

OTIMIZAÇÃO MULTI OBJETIVO COM ESTRATÉGIAS EVOLUTIVAS  
APLICADA A PROJETOS ESTRUTURAIS

Carlos Alberto Bardanachvili

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

---

Prof. Edison Castro Prates de Lima, D.Sc.

---

Prof. Gilberto Bruno Ellwanger, D.Sc.

---

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

---

Prof. Murilo Augusto Vaz, Ph.D.

---

Prof. Hélio José Correa Barbosa, D.Sc.

---

Dr. Carlos Eduardo Luz Riodades de Mendonça, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2006

BARDANACHVILI, CARLOS ALBERTO

Otimização Multiobjetivo com Estratégias  
Evolutivas Aplicada a Projetos Estruturais

[Rio de Janeiro] 2006

VIII, 104 p. 29.7 cm, (COPPE/UFRJ, D.Sc.,  
Engenharia Civil, 2006)

Tese - Universidade Federal do Rio de Janeiro,  
COPPE.

1. Otimização
2. Estratégias Evolutivas
3. Projetos Estruturais

I. COPPE/UFRJ

II. Título (Série)

À minha esposa

***Dalila***

Aos meus filhos

***Júlia e Henrique***

Aos meus pais

***Ezra (in memorian) e Lili***

## AGRADECIMENTOS

Ao professor Edison C. Prates de Lima pela orientação decisiva, em especial na definição do escopo do trabalho e nos ensinamentos transmitidos, e por manter a capacidade de empolgação.

Aos professores da COPPE/UFRJ, em especial aos professores Gilberto B. Ellwanger, Luis V. S. Sagrilo e José Luis D. Alves, pelas discussões sobre o tema e pela preocupação com a concretização do presente trabalho.

À minha esposa, Dalila, pelo carinho, compreensão e por termos vivenciado e concluído juntos mais esse longo desafio.

Aos meus filhos, Júlia e Henrique, que aguardaram ansiosos pelo retorno à “normalidade”.

Aos meus pais, Ezra Bardanachvili (*in memoriam*) e Lili Bardanachvili, e às minhas irmãs, Eliane e Rosane, por terem sempre me incentivado e acompanhado em todas as etapas de minha formação pessoal e profissional.

Aos amigos no CENPES e na PETROBRAS, em especial na gerência de Estruturas Oceânicas, pelo incentivo constante; a Paulo Maurício Videiro, pelo estímulo; a Luiz Henrique M. Alves, Glauco D. Ribeiro, Cláudia A. Claro, Ângelo A. Colafemea, Lena Lerner, Celso Ayabe, Daniel Saito, José Tertuliano S. Filho e Antônio G. V. Neto, pela cobertura durante minha ausência; a Vitor G. Maciel, pelos ensinamentos; a Carlos Eduardo Luz R. Mendonça, pelas discussões sobre o tema; a Salvador S. Filho e Djalene M. Rocha, pelo incentivo mútuo.

À PETROBRAS, pela oportunidade e pelas facilidades oferecidas.

A todos que de alguma forma contribuíram para a conclusão deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

OTIMIZAÇÃO MULTI OBJETIVO COM ESTRATÉGIAS EVOLUTIVAS  
APLICADA A PROJETOS ESTRUTURAIS

Carlos Alberto Bardanachvili

Setembro/2006

Orientadores: Edison Castro Prates de Lima

Gilberto Bruno Ellwanger

Programa: Engenharia Civil

Neste trabalho, procurou-se desenvolver um dos mais simples algoritmos não-triviais de otimização multiobjetivo capaz de gerar um conjunto diversificado de soluções do fronte de Pareto. A metodologia proposta é baseada no método (1+1)-PAES (Estratégia Evolutiva com Pareto Arquivado), que gera um arquivo de referência para armazenar as soluções previamente encontradas do fronte. Este arquivo é utilizado para estabelecer o *ranking* de dominância aproximado entre a solução atual e a candidata gerada. No procedimento proposto (1+1)-PAES-M, foram introduzidas as seguintes modificações: foi desenvolvido um algoritmo de Estratégia Evolutiva (1+1)-ES de pesquisa global, através da utilização de dois níveis de desvio padrão para mutação: um global e um local; o algoritmo (1+1)-ES realiza simulações tanto individuais como conjuntas das variáveis do problema para acelerar a convergência; ele permite que as restrições sejam tratadas de forma *hard* ou *soft* sem o emprego de funções de penalização; a diversidade das soluções ao longo do fronte de Pareto é obtida através de uma métrica, baseada na distância de aglomeração. O algoritmo foi testado para várias funções de referência, alcançando níveis de precisão, robustez e eficiência comparáveis aos melhores algoritmos GA e ES encontrados na literatura. Para contemplar os casos de problemas de otimização estrutural para os quais a obtenção dos valores das funções objetivas e restritivas demandem longo tempo de processamento, foram estudadas algumas técnicas de interpolação multidimensional de resposta para trabalhar com malhas adaptativas a partir de poucos pontos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

MULTIOBJECTIVE OPTIMIZATION USING EVOLUTION STRATEGIES  
APPLIED TO STRUCTURAL DESIGN

Carlos Alberto Bardanachvili

September/2006

Advisors: Edison Castro Prates de Lima

Gilberto Bruno Ellwanger

Department: Civil Engineering

In this work, a procedure to address multiobjective optimization problems related to structural design is presented. The proposed approach is aimed to be the simplest possible non-trivial algorithm capable of generating diverse solutions in the Pareto optimal set. It is based on (1+1)-PAES method (Pareto Archived Evolution Strategy), using an archive of previously found solutions in order to identify the approximate dominance ranking of the current and candidate solution vectors. In relationship with the original (1+1)-PAES, the proposed algorithm (1+1)-PAES-M was modified in the following main aspects: it was developed a global search Evolution Strategy (1+1)-ES algorithm using two levels of mutation strength: one global and one local; the (1+1)-ES allows both individual and simultaneous simulations of the optimization variables to improve the convergence; the optimization problem constraints can be treated as hard or soft and does not require the using of penalty functions; the solution spread of the Pareto-optimum front is achieved with a crowding tournament selection operator using a crowding distance metric. The results obtained are very competitive when comparing the (1+1)-PAES-M against other state-of-the-art GA and ES algorithms found in the references. In order to apply the proposed approach to cases of structural optimization problems, demanding large computer processing time, the objective functions and the constraints may be evaluated by multidimensional interpolation techniques.

## ÍNDICE

1.	Introdução.....	1
2.	Otimização Multiobjetivo.....	7
2.1.	Introdução.....	7
2.2.	Definição do Problema de Otimização Multiobjetivo.....	8
2.3.	Métodos de Abordagem.....	13
2.3.1.	Métodos Clássicos.....	16
3.	Otimização com Algoritmos Evolutivos.....	20
3.1.	Algoritmos Genéticos (GA).....	22
3.2.	Estratégias Evolutivas (ES).....	23
3.2.1.	Tratamento de problemas com restrições.....	28
4.	Otimização Multiobjetivo com Algoritmos Evolutivos.....	32
4.1.	PAES (Estratégia Evolutiva com Pareto-Arquivado).....	33
4.2.	Outros Algoritmos Multiobjetivo.....	38
5.	Testes de Desempenho do Otimizador ( <i>Bench-mark</i> ).....	40
5.1.	Medidas de Desempenho.....	40
5.2.	Problemas ZDT.....	44
5.2.1.	Problema ZDT1.....	45
5.2.2.	Problema ZDT2.....	48
5.2.3.	Problema ZDT3.....	50
5.2.4.	Problema ZDT4.....	51
5.2.5.	Problema ZDT6.....	53
5.3.	Problemas Multiobjetivo com Restrições.....	54
5.3.1.	Problema TNK.....	54
5.3.2.	Problema MEC ( <i>Constr-Ex</i> ).....	56
5.4.	Problemas Uniobjetivo com Restrições.....	57
5.4.1.	Problema g01.....	58
5.4.2.	Problema g02.....	60
5.4.3.	Problema g03.....	62
5.4.4.	Problema g06.....	63
5.4.5.	Problema g08.....	65
5.4.6.	Problema g09.....	66
5.4.7.	Problema g11.....	67

5.4.8. Problema g13 .....	68
6. Interpolação das respostas .....	70
6.1. Interpolação Linear .....	71
6.2. Regressão Linear .....	72
6.3. Considerações sobre a Interpolação .....	77
7. Aplicações em Estruturas .....	79
7.1. Otimização de seção de vigas .....	80
7.2. Cortes para alívio de tensões .....	85
8. Conclusões.....	95
8.1. Algoritmo de Otimização .....	95
8.2. Outras conclusões .....	97
8.3. Sugestões de Trabalhos Futuros .....	99
9. Referências Bibliográficas.....	101

## 1. Introdução

A principal motivação para pesquisar e fazer aplicações envolvendo problemas de otimização provém da minha experiência diária com projetos de estruturas marítimas de produção de petróleo. A grande variedade de concepções estruturais alternativas, de situações e de níveis de detalhe a que se tem que chegar, mostra claramente a necessidade e a conveniência da utilização de ferramentas de otimização para auxiliar o projetista na busca de soluções de projeto que utilizem os recursos da melhor maneira possível.

Neste trabalho, procurou-se desenvolver um dos mais simples algoritmos não-triviais de otimização multiobjetivo capaz de gerar um conjunto suficientemente diversificado de soluções do fronte de Pareto com um nível de precisão, robustez e eficiência comparáveis aos melhores algoritmos GA e ES encontrados na literatura.

O procedimento proposto baseia-se no método (1+1)-PAES (Estratégia Evolutiva com Pareto Arquivado) (Knowles e Corne, 1999), que gera um arquivo de referência para armazenar as soluções previamente encontradas do fronte de Pareto. Este arquivo é utilizado para estabelecer o *ranking* de dominância aproximado entre a solução atual e a solução candidata gerada. No procedimento proposto, denominado (1+1)-PAES-M (Estratégia Evolutiva com Pareto Arquivado, Modificado), foram introduzidas no método (1+1)-PAES original as seguintes modificações principais:

- i. Foi desenvolvido um algoritmo de Estratégia Evolutiva (1+1)-ES de pesquisa global, através da utilização de dois níveis de desvio padrão para mutação: um global e um local;
- ii. O algoritmo (1+1)-ES realiza tanto simulações individuais como simulações conjuntas das variáveis do problema para acelerar a convergência;
- iii. O algoritmo (1+1)-ES permite que as restrições sejam tratadas de forma *hard* ou *soft* sem o emprego de funções de penalização;
- iv. A avaliação da diversidade das soluções ao longo do fronte de Pareto é obtida através de uma métrica baseada na distância de aglomeração (*crowding distance metric*) (Deb, 2001).

Portanto, para a definição do escopo do trabalho, foi necessário, inicialmente, responder as seguintes perguntas:

- 1) Por que foi escolhido um algoritmo baseado em Estratégias Evolutivas ES?
- 2) Por que foi utilizada a versão mais simples (1+1)-ES do algoritmo de ES?
- 3) Por que foi utilizada a versão mais simples (1+1)-PAES do algoritmo de otimização multiobjetivo?

A escolha de um algoritmo do tipo Estratégias Evolutivas ES foi motivada pelo fato de que, conforme mencionado em Mezura-Montes e Coello (2005), os mais recentes e competitivos algoritmos para resolver problemas de otimização global não-linear com um grande número de restrições, tanto do tipo inequação como do tipo igualdade, são baseados em ES, como por exemplo: *Stochastic Ranking (SR)* (Runarsson e Yao, 2000), *Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA)* (Hamida e Schoenauer, 2002) e *A Simple Multimembered Evolution Strategy to Solve Constraint Optimization Problems (SMES)* (Mezura-Montes e Coello, 2005).

Uma das maiores críticas ao algoritmo de Estratégias Evolutivas (1+1)-ES baseia-se no fato de que ele é considerado um algoritmo com **estratégia de busca local**, uma vez que utiliza somente a operação de mutação num único parente (no sentido de solução parental, genitora, matriz) para criar uma única prole.

Porém, observa-se que o caráter de busca local pode ser expandido pela utilização de dois níveis de desvio padrão para mutação: um global e um local. Também é possível utilizar eventualmente como parentes, dentro do processo de simulação, as soluções armazenadas no arquivo de referência que estejam situadas nas regiões mais rarefeitas do fronte de Pareto. Desta forma, obtém-se um algoritmo de Estratégias Evolutivas (1+1)-ES com características mais próximas de uma **estratégia de busca global**.

O algoritmo (1+1)-ES é utilizado para gerar as soluções candidatas a participar do fronte de Pareto, através do procedimento proposto (1+1)-PAES-M.

Os motivos da escolha de um algoritmo de otimização multiobjetivo do tipo (1+1)-PAES são não somente a sua simplicidade computacional e a sua eficiência, quando comparado com outros algoritmos GA e ES do tipo  $(\mu + \lambda)$ , mas, principalmente, a sua confiabilidade. Parece que o algoritmo do tipo (1+1)-PAES funciona bem devido aos mesmos motivos pelos quais ele é um algoritmo computacionalmente simples: é um algoritmo agressivo, testando cada solução candidata de uma forma direta e não gastando muitos recursos nas soluções candidatas que não passaram no teste.

O principal inconveniente do algoritmo (1+1)-PAES, segundo seus idealizadores Knowles e Corne (1999), é que ele pode ficar trancado num ponto de máximo (mínimo) local, quando as mutações produzidas no parente não forem suficientemente grandes para atravessar o espaço existente entre o ponto de máximo global e os diversos pontos de máximos locais.

Este inconveniente foi contornado no algoritmo proposto (1+1)-PAES-M pela utilização do algoritmo de Estratégias Evolutivas (1+1)-ES com características de busca global.

O algoritmo (1+1)-PAES-M foi testado para várias funções de referência, alcançando um nível de precisão, robustez e eficiência comparáveis aos melhores algoritmos GA e ES encontrados na literatura.

Finalmente, para contemplar os casos de problemas de otimização estrutural para os quais a obtenção dos valores das funções objetivas e restritivas demandem longo tempo de processamento, foram estudadas e implementadas algumas técnicas de interpolação de resposta no espaço multidimensional das variáveis do problema, com o objetivo principal de se trabalhar com uma malha adaptativa a partir de poucos pontos.

Uma outra alternativa seria tratar essas variáveis de forma discreta, de modo que o algoritmo possa ser utilizado como um controlador (*Shell*). Neste caso, os valores das funções objetivas e restritivas do problema de otimização são obtidos a partir da resposta de programas de computador de análises estruturais que são chamados de

dentro da *Shell* pelo programa controlador, para cada conjunto simulado de valores das variáveis do problema de otimização.

Os problemas de otimização estrutural, em geral, usam um programa de elementos finitos para avaliação da resposta estrutural. Há, hoje em dia, muitos programas no mercado com a facilidade de serem implementados sob programas de otimização do tipo *Shell*. Em princípio, poder-se-ia deixar o controlador chamar as rodadas por um tempo considerável na busca das soluções. Entretanto, deve-se avaliar cuidadosamente as vantagens e desvantagens de se deixar o programa de análise controlado ou independente do programa de otimização.

Um motivo para mantê-los independentes é que pode não ser o mais adequado deixar para o otimizador estabelecer a priori um número grande de rodadas a partir de funções objetivas e de restrições estabelecidas. A construção do problema de otimização através dessas funções geralmente progride com a interferência do projetista, com sua experiência e capacidade de discernimento, a partir dos resultados de algumas rodadas iniciais. Isso ficou evidenciado nos exemplos práticos considerados apesar de serem aparentemente simples.

Vale observar que, nos problemas multiobjetivo, geralmente, devido à situação de compromisso entre as diversas funções objetivas, a região de interesse do fronte de Pareto não coincide com os pontos de máximo (mínimo) globais de cada uma das funções objetivas consideradas separadamente e pode vir a ser mais facilmente interpolável.

Por outro lado, a automação nas análises e controle de interfaces entre programas é desejável, para se evitarem erros de modelagem e na passagem de resultados, especialmente na fase de projeto. Os prazos dos projetos na indústria *offshore* têm sido bastante reduzidos nas avaliações econômicas, devido ao fato de que cada mês de atraso na entrada em produção de uma unidade representa uma perda significativa de receita. Nesse sentido, o olhar do projetista experiente é importante na condução de um processo de otimização, uma vez que na área *offshore*, é comum encontrarem-se situações bem específicas (*tailor made*).

A seguir apresenta-se a organização do texto do presente trabalho:

No Capítulo 2, serão apresentados a formulação geral do problema de otimização multiobjetivo, as definições dos termos principais utilizados e o resumo de alguns métodos clássicos de otimização multiobjetivo.

No Capítulo 3, serão abordados os princípios básicos do problema de otimização com Algoritmos Evolutivos.

No Capítulo 4, serão destacadas as características do algoritmo de otimização multiobjetivo proposto (1+1)-PAES-M (Estratégia Evolutiva com Pareto Arquivado, Modificado), enfatizando as características originais do mesmo.

No Capítulo 5, serão apresentadas aplicações do procedimento de otimização multiobjetivo (1+1)-PAES-M proposto numa série de problemas do tipo *bench-mark* de otimização com e sem restrições, para demonstrar a robustez e a qualidade do mesmo na determinação das soluções do fronte de Pareto.

No Capítulo 6, será vista a questão da interpolação de funções no espaço multidimensional. Serão abordadas as dificuldades e os cuidados que devem ser tomados para garantir uma boa representação das funções objetivas e restritivas que são definidas por pontos nos problemas práticos.

No Capítulo 7, serão apresentadas duas aplicações simples. A primeira aplicação refere-se a um exemplo clássico de otimização de seção transversal de vigas com perfil I.

A segunda aplicação está relacionada à retirada de material em quinias, de modo a suavizar sua forma e promover o alívio de tensões. Esse conceito de solução tem sido empregado há algumas décadas em estruturas marítimas, sujeitas a cargas cíclicas, que podem provocar fadiga em regiões críticas da estrutura. Essas formas têm sido escolhidas por processo de otimização.

Finalmente, no capítulo das conclusões, será consolidado o procedimento sugerido e feito um resumo das principais conclusões obtidas durante o trabalho. Também serão apresentadas propostas de desenvolvimentos futuros e novas aplicações que poderão dar seqüência ao estudo, de forma a permitir a solução de problemas que mais se aproximem dos casos práticos da Engenharia.

## 2. Otimização Multiobjetivo

### 2.1. Introdução

Os processos de otimização, em geral, se referem à busca das melhores soluções, do ponto de vista dos objetivos associados a um problema, que sejam viáveis em função de restrições impostas ou inerentes a ele. Os objetivos podem ser traduzidos através de funções objetivas. Em projetos de engenharia, os objetivos podem ser, por exemplo, obter o menor custo (fabricação, transporte, montagem), menor prazo, menor peso, maior confiabilidade, menores tensões ao longo de uma estrutura, menores deslocamentos.

Na sua formulação, o problema de otimização pode ser representado por uma única função objetiva, SOOP, (*single-objective optimization problem*) ou por várias funções objetivas, MOOP, (*multi-objective optimization problem*). Para um SOOP, o objetivo do processo de otimização é encontrar a solução viável (*feasible*) que represente o mínimo ou máximo global da função objetiva dentro do domínio permitido pelas restrições impostas ao problema, que pode ser um mínimo local dentro do domínio permitido ou uma solução na fronteira do intervalo. Eventualmente, pode ser interessante investigar a existência de outros picos locais secundários, por exemplo, no caso de respostas muito sensíveis a possíveis variações (imprecisões) nas variáveis de projeto.

Já em um problema de otimização multiobjetivo (MOOP), no caso geral em que pode haver objetivos conflitantes, uma solução pode ser a melhor do ponto de vista de um objetivo, mas não em relação aos demais objetivos. Na Figura 1, é mostrada uma região de objetivos conflitantes, situada entre os pontos de mínimo de cada função. Portanto, não se conseguirá uma única solução e sim um conjunto de soluções viáveis ótimas. Desse modo, para se adotar uma determinada solução ótima viável, será necessário recorrer a informações adicionais que irão contribuir para caracterizar a escolha da solução de compromisso. Essas informações são denominadas de informações de mais alto nível ou subjetivas, podendo ser, inclusive, informações puramente qualitativas ou baseadas na experiência.

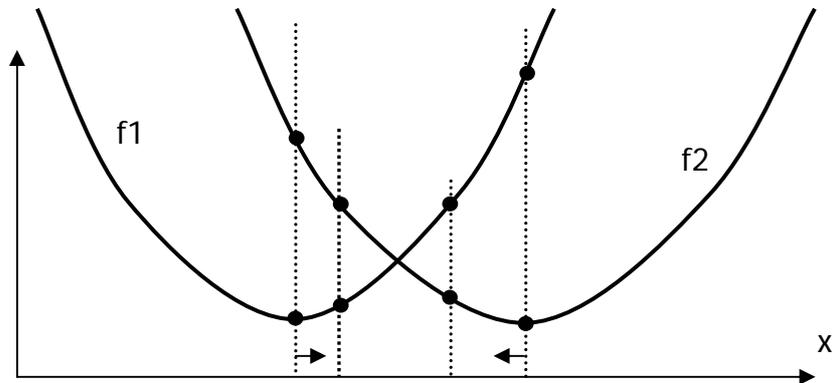


Figura 1 - Objetivos conflitantes das funções  $f_1$  e  $f_2$  – região central, entre mínimos.

## 2.2. Definição do Problema de Otimização Multiobjetivo

A formulação geral do problema de otimização multiobjetivo envolve a minimização ou maximização de certo número de funções objetivas das variáveis de projeto, que, por sua vez, para serem viáveis, os seus valores devem ser tais que atendam às restrições impostas. Como a maximização pode ser transformada numa minimização, bastando multiplicar a função por (-1) (princípio da dualidade), só se trabalhará aqui com minimização. A forma geral o problema de otimização multiobjetivo pode ser apresentada como na expressão (2-1) (Deb, 2001):

$$\begin{aligned}
 &\text{Minimizar: } f_m(\mathbf{x}), && m = 1, 2, \dots, M \\
 &\text{sujeito a: } g_j(\mathbf{x}) \leq 0, && j = 1, 2, \dots, J \\
 & && h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K \\
 & && x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, ND
 \end{aligned} \tag{2-1}$$

Uma solução é um vetor de ND variáveis de projeto  $\mathbf{x} = (x_1, x_2, \dots, x_{ND})$ . O vetor  $\mathbf{x}$  é denominado de viável (*feasible*) se ele atender a todas as restrições em (2-1), caso contrário ele será considerado inviável (*infeasible*). O terceiro conjunto de restrições representa a delimitação do intervalo de validade das variáveis de projeto, formando o domínio ou espaço de decisão dessas variáveis. As expressões para as restrições devem

ser colocadas na forma das expressões para  $g_j(\mathbf{x})$  e  $h_k(\mathbf{x})$ . As diferentes funções objetivas formam um espaço e, portanto, pode ser feito o mapeamento das soluções do espaço ND-dimensional das variáveis de projeto para soluções no espaço M-dimensional das funções objetivas, como mostrado na Figura 2 para espaços bidimensionais.

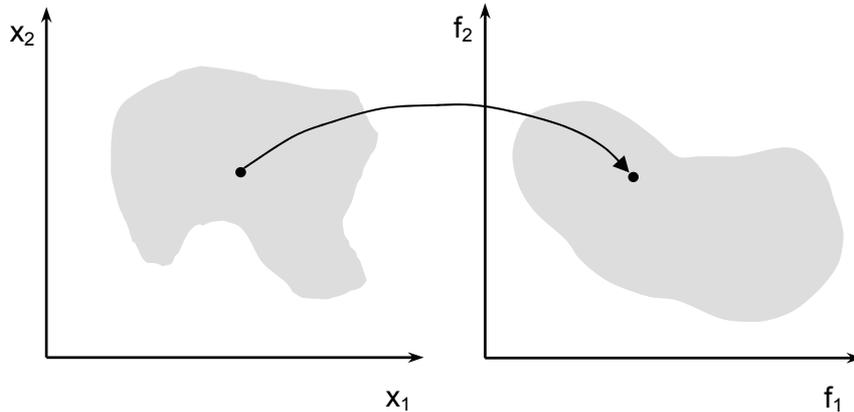


Figura 2 - Mapeamento de solução do espaço das variáveis de projeto para o espaço das funções objetivas.

Caso todas as funções objetivas e restrições sejam lineares, trata-se de um problema de otimização multiobjetivo linear, com características próprias de solução. Caso pelo menos uma dessas funções seja não-linear, o problema passa a ser não-linear, que é o caso geral.

Um problema de otimização multiobjetivo é dito convexo quando todas as funções objetivas são convexas e o espaço de soluções viáveis é convexo. Uma função  $f(\mathbf{x})$  de  $\mathbb{R}^n \rightarrow \mathbb{R}$  é dita convexa quando, para quaisquer pares de soluções  $\mathbf{x}_1$  e  $\mathbf{x}_2$ :

$$f(\lambda \mathbf{x}^{(1)} + (1-\lambda) \mathbf{x}^{(2)}) \leq \lambda f(\mathbf{x}^{(1)}) + (1-\lambda) f(\mathbf{x}^{(2)}), \text{ para qualquer } 0 \leq \lambda \leq 1 \quad (2-2)$$

Isso quer dizer que a aproximação linear de  $f(\mathbf{x})$  superestima uma função convexa no intervalo  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ . Se a função atender à inequação (2-2) com sinal  $>$  em vez de  $\leq$  ela é dita não-convexa. A matriz hessiana  $\nabla^2 f$ , se for positiva definida em todos os pontos de um intervalo de  $n$  variáveis, indica que a função é convexa nesse intervalo, enquanto que se  $-\nabla^2 f$  o for, é dita não-convexa.

Um espaço de soluções com  $ND$  variáveis de projeto, ou de  $M$  funções objetivas ou de  $J$  e  $K$  funções restritivas é convexo numa região, se, ao ligarmos quaisquer 2 pontos da região por um segmento de reta, todos os pontos do segmento estiverem dentro da região; do contrário, o espaço é dito não-convexo.

A relevância desse conceito se tornará evidente ao se descreverem os métodos de busca de soluções, quando se verá a importância de não se perderem regiões do espaço de funções e, por conseguinte, as soluções com as variáveis de projeto que levariam a elas.

A relação entre as soluções mapeadas no espaço de funções objetivas irá guiar a escolha das situações desejáveis ou aceitáveis como soluções de projeto. Essa relação é traduzida pela comparação entre cada duas soluções mapeadas levando-se em conta cada função objetiva.

### **Conceito de Dominação**

Quando uma solução  $\mathbf{x}^{(1)}$  é melhor ou igual a outra solução  $\mathbf{x}^{(2)}$  em relação a todas as funções objetivas,  $f(\mathbf{x}^{(1)}) \leq f(\mathbf{x}^{(2)})$ , e estritamente melhor em relação a pelo menos uma das funções objetivas,  $f(\mathbf{x}^{(1)}) < f(\mathbf{x}^{(2)})$ , diz-se que a solução  $\mathbf{x}^{(1)}$  **domina** a solução  $\mathbf{x}^{(2)}$  ( $\mathbf{x}^{(1)} \prec= \mathbf{x}^{(2)}$ ).

Caso a solução  $\mathbf{x}^{(1)}$  seja estritamente melhor que a solução  $\mathbf{x}^{(2)}$  para todas as funções objetivas, então a relação é dita de **forte dominância** ( $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$ ).

Quando uma solução é estritamente melhor em relação a pelo menos uma das funções objetivas, mas é estritamente pior com relação a pelo menos uma das demais funções objetivas, diz-se que elas são soluções **não-dominadas**.

Pelas definições anteriores, na Figura 3, supondo-se que se trata de um problema de minimização, a solução  $a$  domina a solução  $b$  ( $a \prec= b$ ) e também a solução  $c$  ( $a \prec= c$ ), enquanto que as soluções  $a$  e  $d$  são não-dominadas entre si. Pode-se afirmar que as

soluções  $a$ ,  $d$  e  $e$  formam um conjunto de soluções não-dominadas que domina o conjunto formado pelas soluções não-dominadas  $b$  e  $c$ . Isso significa que um elemento de  $\{b, c\}$  encontrará pelo menos um elemento de  $\{a, d, e\}$  que o domine.

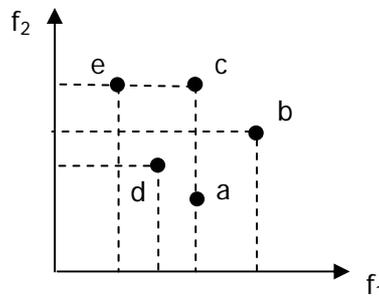


Figura 3 - Ilustração de relação de dominação entre soluções. Exemplos:  $a \prec b$ ;  $a \prec c$ ;  $\{a, d, e\}$ , conjunto de soluções não-dominadas.

A relação de dominação apresenta a propriedade de ser transitiva, ou seja, para soluções  $p$ ,  $q$  e  $r$ , se  $p \prec q$  e  $q \prec r$ , então  $p \prec r$  (ou seja, se  $p$  domina  $q$  e  $q$  domina  $r$ , então  $p$  domina  $r$ ). Não apresentam, por outro lado, propriedades reflexiva, de simetria nem de anti-simetria. Outra propriedade é que se  $p$  não domina  $q$ , não quer dizer que  $q$  domine  $p$ , pois podem ser soluções não-dominadas.

De um conjunto de soluções  $P$ , o conjunto de soluções não-dominadas  $P'$  é aquele tal que seus elementos são não-dominados por qualquer elemento do conjunto  $P$ . Então, quaisquer 2 soluções de  $P'$  são não-dominadas entre si e qualquer solução das demais do conjunto  $P$  são dominadas por pelo menos um elemento de  $P'$ . Se o conjunto  $P$  for o próprio espaço de busca  $S$ , então esse conjunto  $P'$  é chamado de conjunto de soluções **Pareto-ótimas**. Trabalhando somente com funções a serem minimizadas, as soluções Pareto-ótimas tomam o aspecto da fronteira como na Figura 4.

Deduz-se, então, que o objetivo do processo de otimização multiobjetivo será obter os elementos da fronteira de Pareto, já que qualquer solução fora dela encontraria uma solução melhor sobre ela. Assim, o espaço de soluções se divide em soluções ótimas e não-ótimas.

Sobre a fronteira de Pareto, não há, em princípio, preferência por nenhuma das soluções, sendo necessário se acrescentarem **informações subjetivas de alto nível**, ou seja, expressões que contemplem relações entre as funções objetivas para se escolher uma determinada solução da fronteira de Pareto.

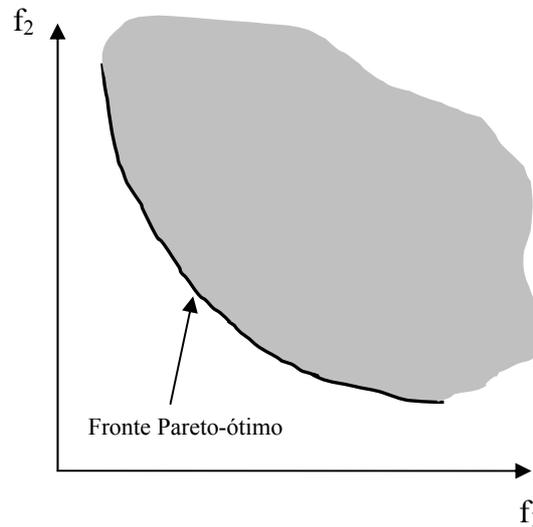


Figura 4 - Fronte Pareto-ótimo

Os pontos no espaço de funções objetivas são normalmente obtidos, nos casos práticos reais, através de análises individuais por modelagem em computador. Então, o que se deve procurar encontrar é um número adequado de pontos o mais próximos possível da fronteira de Pareto. Isso vale também no caso em que as respostas serão interpoladas por funções, porque a fronteira será obtida por simulação das variáveis de projeto dentro de intervalos.

Com um número finito de pontos na fronteira de Pareto, é interessante que eles sejam relativamente bem espaçados, de forma a representá-la bem e permitir que a solução de compromisso, através da relação subjetiva mencionada, seja feita com dados suficientes.

Entretanto, a fronteira obtida está associada à formulação que foi escolhida para o problema de otimização. Algumas vezes pode ocorrer que, ao analisar as soluções obtidas, seja necessário reformular o problema devido à ocorrência de algum aspecto que não foi levado em conta. Pode não ser evidente decidir inicialmente se o mais

conveniente é que determinado aspecto do problema seja tratado como uma função objetiva ou como uma restrição.

Um exemplo seria o caso de decidir se, num problema de otimização estrutural, as deflexões, tensões ou a probabilidade de falha devem ser minimizadas, junto com as demais funções objetivas, tais como o peso da estrutura, ou se bastaria que atendessem a requisitos de norma e fossem representadas por restrições.

### 2.3. Métodos de Abordagem

Um procedimento aparentemente simples seria utilizar as informações subjetivas de alto nível para formar uma única função objetiva, composta, por exemplo, pela soma ponderada das funções objetivas do problema de otimização multiobjetivo original, onde cada peso seria proporcional ao nível de preferência atribuído à respectiva função objetiva. O problema de otimização multiobjetivo seria então reduzido a um problema de otimização com um único objetivo e a abordagem seguiria a seqüência a seguir:

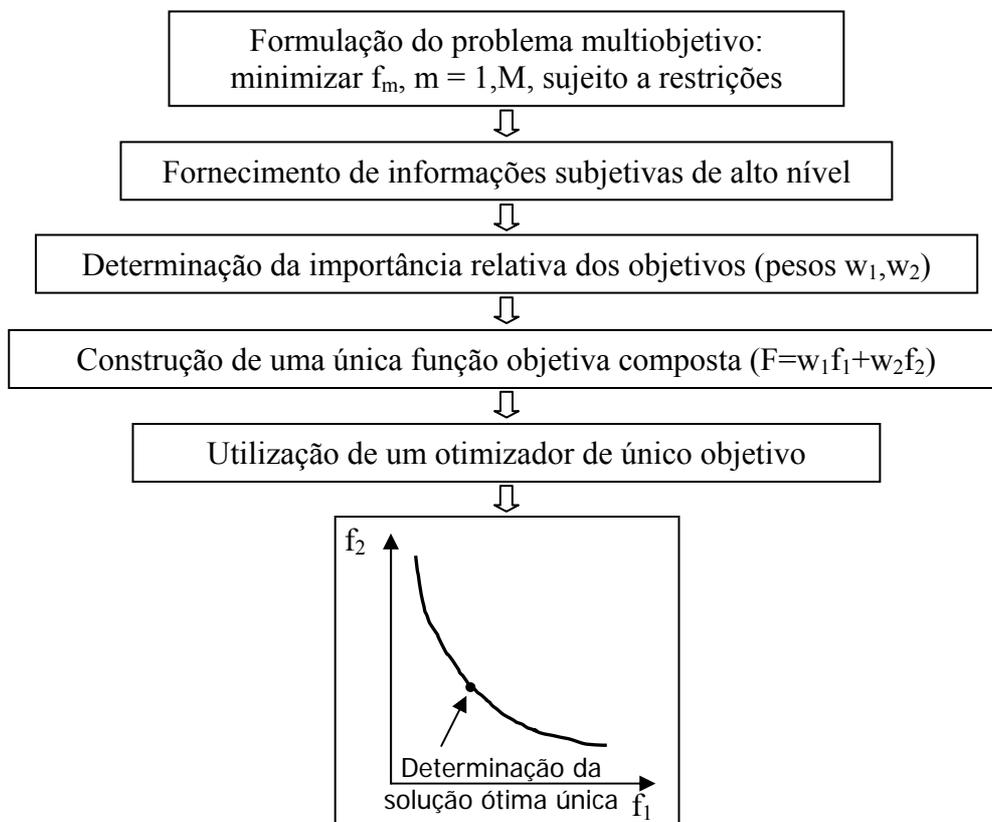


Figura 5 - Seqüência para abordagem do problema de otimização por método clássico.

Esse método, que utiliza informações subjetivas de alto nível, é a base dos chamados **métodos clássicos**, que permitem a utilização de um otimizador para problemas de um único objetivo. Neste caso, encontram-se disponíveis vários algoritmos que podem ser baseados em métodos numéricos tradicionais, como o gradiente conjugado e métodos do tipo *simulated annealing* (Press, 1986), ou métodos baseados em algoritmos evolutivos (Michalewicz, 1996, Bäck e Michalewicz, 1997, Corne et al., 1999)

Cabe ressaltar que, a priori, é geralmente bastante difícil estabelecer o nível relativo de preferência entre os diversos objetivos. Esta informação geralmente é altamente subjetiva, requerendo a consideração de aspectos não-técnicos, qualitativos e econômicos de mais alto nível. As funções objetivas referem-se, no caso geral, a grandezas incomensuráveis e será necessário estabelecer um meio de promover uma equalização entre as funções. Esta dificuldade fica agravada pelo fato de não ser conhecida ainda a correspondência entre os pontos do domínio das funções objetivas e os pontos do domínio das variáveis do problema de otimização.

Deste modo, será geralmente necessário realizar mais de um processo de otimização variando-se os pesos atribuídos às funções objetivas, na esperança de obter mais informações sobre o problema que está sendo analisado. Porém, observa-se que os resultados do problema de otimização podem ser altamente sensíveis aos valores dos pesos.

Outro aspecto dos métodos clássicos é que, dependendo do procedimento empregado para formar uma única função objetiva composta, muitas regiões do espaço das funções objetivas poderão ficar inacessíveis para qualquer combinação possível dos valores dos pesos atribuídos às funções objetivas. Isto pode ocorrer nos chamados problemas não-convexos, como será abordado mais adiante.

Finalmente, observa-se que o resultado de um problema de otimização com um único objetivo corresponde ao ponto ótimo global. Como o problema pode apresentar também vários pontos de ótimo locais, o otimizador deve ser robusto o suficiente para captar o ponto ótimo global e também os demais pontos de ótimo locais, que podem vir a despertar o interesse do projetista.

Para evitar as dificuldades apresentadas pelos métodos clássicos, foi desenvolvida uma outra forma de abordagem dos problemas de otimização multiobjetivo, denominada de **métodos diretos**. Neste caso, um procedimento considerado ideal para abordagem do problema de otimização multiobjetivo seria a obtenção direta do fronte Pareto-ótimo, conforme a seqüência a seguir.

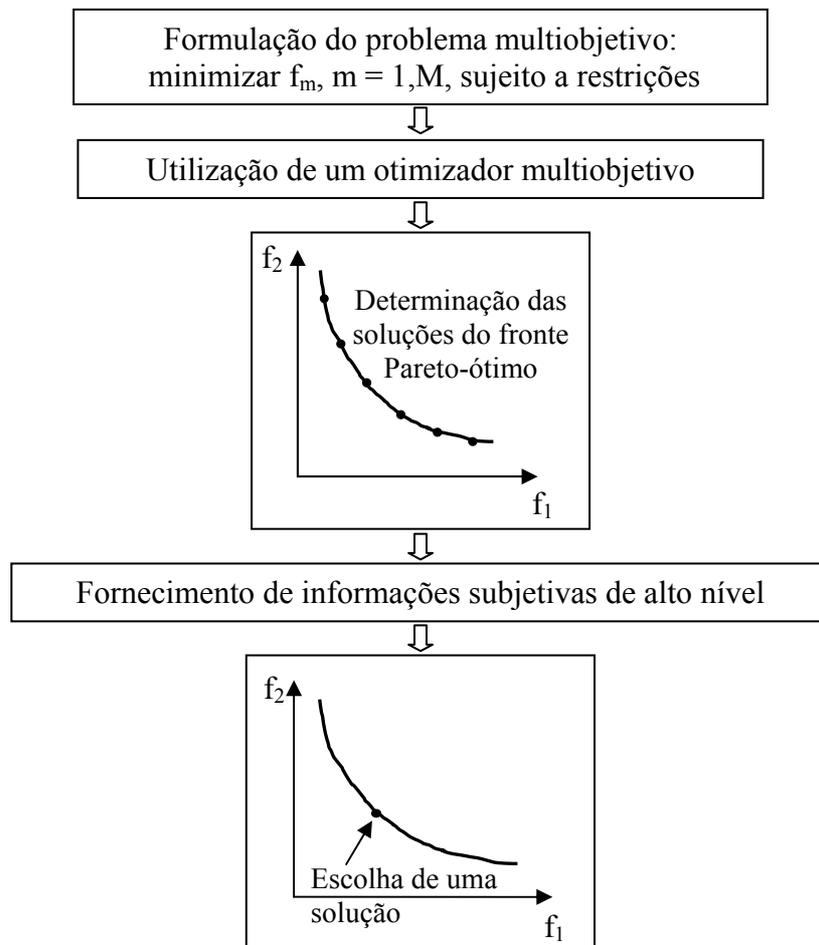


Figura 6 - Seqüência para abordagem do problema de otimização por método direto.

Os métodos baseados nesta seqüência requerem a utilização de um otimizador multiobjetivo.

Observa-se que a obtenção, a priori, das soluções do fronte Pareto-ótimo permite que as informações subjetivas de alto nível sejam aplicadas na etapa final do processo, quando já estiver disponível o universo de soluções possíveis e definida a correspondência entre os pontos do domínio das funções objetivas e os pontos do domínio das variáveis do problema de otimização.

Neste caso, a escolha de uma solução fica bastante mais fácil, quando comparada com os métodos clássicos.

### 2.3.1. Métodos Clássicos

Como já foi mencionado anteriormente, os métodos clássicos transformam o problema multiobjetivo em um problema uniobjetivo, incorporando informações subjetivas adicionais de mais alto nível.

Um dos métodos clássicos mais simples é o **Método da Soma Ponderada** (Deb, 2001), que utiliza as informações subjetivas de mais alto nível para formar uma única função objetiva composta pela soma ponderada das funções objetivas do problema de otimização multiobjetivo original. O valor de cada peso é feito proporcional ao nível de preferência atribuído à respectiva função objetiva. O problema de otimização multiobjetivo fica então reduzido a um problema de otimização com um único objetivo (Michalewicz, 1992). As soluções Pareto-ótimas são obtidas para cada conjunto de pesos. A função objetiva unificada é expressa então por:

$$F(\mathbf{X}) = \sum_{i=1}^m w_i f_i(\mathbf{X}) \quad (2-3)$$

$$\text{com } \sum_{i=1}^m w_i = 1 \quad (2-4)$$

A escolha da solução final poderá ser feita após uma análise de sensibilidade a diferentes distribuições dos pesos pelas funções. A Figura 7, apresenta uma interpretação gráfica de como serão obtidos os pontos da fronteira de Pareto.

Cada ponto será obtido traçando-se segmentos de reta paralelos, com inclinações dadas pela distribuição dos pesos, até se tocar a fronteira do espaço das funções objetivas. As direções desses segmentos de reta podem ser escolhidas da expressão (2-5), fixando-se valores para a função combinada:

$$f_2(\mathbf{x}) = C - \frac{w_1}{w_2} f_1(\mathbf{x}) \quad (2-5)$$

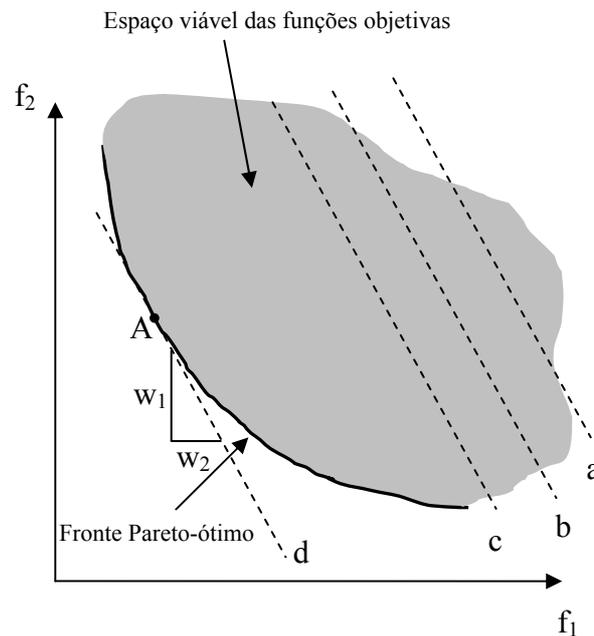


Figura 7 - Interpretação gráfica do método da soma ponderada

Nota-se que esse tipo de abordagem é adequado apenas quando o espaço de funções objetivas for convexa na região da fronteira de Pareto. Nesse caso, pode-se provar que todos os pontos dessa fronteira podem ser obtidos apenas variando-se os pesos (Deb, 2001).

Caso o espaço de funções objetivas seja não-convexo, uma região do espaço das funções objetivas não poderá ser atingida pelo método exposto, perdendo-se soluções que poderiam vir a ser interessantes, como mostrado na Figura 8.

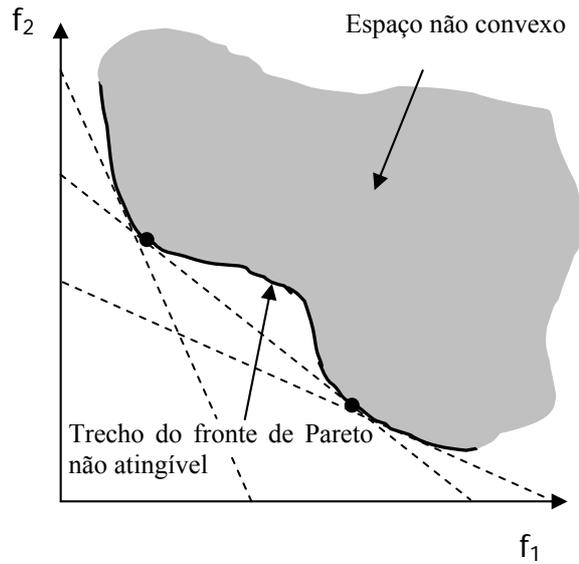


Figura 8 - Trecho do fronte de Pareto não atingível pelo Método da Soma Ponderada.

Um outro método clássico, bastante conhecido, é o **Método da Restrição- $\epsilon$**  que consiste em se tratarem todas as funções objetivas como restrições em níveis  $\epsilon_i$ , a menos de uma, que continuará sendo minimizada (Figura 9). Variando-se os valores das restrições  $\epsilon_i$ , qualquer região do fronte de Pareto poderá ser acessada, tendo-se então transformado o problema multiobjetivo em diversos problemas com um único objetivo.

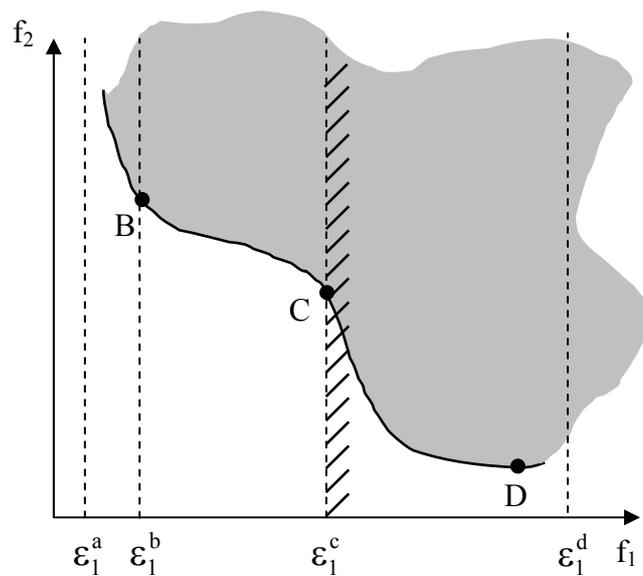


Figura 9 - Representação gráfica do método da restrição- $\epsilon$

Dentre os demais métodos clássicos, citaremos apenas o **Método da Função de Utilidade**, que introduz a preferência do projetista através de uma função; o **Método das Distâncias Ponderadas**, dos quais o Método da Soma Ponderada é um caso particular; o **Método de Benson**; o **Método de Especificação de Objetivos**; e **Métodos Interativos**, todos visando a incorporar informações adicionais antes do processo de otimização, buscando trabalhar com apenas uma função objetiva [Deb, 2001).

### 3. Otimização com Algoritmos Evolutivos

Os algoritmos evolutivos procuram utilizar conceitos e princípios da evolução natural das espécies como estratégia de otimização de problemas.

A aplicação desses conceitos aos problemas de otimização multiobjetivo pode ser vista como uma extensão a partir do entendimento do funcionamento dos mesmos na abordagem de problemas de um único objetivo.

Para a busca de soluções ótimas em problemas de um único objetivo, existem à disposição métodos numéricos tradicionais, como o gradiente conjugado e outros, que têm algumas características fundamentais. Uma delas é trabalhar com regra determinística na transição de uma solução a outra durante a busca ponto-a-ponto. Para obterem maior eficiência, eles utilizam os gradientes da função objetiva. Neste caso, os métodos numéricos tradicionais podem não ser eficientes em problemas com regiões com descontinuidade ou não-diferenciáveis ou ainda com variáveis discretas, o que pode requerer uma implementação para cada tipo de problema. A convergência pode depender da solução inicial escolhida e o ponto ótimo encontrado pode ser apenas um ponto ótimo local.

Os algoritmos evolutivos conseguem contornar essas desvantagens. As regras de transição entre soluções são definidas por operações do tipo genéticas, orientadas em grande parte por simulação, e são implementadas visando à busca do ponto ótimo global.

Uma vantagem adicional dos algoritmos evolutivos em relação aos métodos numéricos clássicos é que podem utilizar as facilidades da computação paralela para avaliação das funções de resposta. Na verdade, isso se torna uma necessidade quando se trabalha com respostas de análises com grande tempo de processamento que serão avaliadas para cada geração de soluções indicadas pelo algoritmo de otimização.

Os algoritmos de otimização que empregam processos evolutivos podem seguir diferentes linhas, das quais algumas têm sido mais investigadas, especialmente os

**Algoritmos Genéticos** (Genetic Algorithms - GA) e os métodos baseados em **Estratégias Evolutivas** (Evolution Strategies - ES), podendo-se citar também **Evolução Diferencial** (Differential Evolution - DE) e **Programação Evolutiva** (Evolutionary Programming - EP).

Os algoritmos evolutivos podem ser elitistas ou não. A preservação do elitismo é feita garantindo-se uma parte mais ajustada da população corrente na descendência, o que pode ser feito com diversos critérios. Em Deb (2001), é justificada a vantagem de se utilizarem os métodos elitistas durante a busca, como uma forma de não permitir que se deteriore nos passos seguintes a aptidão das melhores soluções encontrada até o passo corrente. Também a presença dessas melhores soluções nas gerações seguintes permite que se tenha uma maior probabilidade de se criarem melhores indivíduos.

No caso de problemas com mais de um ponto ótimo local, passa a ser importante a preservação de diversidade na geração de soluções candidatas, de modo a evitar a tendência de convergência prematura para pontos ótimos locais.

A manutenção da diversidade é obtida pela introdução de funções de avaliação da densidade de soluções por regiões do espaço das variáveis de projeto, para se dar preferência a preencher regiões menos densas. Uma opção, sugerida em Deb (2001), é adotar a preservação de nichos baseados em distâncias entre cada 2 soluções da população de  $\mu$  elementos, no espaço das variáveis de projeto, para permitir avaliar a concentração de soluções em torno de uma determinada região, representada por uma função de compartilhamento (*sharing function*), um contador de nicho (*niche count*) e a reavaliação da aptidão (*shared fitness*), como definidos nas expressões (3-1), (3-2) e (3-3). Se uma solução estiver isolada, terá o seu contador de nicho  $n_c$  próximo de 1, não reduzindo o valor de sua aptidão, tendo preferência em relação a outra de mesma aptidão, mas em região mais densa. Essa abordagem necessita de  $O(\mu^2)$  avaliações de distância entre soluções em termos de variáveis de projeto.

$$Sh(d) = \begin{cases} 1 - \left( \frac{d}{\sigma_{share}} \right)^\alpha, & \text{se } d \leq \sigma_{share} \\ 0, & \text{se } d > \sigma_{share} \end{cases} \quad (3-1)$$

onde  $\sigma_{share}$  e  $\alpha$  são ajustados ao problema.

$$nc_i = \sum_{j=1}^{\mu} Sh(d_{ij}) \quad (3-2)$$

$$f'_i = \frac{f_i}{nc_i} \quad (3-3)$$

Será mostrada em seguida a base da formulação dos processos que seguem Algoritmos Genéticos e Estratégias Evolutivas.

### 3.1. Algoritmos Genéticos (GA)

Os Algoritmos Genéticos se utilizam dos conceitos da Genética, para simular a evolução das populações. Cada solução é representada por uma seqüência binária ou, mais recentemente, por parâmetros reais. Cada passo do processo corresponde a uma geração, que é formada por uma população de soluções.

Na representação binária, cada seqüência forma um cromossomo ou genótipo na linguagem da genética, que, decodificada, representa os valores de cada uma das variáveis de projeto ou parâmetros, potenciais soluções. Através de mapeamento, pode-se trabalhar com números decimais e negativos, com a precisão desejada, mas finita, ao custo de seqüências mais longas. Nesse aspecto, o emprego de variáveis reais apresenta vantagens ao se trabalhar com espaços de solução contínuos.

A partir de uma população escolhida randomicamente, é feita uma seleção dos membros que participarão da geração da prole ou próxima geração. A seleção ou reprodução pode ser feita por diferentes métodos, todos visando a que os indivíduos mais aptos à sua função tenham mais chances de ser representados na base de acasalamento (*mating pool*). Esta aptidão é representada por uma função de aptidão (*fitness*), que é avaliada para cada solução. Alguns métodos de seleção citados na

literatura são seleção por torneio (*tournament*), seleção proporcional e seleção por ordenação (*ranking*). Escolhida a base de acasalamento, ela passará, então, pelas operações de recombinação (*crossover*) e de mutação.

A recombinação consiste em criar descendentes de forma tal que algumas das características dos parentes sejam trocadas entre si. A idéia é que se tenha a chance de criar melhores indivíduos, mesmo que se criem também piores, pois os melhores irão ter mais representantes na seleção para reprodução seguinte. Para a representação binária, por exemplo, consiste em trocar trechos da seqüência binária de cada duas soluções numa posição CS escolhida de forma randômica, podendo-se associar também uma probabilidade  $p_c$  de ocorrência de recombinação, de 0 a 1. Para a representação real, há algumas opções de regras de composição das soluções parentes.

Já a mutação, para os Algoritmos Genéticos, é uma perturbação em cada descendente, preferencialmente local. Para a representação binária, por exemplo, um bit (gene) pode ser invertido de 1 a 0 ou vice-versa, segundo uma probabilidade  $p_m$ . Para a representação por parâmetros reais, a idéia é que se gerem por simulação pontos segundo alguma distribuição concentrada na região da solução a sofrer mutação.

De modo geral, nos Algoritmos Genéticos, a recombinação é responsável mais pela procura no espaço de soluções, enquanto que a mutação é responsável mais pela diversidade de soluções.

Após as 3 operações, seleção, recombinação e mutação, nessa ordem, a nova população é avaliada, sendo as novas aptidões calculadas e reiniciado o processo, até atender a um critério de parada.

### **3.2. Estratégias Evolutivas (ES)**

As Estratégias Evolutivas foram desenvolvidas para otimização numérica com vetores de ponto flutuante. Cada indivíduo de uma população será representado por  $(\mathbf{x}, \boldsymbol{\sigma})$ , em que  $\mathbf{x}$  é um ponto no espaço de procura, que deverá otimizar uma ou mais funções objetivas no domínio do problema e  $\boldsymbol{\sigma}$  é um vetor de desvios padrões com a

dimensão de  $\mathbf{x}$ . Cada nova geração pode ser obtida por operações de recombinação (*crossover*) e de mutação.

A recombinação não é essencial nas Estratégias Evolutivas. Algoritmos que as empregam passaram a utilizá-la apenas recentemente, importando a idéia dos Algoritmos Genéticos. A recombinação pode ser realizada por diferentes critérios, como de forma discreta, em que cada variável de projeto que forma um indivíduo descendente  $(\mathbf{x}, \sigma)$  é obtida das variáveis correspondentes dos 2 ou mais indivíduos parentes, com escolha aleatória das parentes para cada variável. Pode também ser obtida de forma intermediária, em que as variáveis de projeto dos indivíduos descendentes incorporam informação parcial de todas as variáveis dos 2 ou mais indivíduos da geração parente.

Já a mutação é um elemento crucial das Estratégias Evolutivas. No caso de não haver recombinação, a mutação deve dar conta tanto do aspecto de busca (convergência) quanto de diversidade. No caso mais simples, ela é aplicada sobre o(s) indivíduo(s) parente como em (3-4):

$$\mathbf{x}' = \mathbf{x} + N(0, \sigma), \quad (3-4)$$

onde  $N(0, \sigma)$  é um vetor de números aleatórios independentes que segue a distribuição de Gauss. A idéia de criar novos indivíduos aleatoriamente segundo a distribuição normal estaria em acordo com a observação biológica de que as pequenas mudanças são mais freqüentes.

O algoritmo mais simples das estratégias evolutivas, somente com mutação, e com somente um indivíduo na população, com notação (1+1)-ES, pode ser descrito como em (3-5) abaixo para o caso de um único objetivo (Michalewicz, 1992):

Inicialize  $\mathbf{X}^t$

Faça até (condição de finalização)

$$\mathbf{X}^{t+1} = \mathbf{X}^t + N(0, \sigma) \quad (3-5)$$

Selecione  $\mathbf{X}^{t+1}$  se  $F(\mathbf{X}^{t+1}) < F(\mathbf{X}^t)$  e  $\mathbf{X}^{t+1}$  satisfaz a todas as restrições

Fim

Nesse processo, é removido o indivíduo menos ajustado da população intermediária de  $1 + 1 = 2$  indivíduos, sendo, portanto, forçosamente elitista. O processo pode ser multimembro, ou seja, baseado em uma população, sendo  $\mu$  o número de indivíduos da geração anterior e  $\lambda$  o número de indivíduos gerados por recombinação e mutação. No caso de ser elitista, são removidos os indivíduos menos ajustados de  $\mu + \lambda$ , representado por  $(\mu + \lambda)$ -ES; no caso de ser não-elitista, são removidos os menos ajustados de somente  $\lambda$  indivíduos, com a notação  $(\mu, \lambda)$ -ES, voltando-se a  $\mu$  indivíduos em ambos os casos. Há ainda variantes como  $(1 + \lambda)$ -ES em que, de apenas um parente, são gerados  $\lambda$  descendentes. Após a seleção, resulta um só parente para a geração seguinte. Neste trabalho, foi utilizado o  $(1 + 1)$ -ES, estendido para o algoritmo de otimização multiobjetivo escolhido, a ser detalhado.

Em qualquer caso, a seqüência para as Estratégias Evolutivas será em cada passo, a partir dos indivíduos parentes, recombinação, se houver, mutação, e, então, seleção.

O vetor de desvios padrões pode sofrer ajustes ao longo do processo, que podem se dar de forma auto-adaptativa ou não. O ajuste auto-adaptativo pode ser do tipo aleatório ou determinístico. Algumas alternativas em que os desvios padrões participam do processo de evolução são mencionadas em Deb (2001). Rechenberg (1973) propôs a famosa a regra determinística da “taxa de sucessos 1/5”, relativa à taxa de sucessos  $p_s$  da descendência numa certa fase, definida como o número de sucessos após  $n$  mutações. Caso essa taxa fique abaixo de 1/5, significando que o processo está caminhando para o ótimo, sendo difícil encontrar novas soluções melhores, reduzem-se os desvios padrões, e vice-versa, como em (3-6). Entretanto, algum cuidado deve ser tomado em problemas em que é difícil encontrar um determinado ótimo local dentre outros com ordenadas similares ou com restrições severas que dificultem o encontro de soluções viáveis.

$$\sigma^{t+1} = \begin{cases} c_d \sigma^t, & \text{se } p_s < 1/5 \\ \frac{1}{c_d} \sigma^t, & \text{se } p_s > 1/5 \\ \sigma^t & \end{cases} \quad (3-6)$$

onde  $c_d$  é o fator de redução ou ampliação, sugerido como 0.817, conforme mencionado em Deb (2001), atribuído a Schwefel (1981).

Como já foi mencionado no Capítulo 1, uma das maiores críticas ao algoritmo Estratégias Evolutivas (1+1)-ES baseia-se no fato de que ele é considerado um algoritmo com **estratégia de busca local**, uma vez que utiliza somente a operação de mutação num único parente para criar uma única prole.

Porém, observa-se que o caráter de busca local pode ser expandido pela utilização de dois níveis de desvio padrão para mutação: um global e um local. Também é possível utilizar eventualmente como parentes, dentro do processo de simulação, as soluções armazenadas no arquivo de referência que estejam situadas nas regiões mais rarefeitas do fronte de Pareto. Desta forma, obtém-se um algoritmo de Estratégias Evolutivas (1+1)-ES com características mais próximas de uma **estratégia de busca global**.

Pare este trabalho, o trecho de pseudo-código apresentado na Figura 10 dá uma idéia da implementação adotada em termos de mutação. No algoritmo, foram utilizados dois vetores com desvios-padrão  $\sigma_1$  e  $\sigma_2$  para mutação das variáveis de projeto. A busca com o desvio padrão maior  $\sigma_1$  tem a função de varrer mais amplamente o espaço, a partir da parente corrente, mantendo a procura por regiões mais promissoras, enquanto que o desvio padrão menor  $\sigma_2$  faz uma busca local. São sugeridos os valores  $\sigma_1 = 1/3$  do intervalo de simulação de cada variável de projeto e um valor inicial  $\sigma_2 = \sigma_1/10$ . Para acelerar a convergência, o valor inicial de  $\sigma_2$  é reduzido ao longo do processo, com decaimento linear.

Percebe-se que a simulação é feita em ciclos de 2 etapas. Na simulação por eixo, cada variável é incrementada através da simulação com os respectivos desvios padrões,  $\sigma_1$  ou  $\sigma_2$ , enquanto que as demais são fixadas, de forma tal que só melhore a resposta

em cada eixo, num total de ND simulações. Em seguida, são feitas simulações independentes, sugeridas  $ND/3 + 3$  simulações, com desvio padrão  $\sigma_1$  ou  $\sigma_2$ . A escolha entre os desvios padrões  $\sigma_1$  ou  $\sigma_2$  para cada variável de projeto em qualquer das etapas é feita de forma aleatória, em que  $\sigma_1$  é adotado caso o resultado de uma simulação com distribuição Uniforme(0,1) seja maior que uma probabilidade PR, previamente escolhida; caso contrário,  $\sigma_2$  é adotado. A probabilidade PR pode variar ao longo das iterações, tendo-se adotado um decaimento linear.

```

jmax = 80 ; nsj = ns/jmax; ND2 = ND+ND/3+3
Para j = 1,jmax Faça
   $\sigma_2 \leftarrow$  decresce linearmente com j, de  $\sigma_{2\_inicial}$  a  $\sigma_{2\_inicial}/10$ 
  PR  $\leftarrow$  decresce linearmente com j, de PR1 a PR2
   $\varepsilon \leftarrow$  decresce linearmente com j, de  $\varepsilon_1$  a  $\varepsilon_2$  (tolerância na soma das violações das
    restrições)
  Atualiza condição de viabilidade da parente para novo  $\varepsilon$ ,  $j \neq j_{max}-jj$ ,  $jj = 5,3,1,0$ .
  Atualiza condição de viabilidade do fronte de Pareto para novo  $\varepsilon$ .
  Para j1 = 1,nsj Faça
    Se (desvio padrão adaptativo em  $j = j_{max}$ ) atualiza  $\sigma_2$ 
    k = resto de  $j1/ND_2$ 
    Se( $1 \leq k \leq ND$ ) Então ! Simulação Individual para cada Variavel
      v = z ! v = descendente; z = parente
20 Se(Uniforme(0,1) < PR) Então
       $\sigma \leftarrow \sigma_{1,k}$ 
      Se não
         $\sigma \leftarrow \sigma_{2,k}$ 
      Fim Se
       $v_k \leftarrow z_k + N(0, \sigma)$ 
      Se( $v_k < a_k$  ou  $v_k > b_k$ ) Vá para 20 (a cada 10 ultrapassagens, adota limite)
    Se não
      Para i = 1,ND Faça ! Simulação Conjunta de todas as Variáveis
25 Se(Uniforme(0,1)<PR) Então
         $\sigma \leftarrow \sigma_{1,i}$ 
        Se não
           $\sigma \leftarrow \sigma_{2,i}$ 
        Fim Se
         $v_i \leftarrow z_i + N(0, \sigma)$ 
        Se( $v_i < a_i$  ou  $v_i > b_i$ ) Vá para 25 (a cada 100 iterações, adota limite)
      Fim Faça
    Fim Se
  Fim Para

```

Figura 10 - Pseudo-código com o trecho do algoritmo implementado, referente à mutação das variáveis de projeto (o tratamento de restrições será visto no próximo item.

Pode-se perceber também que, caso o ponto simulado caia fora dos limites do intervalo de cada variável,  $a_i$  e  $b_i$ , a solução é novamente simulada. Posteriormente, para esses casos, foi introduzida a opção de permitir adotar-se o limite ultrapassado, ao invés de simular novamente a variável de projeto, a cada certo número de ultrapassagens ou de iterações pré-determinado (adotou-se a cada 10 ultrapassagens na simulação individual e a cada 100 iterações na simulação conjunta). Isso fez com que se evitassem perturbações em regiões do fronte nos problemas multiobjetivo do tipo ZDT, como será visto no Capítulo 5. Entretanto, o número de referência não deve ser baixo a ponto de reduzir substancialmente o número de pontos simulados no interior dos intervalos.

Poderíamos chamar, genericamente, a simulação com pelo menos uma variável simulada com o desvio padrão maior,  $\sigma_1$ , de exploração em busca de áreas promissoras, enquanto que a simulação com o desvio padrão menor,  $\sigma_2$ , para todas as variáveis, de exploração de áreas promissoras já encontradas, ou de fase de sintonia fina.

Cabe notar que, em problemas com restrições, é possível que nem todos os passos do ciclo sejam atendidos seqüencialmente, o que ocorre quando as restrições não são atendidas em cada simulação. Porém, poderá haver compensação em ciclos subseqüentes.

A estratégia de se adotar a simulação individual por variável de projeto se mostrou útil para contornar efeitos de alta dimensionalidade. Por exemplo, em problemas com função objetiva do tipo soma de muitas variáveis de projeto, o desvio padrão da perturbação causada na função pela simulação conjunta, com valor esperado nulo, fica pequeno em relação ao intervalo de variação da função. No caso de ND variáveis de projeto com o mesmo intervalo de validade  $[a,b]$  e mesmo desvio padrão  $\sigma$ , o intervalo de variação da função soma seria  $ND(b-a)$ , enquanto que o desvio padrão da função seria  $\sigma_f = \sqrt{ND}\sigma$ .

### 3.2.1. Tratamento de problemas com restrições

Num problema geral de otimização, as restrições são contempladas pelas inequações  $g_j(x) \leq 0$  e igualdades  $h_k(x) = 0$ , de acordo com a expressão geral (2-1).

Existem três enfoques básicos para abordar os problemas com restrições no processo de otimização: no primeiro, são utilizadas apenas soluções viáveis (*feasible*) que respeitem rigorosamente as funções de restrição; no segundo, são utilizadas funções de penalidade adicionadas às funções objetivas; no terceiro, as funções objetivas são substituídas por funções de aptidão (*fitness*) que assumem o valor da respectiva função objetiva para as soluções viáveis e o valor da soma das restrições violadas para as soluções não-viáveis.

No segundo enfoque, as funções de penalidade são adicionadas às funções objetivas (Deb, 1995, Reklaitis et al., 1983, Homaifar et al., 1994, Michalewicz and Attia, 1994), podendo ser tratada de acordo com o indicado na expressão (3-7).

$$F_m(\mathbf{x}) = f_m(\mathbf{x}) + R_m \left[ \sum_{j=1}^J \left\{ \begin{array}{ll} g_j(\mathbf{x}), & g_j(\mathbf{x}) > 0 \\ 0, & g_j(\mathbf{x}) \leq 0 \end{array} \right\} + \sum_{k=1}^K (|h_k(\mathbf{x})|) \right] \quad (3-7)$$

Observa-se que, para a utilização de (3-7), é essencial que as funções de restrição sejam inicialmente normalizadas. Os valores dos parâmetros de penalidade  $R_m$  são então ajustados de forma que o somatório das restrições fique com a mesma ordem de grandeza da respectiva função objetiva  $f_m(\mathbf{x})$ , num problema multiobjetivo. Como as funções objetivas podem ser de diferentes ordens de magnitude, o valor do parâmetro de penalidade  $R_m$  deverá ser ajustado para cada função objetiva  $f_m(\mathbf{x})$ .

Na prática, dependendo dos tipos de funções  $f_m(\mathbf{x})$ , é bastante difícil conseguir o ajuste dos parâmetros de penalidade. A função objetiva penalizada  $F_m(\mathbf{x})$  é, em geral, altamente sensível aos valores dos parâmetros  $R_m$ , tornando-se necessária a realização de várias rodadas para realizar o ajuste fino dos mesmos.

No terceiro enfoque, proposto por Mezura-Montes e Coello (2005) para problemas de otimização de um único objetivo, a função objetiva é substituída por uma função de aptidão (*fitness*) que assume o valor da respectiva função objetiva para as soluções viáveis e o valor da soma das restrições violadas para as soluções não-viáveis.

Para o presente trabalho, foi feita a extensão deste enfoque para problemas de otimização multiobjetivo, conforme mostrado na expressão (3-8), particularizada para o problema de minimização:

$$\text{Fitness}(\mathbf{x}) = \begin{cases} f_m(\mathbf{x}), & m = 1, 2, \dots, M \quad \text{se } \mathbf{x} \text{ é viável} \\ \sum_{j=1}^J \begin{cases} g_j(\mathbf{x}), & g_j(\mathbf{x}) > 0 \\ 0, & g_j(\mathbf{x}) \leq 0 \end{cases} + \sum_{k=1}^K |h_k(\mathbf{x})| & \text{se } \mathbf{x} \text{ é inviável} \end{cases} \quad (3-8)$$

A função de aptidão em (3-8) deve ser utilizada em conjunto com o seguinte critério:

1. Entre duas soluções viáveis, a que tiver menor valor vence;
2. Se uma solução for viável e a outra inviável, a solução viável vence;
3. Caso ambas as soluções sejam inviáveis, a que tiver a menor soma dos valores das funções de restrição referentes às restrições violadas vence.

Observa-se que este enfoque, por não misturar os valores das funções de restrição com os valores das funções objetivas, evita os principais inconvenientes do segundo enfoque que utiliza as funções de penalização.

Cabe notar que a definição da função de aptidão em (3-8) é diferente da proposta de Deb (2000), em que, no caso de  $\mathbf{x}$  ser inviável, o valor da função objetiva da pior solução viável,  $f_{\text{pior}}(\mathbf{x})$ , é adicionado ao somatório das restrições violadas. Esta definição possui os mesmos problemas de normalização e ajuste para que o somatório das restrições fique com a mesma ordem de grandeza da respectiva função objetiva  $f_{\text{pior}}(\mathbf{x})$ , que os observados no segundo enfoque. Também é diferente da proposta de Powell e Skolnick (1993) de mapear soluções viáveis e inviáveis em seqüência, de forma tal que soluções viáveis sejam sempre melhores que as inviáveis.

Para facilitar a convergência, geralmente pode-se adotar algum tipo de relaxamento inicial para as restrições, ou seja:  $g_j(\mathbf{x}) \leq \varepsilon_j$  e  $|h_k(\mathbf{x})| \leq \varepsilon_k$ , sendo  $\varepsilon_j$  e  $\varepsilon_k$  os valores iniciais de tolerância, definidos para cada restrição. Os valores  $\varepsilon_j$  e  $\varepsilon_k$  são progressivamente reduzidos, ao longo das simulações, até atingir os níveis de valores dentro da precisão requerida para o problema que está sendo analisado. No presente

trabalho, são admitidas tolerâncias em relação à soma das violações das restrições, ou seja,  $\mathbf{x}$  é viável se:

$$\sum_{j=1}^J \left\{ \begin{array}{l} \mathbf{g}_j(\mathbf{x}), \quad \mathbf{g}_j(\mathbf{x}) > 0 \\ 0, \quad \mathbf{g}_j(\mathbf{x}) \leq 0 \end{array} \right\} + \sum_{k=1}^K |\mathbf{h}_k(\mathbf{x})| \leq \varepsilon, \quad (3-9)$$

com  $\varepsilon$  sofrendo decaimento linear de um valor inicial  $\varepsilon_1$  a um valor final  $\varepsilon_2$ .

O trabalho de Mezura-Montes e Coello (2005) baseou-se em Estratégias Evolutivas do tipo multimembro  $(\mu + \lambda)$ -ES, onde foi adotado um mecanismo de preservação de diversidade para admitir que permaneçam algumas soluções ligeiramente inviáveis (baixa violação do somatório das restrições e com relativamente boa aptidão) na população para a próxima geração.

No presente trabalho, a diversidade da busca de soluções é obtida pela utilização de dois níveis de desvio padrão. Acredita-se que esta estratégia permita varrer adequadamente o espaço das variáveis de projeto com o método (1+1)-ES. Além disso, a solução parente será intencionalmente mantida fixa, viável ou inviável, nos passos finais  $j_{\max}-jj$ ,  $jj = 5,3,1,0$ ,  $j_{\max}$  definido no pseudo-código da Figura 10, com as soluções descendentes sendo, então, geradas supostamente numa região já próxima da solução ótima. Isso pode ser particularmente útil em problemas com restrições ativas no ótimo ( $\mathbf{g}(\mathbf{x}) = 0$  ou  $\mathbf{h}(\mathbf{x}) = 0$ ).

O algoritmo (1+1)-ES é utilizado para gerar as soluções candidatas a participar do fronte de Pareto, através do procedimento proposto (1+1)-PAES-M como será visto nos capítulos seguintes.

#### 4. Otimização Multiobjetivo com Algoritmos Evolutivos

Os algoritmos evolutivos tornaram-se uma ferramenta importante para solução dos problemas de otimização multiobjetivo pelos métodos diretos, para obter os pontos do fronte de soluções Pareto-ótimas com boa convergência e diversidade de soluções. Existem diversos algoritmos propostos na literatura para implementação de otimizadores multiobjetivo com diferentes graus de sucesso e eficiência (Deb, 2001).

Neste trabalho, foi implementado um método direto de otimização multiobjetivo, baseado no método (1+1)-PAES (Estratégia Evolutiva com Pareto-Arquivado) (Knowles e Corne, 1999), onde foram introduzidas algumas modificações, a serem descritas no próximo item.

O método de otimização multiobjetivo implementado foi denominado (1+1)-PAES-M (Estratégia Evolutiva com Pareto-Arquivado, Modificado).

O principal motivo da escolha de um algoritmo de otimização multiobjetivo do tipo (1+1)-PAES é devido a sua simplicidade computacional, a sua eficiência e a sua robustez, quando comparado com outros algoritmos GA e ES do tipo  $(\mu + \lambda)$ .

Parece que o algoritmo do tipo (1+1)-PAES funciona bem devido aos mesmos motivos pelos quais ele é um algoritmo computacionalmente simples: é um algoritmo agressivo, que testa cada solução candidata de uma forma direta e não gasta muitos recursos nas soluções candidatas que não passaram no teste.

O principal inconveniente do algoritmo original (1+1)-PAES, segundo os autores Knowles e Corne (1999), é que ele pode ficar trancado num ponto de máximo (mínimo) local, quando as mutações produzidas na solução parente não forem suficientemente grandes para atravessar o espaço existente entre o ponto de máximo global e os diversos pontos de máximos locais.

Este inconveniente foi contornado no algoritmo proposto (1+1)-PAES-M pela utilização do algoritmo de Estratégias Evolutivas (1+1)-ES com características de busca

global, onde a diversidade da busca de soluções é obtida pela utilização de dois níveis de desvio padrão. Acredita-se que esta estratégia permite varrer adequadamente o espaço das variáveis de projeto.

Um aspecto interessante, abordado por Deb (2001), seria entender a necessidade da obtenção prévia de vários pontos do fronte de Pareto para, ao final, utilizar no projeto apenas um único ponto do fronte. A principal razão reside no fato de que é muito mais fácil escolher o ponto de projeto quando se tem a possibilidade de analisar o conjunto de soluções possíveis do fronte de Pareto e de estabelecer correspondência entre o espaço das funções objetivas e o espaço das variáveis de projeto. Neste caso, o projetista terá mais condições para avaliar o impacto das informações subjetivas adicionais na escolha dos valores finais das variáveis de projeto, do que seria possível fazer a priori sem essas informações.

#### **4.1. PAES (Estratégia Evolutiva com Pareto-Arquivado)**

Como já foi mencionado anteriormente, neste trabalho, foi implementado um método de otimização multiobjetivo baseado no método (1+1)-PAES (Estratégia Evolutiva com Pareto-Arquivado) (Knowles e Corne, 1999), onde foram introduzidas algumas modificações.

O método de otimização multiobjetivo implementado foi denominado (1+1)-PAES-M (Estratégia Evolutiva com Pareto-Arquivado, Modificado), cujas principais características são descritas a seguir.

O método (1+1)-PAES mantém um arquivo **A** com as melhores soluções encontradas até o momento. Ele começa vazio e é preenchido e atualizado ao longo do processo, porém nunca pode ultrapassar um número máximo de elementos pré-fixado.

O método (1+1)-PAES consiste essencialmente dos seguintes passos:

- Inicialmente, adota-se como parente  $p_0$  uma solução aleatória, dentro do intervalo de validade das variáveis de projeto.

- Gera-se a seguir um  $\mathbf{c}_0$  descendente de  $\mathbf{p}_0$ , obtido pelo método de Estratégias Evolutivas (1+1)-ES.
- O descendente  $\mathbf{c}_0$  é comparado com o parente  $\mathbf{p}_0$  e o vencedor torna-se o parente da próxima geração. A questão central é a forma como o vencedor é escolhido e como o fronte é atualizado num processo de otimização multiobjetivo, o que se apresenta na literatura com algumas alternativas, tendo sido implementado como descrito a seguir.
- Em cada geração  $t$ , o parente  $\mathbf{p}_t$  é comparado com o descendente  $\mathbf{c}_t$ , em termos de relação de dominação, resultando três cenários possíveis:
  1. Se  $\mathbf{p}_t$  domina  $\mathbf{c}_t$ , o descendente  $\mathbf{c}_t$  não é aceito e o processo prossegue com a geração de uma nova solução pelo método (1+1)-ES, a partir do mesmo parente  $\mathbf{p}_t$ .
  2. Se  $\mathbf{c}_t$  domina  $\mathbf{p}_t$ , então  $\mathbf{c}_t$  é aceito como a parente da próxima geração. Compara  $\mathbf{c}_t$  com o arquivo  $\mathbf{A}_t$ , resultando em 3 cenários:
    - 2.1. Se  $\mathbf{c}_t$  é dominado por qualquer dos membros do arquivo  $\mathbf{A}_t$ , então  $\mathbf{A}_t$  não é atualizado;
    - 2.2. Se  $\mathbf{c}_t$  domina um ou mais membros do arquivo  $\mathbf{A}_t$ , os elementos dominados são removidos de  $\mathbf{A}_t$ , deslocando uma ou mais soluções previamente armazenadas no mesmo. Deste modo, o arquivo  $\mathbf{A}$  fica composto somente por soluções não-dominadas.
    - 2.3. Os elementos de  $\mathbf{A}_t$  e  $\mathbf{c}_t$  formam, 2 a 2, soluções não-dominadas. Neste caso, o descendente  $\mathbf{c}_t$  pertence ao fronte de soluções não-dominadas. Se o arquivo  $\mathbf{A}_t$  não estiver preenchido com o número máximo de elementos, uma cópia de  $\mathbf{c}_t$  é enviada diretamente para  $\mathbf{A}_t$ . Porém, se  $\mathbf{A}_t$  já estiver preenchido,  $\mathbf{c}_t$  só será incorporado ao arquivo se estiver situado numa região menos densa de  $\mathbf{A}_t$ , deslocando automaticamente uma solução situada numa região mais densamente povoada de  $\mathbf{A}_t$ .

3. Se  $\mathbf{p}_t$  e  $\mathbf{c}_t$  são soluções mutuamente não-dominadas, o descendente  $\mathbf{c}_t$  é comparado com as soluções armazenadas no arquivo  $\mathbf{A}_t$ , resultando em três situações possíveis:
  - 3.1. Se  $\mathbf{c}_t$  é dominado por qualquer dos membros do arquivo  $\mathbf{A}_t$ , segue passo 2.1;
  - 3.2. Se  $\mathbf{c}_t$  domina um ou mais membros do arquivo  $\mathbf{A}_t$ , segue passo 2.2 e  $\mathbf{c}_t$  é aceito como a parente da próxima geração;
  - 3.3. Se os elementos de  $\mathbf{A}_t$  e  $\mathbf{c}_t$  formam, 2 a 2, soluções não-dominadas, segue passo 2.3. O descendente  $\mathbf{c}_t$  somente será escolhido como novo parente se estiver situado numa região menos densa de  $\mathbf{A}_t$  que o atual parente  $\mathbf{p}_t$ .

Cabe notar que, dessa forma, a solução parente pode não fazer parte do fronte corrente, armazenado no arquivo  $\mathbf{A}_t$ . No caso de problemas com restrições em que se trabalhe com uma tolerância na soma das violações das restrições, a solução parente não é atualizada nos passos  $j_{\max} - jj$ ,  $jj = 5, 3, 1, 0$ ,  $j_{\max}$  definido no pseudo-código da Figura 10. Assim, os descendentes serão gerados nesses passos a partir de uma solução parente localizada em região supostamente já próxima da solução ótima, mesmo que viole as restrições, o que é particularmente útil nos problemas com restrições ativas no ótimo, como já mencionado.

A densidade das soluções do fronte de Pareto, armazenadas no arquivo  $\mathbf{A}$ , pode ser dada de várias formas.

No método PAES original, o espaço das funções objetivas é dividido em um determinado número de hipercubos de tamanhos iguais. Calcula-se então o número de soluções situadas em cada hipercubo.

Caso o descendente esteja situado num hipercubo com menor número de soluções em relação ao hipercubo do parente atual, o descendente será escolhido como parente para a próxima geração.

Observa-se que, nesse método, o cálculo da densidade das soluções do fronte de Pareto fica dependente do tamanho escolhido para os hipercubos, que é fornecido através da criação de um novo parâmetro  $d$ . O número de hipercubos varia exponencialmente com esse parâmetro. Além disso, o comprimento de cada hipercubo depende do conhecimento dos valores máximos e mínimos possíveis para as funções objetivas, o que é difícil de se conhecer a priori.

Para evitar a introdução de mais um parâmetro no problema de otimização multiobjetivo, foi adotado neste trabalho um outro método para avaliação da densidade das soluções do fronte de Pareto, baseado no conceito de distância de aglomeração (*crowding distance*), semelhante ao utilizado pelo método de otimização multiobjetivo através de Algoritmos Genéticos denominado NSGA-II (Non-dominated Sorting Genetic Algorithm) (Deb et al., 2000a, 2000b), definida da seguinte forma:

### **Distância de Aglomeração (*Crowding Distance*)**

Para estimar a densidade das soluções do fronte de Pareto, armazenadas no arquivo **A**, que estão próximas a uma dada solução  $i$ , é calculada a distância média  $d_i$  de duas soluções de cada lado da solução  $i$  ao longo de cada uma das funções objetivas. A distância  $d_i$  serve para avaliar o perímetro de um hipercubo cujos vértices são constituídos pelas soluções vizinhas mais próximas da solução  $i$ . O perímetro deste hipercubo é denominado de Distância de Aglomeração (*Crowding Distance*), como está indicado na Figura 11.

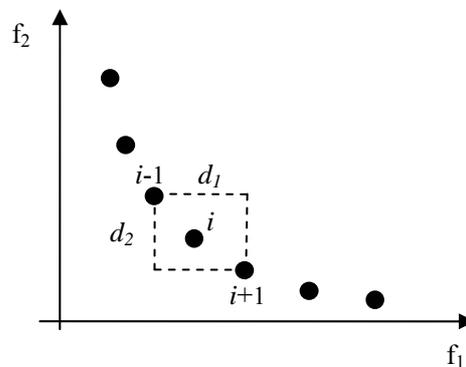


Figura 11 - Distância de Aglomeração (*Crowding Distance*)

A Distância de Aglomeração (*Crowding Distance*), é calculada pelo seguinte algoritmo:

- a) Seja  $n$  o número de soluções do fronte de Pareto armazenadas num dado instante no arquivo  $\mathbf{A}$ ;
- b) Para cada solução  $i$  do arquivo  $\mathbf{A}$ , é atribuída inicialmente uma distância  $d_i = 0$  ;
- c) Para cada função objetiva  $F_m$ ,  $m = 1, 2, \dots, M$ , determinar o vetor dos índices das soluções  $i$  do fronte de Pareto ordenadas em ordem crescente:  $I_i^m$  ;
- d) Atribuir valores infinitos ( $\infty$ ) para as distâncias das soluções situadas nos limites do intervalo de cada função objetiva:  $d_{I_1^m} = d_{I_n^m} = \infty$ ,  $m = 1, 2, \dots, M$ ;
- e) Para as demais soluções:  $j = 2, 3, \dots, n-1$  calcular:

$$d_{I_j^m} = d_{I_j^m} + \frac{F_m^{I_{j+1}^m} - F_m^{I_{j-1}^m}}{F_m^{\max} - F_m^{\min}} \quad (4-1)$$

- f) O elemento  $I_j^m$  do vetor dos índices das soluções indica o valor do índice do membro  $j$  na lista ordenada das soluções da função objetiva  $F_m$  ;
- g) Observa-se que esta métrica fornece, na realidade, o semi-perímetro do retângulo representado na Figura 11.

Resumindo, no procedimento proposto, denominado (1+1)-PAES-M (Estratégia Evolutiva com Pareto Arquivado, Modificado), foram introduzidas no método (1+1)-PAES original as seguintes modificações principais:

- i. Foi desenvolvido um algoritmo de Estratégia Evolutiva (1+1)-ES de pesquisa global, através da utilização de dois níveis de desvio padrão para mutação: um global e um local;
- ii. O algoritmo (1+1)-ES proposto realiza tanto simulações individuais como simulações conjuntas das variáveis do problema para acelerar a convergência;
- iii. O algoritmo (1+1)-ES permite que as restrições sejam tratadas de forma *hard* ou *soft* sem o emprego de funções de penalização;
- iv. A avaliação da diversidade das soluções ao longo do fronte de Pareto é obtida através de uma métrica baseada na distância de aglomeração (*crowding distance metric*).

## 4.2. Outros Algoritmos Multiobjetivo

Na literatura, encontram-se outros algoritmos de otimização multiobjetivo, do qual citaremos alguns mais mencionados em comparações de desempenho. Deb (2001) situa alguns Algoritmos Genéticos (GA) ao longo da história da pesquisa sobre o assunto, como o VEGA (*Vector Evaluated Genetic Algorithm*) (Schaffer, 1984), o MOGA (Multi-objective Genetic Algorithm) (Fonseca e Flemming, 1993), NPGA (*Niched Pareto Genetic Algorithm*) (Horn et al., 1994) e o NSGA (*Non-dominated Sorting Genetic Algorithm*) (Srinivas e Deb, 1994), dentre outros. De modo geral, esses métodos se baseiam em adaptações para que se continue a utilizar as três operações básicas de um GA, especialmente a seleção, que depende da função de aptidão (*fitness*). Assim, por exemplo, o VEGA se baseia em dividir a população em cada geração em M vetores, cada um se concentrando em uma função objetiva na comparação das aptidões. Por outro lado, o NSGA se baseia em dividir a população em frentes não-dominadas e atribuir maiores aptidões aos frentes mais próximos da origem (em problemas de minimização). As comparações são complementadas por funções de avaliação da densidade de soluções por regiões, para se dar preferência a preencher regiões menos densas.

Posteriormente, foi desenvolvido o NSGA-II (Deb et al., 2000a e 2000b), citado como um otimizador com um dos melhores desempenhos. Considera elitismo e contém várias diferenças em relação ao NSGA, dentre as quais passar a avaliar a densidade de soluções no espaço das funções objetivas, como, por exemplo, através da distância de aglomeração (*crowding distance*), e não no de variáveis de projeto.

## 5. Testes de Desempenho do Otimizador (*Bench-mark*)

### 5.1. Medidas de Desempenho

São três os objetivos principais de um processo de otimização multiobjetivo:

- i. Obter soluções as mais próximas possível do fronte ótimo de Pareto;
- ii. Obter soluções as mais diversas possível ao longo do fronte;
- iii. Contemplar a possibilidade de não-convexidade do espaço de soluções.

Na Figura 12, são mostrados os objetivos i e ii.

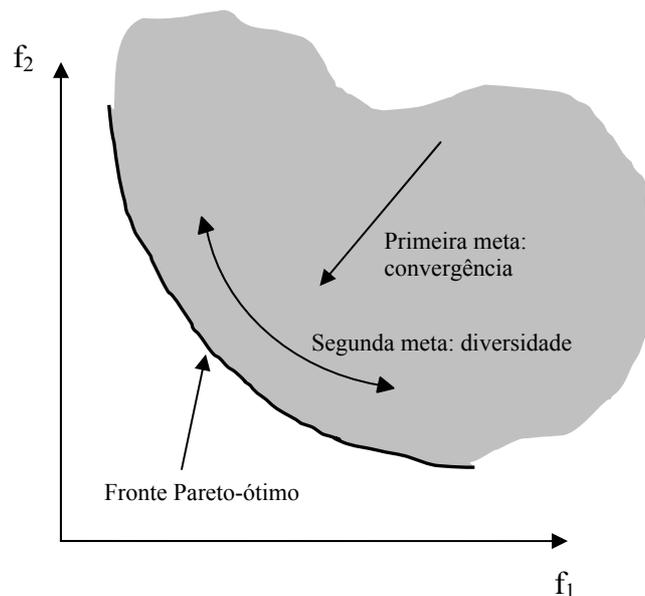


Figura 12 - Metas de um processo de otimização multiobjetivo

Na Figura 13, é mostrado um conjunto ideal de soluções do fronte Pareto-ótimo. Na Figura 14, é mostrado um conjunto de soluções sobre o fronte de Pareto, mas que não contemplam a região central, perdendo-se informações. Na Figura 15, por outro lado, é mostrado um conjunto de soluções com boa diversidade, mas que não convergiram para

o fronte, possivelmente por número insuficiente de simulações do processo evolutivo ou mesmo por dificuldade inerente ao problema ou ao algoritmo.

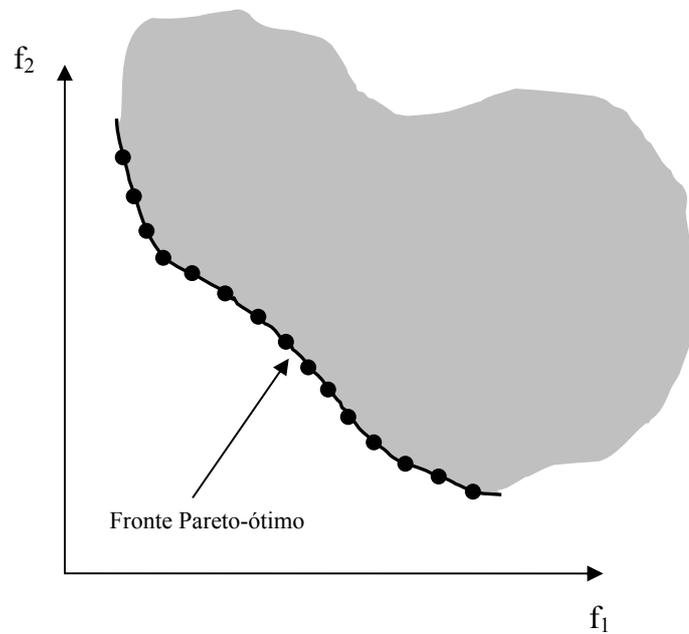


Figura 13 - Fronte Pareto-ótimo ideal.

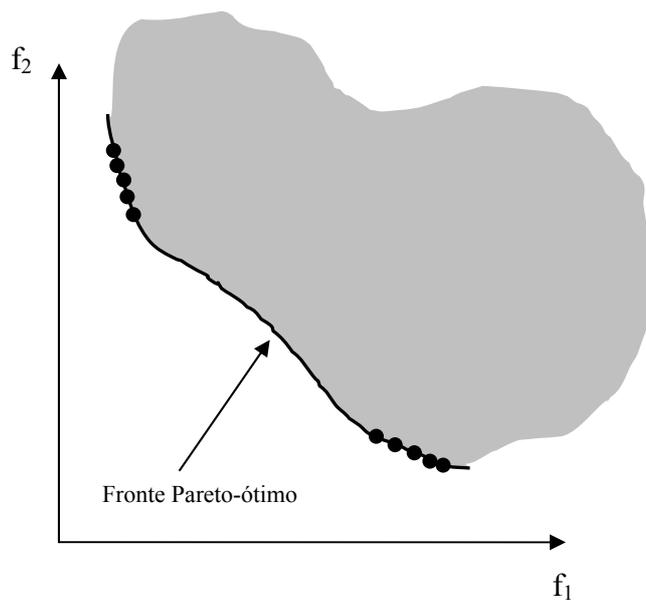


Figura 14 - Fronte Pareto-ótimo obtido com boa convergência, mas má diversidade.

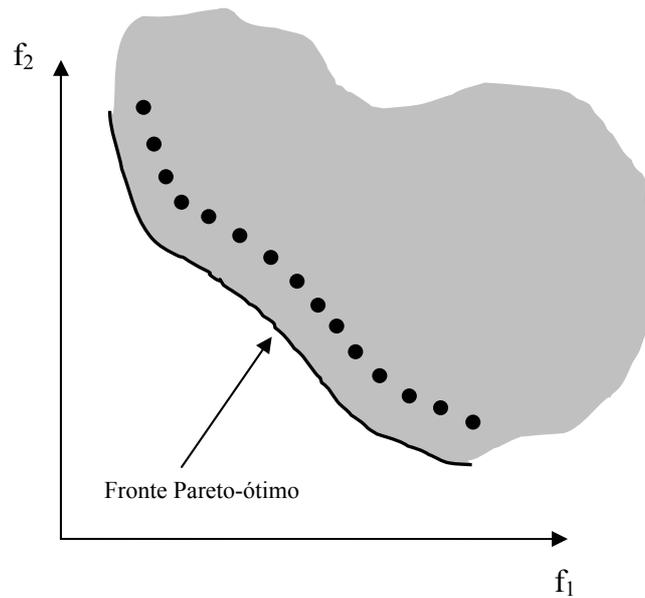


Figura 15 - Frente Pareto-ótimo com boa diversidade, mas má convergência.

Para possibilitar a avaliação do desempenho dos diferentes algoritmos, permitindo uma comparação entre eles, são propostas na literatura funções de referência (*bench-mark*). Essas funções têm diferentes graus de dificuldade e procuram justamente testar a capacidade dos algoritmos de atender aos objetivos enumerados acima.

Também são definidas medidas que permitem a comparação entre diferentes algoritmos. Duas dessas medidas foram implementadas.

Uma é a distância da geração (*generational distance - GD*), que pode ser definida como a distância média dos pontos do frente de Pareto encontrado pelo programa na última geração aos correspondentes pontos do frente de Pareto exato. Essa medida contempla o aspecto de convergência para a solução. Trabalhando com a distância euclidiana, é dada pela expressão (5-1):

$$GD = \frac{\sqrt{\sum_{i=1}^{NFI} d_i^2}}{NFI} \quad (5-1)$$

onde:

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{j=1}^{NFU} (FF_{i,j} - P_{k,j}^*)^2} \quad (5-2)$$

onde  $|P^*|$  é o número de pontos disponíveis do fronte exato  $P^*$ , e a pesquisa é feita de forma a encontrar o ponto disponível mais próximo.

A outra medida implementada foi o espalhamento (*spread* -  $\Delta$ ), que mede a diversidade das soluções no fronte encontrado. É dado pela expressão (5-3) (Deb, 2001):

$$\Delta = \frac{\sum_{j=1}^{NFU} d_j^e + \sum_{i=2}^{NFI-1} |d_i - \bar{d}|}{\sum_{j=1}^{NFU} d_j^e + NFI \cdot \bar{d}} \quad (5-3)$$

onde  $d_i$  é a distância de aglomeração (*crowding distance*) previamente definida, daí se considerarem  $(NFI - 2)$  pontos (poderia ser outra medida de distância entre pontos vizinhos do fronte), e  $\bar{d}$  é a média dessas distâncias. No caso ideal, o valor de  $\Delta$  deve tender a zero. As parcelas com as distâncias  $d_j^e$  extremas de cada função, entre o valor exato e o valor encontrado mais próximo servem para que a medida também detecte se o espalhamento abrange todo o fronte exato ou se está somente bem espaçado numa região específica. A implementação não contemplou essas parcelas adicionais. Entretanto, como será observado, os pontos extremos para os 5 problemas ZDT foram atingidos nas simulações de 24000 gerações, fazendo com que essas parcelas não sejam relevantes para a avaliação do algoritmo.

Para o caso de se desejar comparar desempenhos entre algoritmos sem conhecimento da solução exata, pode ser adotado o critério atribuído a Fonseca e Fleming (1995) baseado em Superfície de Realização (*Attainment Surface*), que é obtida unindo-se os pontos do fronte encontrado formando degraus. O critério se utiliza de linhas de amostragem que cruzam as superfícies de realização obtidas de diversas rodadas, permitindo uma avaliação probabilística de desempenho. Consegue ser um meio objetivo de avaliar os casos em que há regiões do fronte com soluções obtidas com

um dos métodos dominadas pelas do outro método concomitante com regiões em que ocorre o inverso.

Para avaliação do algoritmo de otimização implementado no programa PAES-M (contendo o método (1+1)-PAES com as modificações já descritas), serão utilizados inicialmente problemas multiobjetivo sem restrições (Problemas ZDT), passando-se a dois problemas com restrições (MEC e TNK), apresentados em Deb (2001).

Em seguida, serão utilizados problemas uniobjetivo com condições de busca em regiões de restrição bastante severas, apresentados em Mezura-Montes e Coello (2005). Para isso, serão utilizados os mesmos recursos de Estratégia Evolutiva implementada no programa PAES-M, mas buscando o ótimo para apenas uma função objetiva, ou seja, sem frente Pareto-ótimo.

## **5.2. Problemas ZDT**

São problemas propostos por Zitzler-Deb-Thiler. Serão utilizados os problemas ZDT1 a ZDT4 e ZDT6. Eles trabalham com 2 funções objetivas a serem minimizadas com um aspecto geral similar. A função  $f_1$  deverá testar a habilidade do algoritmo de alcançar diversidade na resposta ao longo do frente de Pareto. A função  $f_2$  será obtida do produto de uma função  $g$ , que deverá testar a capacidade de convergência ao frente, pela função  $h$ , esta, por sua vez, função de  $f_1$  e de  $g$ , e que deverá testar a capacidade do algoritmo de lidar com diferentes formas do frente, em termos de possível não-convexidade e descontinuidade. São problemas sem restrições. O problema ZDT5 não foi utilizado porque trabalha com representação binária, para o qual o algoritmo implementado não é adequado.

Para permitir a comparação com resultados de outros algoritmos citados na literatura, é necessário que o número de avaliações das funções objetivas sejam próximos. Em Deb (2001), trabalhou-se com 250 gerações de 100 indivíduos cada, totalizando 25000 avaliações das funções. Assim, o caso básico para cada problema terá 24000-25000 simulações.

Para obter as medidas como distância da geração (GD) e espalhamento (*spread*), as soluções exatas nos problemas ZDT1 a ZDT4 e ZDT6, para as quais as variáveis  $x_2$  a  $x_n$  devem tender a zero, como será visto em seguida, as soluções exatas  $P_k^*$  mais próximas das soluções encontradas  $FF_{i,j}$  (função objetiva  $j$ , ponto  $i$ ) serão obtidas calculando-se os valores das funções para  $x_1$  do ponto  $i$  do fronte, com as demais variáveis igualadas a zero.

Os resultados para os problemas ZDT serão apresentados para as rodadas com a probabilidade de referência PR variando linearmente de 0.8 a 0.4 em cada uma ao longo da simulação, o que forneceu bons resultados, de modo geral.

### 5.2.1. Problema ZDT1

É um problema com 30 variáveis de projeto, com fronte de Pareto convexo, dado pela expressão (5-4).

$$\text{ZDT1: } \begin{cases} f_1(\mathbf{x}) = x_1, \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, n = 30 \\ h(f_1, g) = 1 - \sqrt{f_1/g}. \\ 0 \leq x_i \leq 1 \end{cases} \quad (5-4)$$

A expressão para a solução exata para o fronte é obtida igualando-se  $x_2$  a  $x_{30}$  a zero, o que faz com que  $g(\mathbf{x})$  seja igual a 1, atingindo seu menor valor, e é dada por:

$$f_2 = 1 - \sqrt{f_1} \quad (5-5)$$

Na Figura 16, é mostrado o fronte de Pareto encontrado com o programa PAES Modificado (PAES-M), com 24000 iterações ou avaliações das funções objetivas, além da solução exata, revelando o bom desempenho do otimizador. Na Figura 17, é apresentado o fronte encontrado após 3000 iterações, que irá avançar para a solução exata. Para ilustrar o papel da simulação individual por eixo de variável de projeto, na

Figura 18 é mostrado o fronte encontrado após 24000 iterações, mostrando que a convergência seria muito lenta. Na Tabela 1, são apresentadas as medidas de desempenho distância de geração e espalhamento para comparação dos casos rodados. Na Tabela 2, é mostrada a comparação do PAES-M com o NSGA-II e com o PAES original, apresentados em Deb (2001), revelando um desempenho bom para o PAES-M.

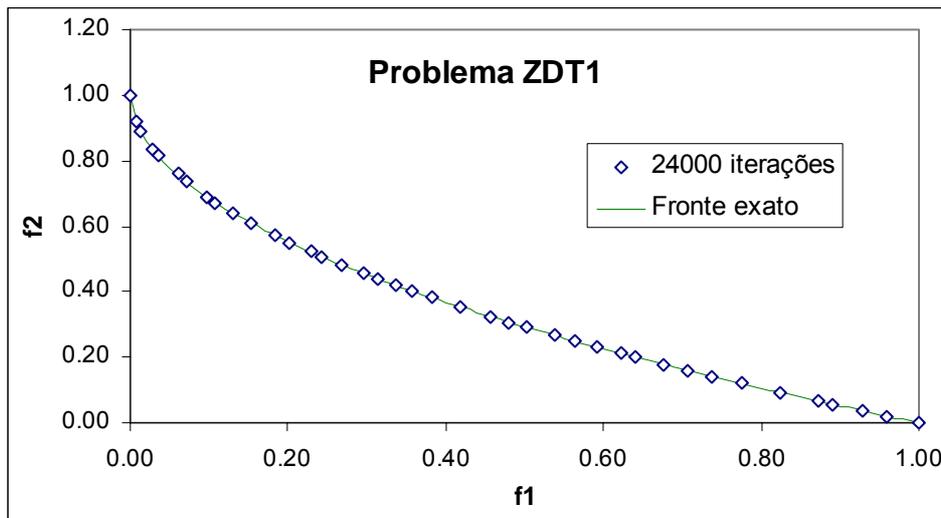


Figura 16 - Problema ZDT1 resolvido com 24000 iterações, 40 pontos no fronte (caso básico).

Tabela 1 - Problema ZDT1 - Medidas de desempenho: comparação entre simulações, para uma mesma semente.

Número de Iterações com PAES-M	Distância de geração (GD)	Espalhamento $\Delta$
24000 (caso básico)	0.000220	0.129089
3000	0.156935	0.665612
24000 (sem simulação individual)	0.174290	0.508011
24000 (s/ critério para limite de intervalo)	0.003766	0.095087

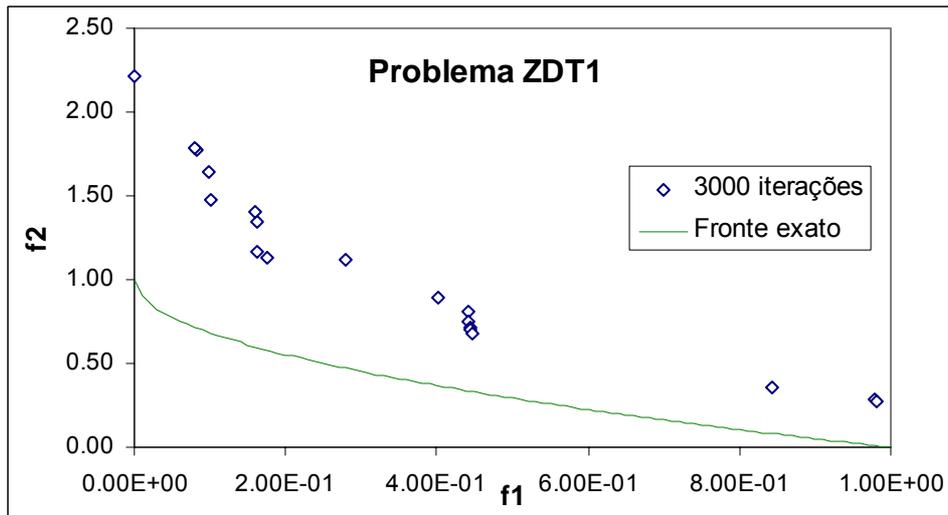


Figura 17 - Problema ZDT1 com 3000 iterações, 15 pontos no fronte.

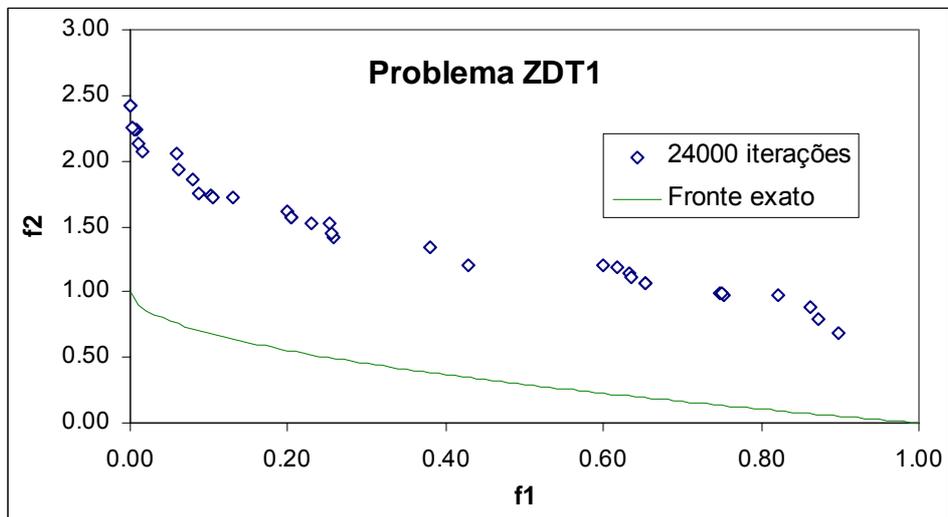


Figura 18 - Problema ZDT1 sem simulação individual por eixo de variável de projeto, com 24000 iterações.

Tabela 2 - Problema ZDT1 - Medidas de desempenho para 10 rodadas com diferentes sementes, comparação entre algoritmos.

Método PAES-M	Avaliações das funções objetivas	Distância de geração (GD)		Espalhamento $\Delta$	
		Média	Desvio padrão	Média	Desvio padrão
PAES-M	24000	0.000143	0.000200	0.137407	0.020873
PAES	250 x 100	0.082085	0.093161	1.229794	0.069563
NSGA-II código binário	250 x 100	0.000894	0.000000	0.463292	0.204015

O resultado satisfatório do PAES-M mostrou a adequação da estratégia adotada na rotina para incluir uma busca por melhores pontos por eixo de variável, pois conforme se percebeu da formulação do problema ZDT1, as variáveis  $x_2$  a  $x_{30}$  devem tender para zero simultaneamente para minimizar o somatório na função  $g(\mathbf{x})$ , o que não seria detectado de forma tão eficiente variando-se simultaneamente todas as variáveis em cada passo da simulação de forma independente a partir da solução parente corrente.

Nesse problema, os valores das variáveis  $x_2$  a  $x_{30}$  que levam ao fronte Pareto-ótimo coincidem com um dos limites do intervalo de simulação, zero. O critério de adotar o valor do limite do intervalo a cada certo número de iterações ou de ultrapassagens do limite foi importante para minimizar uma perturbação na região de  $f_1 = 0$ . Na Figura 19, é mostrado o fronte encontrado sem a adoção desse critério. É provável que o algoritmo encare essa região como uma transição brusca para uma região com restrição ( $f_1 \geq 0$ ).

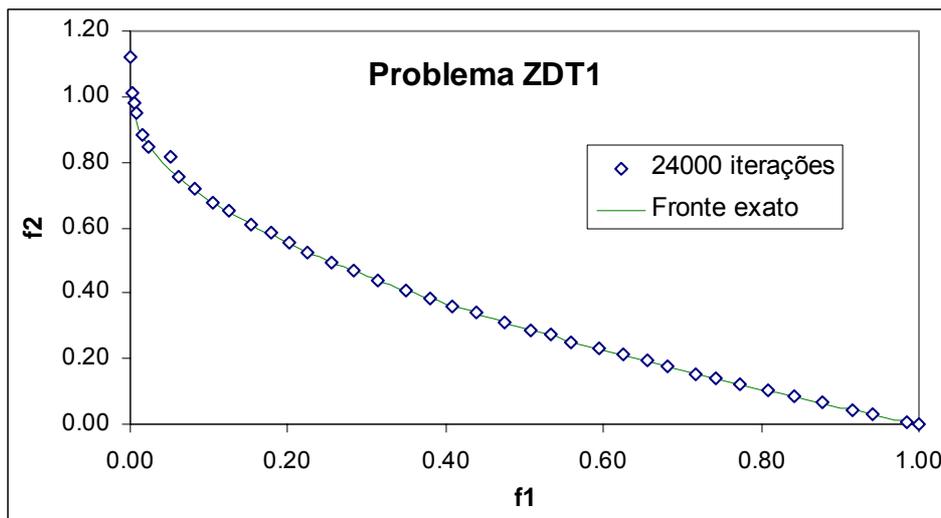


Figura 19 - Problema ZDT1 com 24000 iterações, sem utilizar critério para adotar limite do intervalo quando ultrapassado, frente com 40 pontos.

### 5.2.2. Problema ZDT2

É um problema com 30 variáveis de projeto ( $n=30$ ), com fronte de Pareto não-convexo, formulado pela expressão (5-6).

$$\text{ZDT2:} \begin{cases} f_1(\mathbf{x}) = x_1, \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, n = 30 \\ h(f_1, g) = 1 - (f_1/g)^2 \\ 0 \leq x_i \leq 1 \end{cases} \quad (5-6)$$

Os resultados estão mostrados na Figura 20, onde se pode perceber a boa aderência com a solução exata para o fronte Pareto-ótimo, dada pela expressão (5-7). Na Tabela 3, estão apresentadas as medidas de desempenho para diferentes algoritmos da literatura.

$$f_2 = 1 - f_1^2 \quad (5-7)$$

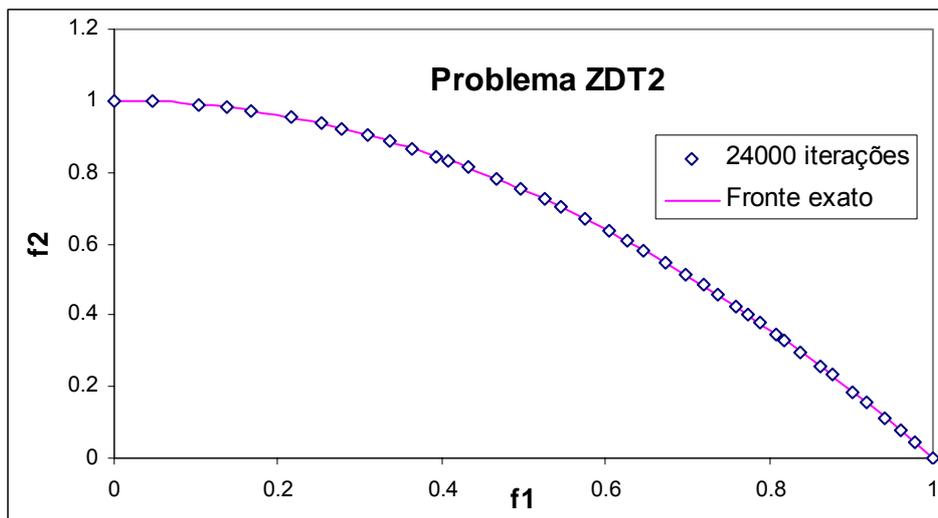


Figura 20 - Problema ZDT2 – Solução com 24000 iterações, frente com 40 pontos.

Tabela 3 - Problema ZDT2 - Medidas de desempenho para 10 rodadas com diferentes sementes, comparação entre algoritmos.

Método PAES-M	Avaliações das funções objetivas	Distância de geração (GD)		Espalhamento Δ	
		Média	Desvio padrão	Média	Desvio padrão
PAES-M	24000	0.000012	0.000022	0.137666	0.024722
PAES	250 x 100	0.126276	0.192034	1.165942	0.087647
NSGA-II código binário	250 x 100	0.000824	0.000000	0.435112	0.156866

### 5.2.3. Problema ZDT3

É um problema com 30 variáveis de projeto ( $n=30$ ), com frente de Pareto não-convexo, nesse caso descontínuo, dado pela expressão (5-8).

$$\text{ZDT3:} \begin{cases} f_1(\mathbf{x}) = x_1, \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, n = 30 \\ h(f_1, g) = 1 - \sqrt{f_1/g} - f_1/g \cdot \sin(10\pi f_1) \\ 0 \leq x_i \leq 1 \end{cases} \quad (5-8)$$

A solução exata do frente de Pareto, entrecortado com 5 trechos, é obtida de  $f_2 = 1 - \sqrt{f_1} - f_1 \cdot \sin(10\pi f_1)$ , nos trechos entre cada mínimo local e o ponto de  $f_1$  menor em que  $f_2$  tem a mesma ordenada que o mínimo local anterior. Pode-se notar na Figura 21 a boa concordância do frente encontrado com o frente exato, mesmo mantendo 40 pontos no frente entrecortado.

Na comparação com outros algoritmos, o proposto novamente apresenta bom desempenho, conforme mostrado na Tabela 4. Com relação ao espalhamento (*spread*), os resultados da literatura foram obtidos por cada segmento do frente e, então, calculada a média. Como o espalhamento médio para o PAES-M relativo a todo o frente já foi menor que o dos resultados da literatura, foi feito apenas um cálculo adicional do espalhamento só para o segundo trecho do frente, delimitando-se o intervalo de  $x_1$  por  $0.18 \leq x_1 \leq 0.30$ , para o que se nota uma redução significativa no espalhamento.

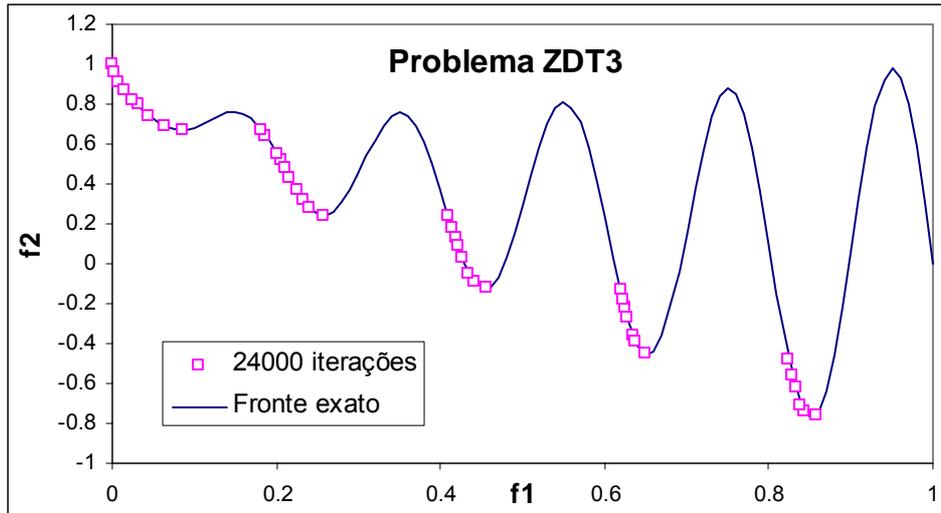


Figura 21 - Problema ZDT3 – Solução com 24000 iterações, frente com 40 pontos, PR = 0.8 a 0.4.

Tabela 4 - Problema ZDT3 Medidas de desempenho para 10 rodadas com diferentes sementes, comparação entre algoritmos.

Método PAES-M	Avaliações das funções objetivas	Distância de geração (GD)		Espalhamento Δ	
		Média	Desvio padrão	Média	Desvio padrão
PAES-M	24000	0.000572	0.000753	0.157638	0.025408
PAES	250 x 100	0.023872	0.003162	0.789920	0.040657
NSGA-II código binário	250 x 100	0.0434111	0.006481	0.575606	0.071260
PAES-M 0.18 ≤ x <sub>1</sub> ≤ 0.30	24000	0.000529	0.000407	0.166643	0.033158

#### 5.2.4. Problema ZDT4

É um problema com 10 variáveis de projeto (n=10), dado pela expressão (5-9), com frente de Pareto convexo, mas com 100 frentes ótimos locais, devendo o programa encontrar o frente ótimo global.

$$\text{ZDT4: } \begin{cases} f_1(\mathbf{x}) = x_1, \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)), n = 10 \\ h(f_1, g) = 1 - \sqrt{f_1/g} \\ 0 \leq x_1 \leq 1 \text{ e } -5 \leq x_i \leq 5, i = 2 \text{ a } 10 \end{cases} \quad (5-9)$$

Com as variáveis  $x_2$  a  $x_{10}$  no intervalo  $[-5,5]$  e  $x_1$  no intervalo  $[0,1]$ , há  $21^9$  soluções locais Pareto-ótimas, obtidas com passo 0.5 para as 9 variáveis  $x_2$  a  $x_{10}$ , independentes. Essas soluções formam 100 fronte Pareto-ótimos. A solução exata do fronte global ótimo de Pareto é obtida para o caso em que  $x_2$  a  $x_{10}$  assumem todas valor zero, quando então  $g(x) = 1$ . O fronte seguinte seria obtido com uma das variáveis igual a  $\frac{1}{2}$ , com as demais iguais a zero, levando a  $g(x) = 1.25$ . O fronte seguinte, com duas das variáveis de 2 a 10 iguais a  $\frac{1}{2}$  e as demais iguais a zero, levando a  $g(x) = 1.5$ . O algoritmo mais uma vez se mostrou robusto, conforme a Figura 22 e a Tabela 5.

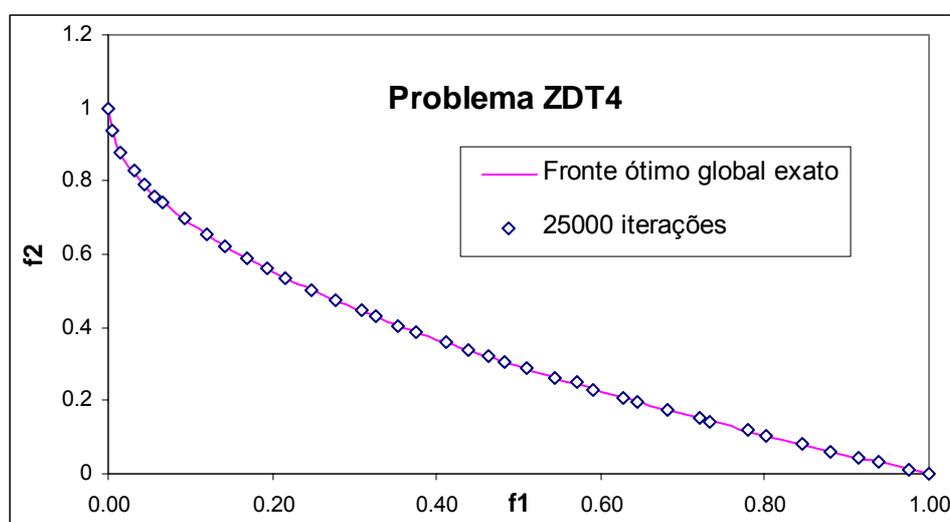


Figura 22 - Problema ZDT4 – Solução com 25000 iterações, frente com 40 pontos, adota limite de intervalo, se ultrapassado, a cada 5 iterações na simulação individual.

Tabela 5 - Problema ZDT4 - Medidas de desempenho para 10 rodadas com diferentes sementes.

Método PAES-M	Avaliações das funções objetivas	Distância de geração (GD)		Espalhamento $\Delta$	
		Média	Desvio padrão	Média	Desvio padrão
PAES-M	24000	0.003185	0.009310	0.139004	0.023742
PAES	250 x 100	0.854816	0.726112	0.870458	0.318432
NSGA-II código binário	250 x 100	3.227636	2.703263	0.479475	0.099202
NSGA-II variáveis reais	250 x 100	0.5130530	0.344180	0.702612	0.254260

### 5.2.5. Problema ZDT6

É um problema com 10 variáveis de projeto ( $n = 10$ ), dado pela expressão (5-10), com frente de Pareto não-convexo, com densidade de soluções ao longo da região Pareto-ótima não-uniforme. O fronte encontrado é apresentado na Figura 23, mostrando aderência ao fronte exato. As medidas de desempenho são apresentadas na Tabela 6, com valores bastante satisfatórios.

$$\text{ZDT6: } \begin{cases} f_1(\mathbf{x}) = 1 - \exp(-4x_1) \text{sen}^6(6\pi x_1), \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + 9 \left[ \left( \sum_{i=2}^n x_i^2 \right) / 9 \right]^{0.25}, n = 10 \\ h(f_1, g) = 1 - (f_1 / g)^2 \\ 0 \leq x_i \leq 1 \end{cases} \quad (5-10)$$

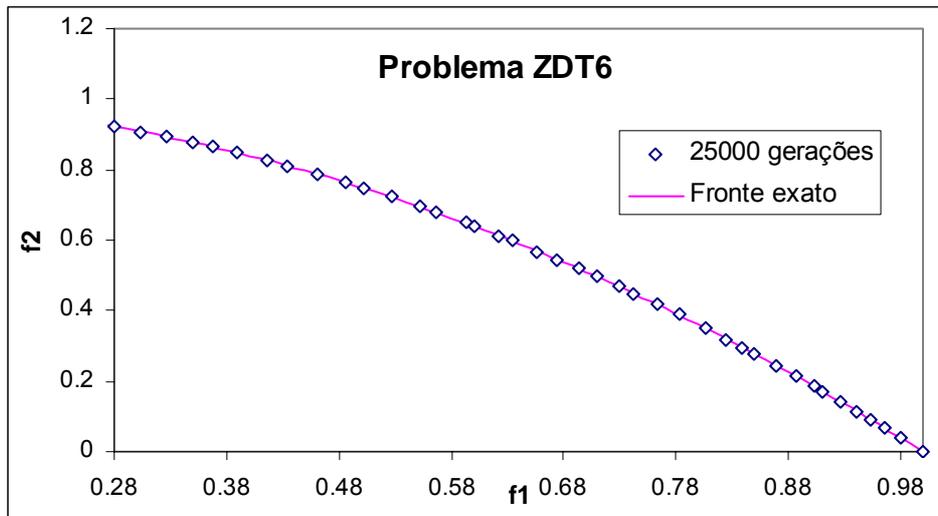


Figura 23 - Problema ZDT6 – Solução com 25000 iterações, frente com 40 pontos.

Conforme descrito em Deb (2001), para valores igualmente espaçados de  $x_1$ , com os valores de  $x_2$  a  $x_{10}$  igualados a zero, as soluções do fronte tendem a se concentrar na extremidade direita da curva. Apesar disso, o programa conseguiu um excelente espalhamento nas soluções em boa parte das rodadas, observando-se que o menor valor possível de  $f_1$  se situa em torno de 0.28. Entretanto, houve casos de concentração dos

pontos à direita do fronte em algumas das rodadas, o que se refletiu nas estatísticas encontradas.

Tabela 6 - Problema ZDT6 - Medidas de desempenho para 10 rodadas com diferentes sementes, comparação entre algoritmos.

Método PAES-M	Avaliações das funções objetivas	Distância de geração (GD)		Espalhamento $\Delta$	
		Média	Desvio padrão	Média	Desvio padrão
PAES-M	24000	0.022886	0.048919	0.319951	0.326727
PAES	250 x 100	0.085469	0.081633	1.153052	0.062578
NSGA-II código binário	250 x 100	7.806798	0.040829	0.644477	0.187195
NSGA-II variáveis reais	250 x 100	0.296564	0.114608	0.668025	0.099614

### 5.3. Problemas Multiobjetivo com Restrições

Serão apresentados a seguir 2 problemas com 2 funções objetivas, encontrados em Deb (2001), problemas TNK e MEC (*Constr-Ex*).

#### 5.3.1. Problema TNK

Proposto por Tanaka, encontrado em Deb (2001), sua formulação é dada pela expressão (5-11).

$$\left\{ \begin{array}{l}
 \text{Minimizar } f_1(x) = x_1 \\
 \text{Minimizar } f_2(x) = x_2, \\
 \text{sujeito a } C_1(x) \equiv x_1^2 + x_2^2 - 1 - 0.1 \cos(16 \arctan(\frac{x_1}{x_2})) \leq 0 \\
 C_2(x) \equiv (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \\
 0 \leq x_1 \leq \pi, 0 \leq x_2 \leq \pi
 \end{array} \right. \quad (5-11)$$

A região válida para busca de soluções no espaço das funções objetivas, nesse caso também as variáveis de projeto, está entre as curvas que representam as restrições 1 e 2,

conforme a Figura 24. Nela, também é apresentado o fronte Pareto-ótimo encontrado, já com a implementação da busca automática por pontos viáveis, mostrando excelente aderência ao fronte exato. As medidas de desempenho são apresentadas na Tabela 7, para 30 rodadas com 50000 e 200000 iterações, mostrando pequena variação entre elas, podendo-se inferir, juntamente com o aspecto do fronte, que estão adequadas. O espalhamento foi calculado apenas para o trecho central do fronte segmentado.

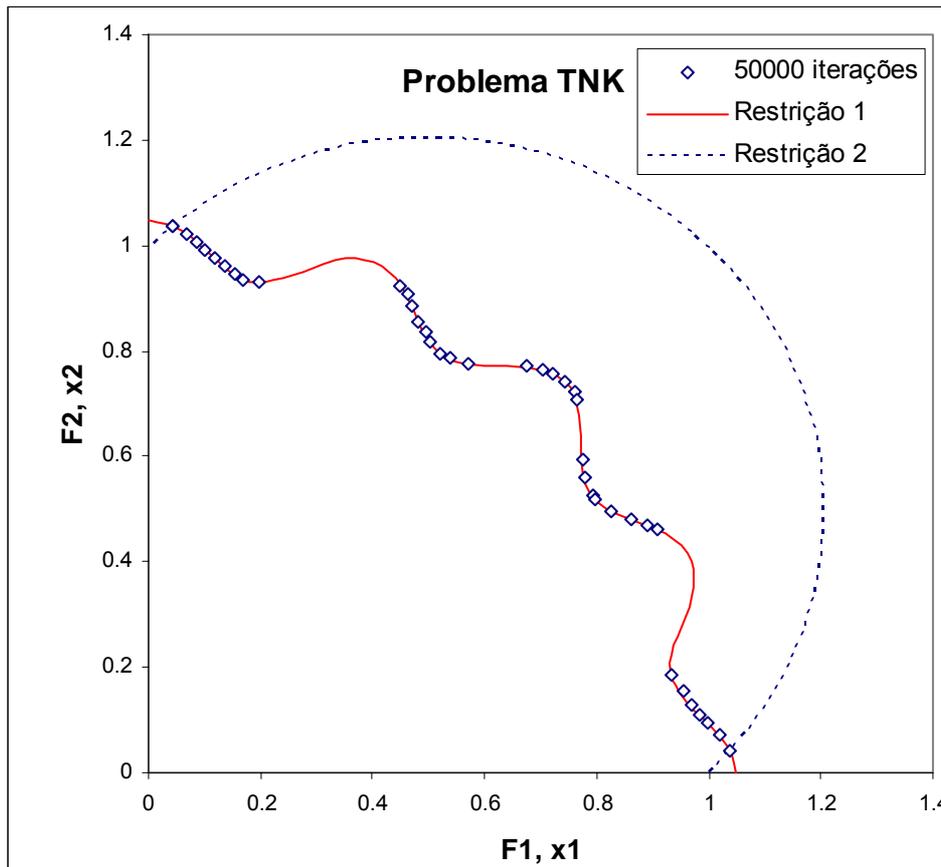


Figura 24 – Problema TNK – simulação com 50000 iterações, fronte com 40 pontos, PR de 0.8 a 0.4, sem tolerância para as restrições.

Tabela 7 – Problema TNK – Medidas de desempenho para 30 rodadas com diferentes sementes, PR 0.8 a 0.4.

Método PAES-M	Distância de geração (GD)		Espalhamento $\Delta$ (trecho central)	
	Média	Desvio padrão	Média	Desvio padrão
50000 iterações	0.111523	0.002243	0.161537	0.038487
200000 iterações	0.111288	0.001573	0.134559	0.021200

### 5.3.2. Problema MEC (*Constr-Ex*)

É o problema Min-Ex, utilizado largamente em Deb (2001), mas com 2 restrições. Sua formulação é dada pela expressão (5-12). A região válida para busca de soluções no espaço das variáveis de projeto está localizada como mostrado na Figura 25.

$$\left\{ \begin{array}{l} \text{Minimizar } f_1(x) = x_1 \\ \text{Minimizar } f_2 = \frac{1+x_2}{x_1}, \\ \text{sujeito a } g_1(x) \equiv -x_2 - 9x_1 + 6 \leq 0 \\ \quad \quad \quad g_2(x) \equiv x_2 - 9x_1 + 1 \leq 0 \\ 0.1 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 5 \end{array} \right. \quad (5-12)$$

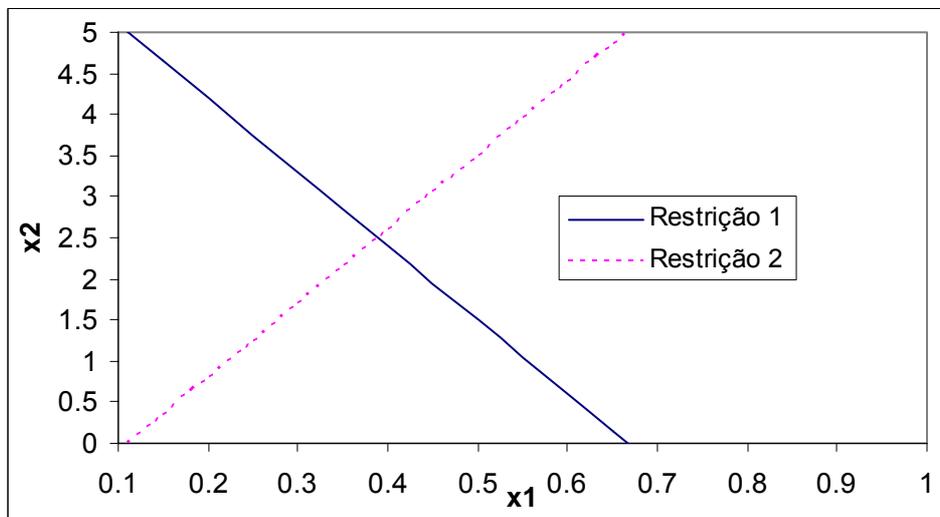


Figura 25 - Delimitação do domínio de validade das variáveis de projeto, localizado na interseção das regiões à direita dos segmentos de reta que definem as restrições 1 e 2.

O fronte de Pareto para o problema Min-Ex sem restrições é dado por uma hipérbole  $f_2 = 1/f_1$ . Com a restrição 1, há uma interrupção no fronte original, para seguir a

fronteira por ela definida. Pela Figura 26, o fronte Pareto-ótimo atingido foi satisfatório, com alguma perturbação nas regiões de transição causadas pelas restrições. Na Tabela 8, são apresentadas medidas de desempenho para a rodada realizada para 8000 e 24000 iterações. Como a diferença entre as medidas foi pequena e o aspecto do fronte é satisfatório, supõe-se que estejam adequadas.

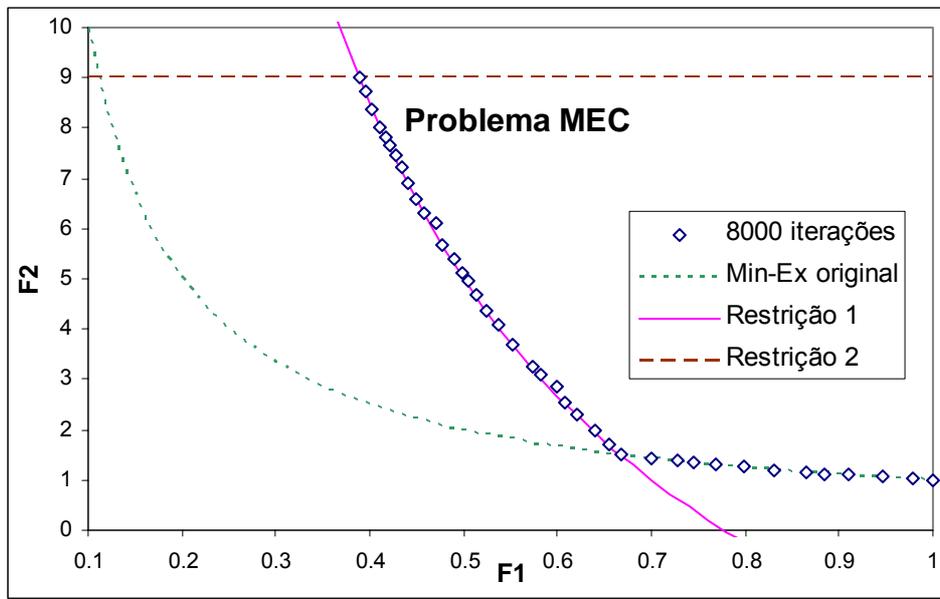


Figura 26 - Problema MEC (*Constr-Ex*) – Solução com 8000 iterações, com 40 pontos no fronte, PR 0.8 a 0.4, sem tolerância para as restrições.

Tabela 8 – Problema MEC – Medidas de desempenho para 30 rodadas com diferentes sementes, PR 0.8 a 0.4.

Método PAES-M	Distância de geração (GD)		Espalhamento $\Delta$	
	Média	Desvio padrão	Média	Desvio padrão
8000 iterações	0.464124	0.023285	0.151443	0.034012
24000 iterações	0.468321	0.012583	0.133778	0.015496

#### 5.4. Problemas Uniobjetivo com Restrições

A seguir, são apresentados alguns dos problemas de um único objetivo que foram utilizados como exemplo para o procedimento descrito em Mezura-Montes e Coello (2005) (Problemas g01, g02, g03, g06, g08, g09, g11 e g13). Não foram utilizadas todas as funções para evitar estender demais o trabalho, mesmo porque a comunidade

internacional sobre o tema já incrementou o número de funções para 24, que poderão ser abordadas futuramente. De qualquer forma, já se pôde ter uma idéia do comportamento do algoritmo. O procedimento utilizado para o artigo, chamado de SMES (*Simple Mulimembered Evolution Strategy*), é um método  $(\mu + \lambda)$ -ES, que utilizou  $\mu = 100$ ,  $\lambda = 300$  e 800 gerações, totalizando 240000 avaliações das funções objetivas. O SMES teve um desempenho excepcional em todos os exemplos. Assim, buscou-se trabalhar com o número de iterações com o PAES-M em torno desse número de avaliações. Quanto à probabilidade de referência PR, procurou-se trabalhar com decaimento linear de 0.4 a 0.0, o que forneceu, em geral, bons resultados, e com testes de sensibilidade nos problemas g01 e g02.

#### 5.4.1. Problema g01

$$\left\{ \begin{array}{l}
 \text{Minimizar } f(\mathbf{x}) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i, \\
 \text{sujeito a:} \\
 g_1(\mathbf{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
 g_2(\mathbf{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
 g_3(\mathbf{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
 g_4(\mathbf{x}) = -8x_1 + x_{10} \leq 0 \\
 g_5(\mathbf{x}) = -8x_2 + x_{11} \leq 0 \\
 g_6(\mathbf{x}) = -8x_3 + x_{12} \leq 0 \\
 g_7(\mathbf{x}) = -2x_4 - x_5 + x_{10} \leq 0 \\
 g_8(\mathbf{x}) = -2x_6 - x_7 + x_{11} \leq 0 \\
 g_9(\mathbf{x}) = -2x_8 - x_9 + x_{12} \leq 0, \\
 \text{com os limites } 0 \leq x_i \leq 1 \text{ (} i = 1, \dots, 9 \text{), } 0 \leq x_i \leq 100 \text{ (} i = 10, 11, 12 \text{) e} \\
 0 \leq x_{13} \leq 1
 \end{array} \right. \quad (5-13)$$

O ótimo global está situado em  $\mathbf{x}^* = (1,1,1,1,1,1,1,1,3,3,3,1)$ , onde  $f(\mathbf{x}^*) = -15$ . Nesse ponto, as restrições  $g_1$  a  $g_3$  e  $g_6$  a  $g_9$  estão ativas (estão sobre a fronteira  $g_i(\mathbf{x}) = 0$ ). O programa PAES-M (apenas a parte de Estratégia Evolutiva do programa, já que o fronte se resume a um ponto) encontrou essa solução para algumas sementes de partida e diferentes probabilidades de referência PR (inicial e final) de escolha dos desvios padrões maior ou menor. Na Tabela 9, são mostrados os resultados para a função

objetiva e as restrições, com 234000 iterações, para uma mesma semente e diferentes probabilidades de referência PR inicial e final em cada rodada. Notou-se que, para esse problema, é possível que o algoritmo fique preso num ponto, necessitando de que duas variáveis específicas (uma delas  $x_{10}$  a  $x_{12}$ ) ao mesmo tempo dêem um salto para levarem a pontos melhores atendendo às restrições. Cabe notar que a solução para as variáveis  $x_{10}$  a  $x_{12}$  é de valor 3, que está próxima do limite inferior do intervalo de 0 a 100 de cada uma. De qualquer modo, com as taxas utilizadas para o critério de adotar o limite do intervalo, quando ultrapassado, mencionadas no Capítulo 3, bem como com a tolerância  $\varepsilon$  nas restrições de 0.05 a 0.00005, os resultados foram satisfatórios.

Tabela 9 - Problema g01 – resultados para uma mesma semente e diferentes probabilidades de referência PR inicial e final em cada rodada, tolerância  $\varepsilon$  nas restrições 0.05 a 0.00005 (inicial e final em cada rodada), 234000 iterações.

Função objetiva e restrições	Probabilidade de referência PR				
	0.2-0.0	0.8-0.2	0.4-0.0	0.8-0.0	0.4-0.2
$f_1$	-14.9947	-14.9907	-14.9894	-14.9956	-14.9862
$g_1$	-0.002923	-0.006900	-0.004469	-0.003641	-0.004966
$g_2$	-0.004880	-0.007041	-0.009248	-0.002611	-0.009956
$g_3$	-0.002867	-0.004705	-0.007409	-0.002553	-0.012760
$g_4$	-5.002470	-5.004620	-5.003150	-5.001850	-5.001080
$g_5$	-5.000460	-5.002280	-5.001310	-5.001790	-5.003880
$g_6$	-5.002410	-5.002420	-5.006090	-5.000760	-5.008870
$g_7$	-0.002468	-0.004618	-0.003154	-0.001850	-0.001082
$g_8$	-0.000455	-0.002282	-0.001315	-0.001791	-0.003885
$g_9$	-0.002412	-0.002423	-0.006094	-0.000761	-0.008875

Para dar uma idéia da contribuição da simulação individual por variável de projeto durante a otimização, ela foi retirada nesse exemplo, obtendo-se os seguintes resultados para a função objetiva, nos casos 1 a 5 acima, respectivamente: -14.3624, -13.9856, -14.3562, -14.5080 e -14.4095. Isso mostra que essa contribuição é útil para a convergência.

Na Tabela 10, são apresentadas as estatísticas dos resultados para a probabilidade de referência PR variando de 0.4 a 0.0, tolerância  $\varepsilon$  nas restrições de 0.05 a 0.00005 (inicial e final em cada rodada), com 234000 iterações.

Tabela 10 - Problema g01 – estatísticas para 30 rodadas com diferentes sementes, cada uma com probabilidade de referência PR 0.4 a 0.0, tolerância  $\varepsilon$  nas restrições de 0.05 a 0.00005, 234000 iterações.

$\sigma_2$ nas 2925 últimas iterações	Valores da função objetiva			
	Média	Desvio padrão	Menor	Maior
Não-adaptativo	-14.9845	0.043583	-14.9993	-14.7561
Adaptativo	-14.9730	0.141589	-14.9999	-14.2234

#### 5.4.2. Problema g02

$$\left\{ \begin{array}{l}
 \text{Minimizar } f(\mathbf{x}) = - \frac{\left| \sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i) \right|}{\sqrt{\sum_{i=1}^n i \cdot x_i^2}}, \\
 \text{sujeito a:} \\
 g_1(\mathbf{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0 \\
 g_2(\mathbf{x}) = \sum_{i=1}^n x_i - 7.5n \leq 0, \\
 \text{onde } n = 20 \text{ e } 0 \leq x_i \leq 10 \text{ (} i = 1, \dots, n \text{)}.
 \end{array} \right. \quad (5-14)$$

O mínimo global é desconhecido; a melhor solução já encontrada foi  $f(\mathbf{x}^*) = -0.803619$ , com a restrição  $g_1$  quase ativa ( $g_1 = -10^{-8}$ ). O melhor valor atingido pelo método SMES após 240000 iterações foi -0.803601 e outros métodos também levaram a valores próximos. Pela Estratégia Evolutiva do programa PAES-M, o melhor valor encontrado para 240000 iterações, para as rodadas realizadas, foi  $f(\mathbf{x}^*) = -0.802531$ .

Na Tabela 11, são apresentados os resultados para diferentes decaimentos das probabilidades de referência PR de escolha dos desvios padrões  $\sigma_1$  e  $\sigma_2$  ao longo de cada rodada. Para cada caso, foram geradas estatísticas a partir de 30 rodadas, permitindo-se tolerância  $\varepsilon$  nas restrições. Nota-se uma certa distância nos valores

médios da função em relação à melhor solução e pode-se detectar uma tendência para deterioração no ótimo encontrado com valores de PR inicialmente mais altos.

Tabela 11 - Problema g02 – resultados para diferentes probabilidades de referencia PR, estatísticas para 30 rodadas com 240000 iterações cada, com tolerância  $\varepsilon$  para as restrições de 0.001 a 0.00005, sem  $\sigma_2$  adaptativo.

Função objetiva e restrições		Probabilidade de referência PR			
		0.2-0.0	0.4-0.0	0.8-0.0	0.8-0.4
f	Média	-0.777155	-0.781307	-0.766619	-0.630583
	Desvio Padrão	0.020430	0.015143	0.021666	0.035303
	Menor	-0.792832	-0.801794	-0.801304	-0.701036
	Maior	-0.694350	-0.730630	-0.700330	-0.538729
g <sub>1</sub>	Média	-0.004156	-0.005116	-0.015675	-0.396682
	Desvio Padrão	0.007150	0.007264	0.048890	0.700029
	Menor	-0.029931	-0.022533	-0.266624	-2.925874
	Maior	0.000048	0.000047	0.000048	0.000039
g <sub>2</sub>	Média	-118.52	-118.12	-117.39	-109.62
	Desvio Padrão	1.68	1.40	1.79	4.55

Na Tabela 12, são apresentados os resultados para algumas combinações de casos (com ou sem tolerância  $\varepsilon$  nas restrições e emprego ou não de  $\sigma_2$  adaptativo nas últimas iterações, além de incremento no número de iterações), fixando-se o decaimento de PR em 0.4-0.0.

Tabela 12 Problema g02 – Estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0.

Número de iterações Niter	$\sigma_2$ nas Niter/80 últimas iterações	Tolerância $\varepsilon$ 0.001-0.00005	Valores da função objetiva			
			Média	Desvio padrão	Menor	Maior
240000	Não-adapt.	Não	-0.780537	0.013674	-0.802208	-0.752803
240000	Adaptativo	Não	-0.781026	0.013612	-0.802752	-0.753154
240000	Não-adapt.	Sim	-0.781307	0.015143	-0.801794	-0.730630
240000	Adaptativo	Sim	-0.781863	0.015118	-0.802402	-0.730729
720000	Não-adapt.	Não	-0.791405	0.009956	-0.802485	-0.768563
720000	Adaptativo	Sim	-0.790390	0.007379	-0.802650	-0.777490

O uso do desvio padrão  $\sigma_2$  adaptativo nas últimas iterações, bem como o emprego de tolerância nas restrições não proporcionaram, neste exemplo, melhoras significativas nos resultados. O incremento no número de iterações em cada rodada proporcionou apenas uma pequena melhora nos resultados. Apesar de se manter a distância da média à melhor solução numa mesma faixa, observaram-se diferenças consideráveis em algumas das variáveis de projeto.

Esse resultado sugere que se aprofunde a investigação do algoritmo.

### 5.4.3. Problema g03

$$\left\{ \begin{array}{l} \text{Minimizar } f(\mathbf{x}) = -\left(\sqrt{n}\right)^n \prod_{i=1}^n x_i, \\ \text{sujeito a:} \\ h(\mathbf{x}) = \sum_{i=1}^n x_i^2 - 1 = 0, \\ \text{onde } n = 10 \text{ e } 0 \leq x_i \leq 1 \text{ (} i = 1, \dots, n \text{).} \end{array} \right. \quad (5-15)$$

É um problema com restrição do tipo igualdade, para o que foi utilizado o procedimento de relaxá-la com uma tolerância  $\varepsilon$  variando de 0.001 a 0.00005. A solução global está em  $x_i^* = 1/\sqrt{n} = 0.31623$ , ( $i = 1, \dots, n$ ) e  $f(\mathbf{x}^*) = -1$ . Pelo programa PAES-M (a parte de Estratégia Evolutiva), o valor encontrado mais próximo do ótimo, com 240000 iterações, foi de -0.999994, com probabilidades de referência PR variando linearmente de 0.4 a 0. Os valores das variáveis de projeto nesse caso se situaram em torno das coordenadas do minimante da solução exata, o que pode ser visto na Tabela 13, e a restrição  $h$  se situou em  $4.9031 \times 10^{-5}$ . Na Tabela 14, são apresentadas estatísticas para 30 rodadas, apontando para convergência para a solução. O resultado, portanto, foi bastante satisfatório.

Tabela 13 Problema g03 – Valores das variáveis de projeto da melhor solução encontrada e das estatísticas para 30 rodadas sem  $\sigma_2$  adaptativo, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 0.001 a 0.00005.

Variável	Melhor Solução Encontrada	Média de 30 rodadas	Desvio padrão de 30 rodadas
$x_1$	0.313687	0.316213	0.003598
$x_2$	0.317494	0.315590	0.002919
$x_3$	0.316453	0.314276	0.008201
$x_4$	0.316567	0.315594	0.002494
$x_5$	0.316408	0.316365	0.003935
$x_6$	0.315873	0.315968	0.002777
$x_7$	0.314515	0.318077	0.004038
$x_8$	0.319524	0.316176	0.002763
$x_9$	0.315838	0.316404	0.002847
$x_{10}$	0.315961	0.317319	0.007945

Tabela 14 Problema g03 – estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 0.001 a 0.00005.

$\sigma_2$ nas 3000 últimas iterações	Valores da função objetiva			
	Média	Desvio padrão	Menor	Maior
Não-adaptativo	-0.998008	0.007337	-0.999994	-0.959222
Adaptativo	-0.998041	0.007347	-0.999981	-0.959222

#### 5.4.4. Problema g06

$$\left\{ \begin{array}{l}
 \text{Minimizar } f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3, \\
 \text{sujeito a:} \\
 \quad g_1(\mathbf{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \text{ e} \\
 \quad g_2(\mathbf{x}) = (x_1 - 6)^2 - (x_2 - 5)^2 - 82.81 \leq 0, \\
 \text{onde } 13 \leq x_1 \leq 100 \text{ e } 0 \leq x_2 \leq 100.
 \end{array} \right. \quad (5-16)$$

Para se ter uma idéia da dificuldade desse problema, na Figura 27 é mostrada a região do domínio de validade das variáveis de projeto, ou seja, onde o otimizador pode encontrar soluções que atendam às 2 restrições impostas. Trata-se da estreita fenda da figura, delimitada por 2 arcos de círculo que são as curvas de nível das funções de restrição onde assumem o valor zero, conforme mostrado na Figura 28. Na região abaixo do cruzamento das linhas dessa figura, uma ou ambas as funções de restrição assumem valores positivos e portanto violam as restrições.

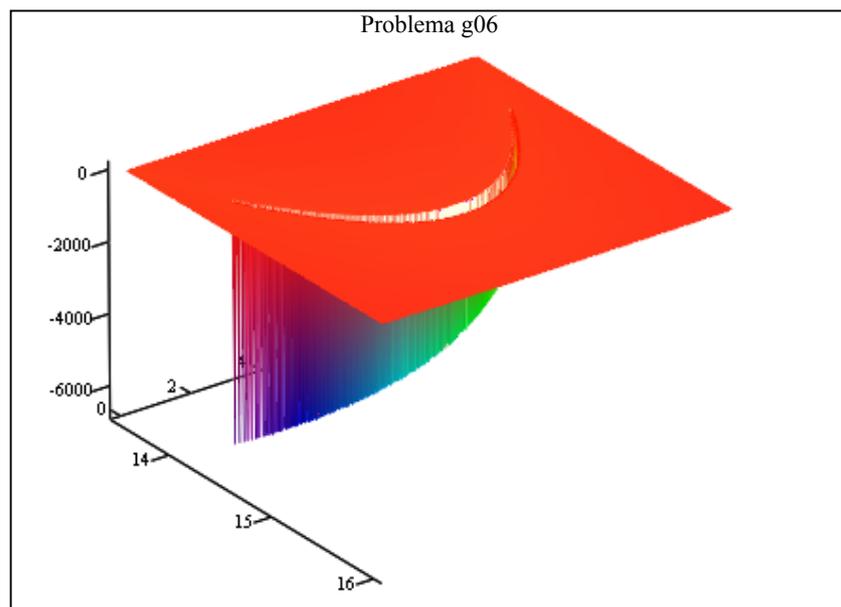


Figura 27 - Domínio de validade das variáveis para a função do problema g06.

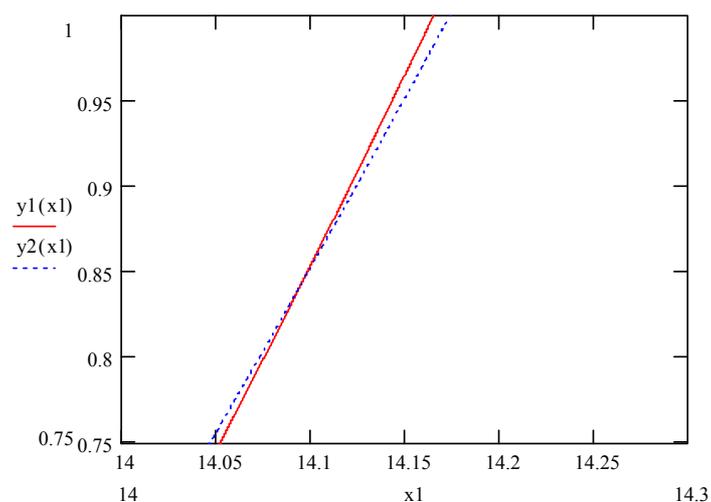


Figura 28 - Curvas de nível das funções de restrição g1 (linha cheia) e g2 (tracejada), onde as  $g1(x_1, x_2) = 0$  e  $g2(x_1, x_2) = 0$ .

Para esse problema, o ótimo global está em  $x^* = (14.095, 0.84296)$  e  $f(x^*) = -6961.81388$ , com as 2 restrições ativas. Pelo método PAES-M (parte de Estratégia Evolutiva), com 240000 iterações, tolerância  $\varepsilon$  nas restrições variando de 0.001 a 0.00005, probabilidade de referência de 0.4 a 0.0, o valor encontrado mais próximo da solução, a partir de 30 rodadas com desvio padrão adaptativo  $\sigma_2$  nas 3000 últimas iterações, foi  $f(x^*) = -6961.839$ , para  $x^* = (14.094991, 0.842938)$ , com  $g_1(x) = -0.217822 \times 10^{-4}$  e  $g_2(x) = 0.400568 \times 10^{-4}$ . O valor encontrado é menor que o ótimo exato, mas a segunda restrição é violada, dentro da tolerância estabelecida para as rodadas. Na Tabela 15, são apresentadas as estatísticas dos resultados com e sem  $\sigma_2$  adaptativo nas 3000 últimas iterações. Pode-se perceber, para esse problema, que o desvio padrão adaptativo melhorou a convergência para a solução.

Tabela 15 Problema g06 – estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\varepsilon$  de 0.001 a 0.00005.

$\sigma_2$ nas 3000 últimas iterações	Valores da função objetiva			
	Média	Desvio padrão	Menor	Maior
Não-adaptativo	-6948.477	6.5466	-6958.707	-6924.094
Adaptativo	-6961.394	0.2839	-6961.855	-6960.790

#### 5.4.5. Problema g08

$$\left\{ \begin{array}{l}
 \text{Maximizar } f(\mathbf{x}) = \frac{\text{sen}^3(2\pi x_1)\text{sen}(2\pi x_2)}{x_1^3(x_1 + x_2)}, \\
 \text{sujeito a:} \\
 \quad g_1(\mathbf{x}) = x_1^2 - x_2 + 1 \leq 0 \text{ e} \\
 \quad g_2(\mathbf{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0, \\
 \text{onde } 0 \leq x_1 \leq 10 \text{ e } 0 \leq x_2 \leq 10.
 \end{array} \right. \quad (5-17)$$

O ótimo global está em  $x^* = (1.2279713, 4.2453733)$  e  $f(x^*) = 0.095825$ . Pelo PAES-M (parte de Estratégia Evolutiva), com 60000 iterações apenas, conseguiu-se atingir a solução exata, com uma ótima precisão nas variáveis de projeto. Entretanto, nesse caso as restrições não estão ativas ( $g_i(x) < 0$  na solução ótima). Além disso, os

métodos citados em Mezura-Montes e Coello (2005) também atingiram a solução desse problema muito rapidamente.

### 5.4.6. Problema g09

É um problema com 4 restrições do tipo inequação, definido por:

$$\left\{ \begin{array}{l}
 \text{Minimizar } f(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + \\
 \quad + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\
 \text{sujeito a:} \\
 \quad g_1(\mathbf{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \text{ e} \\
 \quad g_2(\mathbf{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\
 \quad g_3(\mathbf{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
 \quad g_4(\mathbf{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0, \\
 \text{onde } -10 \leq x_i \leq 10, i = 1,7.
 \end{array} \right. \quad (5-18)$$

O ótimo global é  $f(\mathbf{x}^*) = 680.6300573$ . Com o método PAES-M (parte de Estratégia Evolutiva), após 280000 iterações, atingiu-se o melhor valor de  $f(\mathbf{x}^*) = 680.8537$ . A diferença entre as variáveis de projeto encontradas e as da solução exata pode ser vista na Tabela 16. Nesse ponto, as restrições que devem estar ativas assumem os valores  $g_1 = -1.00633 \times 10^{-4}$  e  $g_4 = -2.4137 \times 10^{-3}$ . Na Tabela 17, são apresentadas as estatísticas para 30 rodadas, revelando um bom desempenho.

Tabela 16 - Problema g09 – Valores das variáveis de projeto para a melhor solução, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 0.001 a 0.00005.

Variável	Solução Encontrada	Solução Exata
$x_1$	2.322337	2.330499
$x_2$	1.966269	1.951372
$x_3$	-0.316931	-0.47754144
$x_4$	4.321575	4.365726
$x_5$	-0.603302	-0.6244870
$x_6$	1.177430	1.038131
$x_7$	1.620967	1.594227

Tabela 17 - Problema g09 – estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 0.001 a 0.00005.

$\sigma_2$ nas 3500 últimas iterações	Valores da função objetiva			
	Média	Desvio padrão	Menor	Maior
Não-adaptativo	681.1876	0.335168	680.8537	682.5967
Adaptativo	681.2040	0.372149	680.8537	682.8270

### 5.4.7. Problema g11

$$\left\{ \begin{array}{l} \text{Minimizar } f(\mathbf{x}) = x_1^2 + (x_2 - 1)^2, \\ \text{sujeito a:} \\ \quad h(\mathbf{x}) = x_2 - x_1^2 = 0, \\ \text{onde } -1 \leq x_1 \leq 1 \text{ e } -1 \leq x_2 \leq 1. \end{array} \right. \quad (5-19)$$

Para dar uma idéia da dificuldade desse problema, na Figura 29 é mostrada a região viável das variáveis do problema, que está situada ao longo trilha criada quando se relaxa a restrição pelo valor  $\epsilon$ .

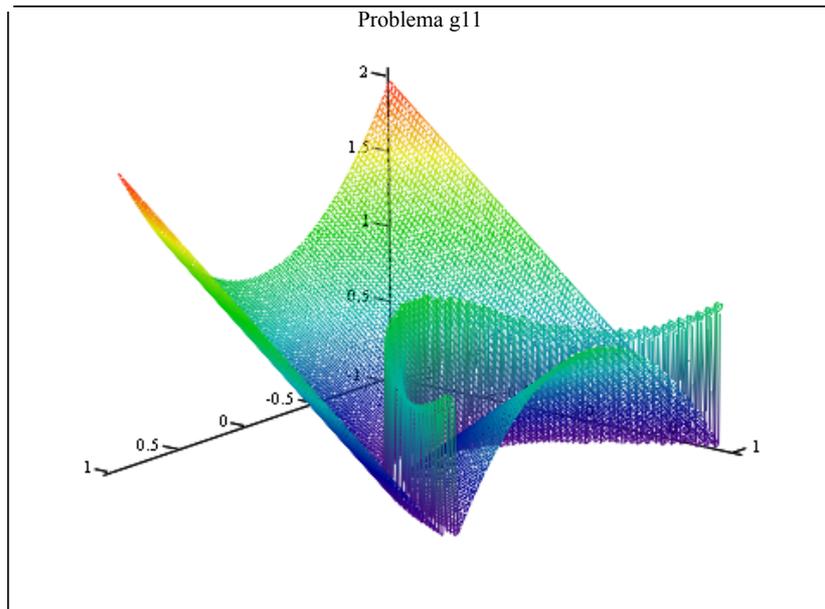


Figura 29 - Problema g11 - região viável situada ao longo da trilha formada quando a restrição é relaxada de um valor  $\epsilon$ .

O ótimo global está em  $x^* = (\pm 1/\sqrt{2}, 1/2) = (\pm 0.707, 0.5)$ , onde  $f(x^*) = 0.75$ . Com o PAES-M (parte de Estratégia Evolutiva), após 240000 iterações, atingiu-se o melhor valor de 0.749950, com a restrição em  $4.994672 \times 10^{-5}$ , com as coordenadas  $x^* = (0.707039, 0.499954)$ , o que é um excelente resultado. Para isso, foi adotada a probabilidade de referência PR variando linearmente de 0.4 a 0.0, a tolerância  $\varepsilon$  na restrição h variando linearmente de 0.001 a 0.00005. Foram encontrados pontos em torno de ambas as soluções. Na Tabela 18, são apresentadas as estatísticas para 30 rodadas, revelando um boa convergência para a solução.

Tabela 18 Problema g11 – estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\varepsilon$  de 0.001 a 0.00005.

$\sigma_2$ nas 3000 últimas iterações	Valores da função objetiva			
	Média	Desvio padrão	Menor	Maior
Não-adaptativo	0.749961	0.000012	0.749960	0.749989
Adaptativo	0.749957	0.000012	0.749950	0.750010

#### 5.4.8. Problema g13

$$\left\{ \begin{array}{l}
 \text{Minimizar } f(\mathbf{x}) = \exp(x_1 x_2 x_3 x_4 x_5), \\
 \text{sujeito a:} \\
 h_1(\mathbf{x}) = \sum_{i=1}^5 x_i^2 - 10 = 0, \\
 h_2(\mathbf{x}) = x_2 x_3 - 5x_4 x_5 = 0 \text{ e} \\
 h_3(\mathbf{x}) = x_1^3 + x_2^3 + 1 = 0 \\
 \text{onde } -2.3 \leq x_i \leq 2.3 \text{ (} i = 1,2 \text{) e } -3.2 \leq x_i \leq 3.2 \text{ (} i = 1,4,5 \text{)}
 \end{array} \right. \quad (5-20)$$

Para esse problema, o valor ótimo global exato é  $f(x^*) = 0.0539498$ . Na Tabela 19, estão apresentados os resultados obtidos com o programa PAES-M (parte de Estratégia Evolutiva) após 250000 iterações, permitindo-se uma tolerância  $\varepsilon$  nas restrições variando de 1.0 a 0.00005, com a probabilidade de referência PR variando de 0.4 a 0.0. Cabe notar que foi necessário adotar essa tolerância nas restrições maior que nos casos anteriores devido à dificuldade de se encontrarem soluções viáveis.

Na Tabela 20, estão apresentados os valores das variáveis de projeto para as melhores soluções encontradas com as rodadas realizadas.

Para esse exemplo, o otimizador não desempenhou bem, com a função objetiva alcançando valores próximos da soluções exata em apenas alguns casos, mas com grande dispersão em geral. Esse resultado aponta para a necessidade de uma investigação adicional do algoritmo.

Tabela 19 - Problema g13 – Estatísticas para 30 rodadas com diferentes sementes, probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 1.0 a 0.00005.

Função objetiva e restrições		Com $\sigma_2$ nas últimas 3125 iterações	
		Não-adaptativo	Adaptativo
f	Média	0.247304	0.235242
	Desvio Padrão	0.274941	0.274920
	Menor	0.053684	0.053973
	Maior	1.001947	1.000055
$h_1$	Média	$4.92 \times 10^{-3}$	$4.45 \times 10^{-5}$
	Desvio Padrão	$3.71 \times 10^{-3}$	$1.29 \times 10^{-5}$
	Menor	$1.97 \times 10^{-5}$	$8.97 \times 10^{-7}$
	Maior	$1.25 \times 10^{-2}$	$5.00 \times 10^{-5}$
$h_1$	Média	$2.47 \times 10^{-3}$	$3.46 \times 10^{-5}$
	Desvio Padrão	$3.22 \times 10^{-3}$	$1.50 \times 10^{-5}$
	Menor	$6.05 \times 10^{-6}$	$1.84 \times 10^{-6}$
	Maior	$1.16 \times 10^{-2}$	$4.97 \times 10^{-5}$
$h_1$	Média	$3.13 \times 10^{-3}$	$3.66 \times 10^{-5}$
	Desvio Padrão	$3.35 \times 10^{-3}$	$3.50 \times 10^{-5}$
	Menor	$2.06 \times 10^{-5}$	$4.37 \times 10^{-7}$
	Maior	$1.38 \times 10^{-2}$	$2.04 \times 10^{-4}$

Tabela 20 - Problema g13 – Valores das variáveis de projeto para as melhores soluções encontradas com probabilidade de referência PR de 0.4 a 0.0, tolerância  $\epsilon$  de 1.0 a 0.00005.

Variáveis de projeto	Solução exata	Com $\sigma_2$ nas últimas 3125 iterações	
		Não-adaptativo	Adaptativo
$x_1$	-1.717143	-1.728404	-1.724511
$x_2$	1.5957091	1.609439	1.604232
$x_3$	1.827247	1.806357	-1.813501
$x_4$	-0.7636413	0.722929	-0.759984
$x_5$	-0.763645	0.804323	0.765628

## 6. Interpolação das respostas

Atualmente, estão disponíveis programas de otimização do tipo controladores (*shell*) que permitem gerenciar o programa utilizado para obter a resposta do problema (estrutura) que está sendo analisado. Os programas de otimização tipo controlador indicam os valores das variáveis de projeto, para as quais deverão ser obtidas as respostas, em cada passo de simulação. Esta estratégia pode se mostrar interessante no caso de problemas com variáveis de projeto naturalmente discretas e com um número relativamente pequeno de intervalos por eixo.

Porém, no caso de problemas com variáveis contínuas o número de análises requeridas pode crescer exponencialmente, alcançando valores significativos. Isto ocorre pelo fato de não se conhecer a priori os valores adequados para o tamanho dos incrementos associados a cada variável contínua. A escolha de incrementos muito grandes pode facilmente prejudicar a detecção de pontos importantes de máximos (mínimos) globais e/ou locais que poderiam ficar escondidos entre os pontos da malha de soluções.

Vale observar que, na análise de problemas requerendo grande tempo de processamento para obter uma resposta, esta estratégia pode demandar dias de análise. Nestes casos, o tempo disponível para a escolha da solução final passa a ser relevante, especialmente com prazos de projeto apertados.

Um outro inconveniente neste modo de trabalhar pode ser o fato de o programa de análise mais adequado (ou a sua versão mais recente) para resolver um determinado problema não estar ainda disponível sob o controlador (*Shell*).

Essas questões podem justificar a conveniência de se dispor de estratégias alternativas de análise, através da utilização de métodos de interpolação multidimensionais adaptativos.

No caso de já haver muitas respostas disponíveis para o problema que está sendo analisado, como ocorre geralmente nos estágios mais avançados da espiral de projeto, pode-se cogitar da utilização de um interpolador do tipo rede neural.

Neste trabalho, buscou-se estudar a viabilidade da utilização de algumas técnicas simples de interpolação adaptativa multidimensional para se trabalhar com um número pequeno de pontos iniciais, compatíveis com o número de análises que seriam realizadas nas etapas iniciais da espiral de projeto. A base de pontos iniciais seria, então, enriquecida gradativamente, através das indicações dos resultados das análises de otimização.

Inicialmente, tentou-se uma interpolação baseada em superposição de *splines* cúbicas por variável de projeto, para o que é necessário definir um sistema de eixos arbitrário. O enriquecimento progressivo se daria nos hiperquadrantes para onde o processo de otimização indicasse os melhores pontos, através da adição de pontos e respectivas respostas com um certo raio de influência, formando hipercones. Entretanto, encontrou-se dificuldade na aplicação referente a quinas com formas suavizadas, em que havia regiões do espaço com restrições de valores das variáveis para montagem do modelo. Além disso, com a indicação de mínimo para um quadrante nos passos iniciais e confirmação com as novas rodadas, o procedimento só enriqueceria nesse quadrante, podendo-se mascarar mínimos locais ainda menores.

Assim, partiu-se para o estudo de outras técnicas de interpolação linear e quadrática para os pontos de resposta em qualquer posição no espaço ND-dimensional. Tanto a linear quanto a quadrática se baseiam na localização dos pontos mais próximos ao ponto simulado para o qual se deseja a resposta, respeitando-se o número mínimo de pontos requeridos em cada caso.

### **6.1. Interpolação Linear**

A interpolação linear é baseada na contribuição da resposta de ND+1 pontos da malha, preferencialmente os mais próximos do ponto simulado, pela proporção, em relação ao hipervolume do hipertetraedro formado por esses pontos, dos hipervolumes

dos hipertetraedros formados substituindo-se cada vértice pelo ponto simulado. Alguns testes são necessários para que a interpolação seja viável (por exemplo, evitar pontos alinhados). Também, são feitos testes para se evitem extrapolações para fora do hipertetraedro, que poderiam acontecer devido ao critério de proximidade dos pontos. Assim, a extrapolação só seria feita em último caso, ainda com a alternativa de se eliminar a possibilidade de simulação que obrigasse a extrapolação, através da criação de restrições adicionais no problema.

Nesse caso, a interpolação utiliza uma malha de pontos dispostos, em princípio, com qualquer arranjo no espaço, já que o programa detectaria e avisaria da eventual impossibilidade de interpolar.

A malha de pontos seria então enriquecida por pontos adicionais, a partir de indicações do programa de otimização, por exemplo a partir da observação do fronte encontrado. Esses pontos adicionais poderiam fazer parte da malha básica ou serem adicionados como hipercones de resposta, contribuindo com a resposta em uma região de influência.

Este tipo de interpolação não seria indicado para problemas uniobjetivo, uma vez que irá indicar como ponto ótimo um dos pontos da malha para os quais já se possui a resposta. Porém, a interpolação linear poderia em princípio ser utilizada em problemas de otimização multiobjetivo, uma vez que geralmente a região de interesse das soluções do fronte de Pareto encontra-se afastada dos pontos de máximo (mínimo) globais das funções objetivas individuais.

## **6.2. Regressão Linear**

Neste trabalho, foi implementada a regressão linear obtida pelo método dos mínimos quadrados, aplicável tanto à interpolação linear quanto à quadrática.

A função quadrática tem a vantagem de ser o polinômio de mais baixo grau que possibilita a criação de uma curvatura entre os pontos da malha. Na região de mínimos (máximos) locais, a função quadrática fornece indicações aproximadas de onde estaria

localizado o mínimo (máximo) da função real. Esta região seria então refinada pela obtenção do valores corretos da resposta para o ponto indicado, a partir dos resultados das análises fornecidas pelo programa de resposta. Obtém-se, deste modo, uma malha adaptativa, convergente para a posição do ponto de mínimo (máximo) da função real.

Para uma interpolação quadrática com todos os termos, são necessários no mínimo  $(ND+1)(ND+2)/2$  pontos, como será visto, ou seja, da ordem de  $ND^2$ , enquanto que para uma interpolação sem os termos cruzados, são necessários  $2ND+1$  pontos, ou seja, da ordem de  $ND$ , solicitando, portanto, menos pontos.

Os termos cruzados são úteis para se evitarem os efeitos negativos que podem advir quando cada par de variáveis de projeto forma uma sela que fuja muito da planicidade. Observou-se que pontos com respostas no mesmo nível podem levar a valores interpolados irreais, por exemplo, muito abaixo dessas respostas, levando-se a indicações de novas rodadas provavelmente desnecessárias. Para se prosseguir trabalhando sem os termos cruzados, um caminho proposto foi elevar o número de pontos até se observar uma estabilidade nos resultados ( $2ND+2$ ,  $2ND+3$ , ...). Nos passos iniciais, seria necessário incrementar o número de rodadas.

Com os respectivos números mínimos de pontos, a hipersuperfície de resposta passa por esses pontos. Com mais pontos na base, o método dos mínimos quadrados minimiza o erro quadrático total da hipersuperfície em relação aos pontos, não passando mais pelos pontos. Isso faz com que a interpolação, nos passos intermediários, já com um número significativo de pontos, caminhe no sentido de fornecer uma tendência do aspecto global, deixando de ser local. Então, o incremento do número de pontos deve ser o mínimo necessário para garantir uma estabilidade nos resultados.

Com relação ao fato de a interpolação não passar pelos pontos, cabe comentar que o método dos mínimos quadrados se baseia em escolher os parâmetros (coeficientes dos termos do polinômio) de forma a maximizar a probabilidade de que o conjunto de pontos disponíveis ocorra (*maximum likelihood*). Assumindo-se que essa probabilidade em cada ponto siga uma distribuição normal com, no caso geral, um desvio padrão  $\sigma_i$  por ponto, e independência entre as ocorrências dos pontos, ela é dada pela expressão (6-1) (Press et al., 1986).

$$P = \prod_{i=1}^{NPI} \left\{ \exp \left[ -\frac{1}{2} \left( \frac{y_i - y(\mathbf{x}_i)}{\sigma_i} \right)^2 \right] \Delta y \right\}, \quad (6-1)$$

onde NPI é o número de pontos de interpolação e  $\Delta y$  o intervalo para calcular probabilidade em torno de cada ponto. Maximizar esse produtório é equivalente a minimizar o negativo de seu logaritmo, como em (6-2).

$$\left[ \sum_{i=1}^{NPI} \left[ \frac{y_i - \sum_{k=1}^M a_k X_k(\mathbf{x}_i)}{\sqrt{2}\sigma_i} \right]^2 \right] - N \cdot \log(\Delta y) \quad (6-2)$$

Os termos  $X_k$  são funções arbitrárias fixas, chamadas de funções básicas. Como mencionado, neste trabalho foram implementadas as funções lineares e quadráticas. Com NPI e  $\Delta y$  constantes, chega-se à função de mérito conhecida (6-3), que representa a soma dos erros quadráticos.

$$\chi^2 = \sum_{i=1}^{NPI} \left[ \frac{y_i - \sum_{k=1}^M a_k X_k(\mathbf{x}_i)}{\sigma_i} \right]^2 \quad (6-3)$$

Com  $\sigma_i$  não conhecido, pode ser tomado como sendo 1.

Então, quando a hipersuperfície não passar pelos pontos, pode-se pensar de duas formas. Uma, é que os resultados do programa de análise, apesar de ser a referência para comparação de valores das funções objetivas, podem apresentar alguma variabilidade, o que se percebe quando se modifica ligeiramente o refinamento da malha de elementos finitos, ou o tipo de geração de malha (por exemplo, mapeada ou livre), já que é muito difícil manter o aspecto da malha na parametrização para todas as regiões do domínio

das variáveis de projeto. Outra maneira de ver é que o aumento no número de pontos é apenas um artifício para conseguir estabilidade nos resultados e que o interesse maior é nas variáveis de projeto, sendo que, de qualquer forma, o valor da função objetiva, ao final, será obtido pelo programa de análise.

Também observou-se que a extrapolação local pode levar a resultados enganosos. Para isso, são adicionados ao conjunto de pontos próximos, pontos já existentes na malha para garantir interpolação em todas as dimensões.

Para o caso de interpolação quadrática completa, a expressão para resposta de cada função objetiva ou restritiva  $y_f$ ,  $f = 1, NFO$ , quando as variáveis de projeto assumem os valores  $x_{j,i}$ ,  $j = 1, ND$  no ponto  $i$ , será dada pela expressão (6-4).

$$y_f(\mathbf{x}_i) = a_0 + \sum_{j=1}^{ND} a_j x_{j,i} + \sum_{j=1+ND}^{ND} a_j x_{j,i}^2 + \sum_{j=1; k=j+1}^{ND; ND} a_{jk} x_{j,i} x_{k,i}, \quad (6-4)$$

Os coeficientes  $a_j$ , no total  $M = 1 + ND + ND + C_{ND}^2 = (ND + 2)(ND + 1)/2$  (reescrevendo  $a_j$ ,  $j = 1, \dots, M$ ), serão obtidos tais que levem o valor de  $\chi^2$  a um mínimo, considerando  $i = 1, \dots, NPI$ . Para trabalhar-se com a interpolação quadrática incompleta, deve ser suprimido o último somatório da expressão (6-4), levando a  $M = 2ND + 1$ . Derivando-se  $\chi^2$  em relação a cada coeficiente  $a_i$  e igualando a zero, teremos  $M$  expressões do tipo da (6-5) abaixo, que são as chamadas equações normais do problema de mínimos quadrados.

$$0 = \sum_{i=1}^{NPI} \frac{1}{\sigma_i^2} \left[ y_i - \sum_{j=1}^M a_j X_j(x_i) \right] \cdot X_k(x_i), \quad k = 1, \dots, M, \quad (6-5)$$

em que os termos com as funções  $X_k$ ,  $k = 1, \dots, M$  serão adaptados com os termos da função quadrática de (6-4).

As expressões de (6-5) formam um sistema de  $M$  equações e  $M$  incógnitas. Então, serão necessários no mínimo  $M$  pontos para a interpolação,  $O(ND^2)$  para a interpolação

completa e  $O(ND)$  para a incompleta. Para resolver o sistema, em Press et al. (1986) são mencionados os métodos padrões, como decomposição LU e eliminação de Gauss. Entretanto, o que é recomendado para os problemas de mínimos quadrados como sendo o mais robusto é o *Singular Value Decomposition*. Ele se baseia em que uma matriz  $\mathbf{A}$   $N \times M$  para a qual o número de linhas é maior que ou igual ao número de colunas pode ser reescrita sempre como em (6-6):

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T, \quad (6-6)$$

em que  $\mathbf{U}$  é uma matriz  $N \times M$  coluna-ortogonal,  $\mathbf{W}$  é uma matriz diagonal de ordem  $M$  e  $\mathbf{V}$  é uma matriz ortogonal de ordem  $M$ . Uma matriz é ortogonal quando suas colunas são ortonormais, satisfazendo (6-7).

$$\mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I}, \text{ sendo que no caso de } \mathbf{V}, \text{ quadrada, também vale } \mathbf{V} \mathbf{V}^T = \mathbf{I}, \quad (6-7)$$

Definindo-se a matriz  $\mathbf{A}$   $N \times M$  e o vetor  $\mathbf{b}$  como em (6-8), tem-se:

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i}, \quad b_i = \frac{y_i}{\sigma_i} \quad (6-8)$$

A função a ser minimizada passa a ser a expressão (6-9).

$$\chi^2 = |\mathbf{A} \cdot \mathbf{a} - \mathbf{b}|^2 \quad (6-9)$$

Assim, os coeficientes para a interpolação são dados por (6-10).

$$\mathbf{a} = \mathbf{V} \cdot \text{diag}(1/w_j) \cdot \mathbf{U}^T \cdot \mathbf{b}, \quad (6-10)$$

onde  $\text{diag}(1/w_j)$ ,  $j = 1, \dots, M$ , é a matriz diagonal com os inversos dos termos das diagonais de  $\mathbf{W}$ .

Como os pontos para interpolação serão, em geral, diferentes para cada ponto simulado, a interpolação será descontínua no domínio de simulação das variáveis de projeto.

A idéia é que, após cada incremento de pontos na base, inicie-se a interpolação da resposta para cada ponto simulado com o número mínimo de pontos próximos necessários em função do número de variáveis de projeto. Observa-se, então, se o comportamento das respostas, no caso a informação dada pelo fronte obtido, está razoável ou se dá sinais claros de que a interpolação não está estável. Essa instabilidade pode ocorrer com a interpolação quadrática incompleta, especialmente quando a função interpolada tem aspecto de sela multidimensional. Nesse caso, aumentar-se-ia o número de pontos para interpolação próximos de cada ponto simulado até que o fronte apresente um comportamento estável. Essa avaliação é quase que subjetiva, lembrando que a função interpolada não irá passar pelos pontos. Por esse mesmo motivo, o número de pontos também não deve ser próximo do total (nos passos seguintes), porque irá forçar uma superfície quadrática global, que pode ser muito diferente da real. Se a indicação for essa para que haja estabilização da interpolação para algumas soluções, é um indício de que a base de pontos deve ser melhorada na região dessas soluções.

### **6.3. Considerações sobre a Interpolação**

Quando se trabalha com malha adaptativa de interpolação, enriquecida a partir de poucos pontos, percebeu-se que a escolha da disposição e do número de pontos para representação adequada das funções não é uma tarefa fácil e pode ser dependente do problema. Nos tipos de interpolação analisados, os pontos que serão efetivamente utilizados na interpolação serão preferencialmente os mais próximos do ponto desejado e são necessárias algumas condições para que a interpolação não leve a respostas e conclusões errôneas, inclusive sobre onde enriquecer a malha.

No caso de problemas com uma função objetiva apenas, o enriquecimento pode ser em torno da indicação do otimizador no passo anterior do processo para algum mínimo local. Nos problemas multiobjetivo, isso se complicaria caso toda a região do fronte Pareto-ótimo tivesse que ser bem representada, além de que o fronte obtido com

respostas interpoladas deve ser conduzido sucessivamente mais próximo do fronte exato com o enriquecimento da malha de pontos. Pode haver regiões do fronte de Pareto em região limite de alguma restrição, em região definida puramente por uma solução de compromisso (*trade-off*) entre as funções, sem restrições, ou em mínimos locais e globais. Nos dois primeiros casos, talvez o requisito para refinamento progressivo da interpolação não seja tão rigoroso quanto para um mínimo local, especialmente para funções relativamente suaves.

Assim, já que num problema multiobjetivo há suposta propensão para interesse na região de *trade-off*, pode ser adequado utilizar a interpolação linear com os  $ND + 1$  pontos próximos do ponto simulado, que, diferentemente da interpolação quadrática, só interpola valores das funções entre os existentes na base, ou seja, não cria curvaturas na resposta entre os pontos, tornando o processo de enriquecimento mais controlável.

Com isso, o fronte Pareto-ótimo a cada passo seria formado por trechos mais facilmente identificáveis através dos correspondentes pontos no espaço das variáveis de projeto e teríamos uma boa representação global do fronte de Pareto.

Com a identificação de alguma região do fronte que desperte maior interesse e que possa conter regiões de mínimos locais das funções objetivas, pode-se iniciar um processo de otimização local com um interpolador quadrático.

Neste trabalho, serão apresentados os resultados dos problemas de otimização estrutural obtidos apenas com interpolação linear das respostas.

## 7. Aplicações em Estruturas

Na otimização de projetos estruturais, é possível fazer a diferenciação entre alguns tipos de abordagem.

Para estruturas do tipo treliça, como mencionado em Deb (2001), que pode ser estendido para pórtico espacial, identificam-se 3 categorias principais: dimensionamento, configuração e topologia.

A otimização puramente do dimensionamento atua sobre as propriedades das seções transversais dos elementos de barra, que são as variáveis de projeto, mantendo-se a posição dos nós (configuração) e a topologia fixos. Essa categoria pode ser estendida a estruturas de cascas, em que as espessuras são variáveis de projeto.

A otimização da configuração, isoladamente, trabalha com as coordenadas dos nós da estrutura como variáveis de projeto. Para manter a conectividade (topologia), será provavelmente necessário introduzir no problema restrições de posições relativas entre os nós. Um exemplo é o posicionamento de luvas de estacas em torno da perna de jaquetas. Cabe ressaltar que cada nó de uma treliça espacial representa 3 variáveis de projeto.

A otimização de topologia envolve a escolha da conectividade entre os nós, o que permitirá identificar os membros realmente necessários.

Os problemas de treliça têm sido abordados com essas 3 categorias tratadas separadamente, ou ainda em níveis, primeiro otimizando a topologia, com as propriedades e coordenadas fixados, e, então, para cada topologia, otimizar as dimensões e configurações. Deb (2001) propôs uma abordagem em que as três categorias são abordadas ao mesmo tempo para uma treliça, em que se destaca que o critério de avaliação da importância de um membro é se a área da seção transversal após a otimização está acima de um valor mínimo. Com isso, a topologia é mantida artificialmente durante o processo de otimização.

Para a otimização de formas de estruturas de cascas, modeladas, por exemplo, pelo método dos elementos finitos, a maneira comum de abordagem é escolher alguns parâmetros utilizados na definição da forma como variáveis de projeto e construir os diversos modelos para obtenção das respostas. Esses parâmetros seriam, dentre outros, raios e posições do centro de arcos, com o cuidado de se procurar que sempre seja possível a montagem. Uma alternativa seria escolher as coordenadas de pontos do contorno da forma, similar à otimização de configuração de treliças.

Em Deb (2001), é proposto um procedimento para esse tipo de problema que trabalha com uma malha unificada, apenas atribuindo-se a condição de presente ou ausente para cada elemento. Esse procedimento seria uma aplicação apropriada para Algoritmos Genéticos, com a formulação binária dos elementos, que irá definir a forma (fenótipo) da estrutura, mas o número de variáveis de projeto cresce consideravelmente.

Neste trabalho, serão apresentadas duas aplicações simples. A primeira refere-se a otimização do dimensionamento da seção transversal de uma viga. A segunda será sobre otimização da forma de um corte na quina de uma figura plana em forma de L, de forma a reduzir a concentração de tensões, utilizando a parametrização da forma para montagem dos modelos. Apenas será mostrada, para esses casos, a conveniência de se utilizar a interpolação linear para aproximação sucessiva do fronte de Pareto na região de *trade-off*, ou seja, longe das regiões de mínimos globais das funções objetivas. Para esses exemplos, os resultados não são sensíveis aos parâmetros das análises de otimização.

### **7.1. Otimização de seção de vigas**

A primeira aplicação se refere a buscar obter uma seção transversal que leve ao menor peso, ao mesmo tempo conseguindo baixas deflexões. Esse problema foi abordado em Castro (2001), como multiobjetivo, e em Mendonça (2004), como uniobjetivo, nesse caso procurando-se minimizar apenas a área da seção, tratando-se o deslocamento admissível como uma restrição.

A viga em estudo está mostrada na Figura 30. As propriedades da viga serão:

Seção tipo perfil I, com B, largura das mesas; D, altura total da seção; Tw, espessura da alma; e Tf, espessura das mesas; A, área da seção transversal; I<sub>y</sub> e I<sub>z</sub>, momentos de inércia em relação aos eixos y e z da seção, respectivamente; W<sub>y</sub> e W<sub>z</sub>, módulos resistentes da seção em relação aos eixos y e z, respectivamente. O módulo de elasticidade será E = 2 x 10<sup>4</sup> kN/cm<sup>3</sup>. A tensão admissível será σ<sub>max</sub> = 16 kN/cm<sup>2</sup>; e o deslocamento máximo admissível será δ<sub>max</sub> = 0.5 cm.

As cargas aplicadas serão: P = 600 kN e Q = 50 kN.

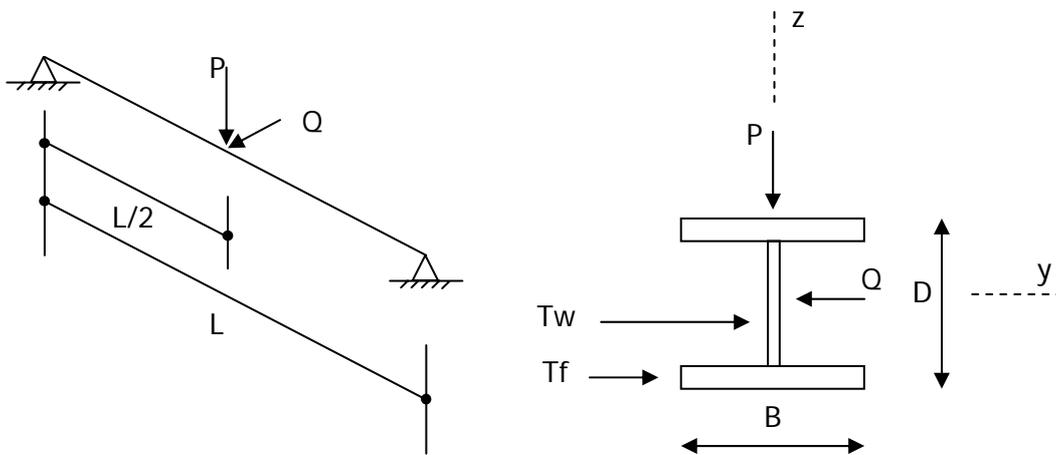
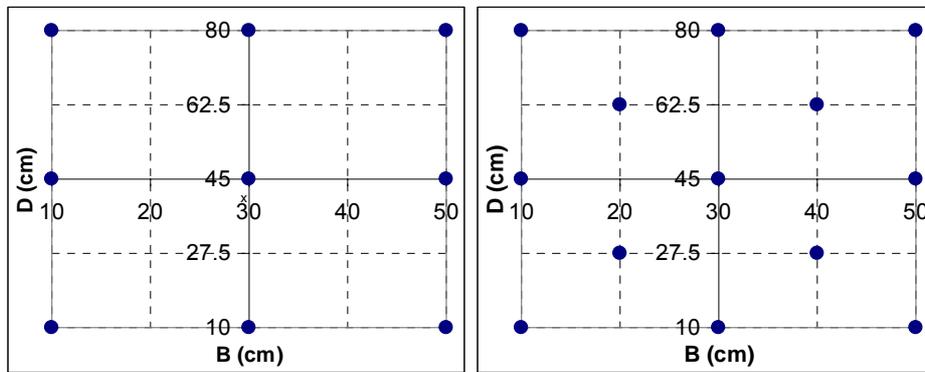


Figura 30 - Viga para estudo de otimização de seção transversal.

Trabalhar-se-á com apenas 2 variáveis de projeto, B e D, mantendo-se as espessuras fixas em Tw = 1.5 cm e Tf = 2.5 cm. A formulação do problema de otimização multiobjetivo será, então, dada pela expressão (7-1).

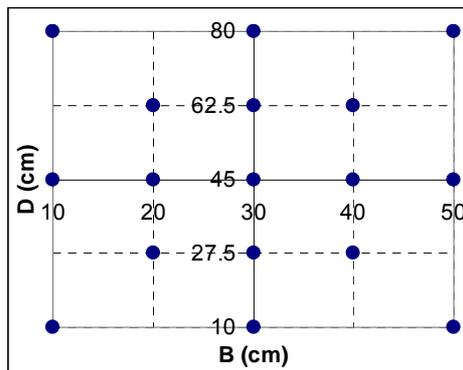
$$\left\{ \begin{array}{l}
 \text{Minimizar } A(\mathbf{x}) = 2BTf + Tw(D - Tf) \text{ e} \\
 \delta(\mathbf{x}) = \sqrt{\left(\frac{PL^3}{48EI_y(\mathbf{x})}\right)^2 + \left(\frac{QL^3}{48EI_z(\mathbf{x})}\right)^2}, \\
 \text{sujeito a: } g_1(\mathbf{x}) = \frac{M_y}{W_y} + \frac{M_z}{W_z} - \sigma_{\max} \leq 0, \\
 g_2(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{\max} \leq 0, \\
 \text{onde: } \mathbf{x} = (B, D, Tw, Tf) \\
 \text{com as limitações de intervalo: } 10 \leq B \leq 50, 10 \leq D \leq 80
 \end{array} \right. \quad (7-1)$$

As respostas serão obtidas através de interpolação linear, conforme o item 6.1, ou seja, com a utilização de 3 pontos da malha de interpolação para cada ponto simulado durante o processo de otimização. Para isso, foram montadas malhas de 9, 13 e 17 pontos, de acordo com a Figura 31. Os valores fornecidos ao programa otimizador estão mostrados na Tabela 22. Com o otimizador PAES-M, foram feitas 3 rodadas de otimização com essas 3 malhas de interpolação, além da rodada para a solução exata de referência. Os fronts de Pareto com 40 pontos encontrados para cada caso estão mostrados na Figura 32.



(a) 9 pontos

(b) 13 pontos



(c) 17 pontos

Figura 31 - Malhas de pontos utilizadas para interpolação linear das respostas.

Tabela 21 - Problema da seção de viga I - valores das funções objetivas e restritivas para a malha de 17 pontos de interpolação das respostas ( $T_f = 1.5$  cm e  $T_w = 2.5$  cm)

Ponto	B (cm)	D (cm)	Área (cm <sup>2</sup> )	Deflexão (cm)	Restrição Tensão (kN/cm <sup>2</sup> )	Restrição Deflexão (cm)
1	30	45	210.00	0.07562	-3.7665	-0.4244
2	10	45	110.00	0.98732	35.2671	0.4873
3	50	45	310.00	0.04208	-9.2227	-0.4579
4	30	10	157.50	2.26981	55.4180	1.7698
5	30	80	262.50	0.04111	-8.3571	-0.4589
6	10	10	57.50	6.78686	215.2977	6.2869
7	10	80	162.50	0.95262	21.9413	0.4526
8	50	10	257.50	1.36560	26.1673	0.8656
9	50	80	362.50	0.01415	-11.9984	-0.4858
10	20	62.5	186.25	0.13193	-0.2992	-0.3681
11	40	62.5	286.25	0.02907	-9.5276	-0.4709
12	20	27.5	133.75	0.31789	15.6073	-0.1821
13	40	27.5	233.75	0.15334	-1.5408	-0.3467
14	20	45	160.00	0.15605	4.1608	-0.3440
15	30	62.5	236.25	0.04857	-6.7694	-0.4514
16	40	45	260.00	0.05315	-7.2672	-0.4468
17	30	27.5	183.75	0.20387	3.8715	-0.2961

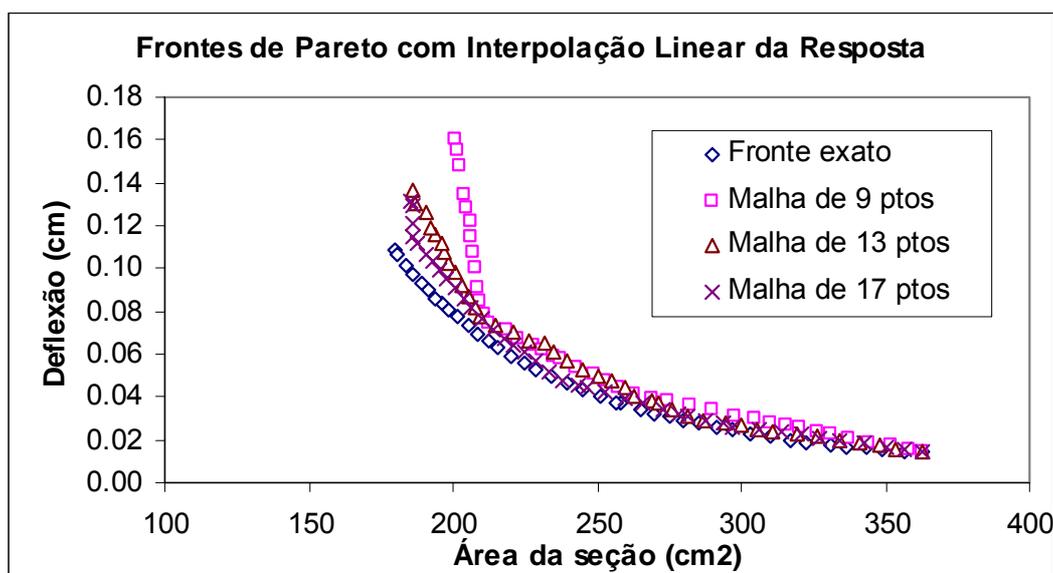


Figura 32 - Frontes de Pareto com 40 pontos, obtido para diferentes refinamentos de malhas, com interpolação linear das respostas.

Na Figura 32, visualmente percebe-se uma tendência para que o fronte inicial, obtido com respostas que utilizaram 4 pontos de interpolação, se aproxime do fronte exato à medida que são acrescentados pontos na malha.

Na Tabela 22, são apresentados os valores dos parâmetros de seção transversal para as soluções encontradas no fronte Pareto-ótimo com 20 pontos, bem como os correspondentes valores das funções objetivas e restritivas, para o caso de interpolação linear com 17 pontos na malha. Pode-se notar que o fronte contém respostas de ampla faixa das variáveis de projeto. Observa-se uma maior frequência dos valores da altura D na faixa da metade superior de seu intervalo de validade, o que também se observou para o mapeamento a partir do fronte exato, apresentado na Tabela 23.

Tabela 22 - Fronte Pareto-ótimo com 20 pontos, encontrado com interpolação linear das respostas com malha de 17 pontos ( $T_f = 1.5$  cm e  $T_w = 2.5$  cm)

Ponto	B (cm)	D (cm)	Área (cm <sup>2</sup> )	Deflexão (cm)	Restrição Tensão (kN/cm <sup>2</sup> )	Restrição Deflexão (cm)
1	20.133	61.177	184.930	0.133	-0.048	-0.367
2	23.416	50.720	185.657	0.121	-0.004	-0.379
3	25.829	45.205	189.451	0.109	-0.495	-0.391
4	27.073	45.042	195.430	0.099	-1.454	-0.401
5	28.025	44.949	200.049	0.092	-2.179	-0.408
6	29.376	44.978	206.845	0.081	-3.262	-0.419
7	29.966	47.741	213.940	0.072	-4.210	-0.428
8	30.014	54.338	224.075	0.061	-5.373	-0.439
9	30.030	58.887	230.979	0.054	-6.158	-0.446
10	30.468	63.664	240.335	0.047	-7.004	-0.453
11	33.240	62.452	252.380	0.042	-7.655	-0.458
12	35.874	62.711	265.938	0.037	-8.409	-0.463
13	38.126	62.536	276.933	0.033	-9.014	-0.467
14	40.305	63.143	288.739	0.029	-9.601	-0.471
15	39.445	71.071	296.330	0.027	-10.152	-0.473
16	44.960	61.942	310.212	0.024	-10.425	-0.476
17	44.142	69.991	318.198	0.023	-10.547	-0.477
18	45.641	73.794	331.394	0.021	-10.974	-0.479
19	47.763	76.399	345.913	0.017	-11.457	-0.483
20	50.000	80.000	362.500	0.014	-11.998	-0.486

Tabela 23 - Fronte Pareto-ótimo com 20 pontos, com as respostas exatas ( $T_f = 1.5$  cm e  $T_w = 2.5$  cm)

Ponto	B (cm)	D (cm)	Área (cm <sup>2</sup> )	Deflexão (cm)	Restrição Tensão (kN/cm <sup>2</sup> )	Restrição Deflexão (cm)
1	22.617	49.305	179.543	0.110	0.000	-0.390
2	22.766	48.827	179.573	0.110	-0.006	-0.390
3	24.302	47.880	185.830	0.098	-1.021	-0.402
4	25.148	49.137	191.947	0.089	-1.957	-0.411
5	25.882	50.308	197.372	0.082	-2.718	-0.418
6	26.447	52.985	204.214	0.074	-3.620	-0.426
7	27.632	53.696	211.204	0.067	-4.409	-0.433
8	28.904	55.442	220.183	0.059	-5.330	-0.441
9	29.285	58.314	226.397	0.055	-5.933	-0.445
10	30.581	58.662	233.399	0.050	-6.510	-0.450
11	31.096	62.568	241.833	0.045	-7.185	-0.455
12	33.147	63.428	253.374	0.039	-7.952	-0.461
13	33.676	67.311	261.845	0.036	-8.483	-0.464
14	35.223	68.649	271.587	0.032	-9.007	-0.468
15	36.183	74.551	285.243	0.028	-9.673	-0.472
16	39.540	72.711	299.264	0.024	-10.222	-0.476
17	40.412	79.085	313.188	0.021	-10.748	-0.479
18	42.995	79.945	327.390	0.018	-11.181	-0.482
19	45.619	80.000	340.593	0.016	-11.524	-0.484
20	50.000	80.000	362.500	0.014	-11.998	-0.486

## 7.2. Cortes para alívio de tensões

Para aliviar tensões em quinas, de forma a reduzir a concentração de tensões, uma opção é melhorar a forma de detalhes estruturais com a retirada de material. Esse tipo de solução tem sido adotada, por exemplo, em plataformas semi-submersíveis, na conexão de contraventamentos com as colunas (estabilizadas por colunas), bem como na ligação de grandes elementos como convés-coluna e flutuador-coluna (*pontoon-coluna*).

O exemplo analisado, mais acadêmico, envolveu a busca de soluções que reduzam as tensões na quina de uma figura plana em forma de L, sujeita a um momento fletor M no plano em uma das pernas do L, com engastamento na outra perna. Foi adotado um L simétrico, com as dimensões indicadas na Figura 33.

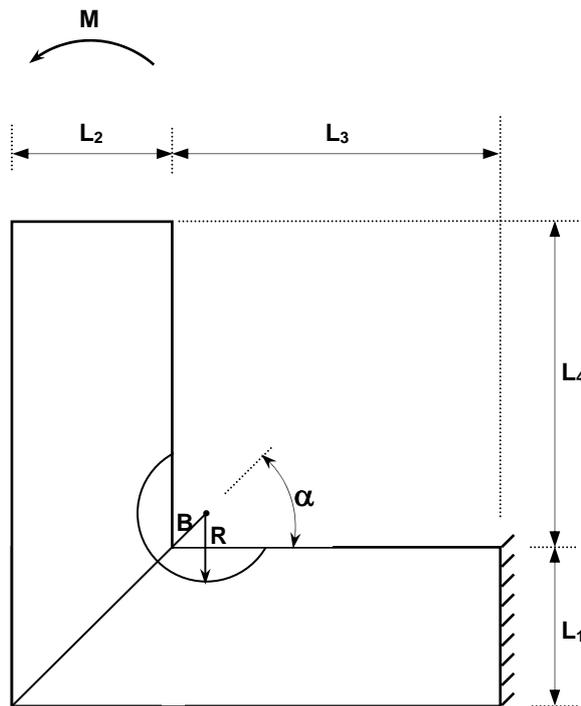


Figura 33 -  $L_1 = L_2 = 200$ ,  $L_3 = L_4 = 400$ ,  $T = 10$  (espessura)

As variáveis de projeto adotadas para representar a abertura foram  $B$ ,  $R$  e  $\alpha$ , sendo  $B$  a distância na direção alfa da quina ao centro do círculo de raio  $R$ . No caso simétrico, com  $\alpha = 45$  graus, trabalha-se, então, com apenas 2 variáveis,  $B$  e  $R$ .

Pela construção, nota-se que  $R$  deve ser maior que  $B$  para que não haja necessidade de adição de material na quina. Se essa for uma premissa a ser seguida, então, diferentemente dos exemplos acadêmicos, essa é uma restrição a ser observada não só durante a busca, mas também será uma região do espaço onde não haverá resposta disponível. Assim, ao mapear o espaço com soluções iniciais, por exemplo, fixando-se uma variável para estudar a variação da outra, não será possível varrer todas as regiões do eixo correspondente, dificultando o emprego de superposição de curvas tipo *spline* cúbica. Essa foi uma das razões que levaram à procura de um interpolador que conseguisse trabalhar com arranjo genérico de pontos no espaço ND-dimensional das variáveis de projeto.

Por algumas rodadas preliminares, observou-se que, para pequenas retiradas de material, a concentração de tensões assume um comportamento local. Notou-se também

que podem-se conseguir maiores reduções nas tensões com a retirada cada vez maior de material, até um certo ponto, quando o efeito de perda global de seção resistente começa a se manifestar, passando a aumentar as tensões. Para fugir da situação de grandes perdas de seção, pensando nas reservas plásticas normalmente desejadas, introduziu-se a função área retirada, que deverá ser mantida em níveis baixos. Poder-se-ia pensar também em associar essa área ao custo de mão-de-obra de retirada de material e de acabamento (que poderia, como alternativa, estar associado ao comprimento do arco do material retirado). Percebe-se, então, que está posto um problema de otimização com dois objetivos antagônicos, tensão e área retirada, caracterizando uma busca de solução de compromisso (*trade-off*). Essa busca será feita justamente ao longo das soluções no fronte de Pareto, para o que alguma informação adicional será necessária para a escolha final. Essas informações adicionais são ditas de mais alto nível ou subjetivas, em contraposição às funções objetivas. A faixa de tensão máxima com que se quer trabalhar em termos de fadiga ou a faixa de orçamento disponível para o trabalho pode auxiliar na tomada de decisão.

O problema de otimização será, então, minimizar a tensão  $T(B,R)$  e a área retirada  $A(B,R)$ , sujeito a  $R - B \leq 0$ . As tensões na borda do corte são obtidas do modelo de elementos finitos, enquanto que a área retirada pode ser obtida da expressão deduzida (7-2).

$$A = \pi \cdot R^2 - R^2 \left( \frac{\pi}{4} + \beta \right) - B \cdot R \cdot \cos \left( \frac{\pi}{4} - \beta \right), \quad (7-2)$$

onde:

$$\beta = \arcsen \left( \frac{B/\sqrt{2}}{R} \right) \quad (7-3)$$

Os resultados de tensões de von Mises para os casos iniciais analisados estão mostrados nos gráficos da Figura 35 à Figura 39. Os pontos estão dispostos como na Figura 34. Foi aplicado um momento que produzisse tensão de 100 na fibra extrema, imaginando o trecho superior do L como um viga de seção T x L<sub>2</sub>, valor que pode ser

considerado uma tensão nominal para cálculo de fator de concentração de tensões (SCF). O valor da tensão máxima, a área do material retirado e a função de restrição para essas rodadas estão mostrados na Tabela 24.

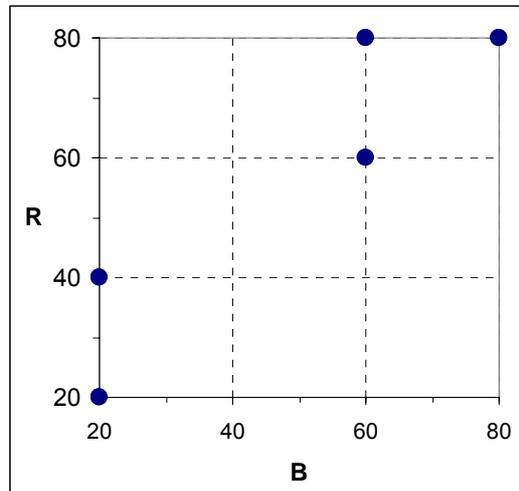


Figura 34 - Disposição dos pontos para rodadas preliminares no problema do L com corte para alívio de tensões.

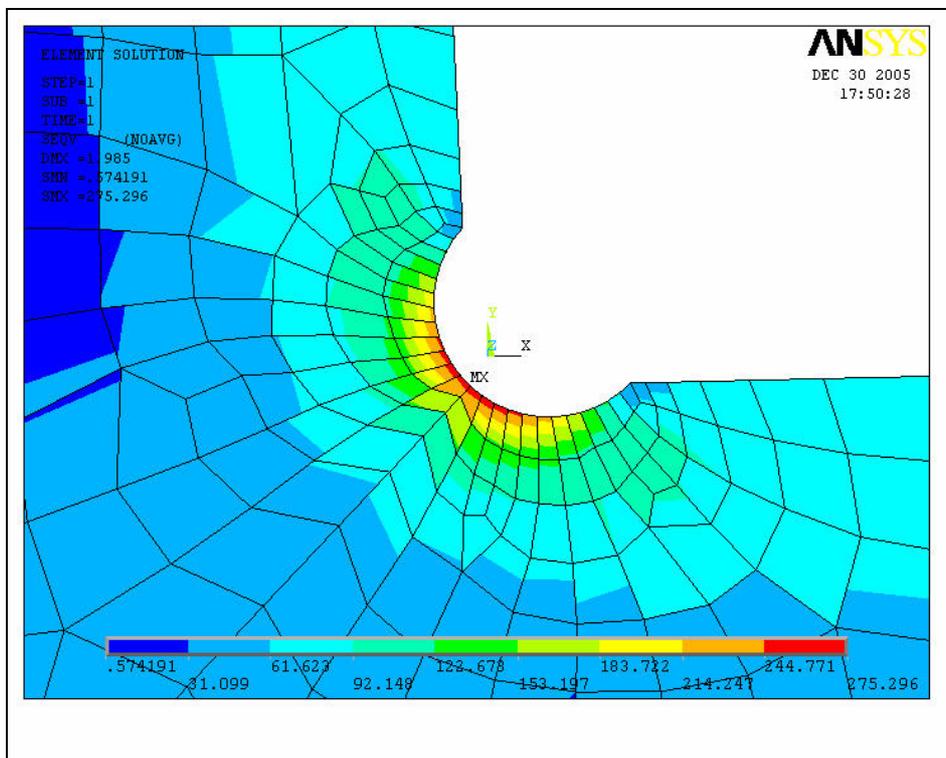


Figura 35 - Gráfico de tensões na região do material retirado da quina, B = 20, R = 20

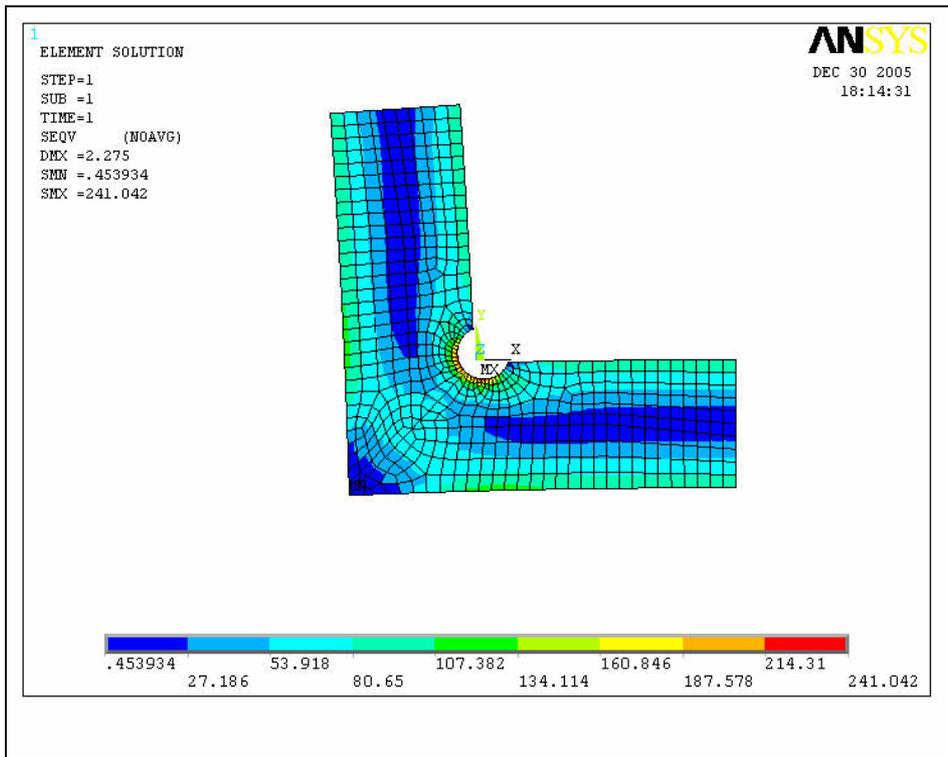


Figura 36 - Gráfico de tensões na região do material retirado da quina, B = 20, R = 40

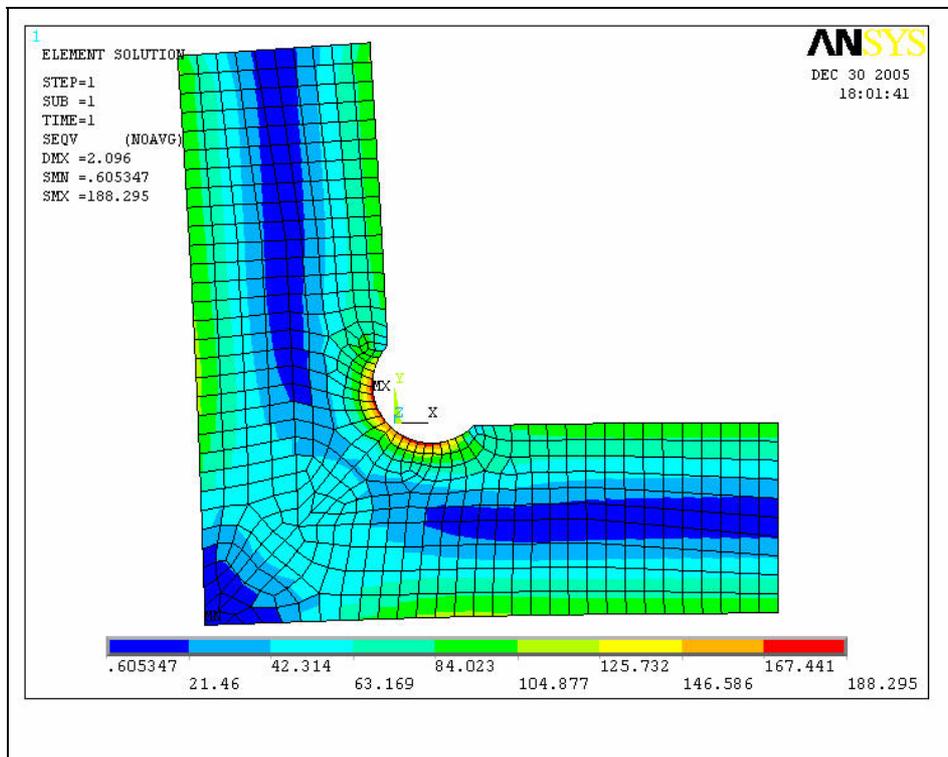


Figura 37 - Gráfico de tensões na região do material retirado da quina, B = 60, R = 60

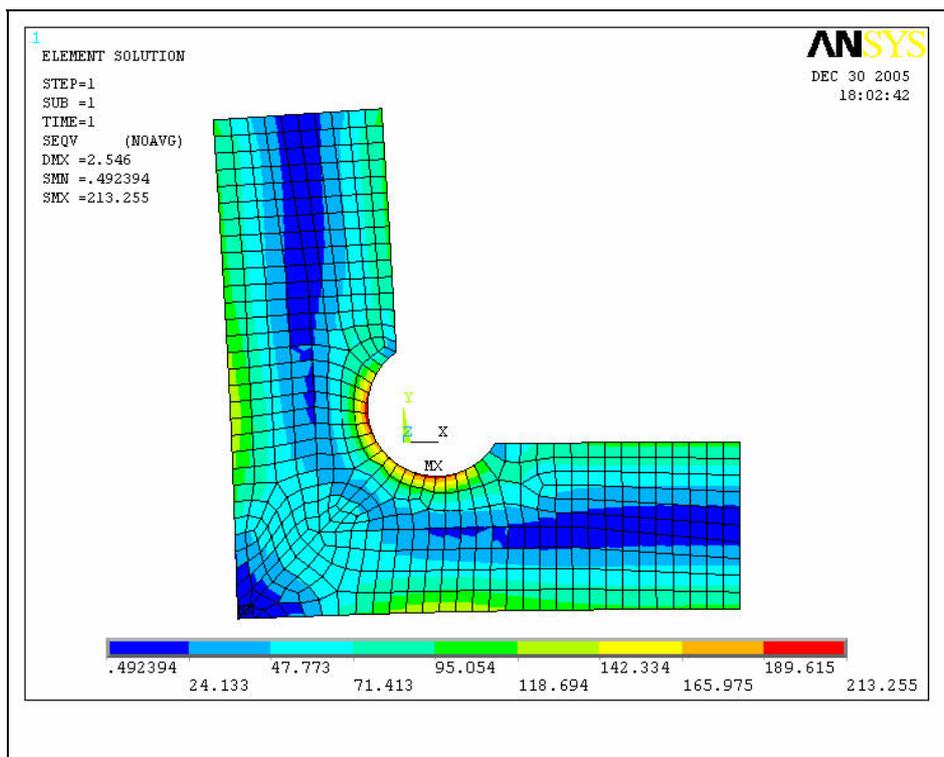


Figura 38 - Gráfico de tensões na região do material retirado da quina, B = 60, R = 80

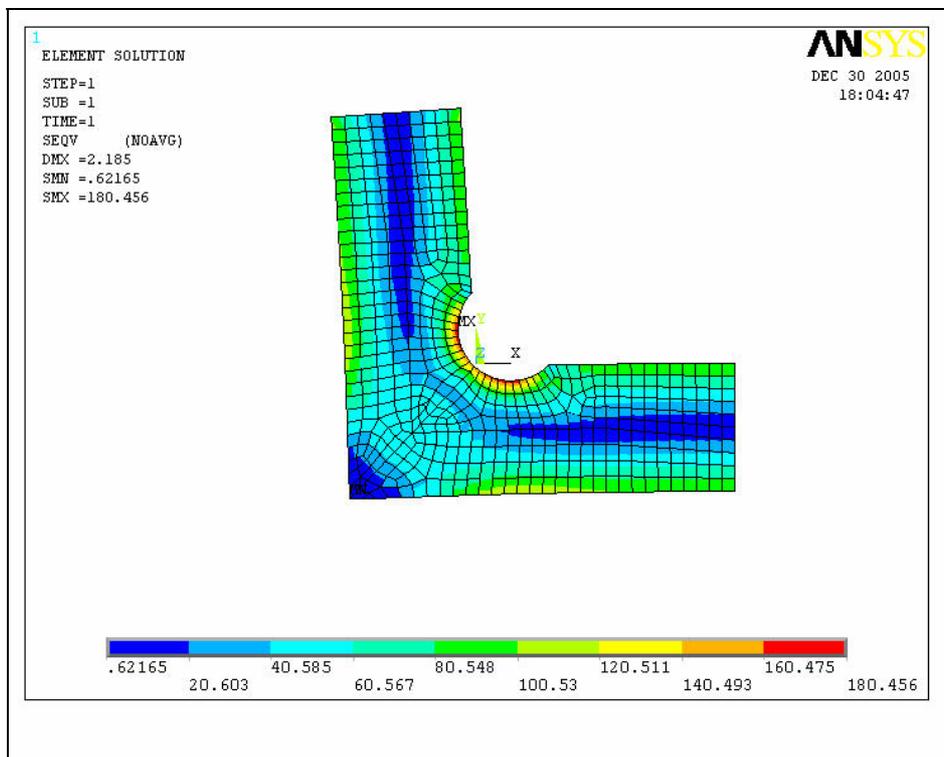
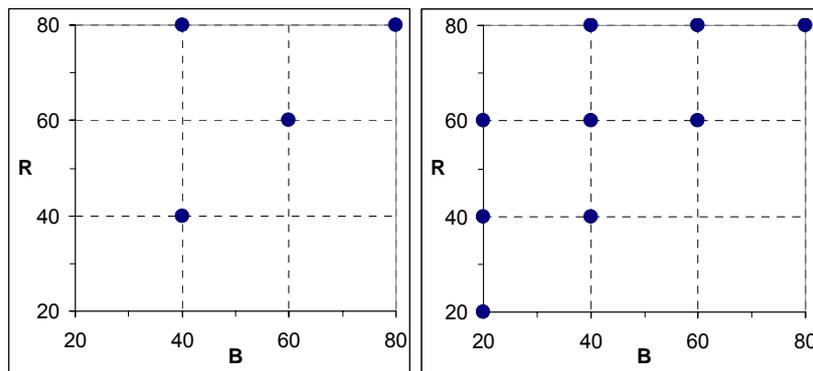


Figura 39 - Gráfico de tensões na região do material retirado da quina, B = 80, R = 80

Tabela 24 - Problema do L com corte de alívio – Análises iniciais de tensões.

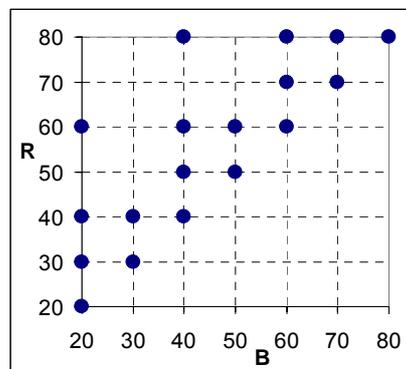
Ponto	Variáveis de projeto		Funções objetivas		Restrição R – B
	B	R	Tensão	Área	
1	20	20	275.3	228.3	-0
2	20	40	241.0	2463	-20
3	60	60	188.3	2055	-0
4	60	80	213.3	6825	-20
5	80	80	180.5	3653	-0

Foi feita, então, uma seqüência de análises de otimização com o programa PAES-M, adotando-se a interpolação linear das respostas de acordo com o item 6.1, com malhas de 4, 9 e 18 pontos, dispostos conforme a Figura 40. Na Tabela 25, estão apresentadas as respostas para a malha de 18 pontos, que contém também os casos das malhas de menos pontos. Na Figura 41, são apresentados os frontes de Pareto com 40 pontos encontrados para os 3 casos de malhas de interpolação. Na Tabela 26, é mostrado o fronte de Pareto com 20 pontos, para o caso da malha de 18 pontos.



(a) Malha de 4 pontos

(b) Malha de 9 pontos



(c) Malha de 18 pontos

Figura 40 - Disposição dos pontos da malha de interpolação para o problema do L com corte, com 4, 9 e 18 pontos.

Tabela 25 - Problema do L com corte de alívio – valores das funções objetivas e restritivas para a malha de 18 pontos de interpolação das respostas.

Ponto	Variáveis de projeto		Funções objetivas		Restrição R – B
	B	R	Tensão	Área	
1	20	20	275.3	228.32	0
2	20	30	249.3	1104.61	-10
3	20	40	241.0	2462.57	-20
4	20	60	255.0	6601.09	-40
5	30	30	233.2	513.72	0
6	30	40	221.4	1706.15	-10
7	40	40	213.9	913.27	0
8	40	50	205.4	2421.14	-10
9	40	60	214.5	4418.46	-20
10	40	80	250.1	9850.29	-40
11	50	50	193.5	1426.99	0
12	50	60	198.1	3249.89	-10
13	60	60	188.3	2054.87	0
14	60	70	197.8	4192.58	-10
15	60	80	213.3	6824.61	-20
16	70	70	183.6	2796.90	0
17	70	80	196.5	5249.27	-10
18	80	80	180.5	3653.10	0

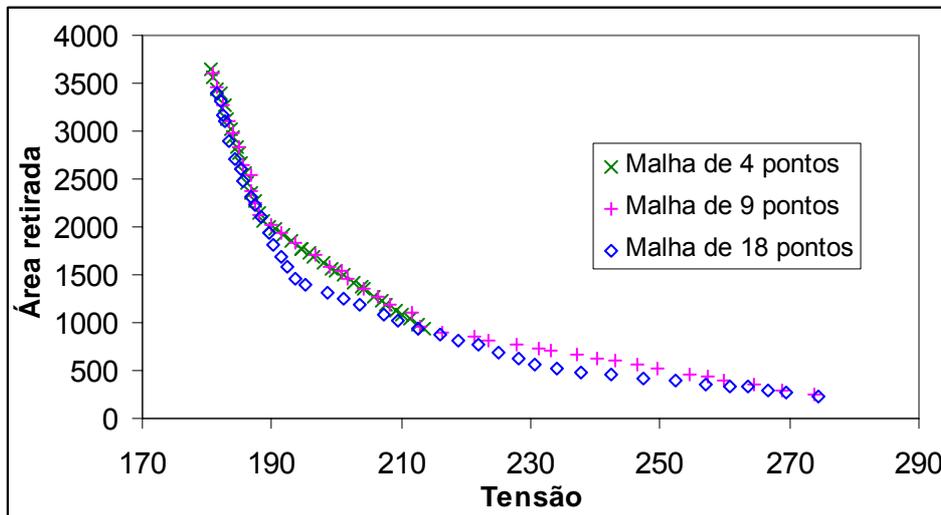


Figura 41 - Frontes de Pareto para o problema do L com corte, trabalhando-se com interpolação linear das respostas, com malhas de 4, 9 e 18 pontos.

Tabela 26 - Problema do L com corte: Fronte Pareto-ótimo com 20 pontos, obtido com interpolação linear das respostas com malha de 18 pontos.

Ponto	Variáveis de projeto		Funções objetivas		Restrição R - B
	B	R	Tensão	Área	
1	79.64	79.66	180.64	3627.90	-0.0236
2	78.11	78.11	181.09	3491.93	-0.0028
3	73.84	73.86	182.44	3132.04	-0.0267
4	71.22	71.24	183.25	2906.09	-0.0189
5	67.49	67.56	184.84	2625.60	-0.0684
6	63.04	63.10	186.93	2293.37	-0.0608
7	59.82	59.82	188.39	2043.53	-0.0004
8	55.01	55.04	190.91	1746.28	-0.0257
9	51.05	51.25	193.05	1529.20	-0.2006
10	46.75	46.83	200.06	1272.48	-0.0835
11	43.48	43.48	206.81	1092.73	-0.0061
12	39.15	39.16	215.53	880.26	-0.0071
13	34.84	34.85	223.85	707.79	-0.0056
14	34.03	34.03	225.42	675.49	-0.0071
15	29.67	29.84	234.15	518.76	-0.1641
16	27.93	27.99	241.76	459.47	-0.0537
17	26.12	26.16	249.44	405.99	-0.0340
18	24.14	24.38	257.23	367.53	-0.2392

Nota-se que todos os pontos do fronte representam os casos em que  $B = R$  e pode-se inferir que todos os pontos nessa condição serão pontos do fronte. Isso ocorre porque a variação na área retirada para uma mesma variação na tensão em torno da região  $B = R$ , onde caminham em sentidos antagônicos, é bem maior na direção do eixo R do que na direção do eixo B. Assim, a linha  $B = R$ , que delimita a restrição  $B \leq R$ , é onde essa variação na área relativa à variação na tensão é a menor possível, como mostrado na Figura 42.

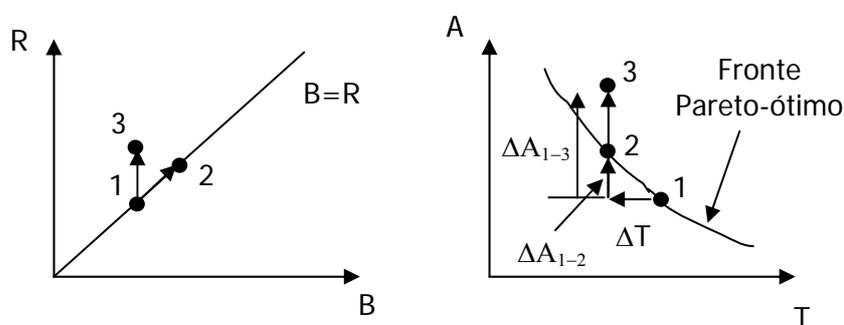


Figura 42 - Incremento na área retirada para uma mesma redução na tensão. Ao longo de  $B = R$ , a variação na área é menor.

Pode-se perceber, também nesse exemplo, em que se utilizou a interpolação linear para obtenção das respostas, a tendência para uma melhor representação do fronte com o incremento de pontos na malha.

Então, quando há interesse pela região de *trade-off* do fronte, ou seja, longe dos pontos de mínimos (máximos) globais, a interpolação linear pode ser adequada, pois permite um enriquecimento da malha mais controlável.

## 8. Conclusões

### 8.1. Algoritmo de Otimização

A principal contribuição deste trabalho foi o desenvolvimento de um dos mais simples algoritmos não-triviais de otimização multiobjetivo capaz de gerar um conjunto suficientemente diversificado de soluções do fronte de Pareto com um nível de precisão, robustez e eficiência comparáveis aos melhores algoritmos GA e ES encontrados na literatura.

O procedimento proposto, denominado (1+1)-PAES-M (Estratégia Evolutiva com Pareto Arquivado, Modificado), baseou-se no método (1+1)-PAES original, com a introdução das seguintes modificações principais:

- i. Foi desenvolvido um algoritmo de Estratégia Evolutiva (1+1)-ES de pesquisa global, através da utilização de dois níveis de desvio padrão para mutação: um global e um local. Desta forma, obteve-se um algoritmo de Estratégias Evolutivas com características mais próximas de uma estratégia de busca global;
- ii. O algoritmo (1+1)-ES realiza tanto simulações individuais como simulações conjuntas das variáveis do problema para acelerar a convergência;
- iii. O algoritmo (1+1)-ES permite que as restrições sejam tratadas de forma *hard* ou *soft* sem o emprego de funções de penalização;
- iv. A avaliação da diversidade das soluções ao longo do fronte de Pareto é obtida através de uma métrica baseada na distância de aglomeração (*crowding distance metric*).

A escolha de um algoritmo do tipo Estratégias Evolutivas (ES) foi motivada pelo fato de que os mais recentes e competitivos algoritmos para resolver problemas de otimização global não-linear com um grande número de restrições, tanto do tipo inequação como do tipo igualdade, são baseados em ES.

Os motivos da escolha de um algoritmo de otimização multiobjetivo do tipo (1+1)-PAES são não somente a sua simplicidade computacional e a sua eficiência, quando comparado com outros algoritmos GA e ES do tipo  $(\mu + \lambda)$ , mas, principalmente, a sua confiabilidade. Cabe ressaltar que o aspecto mais importante num algoritmo de otimização é a robustez e não a rapidez com que fornece a solução, notando-se, ainda, que o seu tempo de processamento será provavelmente muito menor que o do programa de análise de resposta.

O principal inconveniente apontado para o algoritmo (1+1)-PAES é o de que ele pode ficar trancado num ponto de mínimo local quando as mutações produzidas no parente não forem suficientemente grandes para atravessar o espaço existente entre o ponto de máximo global e os diversos pontos de máximos locais. Este inconveniente foi contornado no algoritmo proposto (1+1)-PAES-M pela utilização do algoritmo de Estratégias Evolutivas (1+1)-ES, com características de busca global, para gerar as soluções candidatas a participar do fronte de Pareto.

Na busca do fronte Pareto-ótimo para os exemplos de *bench-mark* de 2 objetivos da literatura, o algoritmo os resolveu com ótimo desempenho, quando comparado com os resultados de outros algoritmos normalmente citados, como o NSGA-II, baseado em algoritmo genético.

Para exemplos de problemas de otimização uniobjetivo com ótimos em regiões de restrições severas, desempenhou bem, de modo geral, com exceção de alguns poucos casos, que merecem alguma investigação adicional.

Assim, concluímos que um otimizador baseado em (1+1)-ES com as modificações introduzidas e algumas melhorias adicionais, poderá ser aplicado na maioria dos problemas relacionados a projetos estruturais e em outras aplicações.

Um aspecto importante é que o (1+1)-ES não permite que se aproveitem os benefícios da computação paralela. Nas comparações entre algoritmos, buscou-se uma equivalência no número de avaliações das funções. No caso de algoritmos que trabalham com populações, é esperado que o número de gerações seja bem menor que o número de avaliações das funções. Isso pode fazer diferença na fase de projeto com

cronograma apertado, quando o tempo para se definir uma solução a ser implementada pode ser bem mais importante que o custo computacional para obtê-las. Quando não se estão utilizando técnicas de interpolação, isso é uma desvantagem do (1+1)-ES. Uma alternativa para contornar esse aspecto e ainda se prosseguir com a linha de gerar a descendência sem a utilização de recombinação seria estudar a implementação do  $(1+\lambda)$ -ES.

## 8.2. Outras conclusões

No trabalho, foram tecidas algumas considerações sobre a obtenção das respostas das funções objetivas e restritivas.

No caso de problemas cujas respostas são obtidas por grande tempo de processamento e cujas variáveis de projeto possam ser tratadas de forma discreta, o algoritmo pode ser utilizado como um controlador (*Shell*). Neste caso, os valores das funções objetivas e restritivas do problema de otimização são obtidos a partir da resposta de análises estruturais por programas de computador que são chamados dentro da *Shell* pelo programa controlador, para cada conjunto simulado de valores das variáveis do problema de otimização.

Para problemas que têm que ser abordados com variáveis contínuas, uma alternativa para se evitar um grande número de análises de resposta com parâmetros indicados pelo otimizador é a interpolação das respostas. A interpolação a partir de uma malha de alguns pontos iniciais pode ser particularmente útil na fase inicial do projeto, quando se quer obter uma primeira avaliação da região de interesse, especialmente quando o custo computacional das análises de resposta é alto. Entretanto, a disposição dos pontos no espaço multidimensional das variáveis de projeto, bem como a escolha do método de interpolação deve ser cuidadosa de forma a propiciar uma interpolação confiável. Observou-se que mesmo a implementação da interpolação linear, procurando-se trabalhar com os  $ND+1$  pontos da malha mais próximos do ponto simulado e ao mesmo tempo evitar-se extrapolação não foi uma tarefa simples e ainda requer prosseguimento nas pesquisas. De qualquer modo, uma conclusão interessante foi a de que a interpolação linear com incrementos progressivos nos pontos na malha para representar

as regiões de *trade-off* do fronte aproximado gerado pode levar a resultados satisfatórios. Identificado o interesse por uma região mais específica, pode-se recorrer à interpolação quadrática local, de forma a contemplar possíveis mínimos (máximos) locais.

Durante o processo de otimização, é esperado que se refaçam simulações com diferentes parâmetros de análise. Assim, no caso de grande número de análises disponíveis de rodadas anteriores em que o programa de análise está sob a *shell* do otimizador, poderá ser mais vantajoso aproveitá-las através de interpolação, por exemplo, do tipo rede neural.

Com relação à formulação de problemas de engenharia, observou-se que ela é um aspecto importante da abordagem dos problemas de otimização. Percebeu-se que a diferenciação entre uma função objetiva e uma função de restrição é tênue, no sentido de que, dependendo do interesse específico do problema ou mesmo das variantes que ele desperta, uma função objetiva pode vir a ser tratada como restrição, haja vista o método clássico da Restrição- $\epsilon$  para solução de problemas multiobjetivo.

Por exemplo, pode-se desejar encontrar uma forma de um corpo que minimize as tensões em pontos críticos ou que as mantenha abaixo de um determinado valor já pré-estabelecido por normas. O mesmo pode valer para deflexões ou para índices de confiabilidade. Também pode-se pensar em delimitar previamente o intervalo de busca de soluções em termos de valores das funções objetivas na montagem do fronte de Pareto, para que fique restrito a uma região de interesse, ainda assim oferecendo solução de compromisso entre as funções.

Com relação a problemas que utilizam o método dos elementos finitos, para o que a parametrização é de grande utilidade, dever-se-á assegurar que o refinamento da malha será adequado para diversas faixas das variáveis de projeto. Isso pode criar alguma dificuldade e requerer elementos de programação para que a malha possa se auto ajustar. Para evitar intervenção do projetista na modelagem quando houver um grande número de rodadas, poderá ser incluída alguma avaliação de condicionamento da malha.

### 8.3. Sugestões de Trabalhos Futuros

Como sugestões para continuidade deste trabalho, podem-se citar:

- a) Prosseguir com as pesquisas sobre a estratégia evolutiva (1+1)-ES, visando a melhorar seu desempenho em problemas com regiões de restrições muito severas. O foco deve-se manter na robustez da busca do ótimo, mesmo que necessite de um número maior de simulações, visando a desfrutar da simplicidade das regras de comparação entre soluções, como meio de alcançar robustez e desempenho na otimização multiobjetivo com o método PAES;
- b) Avaliar alterações na ES de forma a procurar incorporar benefícios da computação paralela, iniciando-se pela implementação do  $(1+\lambda)$ -ES, também sem perder de vista as vantagens da simplicidade nas regras de comparação de solução;
- c) Investigar melhorias na avaliação de densidade de soluções no fronte de Pareto;
- d) Implementar outras medidas de desempenho, como a baseada em superfícies de realização (*attainment surface*);
- e) Aprofundar o estudo de interpolação de funções no espaço ND-dimensional das variáveis de projeto, partindo-se de poucos pontos;
- f) Estudar a alternativa de se trabalhar com o otimizador como uma *shell* dos programas de análise. Existem programas comerciais disponíveis que admitem a utilização do otimizador que se tenha desenvolvido para comandar a atualização de arquivos de entrada e saída para um número crescente de programas de análise. Avaliar a necessidade de se utilizarem seus recursos gráficos, especialmente para os casos de 3 ou mais funções objetivas;
- g) Incluir a avaliação do condicionamento da malha de elementos finitos no problema, de forma a equalizar a qualidade da mesma para as diversas formas geradas com os parâmetros de projeto durante o processo de otimização;

h) Introduzir a confiabilidade estrutural nos problemas de otimização. Dispensar algum cuidado na interpolação da confiabilidade, que deve se apresentar com valores de diferentes ordens de grandeza para diferentes valores das variáveis de projeto, para o que o índice de confiabilidade  $\beta$  pode ser mais adequado;

i) Como sugestões de possíveis aplicações práticas do procedimento proposto, podem-se citar:

i. busca do espaçamento de enrijecedores de cascas mais adequado em estruturas marítimas (transversal,  $s$ , e longitudinal,  $l$ ), levando-se em conta o peso, restrições de nível de tensões, deflexões e/ou confiabilidade, tempo de construção, e outros aspectos. Para unidades de produção, o aspecto de facilidade construtiva, ligada ao tempo de obra é importante quando se pensa no custo da produção adiada;

ii. melhores movimentos de unidades flutuantes de produção, fundamentais para bom desempenho de estruturas de *risers*, da planta e para conforto a bordo, especialmente em situação de mar de través e de estruturas na proa ou popa, sujeitas a grandes deslocamentos e acelerações verticais;

iii. formas de detalhes estruturais reais, especialmente as que visam a alívio de tensões, que costumam se apresentar com muitos parâmetros, como as aberturas encontradas nas unidades flutuantes.

iv. posicionamento de luvas de estacas em torno da perna de jaquetas (otimização de configuração);

v. otimização de topologia de treliças ou pórticos levando-se em conta a confiabilidade.

## 9. Referências Bibliográficas

- Bäck, T., Fogel, D.B. e Michalewicz, Z., 1997, *Handbook of Evolutionary Computation*, Oxford University Press.
- Castro, R.E., 2001, *Otimização de Estruturas com Multi-objetivos Via Algoritmos Genéticos*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- Corne, D., Dorigo, M. e Glover, F., 1999, *New Ideas in Optimization*, The McGraw-Hill Companies.
- Deb, K., 1995, *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall.
- Deb, K., 2000, “An efficient constraint handling method for genetic algorithms”, *Comput. Methods Appl. Mech. Eng.*, v. 186, n. 2/4, pp. 311-338.
- Deb, K., 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd.
- Deb, K., Agrawal, S., Pratap, A. e Meyarivan, T., 2000a, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”. Technical Report 200001, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL).
- Deb, K., Agrawal, S., Pratap, A. e Meyarivan, T., 2000b, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”. In *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI)*, pp. 849-858.
- ESTECO, 2002, *Multi-objective Design Environment - modeFrontier*, User's Manual.

- Fonseca, C.M. e Flemming, P.J., 1993, “Genetic Algorithms for Multiobjective Optimization: Formulation, discussion, and generalization. In *Proceedings of the Fourth International Conference on Evolutionary Programming*, pp. 355-365.
- Fonseca, C.M. e Flemming, P.J., 1995, “On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers”. In *Voigt, H-M., Ebeling, W., Rechenberg, I. and Schwefel, H-P., editors, Parallel Problem Solving From Nature – PPSN IV*, pp. 584-593, Springer.
- Gasser, M., Schuëller e G.I., 1998, “Reliability-based Optimization Utilizing Approximate Methods”, *Structural Safety and Reliability*, Shiraishi, Shinozuka & Wen (eds), Balkema, Rotterdam, pp. 617-624.
- Hamida, S.B. e Schoenauer, M., 2002, “ASCHEA: New Results Using Adaptive Segregational Constraint Handling”, in *Proc. Congr. Evolutionary Computation*, Vol. I, May, pp.884-889.
- Homaifar, A., Lai, S.H.-V. e Qi, X., 1994, “Constrained Optimization via Genetic Algorithms”. *Simulation* 62(4), pp. 242-254.
- Horn, J., Nafploitis, N. e Goldberg, D., 1994, “A Niche Pareto Genetic Algorithm for Multi-objective Optimization”. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp. 82-87.
- Knowles, J.D. e Corne, D.W., 1999, “Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy”, *Evolutionary Computation*, Massachusetts Institute of Technology, v. 7, n. 1, pp. 1-26.
- Mendonça, C.E.L., 2004, “Um Sistema Computacional para Otimização Através de Algoritmos Genéticos e Redes Neurais”. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- Mezura-Montes, E. e Coello, C.A.C., 2003, “Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems”. In *Proc. Congr. Evolutionary Computation 2003 (CEC'2003)*, v. 1, pp. 6-13.
- Mezura-Montes, E. e Coello, C.A.C., 2005, “A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems”. *Transactions on Evolutionary Computation*, v. 9, n. 1, February.
- Michalewicz, Z., 1996, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag.
- Michalewicz, Z. and Attia, N., 1994. “Evolutionary Optimization of Constrained Problems”. In *Proceedings of the Third Annual Conference on Evolutionary Programming*, pp. 98-108.
- Montgomery, D.C., 1991, *Design and Analysis of Experiments*, 3<sup>rd</sup> Edition, John Wiley & Sons.
- Powell, D. e Skolnick, M.M., 1993, “Using Genetic Algorithms in Engineering Design Optimization with Nonlinear Constraints”. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 424-430.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1986, *Numerical Recipes. The Art of Scientific Computing*. Cambridge University Press.
- Rechenberg, I., 1973, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag.
- Reklaitis, G.V., Ravindran, A. e Ragsdell, K.M., 1983. *Engineering Optimization Methods and Applications*. New York: Wiley.

- Rigo, P., 2001, "Least Cost Structural Optimization Oriented Preliminary Design", *Ship Production Symposium*, The Society of Naval Architects and Marine Engineers, Paper 15, pp. 1-18, Ypsilanti, Michigan, 13-15 June.
- Runarsson, T.P. e Yao, X., 2000, "Stochastic Ranking for Constrained Evolutionary Optimization", *IEEE Trans. Evol. Comput.*, v. 4, pp. 284-294, September.
- Runarsson, T.P., 2004, "Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models". In *Parallel Problem Solving from Nature VII (PPSN-2004)*, volume 3242 of LNCS, pp. 401-410. Springer Verlag.
- Schaffer, J.D., 1984, "Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms". Ph. D. Thesis, Nashville, TN: Vanderbilt University.
- Schwefel, H.-P., 1981, *Numerical Optimization of Computer Models*. Chichester, UK: Wiley.
- Srinivas, N. and Deb, K., 1994, "Multi-objective Function Optimization Using Non-dominated Sorting Genetic Algorithms". *Evolutionary Computation Journal* 2(3), pp. 221-248.