarão soluções cuja aptidão deverá ser maior que a de qualquer solução fora desse conjunto. O segundo requisito é também indispensável quando se assume que o propósito de um procedimento de otimização multiobjetivo seja a geração de um conjunto de amostras do conjunto Pareto-ótimo que seja representativo.

São possíveis diferentes maneiras de construir mecanismos de atribuição de aptidão que atendam a tais requisitos. Esses mecanismos devem partir de um índice que garanta o ordenamento de acordo com as relações de precedência. Alguns diferentes índices utilizados na literatura são:

**Índice de Fronteira ( $I_f$ ):** Atribui-se o índice 0 a todas as soluções não-dominadas da população (fronteira não-dominada). Essas soluções são então excluídas do conjunto de trabalho e, sobre as soluções restantes, determinam-se as soluções não-dominadas relativas a esse conjunto, que corresponde à nova fronteira não-dominada, relativa ao novo conjunto. A essas soluções atribui-se o índice 1, e à fronteira subsequente, obtida no passo seguinte, o índice 2, e assim por diante, aumentando-se o índice de uma unidade para cada nova fronteira, até que todos os pontos da população estejam rotulados.

**Índice de Dominação ( $I_d$ ):** Realiza-se a contagem do número de soluções que dominam cada uma das soluções contidas na população. Às soluções não dominadas por nenhuma outra, atribui-se o índice 0. Às soluções dominadas por uma única outra solução, atribui-se o índice 1. Às soluções dominadas por duas outras soluções, atribui-se o índice 2, e assim por diante.

**Índice de Força-Pareto ( $I_p$ ):** Realiza-se a contagem do número de soluções que são dominadas por cada uma das soluções contidas na população. Às soluções não dominadas por nenhuma outra, atribui-se o índice 0. Como índice de cada solução dominada  $\mathbf{x}^i$ , totaliza-se a soma do número de soluções que são dominadas pelas soluções que dominam  $\mathbf{x}^i$ .

O índice  $I_f$  foi inicialmente empregado no algoritmo NSGA (Srinivas e Deb, 1994), e integra também o algoritmo NSGA-II (Deb et al., 2002). O índice  $I_d$  foi utilizado no algoritmo MOGA (Fonseca e Fleming, 1993, 1995). O índice  $I_p$  foi proposto junto com o algoritmo SPEA2 (Zitzler et al., 2001). Além dessas duas alternativas, são possíveis outras formas de se criarem índices que preservem as relações de precedência. Chamamos particularmente a atenção para a metodologia utilizada no algoritmo IBEA (Zitzler e Kunzli, 2004), que utiliza uma lógica bem distinta dessas acima.

A figura 17.7 ilustra os índices obtidos dessas formas. Deve-se notar que os índices atribuídos às soluções não-dominadas da população coincidem nos três casos, sendo sempre 0. Já as soluções de alguma forma dominadas são ordenadas de maneiras diferentes segundo os diferentes índices. Nos três casos, entretanto, garante-se que soluções de mesmo índice não tenham relação de precedência entre si. Também se garante que, sempre que houver relação de precedência entre duas soluções, a solução que domina irá possuir índice menor que o da solução que é dominada, ou seja:

$$\mathbf{x}_a \prec \mathbf{x}_b \Rightarrow I_*(\mathbf{x}_a) < I_*(\mathbf{x}_b),$$

sendo isso válido para  $I_*$  dado por  $I_f$ , por  $I_d$  ou por  $I_p$ . A partir deste ponto, para falar genericamente de um indicador que pode ser qualquer desses três, utilizaremos a notação  $I_*$ . Nos algoritmos a serem apresentados a seguir, a notação  $I_*(\mathcal{P})$  será utilizada para representar o vetor de resultados do cálculo do valor do índice  $I_*$  para cada elemento de um conjunto  $\mathcal{P}$  de pontos do espaço de variáveis de decisão.

Após o cômputo do índice  $I_*$  sobre o conjunto  $\mathcal{P}$  dos pontos pertencentes à população de um AE multiobjetivo, já passa a existir uma ordenação das soluções pertencentes a essa população. Se um desses índices fosse utilizado como valor de aptidão, no algoritmo, pode-se perceber que seria induzida uma pressão seletiva que iria favorecer a seleção de indivíduos de maior precedência, na ordem parcial  $\preceq$ . Dessa forma, a população iria convergir em direção ao conjunto Pareto-ótimo do problema de otimização multiobjetivo.

Entretanto, deve-se notar que a simples aplicação de um desses índices como valor da aptidão dos indivíduos não iria produzir o efeito de que a população convergisse para amostras bem distribuídas no conjunto Pareto-ótimo, pois não haveria nenhum mecanismo causando o efeito de uma distribuição

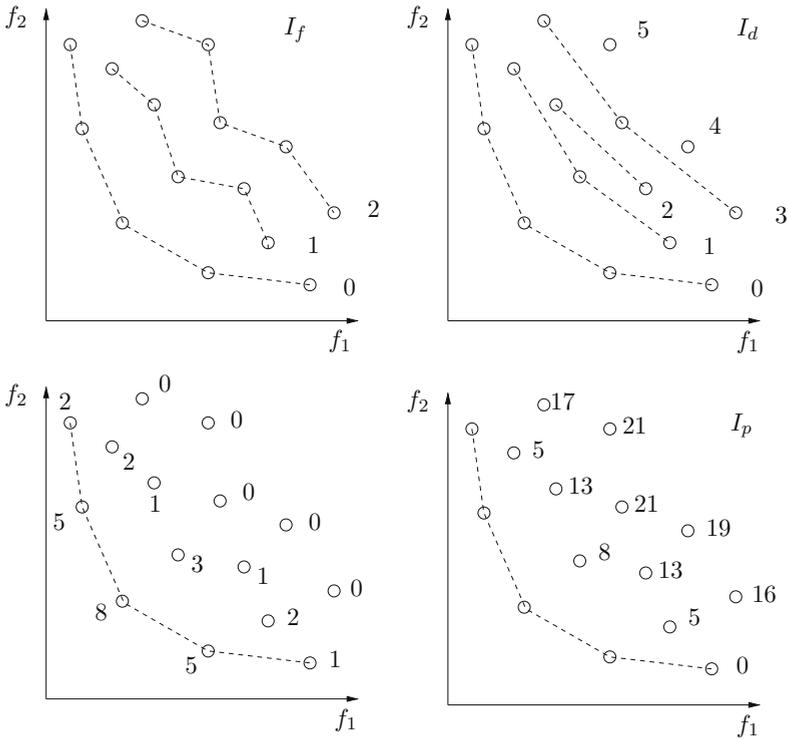


Figura 17.7: Exemplo de cômputo do Índice de Fronteira ( $I_f$ , no canto superior esquerdo), do Índice de Dominação ( $I_d$ , no canto superior direito) e do Índice de Força-Pareto ( $I_p$ , no canto inferior esquerdo), para o mesmo conjunto de pontos. No canto inferior esquerdo, estão registrados, para cada ponto, o número de outros pontos que ele domina. Esse número é utilizado no cômputo no índice  $I_p$ .

representativa do conjunto<sup>3</sup>. Por esse motivo, torna-se necessário acrescentar algum fator que produza um refinamento do ordenamento baseado no índice  $I_*$  que seja capaz de induzir essa boa distribuição. Esse fator de correção deve preservar a ordem relativa entre as soluções que tiverem índices  $I_*$  diferentes, pois essa ordem é a garantia de que as relações de precedência sejam atendidas; assim, o fator de correção usualmente é aplicado de forma a apenas desempatar as soluções que tiverem o mesmo índice. Duas maneiras de realizar tal correção são:

**Distância de Aglomeração ( $\delta_a$ ):** Em problemas com dois objetivos, faz-se a ordenação das  $q$  soluções de mesmo índice  $I_*$  (essas soluções, portanto, não têm relação de dominância entre si), de acordo com a ordem crescente de um dos objetivos (por exemplo  $f_1$ ), ficando estas soluções rotuladas com índices que vão de  $i = 1$  até  $i = q$ . Conseqüentemente, essas soluções ficarão ordenadas pela ordem decrescente do outro objetivo ( $f_2$ ). Para cada solução de  $i = 2$  até  $i = q - 1$  calcula-se o número:

$$\delta_a(\mathbf{x}^i) = [f_2(\mathbf{x}^{i-1}) - f_2(\mathbf{x}^{i+1})] + [f_1(\mathbf{x}^{i+1}) - f_1(\mathbf{x}^{i-1})]$$

Esse número é proporcional ao perímetro do retângulo cujos vértices opostos encontram-se sobre as soluções vizinhas da solução  $i$ , no espaço de objetivos. Quanto maior o número  $\delta_a(\mathbf{x}^i)$ , mais distante a solução  $i$  estará das vizinhas, e portanto mais importante será esta solução para

<sup>3</sup> O índice  $I_p$  até possui algum efeito de evitar a aglomeração de pontos em uma pequena região, causado pela contagem das dominâncias herdadas pelos pontos. No entanto, deve-se notar que esse índice não consegue, por exemplo, discriminar entre duas soluções não-dominadas diferentes.

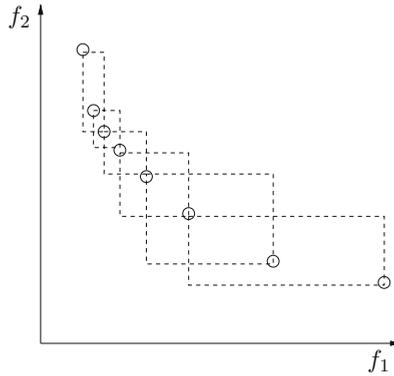


Figura 17.8: Interpretação geométrica da distância de aglomeração  $\delta_a$ . Cada solução  $\delta_a(\mathbf{x}^i)$  tem sua imagem no espaço de objetivos,  $\mathbf{y}^i$ , no interior de um retângulo cujos vértices opostos estão localizados nas imagens vizinhas  $\mathbf{y}^{i-1}$  e  $\mathbf{y}^{i+1}$ . O indicador  $\delta_a(\mathbf{x}^i)$  corresponde à metade do perímetro desse retângulo (ou seja, a soma de sua base mais sua altura).

garantir a representatividade do conjunto de soluções. Por convenção, as soluções extremas, com  $i = 1$  e  $i = q$ , terão  $\delta_a(\mathbf{x}^1) = \infty$  e  $\delta_a(\mathbf{x}^q) = \infty$ , o que significa que essas soluções são consideradas mais importantes que quaisquer outras, para garantir a representatividade do conjunto. A figura 17.8 ilustra o significado de  $\delta_a$ .

**k-Vizinhança ( $\delta_k$ ):** Estabelece-se inicialmente um valor para  $k$ . Um possível critério para se definir  $k$  pode ser, por exemplo, fazer  $k = \sqrt{N}$ , sendo  $N$  o tamanho da população. Calcula-se então a distância (por exemplo euclidiana), no espaço de objetivos, entre cada par de soluções pertencentes à população. Para cada solução  $\mathbf{x}^i$  determina-se a distância  $\delta_k(\mathbf{x}^i)$  que corresponde a  $k$ -ésima menor distância de  $\mathbf{x}^i$  em relação às demais soluções da população.

Tanto o número  $\delta_a(\mathbf{x}^i)$  quanto o número  $\delta_k(\mathbf{x}^i)$  podem ser interpretados como medidas da densidade da amostragem existente na região ao redor do ponto  $\mathbf{x}^i$ . Assim, para que ocorra uma melhor amostragem do conjunto Pareto-ótimo, o algoritmo deverá privilegiar, a cada momento, a seleção de soluções em regiões menos densamente ocupadas, ou seja, soluções com menor valor de  $\delta_a$  ou de  $\delta_k$ . O indicador  $\delta_a$  foi proposto junto com o algoritmo NSGA-II (Deb et al., 2002). Já o indicador  $\delta_k$  foi empregado no algoritmo SPEA2 (Zitzler et al., 2001).

Deve estar claro que é possível definir outras medidas de densidade de amostragem capazes de executar função semelhante à desempenhada por  $\delta_a$  e  $\delta_k$ . Por exemplo, o algoritmo RPSGA (Gaspar-Cunha, 2000; Gaspar-Cunha e Covas, 2004) utiliza uma técnica de *clusterização* proposta por (Roseman e Gero, 1985) para gerar uma medida desse tipo. Para fazer referência, genericamente, a qualquer medida de densidade com tais propriedades, utilizaremos aqui a notação  $\delta_*$ . A notação  $\delta_*(I_*, \mathcal{P})$  será utilizada para representar o vetor de resultados do cálculo do valor do indicador  $\delta_*$  para cada elemento de um conjunto  $\mathcal{P}$  de pontos do espaço de variáveis de decisão, após a aplicação prévia da operação  $I_*(\mathcal{P})$  sobre os elementos desse conjunto.

Finalmente, dados um índice  $I_*$ , relativo às relações de precedência entre soluções, e um índice  $\delta_*$ , ligado à densidade de amostragem, pode-se construir um único ranqueamento das soluções pertencentes à população do AE, pela aplicação primeiro de  $I_*$ , e depois de  $\delta_*$  para desempatar as soluções de mesmo valor de  $I_*$ . A ordem total implícita nesse ranqueamento será utilizada para criar uma pressão seletiva dirigida à obtenção de um conjunto de amostras pertencentes ao conjunto Pareto-ótimo e que ainda constitua uma amostragem representativa desse conjunto. Assim, dadas  $N$  soluções  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , um ranqueamento é produzido de forma a associar um número inteiro de 1 até  $N$  a cada solução. Essa função de ranqueamento  $R(\mathbf{x})$  é tal que:

- Se  $I_*(\mathbf{x}_a) < I_*(\mathbf{x}_b)$  então  $R(\mathbf{x}_a) < R(\mathbf{x}_b)$
- Se  $I_*(\mathbf{x}_a) = I_*(\mathbf{x}_b)$  então:
  - Se  $\delta_*(\mathbf{x}_a) < \delta_*(\mathbf{x}_b)$  então  $R(\mathbf{x}_a) < R(\mathbf{x}_b)$
  - Se  $\delta_*(\mathbf{x}_b) < \delta_*(\mathbf{x}_a)$  então  $R(\mathbf{x}_b) < R(\mathbf{x}_a)$
  - Se  $\delta_*(\mathbf{x}_b) = \delta_*(\mathbf{x}_a)$  então sorteia-se  $R(\mathbf{x}_b) < R(\mathbf{x}_a)$  ou  $R(\mathbf{x}_a) < R(\mathbf{x}_b)$

Deve-se notar que a aplicação de  $I_*$  e  $\delta_*$  pode ainda deixar lugar a algumas soluções ficarem empatadas, com mesmo valor dos dois índices. No entanto, este será um evento relativamente raro, ao contrário do que ocorreria pela aplicação apenas de  $I_*$ . No caso desse tipo de empate, o desempate é feito aleatoriamente.

O ranqueamento  $R(\mathbf{x})$  pode enfim ser utilizado para definir a aptidão<sup>4</sup>, que será utilizada em um mecanismo de seleção, seja ele do tipo *roleta* ou *torneio*, exatamente como nos AG's mono-objetivo. A notação  $\mathbf{R}(I_*, \delta_*, \mathcal{P})$  será utilizada aqui para representar o vetor de resultados do ranqueamento dos pontos pertencentes ao conjunto  $\mathcal{P}$ , após a aplicação prévia das operações  $I_*(\mathcal{P})$  e  $\delta_*(I_*, \mathcal{P})$  sobre o conjunto. Já a notação

$$\mathcal{V} \leftarrow \text{seleciona}(I_*, \delta_*, \mathbf{R}, \mathcal{P}, n)$$

irá representar a operação de atribuir ao conjunto  $\mathcal{V}$  as  $n$  soluções pertencentes ao conjunto  $\mathcal{P}$  que ocupam as  $n$  primeiras soluções no ranqueamento  $\mathbf{R}(I_*, \delta_*, \mathcal{P})$ .

## Arquivamento em AE's Multiobjetivo

No caso dos algoritmos evolutivos mono-objetivo, o arquivamento da melhor solução é uma operação simples, que envolve guardar o vetor de variáveis de decisão  $\mathbf{x}$  que represente, em cada momento, a melhor solução até então encontrada pelo algoritmo durante sua execução.

No caso de algoritmos multiobjetivo, essa função de arquivamento é significativamente mais complexa. Para situar a dificuldade, suponha-se inicialmente o seguinte procedimento ingênuo de arquivamento que poderia ser empregado pelo menos como conceito:

- Seja um conjunto  $\mathcal{P}_{k-1}$  que constitui o arquivo de todas as soluções não-dominadas encontradas até a iteração  $k - 1$  do algoritmo.
- Na iteração  $k$ , para cada nova solução  $\mathbf{x}_i$  encontrada, são feitas as seguintes operações:
  - Se  $\mathbf{x}_i$  é dominada por algum elemento de  $\mathcal{P}_{k-1}$ , então o arquivo é mantido inalterado (a solução  $\mathbf{x}_i$  não é introduzida no arquivo).
  - Se  $\mathbf{x}_i$  domina uma ou mais soluções contidas em  $\mathcal{P}_{k-1}$ , então as soluções dominadas são excluídas do arquivo, e a solução  $\mathbf{x}_i$  é incluída no arquivo.
  - Se  $\mathbf{x}_i$  não tem relação de dominância (nem domina nem é dominada) com nenhuma das soluções contidas em  $\mathcal{P}_{k-1}$ , então  $\mathbf{x}_i$  trata-se de uma nova estimativa de solução não-dominada, sendo incluída no arquivo.
- A execução dessas operações para todas as novas soluções geradas na iteração  $k$ , sobre o arquivo  $\mathcal{P}_{k-1}$  vindo da iteração anterior, irá gerar o novo arquivo  $\mathcal{P}_k$  ao final da iteração  $k$ .

<sup>4</sup> Da maneira como foi definido  $R(\mathbf{x})$ , quanto menor o seu valor, maior será a aptidão do indivíduo, e portanto maior a chance dele ser selecionado.

A dificuldade com esse procedimento é que muito frequentemente a cardinalidade do conjunto Pareto-ótimo de um problema de otimização multiobjetivo será muito elevada, ou até mesmo infinita, como no caso da maioria dos problemas de variáveis contínuas. Assim, haverá uma tendência para que esse procedimento gere um arquivo  $\mathcal{P}_k$  que irá crescer muito rapidamente. Isso causa problemas não apenas relacionados com o armazenamento de um grande número de soluções, mas também com o esforço computacional para avaliar novas soluções, pois cada nova solução tem de ser comparada com todas as soluções do arquivo.

É preciso, portanto, de alguma forma limitar o número de soluções a serem armazenadas no arquivo. Conseqüentemente, a questão da *amostragem representativa*, discutida anteriormente no contexto da construção da função de aptidão  $\mathbf{R}(I_*, \delta_*, \mathcal{P})$ , também será relevante para o arquivamento, pois será necessário, dentre as diferentes alternativas para arquivar soluções não-dominadas, escolher uma estratégia que garanta essa representatividade, ou diversidade, do conjunto arquivado.

Suponha-se que deva ser montado um arquivo que irá conter as  $q$  melhores soluções que o algoritmo tiver encontrado até um dado momento. As seguintes hipóteses podem ocorrer:

- É possível que, até o momento em questão, ainda não tenham sido encontradas  $q$  soluções não-dominadas. Nesse caso, uma alternativa é deixar o arquivo incompleto, contendo apenas as soluções não-dominadas até então determinadas. Outra alternativa é também incluir no arquivo algumas das soluções dominadas. Para se proceder a essa inclusão, é possível simplesmente utilizar o ranqueamento  $\mathbf{R}(I_*, \delta_*, \mathcal{P})$ , conforme construído anteriormente, para escolher as soluções a serem acrescentadas.
- É possível também que num dado momento tenham sido determinadas exatamente  $q$  soluções não-dominadas. Nesse caso, essas soluções deverão constituir o arquivo.
- Por fim, é possível ainda que o número de soluções não-dominadas disponível, num dado momento, seja maior que o tamanho do arquivo,  $q$ . Nesse caso, algumas soluções não-dominadas deverão ser descartadas. Novamente, uma solução simples é utilizar o ranqueamento  $\mathbf{R}(I_*, \delta_*, \mathcal{P})$ , preservando as soluções que ocupam as  $q$  primeiras posições do ranking e descartando as demais. No entanto, é comum o uso de outro tipo de estratégia para efetuar esse descarte.

No caso do algoritmo NSGA-II (Deb et al., 2002), o arquivamento é feito utilizando-se a própria população como arquivo de tamanho  $N$ . Após a geração de  $N$  novas soluções por meio dos operadores de variação, as  $N$  melhores soluções do conjunto de  $2N$  elementos que contém as  $N$  novas soluções e mais as  $N$  soluções da população anterior são diretamente selecionadas utilizando o ranqueamento  $\mathbf{R}(I_f, \delta_a, \mathcal{P})$ , de forma determinística. Assim, o arquivamento das melhores soluções fica implícito no próprio mecanismo de seleção da nova população.

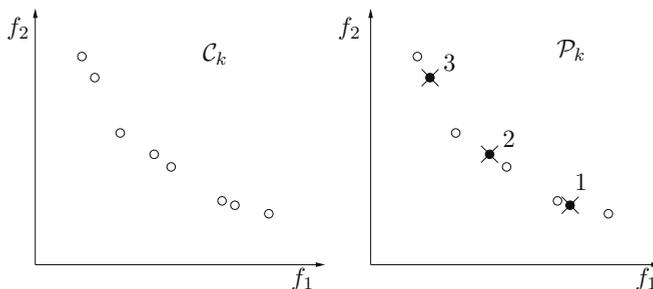


Figura 17.9: Representação da seqüência de descarte de pontos utilizada no arquivamento do algoritmo SPEA2.

No caso do algoritmo SPEA2 (Zitzler et al., 2001), é empregado um arquivo das melhores soluções, mantido separado da população. Em caso do número de soluções não-dominadas ser menor que  $q$ , o

arquivo é completado simplesmente utilizando o ranqueamento  $\mathbf{R}(I_p, \delta_k, \mathcal{P})$ . Já em caso do número de soluções não-dominadas exceder o tamanho do arquivo, ao invés de se utilizar o ranqueamento, emprega-se um critério baseado num índice que mede a densidade das soluções do arquivo apenas em relação às outras soluções também pertencentes ao arquivo, não mais considerando toda a população. O procedimento, chamado de *truncagem*, é descrito a seguir:

- O processo se inicia com um conjunto  $\mathcal{C}$  de soluções não-dominadas, de cardinalidade  $c > q$ , do qual devem ser excluídas  $c - q$  soluções, de forma a constituir o conjunto  $\mathcal{Q}$ , que deverá ter cardinalidade  $q$ .
- Determinam-se as distâncias dois-a-dois, no espaço de objetivos, entre todos os elementos de  $\mathcal{C}$ .
- Tomam-se os elementos aos quais se associa a menor distância. Esses serão pelo menos dois elementos, distantes um do outro dessa mínima distância. Desses elementos, escolhe-se aquele que apresentar a menor distância em relação a um outro elemento, para ser excluído do conjunto  $\mathcal{C}$ . Caso persista o empate entre elementos (ou seja, mais de um elemento cuja segunda menor distância a outro ponto do conjunto é igual) considera-se a terceira menor distância como critério de desempate, e assim por diante.
- O procedimento se repete, até que restem apenas  $q$  soluções em  $\mathcal{C}$ , as quais são então armazenadas em  $\mathcal{P}$ .

Esse procedimento é ilustrado na figura 17.9. A notação

$$\mathcal{P} \leftarrow \text{truncagem}(\mathcal{C}, q)$$

será utilizada para designar a aplicação dessa operação de truncagem sobre o conjunto  $\mathcal{C}$  de maneira a que a cardinalidade do conjunto  $\mathcal{P}$  resultante seja igual a  $q$ .

## Estrutura de AE's Multiobjetivo

Para exemplificar a construção de algoritmos evolutivos multiobjetivo, são agora apresentadas as estruturas do NSGA-II e do SPEA2. O pseudo-código do algoritmo NSGA-II encontra-se representado no Algoritmo 1, e o pseudo-código do algoritmo SPEA2 encontra-se descrito no Algoritmo 2. As seguintes definições serão empregadas nos algoritmos:

$\mathcal{B} \leftarrow \text{variação}(\mathcal{A}, n)$

irá representar a aplicação de operadores de mutação e de cruzamento aos indivíduos do conjunto  $\mathcal{A}$ , resultando em  $n$  novos indivíduos que são armazenados no conjunto  $\mathcal{B}$ .

$\mathcal{B} \leftarrow \text{aleatório}(n)$

irá representar a geração aleatória de  $n$  indivíduos, que serão armazenados no conjunto  $\mathcal{B}$ .

$\mathcal{B} \leftarrow \text{não-dominado}(\mathcal{A})$

irá indicar a operação de identificar os elementos não-dominados pertencentes ao conjunto  $\mathcal{A}$ , atribuindo o resultado ao conjunto  $\mathcal{B}$ .

$\mathcal{B} \leftarrow \text{torneio}(\mathcal{A}, \mathbf{R}, I_*, \delta_*, n)$

indicará a realização de um torneio binário com reposição<sup>5</sup>, realizado com os elementos do conjunto  $\mathcal{A}$ , utilizando como função de aptidão o ranqueamento  $\mathbf{R}(I_*, \delta_*, \mathcal{A})$ , resultando na escolha de  $n$  indivíduos que são armazenados no conjunto  $\mathcal{B}$ .

---

**Algoritmo 1** Pseudocódigo do NSGA-II

---

```

1:  $\mathcal{X} \leftarrow \text{aleatório}(N)$ 
2: enquanto critério de parada não satisfeito faça
3:    $\bar{\mathcal{X}} \leftarrow \text{variação}(\mathcal{X}, N)$ 
4:    $\mathcal{P} \leftarrow \bar{\mathcal{X}} \cup \mathcal{X}$ 
5:    $\mathcal{X} \leftarrow \text{seleciona}(I_f, \delta_a, \mathbf{R}, \mathcal{P}, N)$ 
6: fim enquanto

```

---



---

**Algoritmo 2** Pseudocódigo do SPEA2

---

```

1:  $\mathcal{X} \leftarrow \text{aleatório}(N)$ 
2:  $\mathcal{Q} \leftarrow \emptyset$ 
3: enquanto critério de parada não satisfeito faça
4:    $\mathcal{P} \leftarrow \mathcal{X} \cup \mathcal{Q}$ 
5:    $\mathcal{Q} \leftarrow \text{não-dominado}(\mathcal{P})$ 
6:   se  $|\bar{\mathcal{Q}}| > q$  então
7:      $\mathcal{Q} \leftarrow \text{truncagem}(\mathcal{Q}, q)$ 
8:   senão se  $|\mathcal{Q}| < q$  então
9:      $\mathcal{Q} \leftarrow \text{seleciona}(I_p, \delta_k, \mathbf{R}, \mathcal{Q}, q)$ 
10:  fim se
11:   $\mathcal{X} \leftarrow \text{torneio}(\mathcal{Q}, \mathbf{R}, I_p, \delta_k, N)$ 
12:   $\mathcal{X} \leftarrow \text{variação}(\mathcal{X}, N)$ 
13: fim enquanto

```

---

Além disso, a notação  $|\mathcal{A}|$  indica a cardinalidade do conjunto  $\mathcal{A}$ .

É importante mencionar que o algoritmo NSGA-II e o algoritmo SPEA2 incluem em sua especificação, além das macro-operações a serem realizadas sobre os conjuntos descritas nesses pseudocódigos, também a especificação de estruturas de dados eficientes, bem como de algoritmos detalhados particularmente eficazes para a realização dessas macro-operações. Isso quer dizer que, embora todo algoritmo que implemente os pseudo-códigos descritos no Algoritmo 1 e no Algoritmo 2 vá certamente produzir sequências de populações e resultados compatíveis com o NSGA-II e o SPEA2, não necessariamente isso será feito com a mesma eficiência computacional dos algoritmos originais.

---

### 3. Desenvolvimentos Recentes e Futuros

---

A computação evolutiva multiobjetivo foi, ao longo da década dos 2000, um tema no qual foi concentrado intenso esforço de pesquisa. Uma grande variedade de detalhes a respeito do funcionamento dos algoritmos dessa classe vêm sendo investigada, e encontra-se descrita em extensa literatura, constituída de quase seis mil artigos, publicados nos últimos dez anos, que se encontram hoje indexados<sup>6</sup>. Nesta seção, ao invés de procurar apresentar uma lista abrangente de temas que vêm sendo abordados, faremos uma discussão de apenas quatro assuntos, que se destacam por concentrarem o trabalho de muitos grupos de pesquisadores, em vários locais diferentes, e que parecem, neste momento, ter potencial para causar mudanças nas estruturas consideradas padrão na área de algoritmos evolutivos multiobjetivo: a hibridização com *mecanismos de busca local*, o desenvolvimento de algo-

<sup>5</sup> Em um torneio binário com reposição, dois indivíduos são escolhidos a cada vez, aleatoriamente. Esses dois indivíduos são comparados, utilizando uma função de aptidão, sendo o melhor dos dois escolhido. Os dois indivíduos são retornados ao conjunto original, e o processo se repete, até que  $n$  escolhas tenham sido feitas.

<sup>6</sup> Consulta realizada na base Scopus, em 14/06/2011, indicou 5936 artigos em 10 anos.

ritmos baseados no *indicador de hipervolume*, a introdução de ferramentas para *indicação progressiva de preferências*, e a questão teórica do tratamento de *muitos objetivos*.

## Mecanismos de Busca Local

Sabe-se hoje que, em geral, os algoritmos evolutivos irão requerer algum mecanismo de busca local, em adição aos operadores evolutivos básicos, para realizar eficientemente as tarefas de determinação de mínimos de funções. Isto ocorre porque os operadores evolutivos tradicionais são construídos a partir de movimentos aleatorizados, que geram novos pontos que serão aproveitados se forem melhores que os anteriores. Esse tipo de movimentação é importante para a realização de buscas globais, principalmente no caso de funções multimodais. No entanto, mais cedo ou mais tarde, o processo de busca termina por chegar em pontos a partir dos quais seria possível atingir o ponto de ótimo por meio da escolha da direção do movimento como aquela em que a função decresce mais rapidamente. Quando o processo de otimização atinge regiões em que essa busca é possível, fazer a busca utilizando a informação local sobre o decrescimento da função torna-se muito mais eficiente que a busca aleatorizada típica dos operadores evolutivos. Isso vale tanto para algoritmos para problemas de variáveis discretas quanto para problemas de variáveis contínuas. Essa hibridização de algoritmos evolutivos com técnicas de busca determinística local poderia ser realizada em dois estágios: primeiro se faz a busca aleatorizada e, a partir de um certo ponto, inicia-se uma busca por meio de algoritmos determinísticos. No entanto, há uma tendência hoje para a construção de algoritmos que empregam operadores de busca local como um operador constituinte do algoritmo evolutivo, que é executado ao longo das iterações do AE. Esse tipo de esquema tem sido denominado *Algoritmo Memético* (Moscatto, 1999; Hu et al., 2003).

O primeiro algoritmo memético multiobjetivo foi apresentado por (Ishibuchi e Murata, 1998). Outro algoritmo memético importante foi proposto por (Knowles e Corne, 2000b). O algoritmo denominado *Memetic Pareto Archive Evolutionary Strategy* (M-PAES) associa um esquema evolutivo de busca global, baseada em operadores de mutação e de recombinação, com o algoritmo PAES (Knowles e Corne, 1999), que realiza buscas locais. Uma revisão sobre algoritmos meméticos multiobjetivo é apresentada em (Knowles e Corne, 2004).

Neste momento, podem ser identificados três tipos de estratégias para a realização de buscas locais em algoritmos evolutivos multiobjetivo:

- Durante a execução do algoritmo, são escolhidos alguns pontos do arquivo corrente, e são aplicadas operações de busca local sobre esses pontos, na tentativa de gerar novos pontos que dominem os anteriores, ou ainda pontos sem relação de dominância com os anteriores que sirvam para melhorar a amostragem da estimativa do conjunto Pareto-ótimo. (Lara et al., 2010) apresentam um tipo de operador que procura executar essas operações de maneira automática. (Sindhya et al., 2011), por outro lado, apresentam estratégias baseadas em algoritmos clássicos de programação não-linear para realizar essa tarefa.
- Durante a execução do algoritmo evolutivo, vão sendo construídos modelos das funções-objetivo do problema multiobjetivo. Esses modelos são então empregados para a geração de estimativas melhoradas de soluções Pareto-ótimas. (Wanner et al., 2008) mostram uma estratégia baseada em modelos de aproximação quadrática para realizar essa operação.
- Outra alternativa é a construção de modelos do próprio conjunto Pareto-ótimo, como forma de articular a informação a ser utilizada na busca local. Os autores (Gaspar-Cunha e Vieira, 2004) e (Loshchilov et al., 2010) mostram alternativas de realização desta estratégia.

## Indicador de Hipervolume

Neste capítulo foi visto que a tarefa de avaliar a qualidade das soluções de algoritmos de otimização multiobjetivo é bem mais complexa que no caso mono-objetivo. No caso multiobjetivo, essa tarefa diz

respeito a avaliar a qualidade de um conjunto de pontos, que deve constituir uma amostragem representativa do conjunto Pareto-ótimo. Não basta avaliar isoladamente o quão cada ponto se aproxima do conjunto Pareto-ótimo exato, mas também de julgar o quão bem representado está cada ponto desse conjunto exato, naquele conjunto-solução que procura representá-lo. Uma discussão abrangente sobre essa questão de avaliar os conjuntos-solução de algoritmos de otimização multiobjetivo foi apresentada por (Zitzler et al., 2003).

Uma maneira engenhosa de executar essa tarefa foi proposta por (Zitzler, 1999), baseada na medição do tamanho da região dominada pelo conjunto-solução a ser avaliado. O indicador que resulta dessa medição, chamado de *indicador de hipervolume*<sup>7</sup>, é ilustrado na figura 17.10. O cálculo

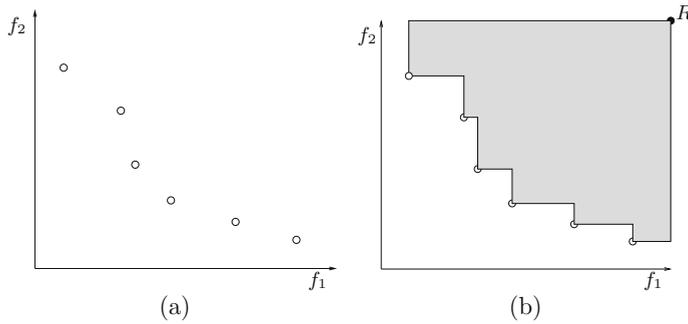


Figura 17.10: Indicador de hipervolume. (a) Conjunto de pontos a ser avaliado, no espaço de objetivos. (b) O valor do indicador de hipervolume corresponde ao volume da região constituída pelos pontos que simultaneamente dominam um ponto de referência  $R$  e são dominados por pelo menos um ponto do conjunto a ser avaliado.

desse indicador é feito da seguinte forma:

- Toma-se o conjunto de pontos a ser avaliado, no espaço de objetivos.
- Acrescenta-se um ponto de referência  $R$ , dominado por todos os pontos do conjunto a ser avaliado.
- Calcula-se o volume da região constituída pelos pontos que simultaneamente dominam  $R$  e são dominados por pelo menos um ponto do conjunto a ser avaliado.

No caso de duas dimensões (dois objetivos), esse volume se reduz à área da união dos retângulos, com lados paralelos aos eixos coordenados, que têm um vértice em  $R$  e o vértice oposto em um ponto do conjunto.

Uma propriedade desejável para um indicador que se proponha a avaliar conjuntos de soluções fornecidas por algoritmos de otimização multiobjetivo seria a chamada *Pareto-conformidade*<sup>8</sup>. Sejam  $A$  e  $B$  dois conjuntos sendo avaliados, e  $I(A)$  e  $I(B)$  os valores de um indicador  $I$  calculado sobre esses conjuntos. A propriedade de Pareto-conformidade é expressa como (Zitzler et al., 2007):

$$A \preceq B \text{ e } B \not\preceq A \Rightarrow I(A) > I(B) \tag{17.6}$$

A relação  $A \preceq B$  significa que para todo ponto  $b_i$  pertencente a  $B$ , existe pelo menos um ponto  $a_j$  pertencente a  $A$  tal que  $a_j \preceq b_i$ . Já a relação  $B \not\preceq A$  significa que existe pelo menos um ponto  $a_j$  pertencente a  $A$  tal que nenhum ponto  $b_i$  pertencente a  $B$  satisfaz a relação  $b_i \preceq a_j$ . Caso  $A$  e  $B$  sejam

<sup>7</sup> Na literatura, esse indicador também é chamado de *métrica S*, (ou *S-metric*, em inglês), seguindo a nomenclatura originalmente empregada na referência em que o indicador foi proposto.

<sup>8</sup> Do inglês: *Pareto-compliance*.

relacionados dessa forma, então um indicador  $I(\cdot)$  Pareto-conforme deverá retornar valores escalares tais que  $I(A) > I(B)$ .

O indicador de hipervolume é a única métrica para avaliar conjuntos de estimativas do conjunto Pareto-ótimo hoje conhecida que possui a propriedade de Pareto-conformidade (Zitzler et al., 2007). Por esse motivo, ao longo dos últimos 10 anos, o indicador de hipervolume veio se tornando uma métrica relativamente consensual, quando se trata de avaliar o desempenho de algoritmos evolutivos de otimização multiobjetivo.

Dois tipos de questões relacionadas com o indicador de hipervolume têm merecido significativos esforços de pesquisa nos anos recentes. O primeiro diz respeito à dificuldade computacional para o cálculo desse indicador. Embora no caso de apenas duas dimensões esse cálculo seja bastante trivial, quando o número de dimensões cresce o problema se torna muito mais complexo. Têm sido feitos esforços tanto para a construção de algoritmos rápidos para o cálculo do indicador em dimensões elevadas (Beume et al., 2009; Emmerich e Fonseca, 2011; While et al., 2011) quanto para a obtenção de versões estocásticas ou heurísticas desse cálculo que aproximem o valor do indicador, sem calculá-lo de forma exata (Bringmann e Friedrich, 2009; Bader, 2009; Ishibuchi et al., 2010). O segundo tipo de questão que vem sendo estudado diz respeito à construção de algoritmos evolutivos que utilizem diretamente o indicador de hipervolume, como critério empregado no mecanismo de seleção (Zitzler e Kunzli, 2004; Emmerich et al., 2005; Bader, 2009). Diferentes estratégias podem ser empregadas para que, ao se substituírem algumas soluções de um conjunto por outras, o conjunto resultante tenha um indicador de hipervolume com um valor maior que o inicial. A fórmula mais simples seria a exclusão dos pontos com menor contribuição individual para o indicador, e a inclusão dos novos pontos que individualmente causem o maior incremento do mesmo. Mecanismos mais sofisticados, no entanto, procuram verificar a contribuição conjunta dos pontos, que é diferente da soma das contribuições individuais, para a escolha dos pontos a serem introduzidos ou excluídos.

### Indicação de Preferências Progressiva

Uma questão que se manifesta em parte significativa dos problemas de otimização multiobjetivo é a de que o custo computacional para a geração de um conjunto de soluções que represente a fronteira Pareto-ótima pode ser muito elevado, seja pelo fato de cada avaliação das funções objetivo ser muito elevado, seja pela necessidade de que seja gerado um número muito grande de soluções para que se constitua um conjunto representativo do conjunto Pareto-ótimo do problema.

Nessas situações, torna-se conveniente que a exploração do espaço de soluções, ao invés de conduzir a soluções que cubram todo o conjunto Pareto-ótimo, promova a geração de soluções que representem apenas um sub-conjunto do conjunto Pareto-ótimo, que inclua a solução que será finalmente a preferida pelo tomador de decisão, e mais algumas soluções na vizinhança desta. Esse procedimento, claramente, simplifica o problema computacional em questão, permitindo a geração de um número muito menor de soluções Pareto-ótimas, e requerendo que as soluções geradas sejam representativas desse conjunto em um sentido apenas local.

Para que isto se torne possível, é necessário que o algoritmo disponha da informação a respeito da estrutura de preferências do tomador de decisão (decisor). Frequentemente, essa estrutura de preferências não se encontra disponível a priori, antes que se inicie o processo de otimização. Nesse caso, para que a busca possa ser dirigida para a região onde se encontra a solução preferida pelo tomador de decisão, será necessário que o algoritmo de otimização promova alguma interação com esse tomador de decisão ao longo do processo de otimização. Isto é particularmente conveniente, muitas vezes, porque a informação produzida pela execução do algoritmo de otimização, com a avaliação de um certo número de soluções em regiões diferentes do espaço de variáveis de decisão, pode ser empregada como informação capaz de subsidiar o tomador de decisão. Ao aprender sobre a estrutura do problema, o tomador de decisão forma sua opinião, e pode assim indicar suas preferências. Isso

ocorre progressivamente porque ao longo do processo de otimização ocorrem várias interações com o tomador de decisão, à medida em que a população do algoritmo evolutivo se aproxima de uma determinada região do conjunto Pareto-ótimo, sendo guiada pelas preferências indicadas por este tomador de decisão.

Um expressivo número de trabalhos foi publicado, de poucos anos para cá, a respeito de esquemas de indicação progressiva de preferências em algoritmos evolutivos. A título de exemplo, citamos aqui os trabalhos de (Battiti e Passerini, 2010), de (Deb et al., 2010), de (Karahana e Koksalan, 2010) e de (Kim et al., 2011).

### Problemas com Muitos Objetivos

No início dos anos 2000, depois que os métodos evolutivos de otimização multiobjetivo passaram a ser utilizados em um número crescente de aplicações, uma dificuldade inicialmente não identificada passou a preocupar os pesquisadores da área: a perda de eficiência dos métodos evolutivos à medida em que crescia o número de objetivos a serem otimizados simultaneamente. Os problemas com tal dificuldade passaram a ser denominados *problemas com muitos objetivos*<sup>9</sup>.

A natureza geométrica dos problemas com muitos objetivos encontra-se ilustrada na figura 17.11. Nessa figura, a origem do sistema de coordenadas do espaço de objetivos encontra-se deslocada para

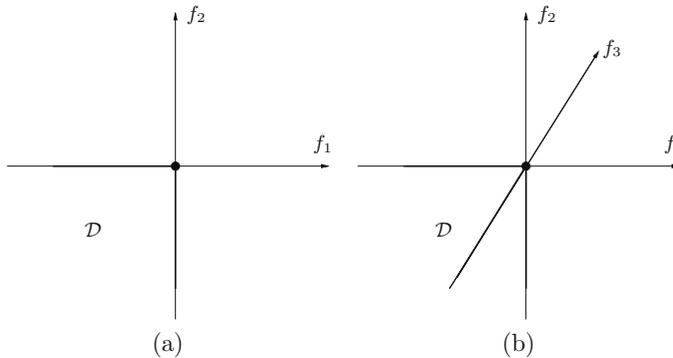


Figura 17.11: Geometria do problema com muitos objetivos. (a) Problema com dois objetivos. (b) Problema com três objetivos.

sobre um ponto-solução corrente, que irá sofrer uma mutação, tanto (a) para um caso com dois objetivos, como (b) para um caso com três objetivos. Em ambos os casos, a região indicada como  $\mathcal{D}$  corresponde à região dos pontos que dominam o ponto corrente. Assim, se a mutação levar a um novo ponto em  $\mathcal{D}$ , ela será bem sucedida em melhorar o ponto atual. Do contrário, não conduzirá a melhorias na solução atual. O aspecto relevante nesta análise é que, no caso de dois objetivos, a região  $\mathcal{D}$  corresponde a um quadrante do espaço (o quadrante de coordenadas negativas), ou seja, a uma fração de  $\frac{1}{4}$  das direções no espaço de objetivos. Já no caso de três objetivos,  $\mathcal{D}$  corresponde a um octante do espaço (o octante de coordenadas negativas), ou a uma fração de  $\frac{1}{8}$  das direções no espaço de objetivos. Para um número de objetivos  $m$ , a fração correspondente a  $\mathcal{D}$  será de  $\frac{1}{2^m}$ .

Embora o mapeamento entre o espaço de variáveis de decisão e o espaço de objetivos seja não-linear, o que significa que essas proporções não serão as mesmas no espaço de variáveis de decisão (no qual efetivamente será processada a mutação da solução corrente), esse efeito de rápida redução do volume da região na qual ocorre melhoria da solução, comparado ao volume da região em que não ocorre melhoria, se transpõe também para esse espaço. Como consequência, a probabilidade de uma mutação efetivada em uma direção inteiramente aleatória levar a uma melhoria de solução pode

<sup>9</sup> Do inglês: *many-objective problems*.

ser muito pequena, para problemas com um grande número de objetivos, ficando cada vez menor à medida em que aumenta a dimensão do espaço de objetivos. Isso explica a perda da eficiência dos algoritmos evolutivos quando o número de objetivos cresce.

Vários trabalhos publicados nos últimos anos têm procurado compreender, de maneira mais precisa, a natureza das dificuldades relacionadas com os problemas com muitos objetivos. Tem sido tentada a proposição de técnicas diferentes de seleção, de atribuição de aptidão, ou ainda de busca local, ou ainda técnicas de redução da dimensão do espaço de objetivos, para tratar esse tipo de problema. Citamos aqui as referências (Purshouse e Fleming, 2007; Adra e Fleming, 2011; Bader e Zitzler, 2011; Lopez-Jaimes et al., 2011; Pasia et al., 2011; Purshouse et al., 2011; Schutze et al., 2011), que em conjunto apresentam um panorama da questão.