

THIAGO PONTIN TANCREDI

Otimização Multidisciplinar Distribuída  
aplicada a Projetos de Engenharia

v. 1

São Paulo  
2008

THIAGO PONTIN TANCREDI

Otimização Multidisciplinar Distribuída  
aplicada a Projetos de Engenharia

Tese apresentada à Escola Politécnica da  
Universidade de São Paulo para a obtenção  
do título de Doutor em Engenharia

v. 1

São Paulo  
2008

THIAGO PONTIN TANCREDI

Otimização Multidisciplinar Distribuída  
aplicada a Projetos de Engenharia

Tese apresentada à Escola Politécnica da  
Universidade de São Paulo para a obtenção  
do título de Doutor em Engenharia

Área de Concentração:  
Engenharia Naval e Oceânica

Orientador: Prof. Livre-Docente  
Oscar Brito Augusto

Coorientador: Prof. Doutor  
Fouad Bennis

v. 1

São Paulo  
2008

# AGRADECIMENTOS

*Aos professores Oscar Brito Augusto e Fouad Bennis, pela paciência e liberdade que permitiram desenvolver essa pesquisa a seu tempo; errando, acertando e aprendendo. E que sempre dispunham de tempo e motivação para me atender.*

*Ao professor Philippe Dépincé e à todos os colegas do IRCCYN, pela excelente acolhida na cidade de Nantes. Em poucos lugares tive tanta tranquilidade para trabalhar.*

*Aos amigos José Afonso Pontin e Monica Dinah, pelo auxílio na difícil arte de escrever em dois idiomas.*

*À toda minha família e amigos, por estarem sempre ao meu lado em todos esses anos de pesquisa.*

*"It's been a long road, getting from there to here,  
it's been a long time, but my time is finally near."*

Russell Watson

# RESUMO

Diante do moderno paradigma de projeto otimizado de grandes sistemas de engenharia, este trabalho apresenta um ambiente de projeto que inclui uma série de inovações que permitem otimizar problemas multidisciplinares utilizando colaboração remota pela Internet.

Muitos trabalhos têm sido desenvolvidos nesse sentido, o que motivou uma parceria internacional entre o Departamento de Engenharia Naval da Universidade de São Paulo e a Ecole Centrale de Nantes. O objetivo desta parceria foi unir a experiência brasileira em projetos de sistemas complexos de engenharia com a experiência francesa em otimização multidisciplinar.

Sendo este trabalho fruto de uma colaboração internacional, é importante ter em mente a obrigação existente para que a pesquisa desenvolvida atenda a ambas as expectativas.

O objetivo é, a um só tempo, viabilizar a integração de subsistemas conhecidos em um ambiente de projeto que permita a otimização multidisciplinar de sistemas complexos de engenharia e desenvolver contribuições pontuais relevantes no uso de superfícies de resposta, projeto distribuído e no desenvolvimento de estratégias de otimização multidisciplinares.

O relatório apresentará uma descrição do problema, seguida da revisão bibliográfica sobre os principais assuntos envolvidos na pesquisa. A descrição do trabalho desenvolvido ocupará os dois próximos capítulos. No primeiro, o trabalho é apresentado privilegiando a visão global do sistema e a integração dos diferentes subsistemas desenvolvidos em um enfoque claramente influenciado pela tradicional visão sistêmica da Engenharia Naval. No capítulo seguinte as principais contribuições pontuais desenvolvidas são detalhadas ao estilo Francês. Por fim é apresentada a aplicação deste trabalho em diversos problemas matemáticos e de engenharia.

# RESUME

En face de la modernité du paradigme de conception optimisée des grands systèmes d'ingénierie, ce travail présente une ambiance de projet qui comprend une série d'innovation qui permettent d'optimiser des problèmes multidisciplinaires à l'aide de la collaboration à distance sur l'internet.

Dans ce sens, de nombreux travaux ont été développés, inspirant un partenariat international entre le Département de l'Université de São Paulo et de l'École Centrale de Nantes. L'objectif de ce partenariat est d'unir l'expérience brésilienne dans la conception de systèmes complexes d'ingénierie avec l'expérience française dans l'optimisation multidisciplinaire.

Étant donné que ce travail est le résultat d'une collaboration internationale, il est important de mettre en relief l'obligation existante pour que cette recherche atteigne aux attentes de tous les organismes y comprise.

Le but de cette recherche est, en même temps, donner les moyens nécessaires pour l'intégration des sous-systèmes connus dans une ambiance de travail pour y permettre l'optimisation multidisciplinaire des systèmes complexes d'ingénierie à l'élaboration des contributions ponctuelles pertinentes dans l'utilisation des surfaces de réponse, optimisation distributive et dans le développement des stratégies d'optimisation multidisciplinaire.

Dans le rapport il y en a donc une description du problème, suivie d'une révision du corpus sur les principaux enjeux abordés dans cette recherche. La description du travail entrepris sera donc abordée dans les deux chapitres suivants. Dans le premier chapitre, on met en relief la vue d'ensemble du système et l'intégration des différents sous-systèmes développés mis au point clairement influencé par le traditionnel point de vue de l'Ingénierie Navale. Dans le deuxième chapitre, les principales contributions ponctuelles développées sont détaillées à la française. Bref, on présente l'application de ce travail dans de divers problèmes en mathématiques également qu'en ingénierie.

# ABSTRACT

Facing the modern paradigm of optimized projects of large engineering systems, this research work presents a project environment that includes a series of innovations that allow the optimization of multidisciplinary problems using remote collaboration through the internet.

Many research works have been done around this subject, which have motivated an international partnership between the Departamento de Engenharia Naval da Universidade de Sao Paulo and Ecole Centrale de Nantes. The objective of this partnership was mix the Brazilian expertise in design of engineering systems with the French expertise in multidisciplinary optimization.

Being the result of an international collaboration, it is important to keep in mind the existing compromise to fulfill both expectations.

The objective is altogether to make the integration of known systems viable in a project environment that allows multidisciplinary optimization of complex engineering systems with the creation of focused contributions that are relevant on the use of response surfaces, design collaborative and on the development of strategies of multidisciplinary optimization.

The report will present a description of the problem, followed by a bibliographic revision about the main issues involved in the research. The description of the work is covered on the two following chapters. On the first one, the work approaches a global view of the system and the integration of different developed sub-systems under a perspective clearly influenced by the traditional systemic vision of Naval Engineering. On the next chapter, the main punctual contributions are detailed in the French style. Finally, the application of this work is presented in several mathematical and engineering problems.

# LISTA DE ILUSTRAÇÕES

<b>Figura 1.1.</b> Projeto multidisciplinar do hidrofólio de um catamarã. Dreyer (2005).....	20
<b>Figura 1.2.</b> Esquematização do projeto distribuído. Rabeau (2006). ....	21
<b>Figura 1.3.</b> Esquema dos principais elementos que motivam esta pesquisa. ....	21
<b>Figura 2.1.</b> Ilustração sobre a otimização de problemas com restrição. Silva (2007).....	28
<b>Figura 2.2.</b> Exemplo de várias alternativas para a compra de um caminhão. ....	31
<b>Figura 2.3.</b> Exemplos de distribuição das soluções ao longo da Fronteira de Pareto.....	32
<b>Figura 2.4.</b> Intersecção da Fronteira de Pareto com a função de ponderação. ....	34
<b>Figura 2.5.</b> Comparação entre a otimização mono-objetivo e multiobjetivo. Rabeau (2006).....	35
<b>Figura 2.6.</b> Representação do algoritmo genético na forma de fluxograma.....	36
<b>Figura 2.7.</b> Operador de cruzamento em um ponto.....	38
<b>Figura 2.8.</b> Cronologia de aparecimento dos diferentes algoritmos evolucionários. Rabeau (2006)....	43
<b>Figura 2.9.</b> Otimização multidisciplinar com objetivos estruturais e hidrodinâmicos. Birk (2001). ....	45
<b>Figura 2.10.</b> Fluxograma do problema multidisciplinar.....	47
<b>Figura 2.11.</b> Principais estratégias de otimização multidisciplinar. Adaptado de Rabeau (2006). ....	49
<b>Figura 2.12.</b> Fluxograma do método MDF. ....	50
<b>Figura 2.13.</b> Fluxograma do método IDF.....	51
<b>Figura 2.14.</b> Fluxograma do método AAO. ....	51
<b>Figura 2.15.</b> Fluxograma do método CO. ....	53
<b>Figura 2.16.</b> Fluxograma do método MORDACE. ....	56
<b>Figura 2.17.</b> Fluxograma do método COSMOS.....	56
<b>Figura 2.18.</b> Representação de um neurônio. ....	60
<b>Figura 2.19.</b> Modelagem de um neurônio proposta por McCulloch (1943).....	61
<b>Figura 2.20.</b> Função de ativação do tipo <i>Sigmoid</i> . ....	62
<b>Figura 2.21.</b> Modelo <i>Perceptron</i> de aprendizado supervisionado. Roseblatt (1958).....	63
<b>Figura 2.22.</b> Rede neural artificial do tipo <i>Multilayer Perceptrons</i> . ....	64
<b>Figura 2.23.</b> Diferença entre o paralelismo lógico e o paralelismo real. Junior (2002). ....	70
<b>Figura 2.24.</b> Tempo de comunicação x Tempo de computação. Quinn (1994). ....	72
<b>Figura 2.25.</b> Programação com memória compartilhada e não compartilhada. Brasil (2006). ....	73
<b>Figura 3.1.</b> Abordagem em espiral do ambiente desenvolvido. ....	77
<b>Figura 3.2.</b> Interface multidisciplinar do módulo de otimização. ....	80
<b>Figura 3.3.</b> Exemplos de gráficos construídos utilizando o objeto de gráficos desenvolvido.....	81
<b>Figura 3.4.</b> Objeto para edição do problema por blocos. ....	82
<b>Figura 3.5.</b> Comparação entre métodos de resolução de expressões na forma de texto.....	84
<b>Figura 3.6.</b> Função $f(x) = \text{seno}(x)$ . ....	85
<b>Figura 3.7.</b> A pesquisa por soluções equivalentes usando regiões tabus. ....	86
<b>Figura 3.8.</b> Abordagens utilizadas pelo programa de otimização. ....	87



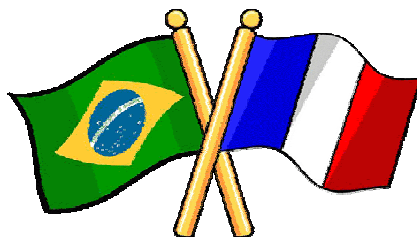
<b>Figura 3.9.</b> Função com singularidade. ....	89
<b>Figura 3.10.</b> Menu de configuração das superfícies de resposta. ....	90
<b>Figura 3.11.</b> Opções para a exportação de dados. ....	91
<b>Figura 3.12.</b> Superfície de resposta exportada para o <i>Excel</i> . ....	92
<b>Figura 3.13.</b> Representação do ambiente desenvolvido. ....	94
<b>Figura 3.14.</b> Interface do servidor de infra-estrutura. ....	96
<b>Figura 3.15.</b> Interface do cliente de redes neurais executando um treinamento remoto. ....	97
<b>Figura 3.16.</b> Comunicação entre o cliente de redes neurais e o servidor de infra-estrutura. ....	99
<b>Figura 4.1.</b> Determinação da Topologia da Rede Neural. ....	103
<b>Figura 4.2.</b> Exemplos de superfícies de resposta. ....	107
<b>Figura 4.3.</b> Redução do módulo dos valores ótimos na superfície de resposta. ....	108
<b>Figura 4.4.</b> Desempenho da superfície de resposta em função do tempo de cada análise. ....	110
<b>Figura 4.5.</b> Projeção do <i>Speedup</i> e eficiência dos algoritmos de otimização utilizando M.E.F. ....	116
<b>Figura 4.6.</b> <i>Speedup</i> em função do tempo de análise. Algoritmo Vértices com 2 processadores. ....	117
<b>Figura 4.7.</b> Superfície de resposta anômala. ....	121
<b>Figura 4.8.</b> Superfície anômala com 3 pontos iniciais. ....	122
<b>Figura 4.9.</b> Superfície de resposta viciada. ....	122
<b>Figura 4.10.</b> Abordagem percentual para a utilização de superfícies de resposta. ....	123
<b>Figura 4.11.</b> Região tabu utilizada no tratamento de superfície viciada. ....	124
<b>Figura 4.12.</b> Comparação entre diferentes abordagens para a utilização de superfícies de resposta. ....	126
<b>Figura 4.13.</b> Formulação geral do problema multidisciplinar. ....	128
<b>Figura 4.14.</b> Esquema geral da estratégia multidisciplinar. ....	129
<b>Figura 4.15.</b> Módulo de análise para a estratégia multidisciplinar. ....	130
<b>Figura 4.16.</b> Módulo de análise para a estratégia multidisciplinar com superfícies de resposta. ....	131
<b>Figura 4.17.</b> Exemplo de utilização de superfícies de resposta para uma disciplina multiobjetivo. ....	132
<b>Figura 5.1.</b> Curvas de nível para o problema Planejamento da Produção. ....	135
<b>Figura 5.2.</b> Solução do problema Planejamento da Produção. a) Vértices. b) Genético. ....	136
<b>Figura 5.3.</b> Formulação do problema de Designação. ....	137
<b>Figura 5.4.</b> Curvas de nível para o problema Rosenbrock. ....	138
<b>Figura 5.5.</b> Solução do problema Rosenbrock. a) Vértices. b) Genético. ....	138
<b>Figura 5.6.</b> Ilustração do problema Tampa de Escotilha. ....	139
<b>Figura 5.7.</b> Curvas de nível para o problema Tampa de Escotilha. ....	140
<b>Figura 5.8.</b> Solução do problema Tampa de Escotilha. a) Vértices. b) Genético. ....	140
<b>Figura 5.9.</b> Ilustração do problema Grelha Simples. ....	141
<b>Figura 5.10.</b> Curvas de nível para o problema Grelha Simples. ....	142
<b>Figura 5.11.</b> Solução do problema Grelha Simples. a) Vértices. b) Genético. ....	142
<b>Figura 5.12.</b> Diagrama de cavitação simples de Burril. ....	143
<b>Figura 5.13.</b> A superfície neural para o diagrama de cavitação de Burril, apud Ugarte. ....	143

<b>Figura 5.14.</b> Representação do problema Viga em balanço. ....	144
<b>Figura 5.15.</b> Solução do problema Viga em balanço. ....	145
<b>Figura 5.16.</b> Distribuição da tensão máxima na solução final.....	145
<b>Figura 5.17.</b> Representação do problema Grelha ortogonal com vigas de seção I.....	146
<b>Figura 5.18.</b> Solução do problema Grelha.....	147
<b>Figura 5.19.</b> Estrutura deformada na configuração otimizada do problema Grelha. ....	148
<b>Figura 5.20.</b> Representação do problema Vaso de Pressão. ....	148
<b>Figura 5.21.</b> Solução do problema Vaso de Pressão. ....	149
<b>Figura 5.22.</b> Distribuição da tensão máxima na solução otimizada. ....	149
<b>Figura 5.23.</b> Distribuição da tensão máxima ao longo de uma seção otimizada do vaso. ....	150
<b>Figura 5.24.</b> Representação gráfica do problema Multiobjetivo Simples. ....	151
<b>Figura 5.25.</b> Solução do problema Multiobjetivo Simples.....	151
<b>Figura 5.26.</b> Representação da função $f_2(X)$ supondo $x_1$ constante. Kalyanmoy (1999). ....	152
<b>Figura 5.27.</b> Solução do problema Multiobjetivo Complexo.....	153
<b>Figura 5.28.</b> Ilustração do problema Viga I. ....	154
<b>Figura 5.29.</b> Solução do problema Viga I. ....	154
<b>Figura 5.30.</b> Comparação entre a solução obtida e solução de referência.....	155
<b>Figura 5.31.</b> Topologia do problema Multidisciplinar Simples. ....	156
<b>Figura 5.32.</b> Solução do problema Multidisciplinar Simples.....	156
<b>Figura 5.33.</b> Comparação entre a abordagem Multiobjetivo e Multidisciplinar. ....	157
<b>Figura 5.34.</b> Topologia do problema Multidisciplinar Acoplado.....	158
<b>Figura 5.35.</b> Solução do problema Multidisciplinar Acoplado. ....	158
<b>Figura 5.36.</b> Topologia do problema Múltiplas Funções Objetivo ....	159
<b>Figura 5.37.</b> Solução do problema Múltiplas Funções Objetivo.....	159
<b>Figura 5.38.</b> Representação tridimensional do Redutor .....	161
<b>Figura 5.39.</b> Topologia do problema Redutor.....	163
<b>Figura 5.40.</b> <i>Layout</i> básico de um catamarã com hidrofólios. Dreyer (2005). ....	164
<b>Figura 5.41.</b> Fluxograma do processo de projeto. Dreyer (2005). ....	165
<b>Figura 5.42.</b> As duas disciplinas envolvidas no projeto e o acoplamento entre elas.....	166
<b>Figura 5.43.</b> Hidrodinâmica de uma embarcação com hidrofólios. Adaptado de Molland (1996). ....	167
<b>Figura 5.44.</b> Variáveis de projeto para a seção do hidrofólio.....	168
<b>Figura 5.45.</b> Deflexão do hidrofólio e distribuição das tensões de von Mises. ....	169
<b>Figura 5.46.</b> Variáveis do projeto estrutural do hidrofólio.....	170
<b>Figura 5.47.</b> Decomposição das forças hidrodinâmicas.....	171
<b>Figura 5.48.</b> Função objetivo maximizada para velocidade do vento de 20 knots.....	173
<b>Figura 5.49.</b> Superfície de resposta calculada para representar o modelo de resistência de Holtrop..	174

# LISTA DE ABREVIATURAS E SIGLAS

<b>AAO</b>	All-at-Once
<b>API</b>	Application Programmer's Interface
<b>ATC</b>	Analytic Target Cascading
<b>BLISS</b>	Bi-Level Integrated System Synthesis
<b>CAD</b>	Computer Aided Design
<b>CO</b>	Collaborative Optimization
<b>COSMOS</b>	Collaborative Optimization Strategy for Multi-Objective Systems
<b>CSSO</b>	Concurrent Sub-Space Optimization
<b>DCF</b>	Disciplinary Constraint Feasible
<b>GSE</b>	Equações de Sensibilidade Global
<b>HLGA</b>	Hajela and Lin Genetic Algorithm
<b>IDF</b>	Individual Disciplinary Feasible
<b>IRCCYN</b>	Institut de Recherche en Communications et Cybernétique de Nantes
<b>MBA</b>	Modified Box Algorithm
<b>MEF</b>	Método dos Elementos Finitos
<b>MDA</b>	Módulo de Análise Multidisciplinar
<b>MDF</b>	Multidisciplinary Feasible
<b>MDO</b>	Otimização Multidisciplinar
<b>MLP</b>	Multilayer Perceptrons
<b>MOGA</b>	Multiobjective Optimization Genetic Algorithm
<b>MOO</b>	Otimização Multiobjetivo
<b>MORDACE</b>	Multidisciplinary Optimization and Robust Design Approaches Applied to Concurrent Engineering
<b>MPI</b>	Message Passing Interface
<b>MPP</b>	Massively Parallel Processors
<b>NACA</b>	National Advisory Committee for Aeronautics
<b>NASA</b>	Aeronautics and Space Administration
<b>NPGA</b>	Niched Pareto Genetic Algorithm
<b>NSGA</b>	Nondominated Sorting Genetic Algorithm
<b>OT</b>	Otimização Topológica
<b>PAES</b>	Pareto Archived Evolution Strategy
<b>PESA</b>	Pareto Envelope based Selection Algorithm
<b>POP3</b>	Post Office Protocol v3
<b>PRAM</b>	Parallel Random Access Machine
<b>PVM</b>	Parallel Virtual Machine
<b>SPEA</b>	Strength Pareto Evolutionary Algorithm
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>SUMT</b>	Sequential Unconstrained Minimization Technique
<b>TCP/IP</b>	Transmission Control Protocol/ Internet Protocol
<b>WBGA</b>	Weight Based Genetic Algorithm
<b>VEGA</b>	Vector Evaluated Genetic Algorithm
<b>VOES</b>	Vector Optimized Evolution Strategy

# ESTRUTURA DO TRABALHO



Este trabalho é resultado de um programa de co-orientação, fruto de uma parceria entre o Departamento de Engenharia Naval da Universidade de São Paulo, a Ecole Centrale de Nantes e a Universidade de Nantes.

Para atender ao convênio fixado, o relatório apresentado foi dividido em dois volumes. O primeiro volume apresenta a totalidade do trabalho realizado e foi redigido em português, em acordo com as normas da instituição na qual a primeira matrícula foi realizada. Um segundo volume, redigido em francês, apresenta um resumo substancial dos aspectos mais relevantes da pesquisa, apresentando referências ao primeiro volume.

Em adição aos volumes impressos, inclui-se ainda um CD-ROM contendo uma versão digital deste relatório, uma cópia dos artigos usados como referência desta pesquisa, uma versão funcional do ambiente de otimização multidisciplinar distribuída desenvolvido, os arquivos de entrada dos exemplos apresentados e uma cópia dos artigos publicados durante o trabalho.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>16</b>
1.1	DESCRIÇÃO DO PROBLEMA.....	19
<b>2</b>	<b>ESTADO DA ARTE .....</b>	<b>22</b>
2.1	OTIMIZAÇÃO .....	22
2.1.1	Definição de otimização .....	22
2.1.2	Breve histórico sobre otimização .....	24
2.1.3	A terminologia do problema de otimização.....	25
2.1.4	Principais algoritmos de otimização mono objetivo .....	27
2.1.5	Comparação entre métodos de otimização.....	28
2.1.6	Problema com restrições .....	28
2.1.7	Problemas com múltiplos objetivos .....	30
2.1.8	Operadores genéticos .....	36
2.1.9	O desenvolvimento histórico dos algoritmos evolucionários .....	39
2.1.10	Problemas multidisciplinares .....	44
2.1.11	O desenvolvimento histórico das estratégias multidisciplinares .....	48
2.1.12	Métodos de otimização multidisciplinar mono-nível.....	49
2.1.13	Métodos de otimização multidisciplinar multi-nível.....	52
2.1.14	As estratégias MORDACE e COSMOS .....	55
2.2	SUPERFÍCIES DE RESPOSTA .....	58
2.2.1	Principais aplicações de superfícies de resposta .....	58
2.2.2	Abordagens para a utilização de superfícies de resposta.....	59
2.2.3	Conceitos fundamentais sobre redes neurais artificiais .....	60
2.2.4	O uso de redes neurais na representação de superfícies de resposta.....	65
2.3	PROCESSAMENTO PARALELO E DISTRIBUÍDO .....	66
2.3.1	Breve histórico sobre processamento paralelo e distribuído .....	66
2.3.2	A computação de grande porte.....	67
2.3.3	Principais conceitos do processamento paralelo e distribuído .....	69
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>76</b>
3.1	ABORDAGEM CONCEITUAL.....	76
3.2	IMPLEMENTAÇÃO .....	78
3.2.1	Programação orientada a objetos .....	79
3.3	O CLIENTE DE OTIMIZAÇÃO .....	79
3.3.1	Interface.....	80
3.3.2	Objeto para geração de gráficos.....	81
3.3.3	Objeto para edição do problema.....	82
3.3.4	Objeto para resolução vetorial de expressões strings .....	83
3.3.5	Regiões tabus.....	85
3.3.6	Algoritmos .....	86
3.3.7	A otimização em regiões viáveis normalizadas .....	87
3.3.8	A otimização em regiões viáveis discretas .....	88
3.3.9	Superfícies de Resposta .....	89
3.3.10	Módulos de análises .....	90
3.3.11	Exportação de Dados .....	91
3.4	O AMBIENTE DE PROGRAMAÇÃO DISTRIBUÍDA DESENVOLVIDO.....	92
3.4.1	Requisitos do sistema.....	93
3.4.2	Implementação da programação distribuída.....	95
3.4.3	Iniciando o ambiente de programação distribuída. ....	95
3.4.4	A otimização utilizando ambiente distribuído .....	98
<b>4</b>	<b>CONTRIBUIÇÕES RELEVANTES.....</b>	<b>101</b>

4.1	O USO DE REDES NEURAIS ARTIFICIAIS COM TREINAMENTO DINÂMICO .....	102
4.1.1	<i>O modelo e topologia da rede neural utilizada .....</i>	<i>103</i>
4.1.2	<i>A geração de superfícies de resposta inicial .....</i>	<i>104</i>
4.1.3	<i>Otimização e adaptação dinâmica .....</i>	<i>107</i>
4.1.4	<i>Readaptação dinâmica da superfície de resposta .....</i>	<i>108</i>
4.1.5	<i>Avaliação do uso de superfícies de resposta .....</i>	<i>109</i>
4.1.6	<i>O problema de interpolação de gráficos e diagramas .....</i>	<i>111</i>
4.1.7	<i>Conclusão .....</i>	<i>111</i>
4.2	OTIMIZAÇÃO UTILIZANDO AMBIENTE DISTRIBUÍDO PELA INTERNET .....	112
4.2.1	<i>O ambiente desenvolvido .....</i>	<i>112</i>
4.2.2	<i>Otimização e sincronização .....</i>	<i>113</i>
4.2.3	<i>Considerações sobre a eficiência do ambiente distribuído .....</i>	<i>113</i>
4.2.4	<i>Estimativa do tempo total de processo .....</i>	<i>118</i>
4.2.5	<i>Conclusão .....</i>	<i>119</i>
4.3	SUPERFÍCIES ANÔMALAS E VICIADAS, TRATAMENTO COM REGIÕES TABUS .....	120
4.3.1	<i>Superfícies anômalas de resposta .....</i>	<i>120</i>
4.3.2	<i>Superfícies viciadas de resposta .....</i>	<i>122</i>
4.3.3	<i>Abordagens tradicionais para superfícies anômalas e viciadas .....</i>	<i>123</i>
4.3.4	<i>Abordagem baseada em regiões tabus .....</i>	<i>124</i>
4.3.5	<i>Avaliação da utilização de regiões tabus com superfícies de resposta .....</i>	<i>125</i>
4.3.6	<i>Conclusão .....</i>	<i>127</i>
4.4	OTIMIZAÇÃO MULTIDISCIPLINAR BASEADA NOS VALORES DE ACOPLAMENTO .....	127
4.4.1	<i>Diferença entre acoplamentos fortes e fracos .....</i>	<i>128</i>
4.4.2	<i>A estratégia multidisciplinar baseada em valores de acoplamento .....</i>	<i>128</i>
4.4.3	<i>A utilização de superfícies de resposta .....</i>	<i>131</i>
4.4.4	<i>Conclusão .....</i>	<i>132</i>
<b>5</b>	<b>EXEMPLOS DE APLICAÇÕES .....</b>	<b>134</b>
5.1	PROGRAMAÇÃO LINEAR .....	135
5.1.1	<i>Planejamento da Produção .....</i>	<i>135</i>
5.1.2	<i>Problema de Designação .....</i>	<i>136</i>
5.2	OTIMIZAÇÃO MONO-OBJETIVA .....	137
5.2.1	<i>Problema Rosenbrock .....</i>	<i>137</i>
5.2.2	<i>Tampa de Escotilha .....</i>	<i>139</i>
5.2.3	<i>Grelha Simples .....</i>	<i>141</i>
5.3	INTERPOLAÇÃO DE GRÁFICOS E DIAGRAMAS .....	143
5.4	OTIMIZAÇÃO COM ANÁLISES NUMÉRICAS .....	144
5.4.1	<i>Viga em balanço .....</i>	<i>144</i>
5.4.2	<i>Grelha .....</i>	<i>146</i>
5.4.3	<i>Vaso de Pressão .....</i>	<i>148</i>
5.5	OTIMIZAÇÃO MULTIOBJETIVO .....	150
5.5.1	<i>Multiobjetivo Simples .....</i>	<i>150</i>
5.5.2	<i>Multiobjetivo Complexo .....</i>	<i>152</i>
5.5.3	<i>Viga I .....</i>	<i>153</i>
5.6	OTIMIZAÇÃO MULTIDISCIPLINAR .....	155
5.6.1	<i>Multidisciplinar Simples .....</i>	<i>156</i>
5.6.2	<i>Multidisciplinar Acoplado .....</i>	<i>157</i>
5.6.3	<i>Múltiplas Funções Objetivo .....</i>	<i>158</i>
5.7	REDUTOR DE VELOCIDADE .....	160
5.8	PROJETO DO HIDROFÓLIO DE UMA EMBARCAÇÃO CATAMARÃ .....	163
5.8.1	<i>Descrição do problema multidisciplinar .....</i>	<i>165</i>

5.8.2	<i>As características da embarcação projetada .....</i>	171
5.8.3	<i>A abordagem racional do problema.....</i>	171
5.8.4	<i>Resultados da otimização .....</i>	172
<b>6</b>	<b>CONCLUSÕES.....</b>	<b>175</b>
6.1	CONCLUSÕES .....	175
6.2	PERSPECTIVAS DE PESQUISA.....	178
6.2.1	<i>Contribuições teóricas.....</i>	178
6.2.2	<i>Implementações computacionais.....</i>	180
<b>7</b>	<b>REFERENCIAS.....</b>	<b>182</b>

# 1 Introdução

---

*"Uma estrutura maciça será sempre mais resistente que uma estrutura vazada, mas a estrutura vazada pode ser mais eficiente que a estrutura maciça."*

A última década consagrou o processo de otimização como de vital importância para a busca de novos paradigmas nos projetos de engenharia. O trabalho dos engenheiros não se limitou apenas a superar barreiras, mas fazê-lo da forma mais eficiente possível.

Da construção de pontes à exploração de petróleo, passando pela distribuição de produtos e pelo projeto de motores, praticamente qualquer projeto de engenharia atual procura, por meio de um processo de otimização, um diferencial competitivo ou tecnológico que propicie redução de custos ou ganhos de produtividade.

Em geral, essa busca pela “melhor solução” é empreendida de forma intuitiva e depende fortemente da experiência profissional do engenheiro. Nos últimos anos, entretanto, uma corrente crescente de pesquisadores tem buscado alternativas para sistematizar a otimização de projetos de engenharia. Motivados pelo crescente avanço e pela redução de custos dos recursos computacionais, a busca por metodologias e algoritmos de otimização tem se intensificado.

O computador não substituirá o engenheiro; ao contrário, o poder de cálculo do computador permitirá ao engenheiro analisar um número maior de alternativas em menor tempo. Essa maior capacidade de análise permitirá ao projetista ser mais arrojado, propor soluções inovadoras ou mesmo ter a certeza de que a solução escolhida seja a “melhor” solução para o problema em questão.

Evidentemente, o conceito de melhor solução deve ser algo bem definido e deve ter por base um ou mais critérios de qualificação. Existem muitos critérios possíveis para avaliar um projeto, como, por exemplo, custo, peso, tempo, facilidade de construção, etc.; é função do engenheiro definir a importância desses critérios para o projeto em análise.

Em alguns casos, a busca pela melhor solução pode mesmo ser realizada de forma intuitiva ou por meio de processos relativamente simples, pois em geral, estes problemas



apresentam poucas variáveis. Nenhum entregador executa um algoritmo de otimização para encontrar a melhor rota entre três pontos de entrega, pois nesse caso sua experiência permite que ele encontre intuitivamente a melhor solução para o problema. Esse é um exemplo simples do clássico problema do “caixeiro viajante”, muito estudado nos setores de logística.

Nesse caso existem poucas variáveis e um único critério de qualificação, que pode ser a distância percorrida ou ainda o tempo gasto pelo entregador.

Imaginemos, porém, que, em vez de um, sejam trinta entregadores responsáveis por cumprir nove mil entregas num período de um mês. Embora exista ainda um único critério de qualificação, o número de variáveis já não permite que a melhor resposta seja obtida de forma intuitiva.


Existem ainda situações em que um único critério pode não ser suficiente para avaliar uma solução. O projeto de um motor pode ser analisado segundo muitos critérios e depende de muitos parâmetros. Definido que o motor deva ser o mais econômico possível, devem existir combinações diferentes que resultem em níveis de economia relativamente próximos. Se, entretanto, o objetivo for o projeto de um motor econômico, com máxima potência e menor índice de poluição possível? Nesse caso o conjunto de possibilidades em análise deve prever uma solução de compromisso que não será necessariamente a melhor solução para cada um dos três critérios específicos.

Esses critérios são influenciados por diversos parâmetros, e a obtenção da melhor solução depende de ciclos sucessivos de projeto. Mesmo com algumas etapas automatizadas do processo, a variação dos parâmetros depende unicamente da experiência do projetista. Nesse caso, um algoritmo de otimização deve ser capaz de realizar mudanças no projeto, direcionando-o para a solução otimizada.

Esse tipo de relação de compromisso<sup>1</sup> ocorre com grande frequência em projetos de sistemas complexos de engenharia, tais como grandes plataformas de petróleo e navios.

Em geral, a busca pela melhor solução para sistemas sem propulsão (tais como as plataformas de petróleo) passa pelo compromisso entre estabilidade e comportamento dinâmico da unidade sob ações ambientais. Por outro lado, projetos de navios apresentam

---

<sup>1</sup>  Normalmente uma relação de compromisso entre objetivos conflitantes é chamada de “*trade-off*”.

relações envolvendo estabilidade direcional e capacidade de manobra, deformação estrutural e comportamento hidrodinâmico, entre outras.

Na Engenharia de Estruturas o objetivo central de um projeto é a minimização do peso estrutural. Diferentes soluções são comparadas verificando-se o nível de esforços suportados por unidade de massa utilizada. Nesse caso, são poucos os parâmetros envolvidos, e o aspecto mais importante do projeto é a distribuição de massa ao longo da estrutura, seja uma ponte, seja um prédio, seja, enfim, um navio ou um avião.

Uma estrutura maciça será sempre mais resistente que uma estrutura vazada, mas a estrutura vazada pode ser mais eficiente que a estrutura maciça.

A distribuição de massa ao longo da estrutura deve se alinhar com as direções principais das tensões resultantes da excitação externa à qual a estrutura está submetida. Mesmo um engenheiro experiente tem dificuldades em intuir a distribuição de tensões ao longo de estruturas complexas; na maioria das vezes adota soluções conservadoras de projeto.


Seguindo o mesmo princípio, a indústria mecânica busca constantemente aumentar a eficiência da matéria prima gasta na produção de peças, o que demanda um algoritmo capaz de modificar a distribuição de massas para maximizar sua eficiência.

Os problemas expostos até aqui são uma pequena amostra das questões envolvidas nos modernos projetos de engenharia, e como eles apresentam o desafio comum de otimizar a solução adotada.

Embora sejam problemas de otimização de natureza diferente (otimização paramétrica, otimização de formas e otimização topológica<sup>2</sup>) o desafio de avaliar uma solução e dirigi-la a uma “otimização” do projeto é comum a todos eles e é um interessante objeto de estudo.

Assim, o novo paradigma atual deixa de ser a análise de grandes projetos de engenharia (destaques nas décadas de 70 e 80) e passa a ser a otimização global dos

---

<sup>2</sup>  O método de otimização topológica destina-se a otimização de problemas estruturais. Partindo de um domínio contínuo bi ou tridimensional, o processo de otimização topológica retira material de regiões menos solicitadas, otimizando a distribuição das tensões ao longo da estrutura. Atualmente existe ampla bibliografia sobre o assunto, como por exemplo, os trabalhos de Coutinho (2006) e Stump (2006). Os trabalhos de Huang (2007) e Novotny (2002) permitem observar as diferenças entre o método de otimização topológica e a otimização paramétrica. Ambos os trabalhos apresentam o mesmo problema: o projeto do quadro de uma bicicleta. Porém, enquanto Novotny utiliza a otimização topológica, Huang utiliza a otimização paramétrica.

projetos. Propor, desenvolver e implementar algoritmos de otimização que, dentro de um conjunto de soluções possíveis, analise uma solução e determine uma mudança “otimizadora” no projeto, é tópico constante nos fóruns internacionais de engenharia e ganha uma complexidade desafiadora ao envolver problemas com múltiplas disciplinas.

## 1.1 Descrição do problema

Recentemente um novo tipo de otimização vem ganhando corpo junto aos escritórios de projeto e aos laboratórios de pesquisa. Esses novos paradigmas são caracterizados por agregar múltiplos objetivos pertencentes a diferentes campos do conhecimento em um mesmo projeto. No caso dos projetos navais, os objetivos do projeto estrutural estão sendo vinculados a outros objetivos da engenharia naval e oceânica (em especial aos objetivos hidrodinâmicos).

Essa estratégia de vincular múltiplos objetivos pertencentes a diferentes campos do conhecimento ficou conhecida pelo termo “Otimização Multidisciplinar”; como será mostrada na revisão bibliográfica, é assunto em pleno desenvolvimento na última década.

Seguindo essa tendência, a Escola Politécnica da Universidade de São Paulo e a Ecole Centrale de Nantes formaram uma parceria para o desenvolvimento de estratégias de otimização para problemas multidisciplinares, sendo este trabalho um dos frutos dessa parceria. A implementação de tais estratégias tem encontrado, porém, duas barreiras cruciais cuja superação constituem a motivação deste trabalho.

O processo de otimização é composto de dois elementos centrais. Uma ferramenta de análise que deve ser confiável e um algoritmo de otimização. Em geral a ferramenta de análise varia em função do tipo de problema que se pretende otimizar; no caso de problemas multidisciplinares, é comum a utilização de mais de uma ferramenta de análise.

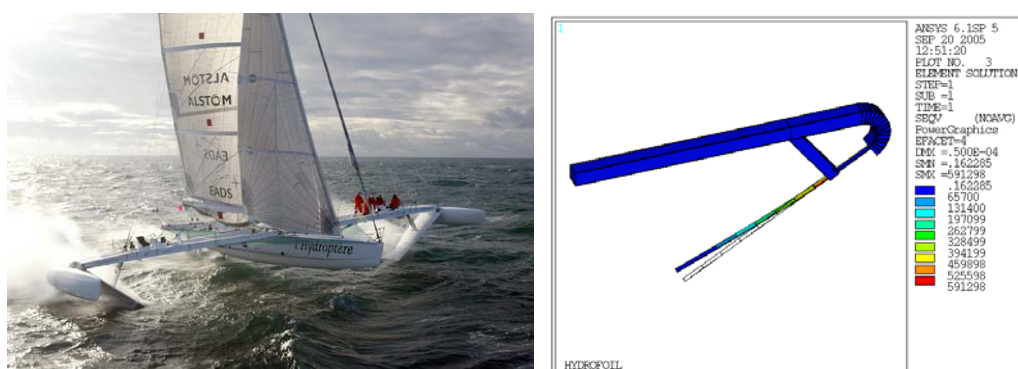
Em engenharia de estruturas é comum utilizar ferramentas baseadas no Método de Elementos Finitos em função da robustez e da eficiência dessas ferramentas. Muito esforço foi despendido na última década no desenvolvimento de ferramentas de análise baseadas no Método dos Elementos Finitos, permitindo que atualmente os esforços se intensifiquem nos algoritmos de otimização.

A utilização de ferramentas baseadas no Método dos Elementos Finitos para análises estruturais introduz, no entanto, um problema sério no processo de otimização, o tempo demandado a cada ciclo de análise muitas vezes inviabiliza a otimização de determinados projetos. Portanto, a primeira barreira observada, por exemplo, em **Dreyer**

(2005) para a otimização de complexos sistemas de engenharia é essencialmente de uma barreira tecnológica.

**Dreyer (2005)** foi um dos primeiros trabalhos frutos da parceria franco-brasileira e demonstra como os mais recentes métodos de otimização esbarram nas limitações computacionais em realizar dezenas e, às vezes, milhares de iterações de análises numéricas, especialmente nos casos em que o Método dos Elementos Finitos é utilizado.

Na figura 1.1, apresentada por **Dreyer (2005)**, é mostrado o projeto multidisciplinar de um catamarã com hidrofólios e a demorada análise utilizando o Método dos Elementos Finitos (figura 1.1b), importante para o projeto estrutural da embarcação.



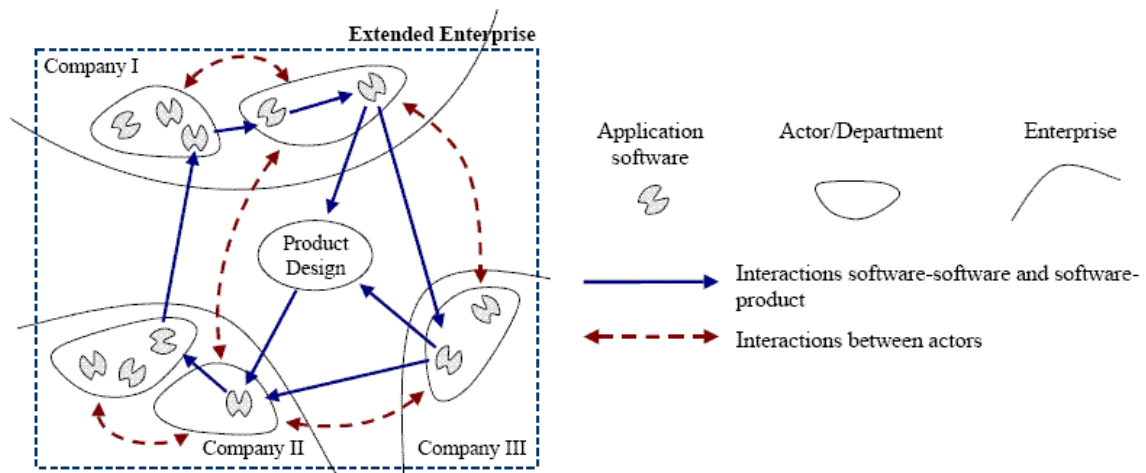
**Figura 1.1.** Projeto multidisciplinar do hidrofólio de um catamarã. Dreyer (2005).

Além de exigir elevados recursos computacionais, a otimização multidisciplinar de sistemas complexos de engenharia pode exigir a cooperação entre diversos profissionais ou grupos de pesquisa que, em alguns casos, atuam fisicamente separados.

Em outro fruto dessa parceria franco-brasileira que motiva este trabalho, **Rabeau (2006)** analisa de forma extensa esta tendência de projetos de sistemas complexos de engenharia em ambientes “colaborativos”.<sup>3</sup> Seu trabalho discute vários aspectos do projeto distribuído<sup>4</sup> em ambientes empresariais e acadêmicos, cuja representação esquemática pode ser vista na figura 1.2.

<sup>3</sup> Deve-se aqui observar que não existe concordância em relação à nomenclatura utilizada. Em muitos trabalhos, como Rabeau (2006) e Giassi (2004), o termo “colaborativo” descreve a utilização de recursos geograficamente separados. No entanto, esse mesmo termo é utilizado para descrever a estratégia de otimização multidisciplinar vista em Braun (1996) e em inúmeros outros trabalhos sobre o assunto. Tentar-se-á, sempre que possível, utilizar o termo “distribuído” para indicar a utilização de recursos geograficamente dispersos, mesmo que não exista paralelismo real do processo.

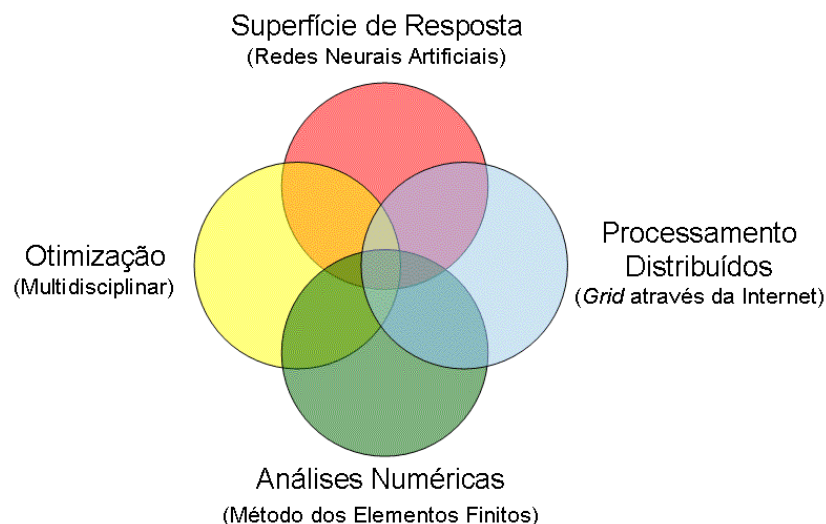
<sup>4</sup> O conceito de projeto distribuído não deve ser confundido com o conceito de engenharia simultânea ou projeto simultâneo. Enquanto a engenharia simultânea se contrapõe ao projeto linear ou em espiral, por permitir que diferentes aspectos de um projeto possam ser elaborados de forma concomitante, o projeto distribuído isola aspectos diferentes do projeto em grupos independentes que podem atuar de forma linear, espiral ou simultânea.



**Figura 1.2.** Esquematisação do projeto distribuído. Rabeau (2006).

A segunda barreira existente é, portanto, como viabilizar que um grupo possa, utilizando seus profissionais, computadores e ferramentas computacionais, atuar em tempo real e de forma colaborativa com outros grupos separados fisicamente.

O objetivo deste trabalho é desenvolver um ambiente de projeto que incorpore uma estratégia de otimização multidisciplinar que possa superar essas barreiras. Propõe-se uma nova estratégia baseada em valores de acoplamento que utiliza superfícies de resposta e processamento distribuído para acelerar a convergência do processo. O esquema dessa abordagem é mostrado na figura 1.3.



**Figura 1.3.** Esquema dos principais elementos que motivam esta pesquisa.

O ambiente computacional desenvolvido e as contribuições teóricas apresentadas são os principais resultados obtidos por esta pesquisa, cujos inúmeros exemplos de aplicação descritos, demonstram a potencialidade da metodologia na realização de projetos assistidos por computador.

## 2 Estado da Arte

---

2.1	Otimização	22
2.1.1	Definição de otimização	22
2.1.2	Breve histórico sobre otimização	24
2.1.3	A terminologia do problema de otimização	25
2.1.4	Principais algoritmos de otimização mono objetivo	27
2.1.5	Comparação entre métodos de otimização	28
2.1.6	Problema com restrições	28
2.1.7	Problemas com múltiplos objetivos	30
2.1.8	Operadores genéticos	36
2.1.9	O desenvolvimento histórico dos algoritmos evolucionários	39
2.1.10	Problemas multidisciplinares	44
2.1.11	O desenvolvimento histórico das estratégias multidisciplinares	48
2.1.12	Métodos de otimização multidisciplinar mono-nível.	49
2.1.13	Métodos de otimização multidisciplinar multi-nível.	52
2.1.14	As estratégias MORDACE e COSMOS	55
2.2	Superfícies de resposta	58
2.2.1	Principais aplicações de superfícies de resposta	58
2.2.2	Abordagens para a utilização de superfícies de resposta	59
2.2.3	Conceitos fundamentais sobre redes neurais artificiais	60
2.2.4	O uso de redes neurais na representação de superfícies de resposta	65
2.3	Processamento paralelo e distribuído	66
2.3.1	Breve histórico sobre processamento paralelo e distribuído	66
2.3.2	A computação de grande porte	67
2.3.3	Principais conceitos do processamento paralelo e distribuído	69

*"If I have seen a little farther than others it is because  
I have stood on the shoulders of giants."  
Isaac Newton*

Neste capítulo será apresentada extensa revisão bibliográfica, bem como os conceitos fundamentais para o desenvolvimento desta pesquisa. O capítulo está estruturado em três seções que reúnem os principais assuntos que compõem a otimização multidisciplinar distribuída sobre superfícies de resposta neurais. Embora cada tópico represente um grande conteúdo, procurou-se apresentar os trabalhos e informações mais relevantes à pesquisa. Alguns assuntos correlatos, tais como o Método dos Elementos Finitos e o uso de Redes Neurais, estão incluídos nas seções pertinentes.

### 2.1 Otimização

#### 2.1.1 Definição de otimização

Como ensina **Silva (2007)**, a melhor forma de entender a importância de se estudar técnicas de otimização é analisar um problema típico de engenharia.

*“Consideremos o projeto da asa de um avião onde se deseja maximizar a rigidez flexional. Suponhamos que os parâmetros que possam ser alterados no projeto sejam a espessura das nervuras, momento de inércia das longarinas, distância entre as nervuras, distância entre as longarinas, espessura da chapa em diferentes pontos, material da estrutura, totalizando 10 parâmetros. Suponha ainda que cada parâmetro possa assumir 10 valores definidos devido a restrições de fabricação e que será usado um software de elementos finitos (MEF) para a simulação da estrutura da asa”.*

*“A abordagem tradicional seria construir um modelo parametrizado de elementos finitos da asa com os parâmetros acima e rodar várias análises de MEF considerando as diversas combinações dos parâmetros acima. Para cada resultado de análise (obtido para uma combinação de parâmetros), calcula-se o valor correspondente da rigidez da asa. De posse dos valores de rigidez da asa para as diferentes combinações de parâmetros, pode-se tomar uma decisão sobre a combinação ótima dos valores desses parâmetros. Essa abordagem é denominada abordagem de análise. No entanto, essa abordagem apresenta uma grande limitação. Suponha que tivéssemos apenas 3 parâmetros (dos 10 acima) para considerar no projeto da asa. Como cada parâmetro pode assumir 10 valores, teríamos  $10^3$  combinações possíveis para serem analisadas. Se cada análise de MEF demora 0,1s (supor um modelo simples de MEF) o tempo total para analisar essas 1000 combinações seria 100s. Agora, considerando os 10 parâmetros, teríamos  $10^{10}$  combinações de parâmetros para serem analisadas, e considerando agora que cada análise de MEF demora 10s, o tempo total passa a ser  $10^{11}$ s, ou seja, 3200 anos!”*

Assim, a abordagem de análise somente faz sentido quando se considera um número muito reduzido de casos a serem estudados; do contrário é necessária uma busca racionalizada da solução por meio de algoritmos de otimização, o que reduz drasticamente o tempo para encontrar a solução ótima.

Pode-se, portanto, definir o processo de otimização como a busca automática, sistemática e dirigida para obter a melhor solução dentro do universo de soluções existentes.

Continuando em seu esforço para definir o processo de otimização, **Silva (2007)** alerta para o fato de que no Brasil o termo “otimização” é constantemente mal utilizado. É comum encontrar trabalhos publicados sob o título de “otimização” de um elemento

estrutural ou mecânico, quando na verdade o que se constata é que o autor, de posse de um método ou ferramenta computacional melhor, realiza uma análise diferenciada que permite verificar que determinada modificação apresenta maior eficiência, sendo, por isso, denominada “estrutura otimizada”.<sup>5</sup>

### 2.1.2 Breve histórico sobre otimização


Uma das áreas de pesquisa em que a otimização tem sido mais intensivamente estudada desde o século XIX é a área do cálculo estrutural. O objetivo básico é reduzir o peso da estrutura mantendo o seu desempenho (rigidez, frequência de ressonância, etc..). Como consequência, é natural que a maior parte das contribuições na otimização de sistemas mecânicos tenham sido desenvolvidas na área de otimização estrutural.

Os primeiros problemas de otimização estrutural foram resolvidos por Maxwell em 1872 e, posteriormente, por Michell, em 1904. Consistiam essencialmente em inicialmente calcular o campo das tensões mecânicas principais, utilizando a teoria da elasticidade, de uma força aplicada num ponto de um domínio infinito que esteja sujeito a restrições de deslocamento em outros pontos. Obtidas as linhas de isotensão principais, a idéia básica, então, era propor nesse domínio uma estrutura formada por barras (treliça), em que cada barra (elemento de treliça) estivesse alinhada com as direções principais de tensão calculadas no domínio. Ou seja, a estrutura ótima (em que o material fosse melhor aproveitado) seria aquela em que os elementos estariam sujeitos apenas a tração e compressão não havendo momentos fletores.

Após os resultados de Michell em 1904, não houve praticamente evolução na otimização estrutural até a década de 60. Durante esse período eram estudados apenas problemas acadêmicos em estruturas simples (vigas, treliças) ainda sem aplicação prática. Embora surpreendentes, esses resultados ficaram esquecidos por um bom tempo por serem considerados na época apenas acadêmicos sem aplicação prática. Na década de 90, no entanto, viriam a ser reproduzidos novamente em domínios de dimensões finitas com o uso de métodos computacionais baseados em otimização topológica.

Ainda que simples, esse tipo de critério de projeto fornece o mesmo resultado que o critério de máxima rigidez com mínimo volume de material e atualmente já é provado (como visto, por exemplo, em **Smith, 2002**) que a configuração ótima para esse critério é uma estrutura de treliças. Assim, utilizando-se do conceito de alinhar as barras com as tensões

---

<sup>5</sup>  Uma revisão bibliográfica atualizada mostra que esse fenômeno foi mais evidente na década de 90.



principais no domínio, Michell obteve resultados surpreendentes de estruturas de treliça para domínios bi e tridimensionais.

Na década de 60, com o surgimento dos computadores e do Método dos Elementos Finitos, problemas práticos de otimização estrutural passaram a ser estudados usando otimização paramétrica, ou seja, alterando-se apenas as dimensões (ou as razões de dimensão da estrutura). Assim, por exemplo, é desenvolvido o método Simplex para a solução de problemas de programação linear.

Na década de 70 são implementados vários algoritmos de otimização para problemas não-lineares bastante utilizados atualmente. Na verdade a formulação teórica de alguns algoritmos já havia sido desenvolvida anteriormente, no entanto somente com o desenvolvimento das linguagens de programação, eles foram implementados. É implementado também o método de otimização de forma, além de métodos probabilísticos como os algoritmos genéticos.

Na década de 80 aparecem os primeiros *softwares* comerciais de otimização estrutural, e alguns *softwares* de elementos finitos passam a incluir módulos de otimização em seu pacote.

Na década de 90, o Método de Otimização Topológica é implementado em *softwares* comerciais, provocando grande repercussão na indústria automotiva e aeronáutica nos Estados Unidos, Japão, e Europa, além de exigir dos engenheiros a capacidade de visualizar complexas estruturas tridimensionais vazadas. Nesse caso ocorre o mesmo processo de redistribuição de massas, no qual elementos de massa são retirados de regiões pouco solicitadas da estrutura, num processo contínuo até se obter uma distribuição otimizada.

A última década tem se caracterizado pelo desenvolvimento de algoritmos e estratégias de otimização para problemas com múltiplos objetivos que podem ser tratados ou não, como problemas multidisciplinares.

A seguir serão discutidos alguns desses algoritmos, em especial os algoritmos e estratégias destinadas aos problemas com múltiplos objetivos.

### **2.1.3 A terminologia do problema de otimização**

O problema de otimização pode ser formulado de maneira padronizada uma vez que a descrição é sempre encontrada na farta literatura sobre o assunto; em **Silva (2007)**, por exemplo.

Visto que a formulação básica do problema de otimização será útil para a compreensão do problema multidisciplinar, ela será apresentada a seguir.

### 2.1.3.1 Variáveis de projeto

As variáveis de projeto, ou variáveis de decisão, são aquelas que caracterizam o projeto e precisam ser definidas pelo engenheiro ou projetista. Essas variáveis agrupam-se em um vetor, constituído de  $n$  variáveis independentes, definido como:

$$X = (x_1, x_2, x_3, x_4, \dots, x_n)^T \quad (2.1)$$

Essas variáveis serão consideradas determinísticas, isto é, não estarão sujeitas a uma distribuição de probabilidades. Podem ter um espectro contínuo ao longo de um intervalo, como no caso do comprimento de um navio, ou podem estar restritas a valores discretos, como no caso da espessura de chapas (definidas por restrições do fabricante).

Alguns autores (Rabeau, 2002, por exemplo), definem outros tipos de variáveis, chamadas de variáveis de estado ou transitórias. De forma genérica pode-se definir como “parâmetros” qualquer grandeza quantificável que não seja uma variável de decisão, uma restrição ou uma função objetivo. A utilização de parâmetros e o tratamento das variáveis discretas merecerão atenção especial no capítulo seguinte deste relatório.

### 2.1.3.2 Função objetivo

A função objetivo é uma função escalar das variáveis de projeto a ser otimizada. Esta função pode ser definida, genericamente, como:

$$f(X) = f(x_1, x_2, x_3, x_4, \dots, x_n) \quad (2.2)$$

Para a maioria dos problemas de engenharia, no entanto, a função objetivo é do tipo não explícita, pois os modelos de projeto não só envolvem equações, como também tabelas, gráficos, simulações ou análises numéricas.

O ponto  $X$  que apresenta o menor valor da função objetivo dentro de toda a região viável, delimitada pelas restrições, é denominado mínimo absoluto ou mínimo global. Já o ponto que apresenta um valor da função objetivo mínimo entre todos os pontos situados numa sub-região, convenientemente definida, é denominado mínimo local.

É importante observar que se tenta, sempre que possível utilizar critérios objetivos ao se estabelecerem as funções objetivo, evitando aspectos como: estética, custos, fabricação, entre outros que possuem aspectos subjetivos ou de difícil formulação.

### 2.1.3.3 Funções de restrição

Os problemas práticos de projeto geralmente estão sujeitos a uma série de restrições que podem ser expressas, de modo geral, como:

$$\begin{aligned} g_j(X) &\geq k \text{ ou } g_j(X) \leq k \\ \text{para } j &= 1, 2, \dots, m \end{aligned} \quad (2.3)$$

Assim como a função objetivo, para a maioria dos problemas de engenharia estas restrições são do tipo não explícitas. O tratamento das funções de restrições (incluindo as restrições homogêneas) será aprofundado no capítulo seguinte.

### 2.1.4 Principais algoritmos de otimização mono objetivo

Na literatura é possível encontrar grande número de algoritmos ligados à otimização, que se dividem em grupos de acordo com o tipo de busca que realizam. Esses métodos estão divididos em métodos para a solução de problemas sem restrição e com restrição.

Considerando que os problemas de otimização, em sua grande maioria, apresentam restrições, é natural se perguntar por que estudar métodos para solução de problemas sem restrições. Basicamente existem duas razões: primeiro, porque os métodos para a solução de problemas com restrições baseiam-se nos métodos para a solução de problemas sem restrições; segundo, porque, como visto em **Wangdahl (1972)**, existem métodos que convertem o problema com restrição num problema sem restrição para depois resolvê-lo, como é o caso do método de penalização.

O primeiro grupo compreende os algoritmos de “Busca Linear” ao qual pertencem os algoritmos “*Bounding the Minimum*”, “*Interval Halving*”, “*Aproximação polinomial de segunda e terceira ordens*” entre outros. Destacam-se ainda os algoritmos de busca irrestrita de “*Primeira Ordem*”, de “*Newton*” e de “*Marquardt*” que utilizam métodos baseados em gradientes; e os algoritmos “*Patter Search*”, “*Simplex Method*” e “*Powell’s Conjugate Direction*” que não utilizam métodos baseados em gradientes. Por fim, encontram-se os algoritmos de busca restrita “*Reduced Gradient*”, “*Box’s Complex*”, “*Penalty Function*” e “*Griffith-Stewart*” (também chamado de “*Linearizações Sucessivas*”). Estes são exemplos de algoritmos tradicionalmente usados na resolução de problemas de otimização paramétrica.

Os detalhes de cada algoritmo são omitidos, pois fogem ao escopo deste trabalho. O leitor mais exigente pode, no entanto, encontrar tais detalhes em farta bibliografia disponível sobre o assunto, como, por exemplo: **Vanderplaats (1984)** e **Haftka (1993)**.

### 2.1.5 Comparação entre métodos de otimização

Tradicionalmente encontram-se na literatura exemplos de *benchmark* bem documentados e que servem como elementos de comparação para analisar e validar a eficiência dos algoritmos de otimização. A comparação em geral é feita levando em consideração o número de iterações e o tempo de cada iteração (medidos sobre o mesmo índice de performance) para a resolução de problemas clássicos.

Um exemplo de problema utilizado neste tipo de comparação é o clássico problema proposto por **Rosenbrock (1960)**. O recente trabalho de **Chase (2008)** descreve este e outros problemas, bem como procedimentos para a comparação de diferentes métodos de otimização.

É importante observar que esta pesquisa não se propõe a desenvolver um novo método de otimização, mas sim desenvolver um ambiente que permita utilizar diferentes métodos de otimização bem documentados em uma estratégia destinada ao projeto de sistemas complexos de engenharia que normalmente apresentam problemas multidisciplinares. Entretanto, a seleção dos algoritmos utilizados no ambiente desenvolvido segue o mesmo critério utilizado na classificação dos algoritmos de otimização.

### 2.1.6 Problema com restrições

Os métodos citados anteriormente resolvem problemas de otimização sem restrições. Para entender melhor a existência de restrições, considere-se a analogia com a escalada de uma montanha; imagine-se que durante a subida encontram-se obstáculos como cercas e grandes pedras que não fazem parte da topologia do terreno em si e que não permitem caminhar sempre no sentido de inclinação máxima, como mostrado na figura 2.1.

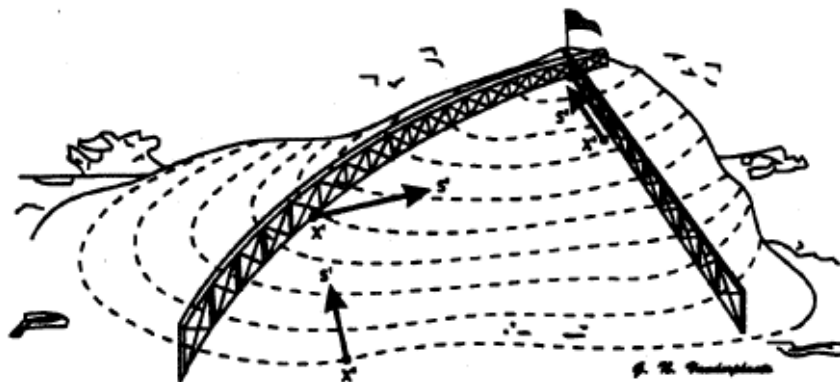


Figura 2.1. Ilustração sobre a otimização de problemas com restrição. Silva (2007).

Esses obstáculos representam as restrições e exigirão que se façam grandes desvios para chegar ao pico da montanha; eventualmente eles nem permitam que se chegue ao pico da montanha, pois esse objetivo pode se encontrar em uma parte inacessível do terreno.

É importante notar que, nesse último caso, a máxima altura que pode ser atingida está situada em algum ponto da cerca. Ou seja, o ponto de mínimo (ou de máximo) não mais está associado com a condição de que o gradiente da função deva ser nulo (estacionaridade).

Existem, essencialmente, dois tipos de métodos para solucionar problemas com restrições: métodos diretos e indiretos. Os métodos diretos verificam as restrições após seguirem o procedimento do algoritmo de otimização sem restrição. Já os métodos indiretos transformam o problema com restrição em um problema sem restrição para em seguida resolvê-lo.

Como demonstra **Wangdahl (1992)**, o método indireto de penalização é uma estratégia robusta e flexível para tratar problemas com restrições, pois pode convertê-los em problemas sem restrição, permitindo que uma grande gama de métodos de otimização possa ser utilizada.

**Fletcher (1973)** sugere que se utilize o método de busca com penalidades. No método das penalidades, define-se uma função auxiliar, que incorpora a função objetivo original e as de restrições, convertendo o problema com restrições num processo de otimização sem restrições.

Seja o problema de otimização como o definido anteriormente. No método das penalidades define-se uma função auxiliar  $\varphi(X, rk)$  apresentada na equação 2.4

$$\varphi(X, rk) = f(X) + \sum_j \rho(rk, g_j(X))$$

(2.4)

*para  $j = (1, 2 \dots n \text{ restrições})$*

Na equação 2.4 a parcela  $p(rk, g(X))$  é chamada de função de penalização, e  $rk$  é uma variável escalar, usualmente denominada fator de penalização. Para problemas de minimização a função de penalização deve ser positiva, enquanto que para problemas de maximização, a função de penalização deve ser negativa.

O problema, antes com restrições, se transforma em diversos problemas sem restrições, já que, para diferentes valores de  $rk$ , pode-se chegar a pontos diferentes na

minimização da função  $\varphi(X, rk)$ . O processo global de busca termina quando entre dois valores distintos de  $rk$  chega-se ao mesmo ponto ótimo de  $\varphi(X, rk)$ . Esse processo foi batizado de SUMT (*Sequential Unconstrained Minimization Technique*).

Conforme originalmente proposto por **Fletcher (1973)**, existem algumas formas de se definir a função de penalização, e a escolha, por uma ou outra, afetará a maneira como o fator de resposta deve ser sequencialmente avaliado.

Por razões óbvias, é importante normalizar os valores das funções objetivo e de restrição. Quando as penalizações, associadas com varias restrições, são combinadas, é necessário ter valores com mesma ordem de grandeza, assegurando-se assim a satisfação de todas as restrições de modo paritário.

Durante a fase de testes, observou-se que, se o valor de  $rk$  for suficientemente grande, uma única busca para o mínimo da  $\varphi(X, rk)$  é necessária. No entanto, valores elevados de  $rk$  podem teoricamente conduzir a um mau condicionamento da superfície  $\varphi(X, rk)$  devido à descontinuidade na fronteira com as restrições. Neste caso, as técnicas de otimização, que dependem da continuidade da superfície  $\varphi(X, rk)$  para obterem um bom desempenho, podem ser pouco eficientes ou até falharem.

## **2.1.7 Problemas com múltiplos objetivos**

### **2.1.7.1 Conceitos Básicos**

Como discutido anteriormente, em muitos casos uma única função objetivo não é suficiente, visto que a tomada de decisões depende do balanceamento de vários critérios. Nestes casos o processo de otimização é denominado otimização multiobjetivo pois trabalha com mais de uma função objetivo.

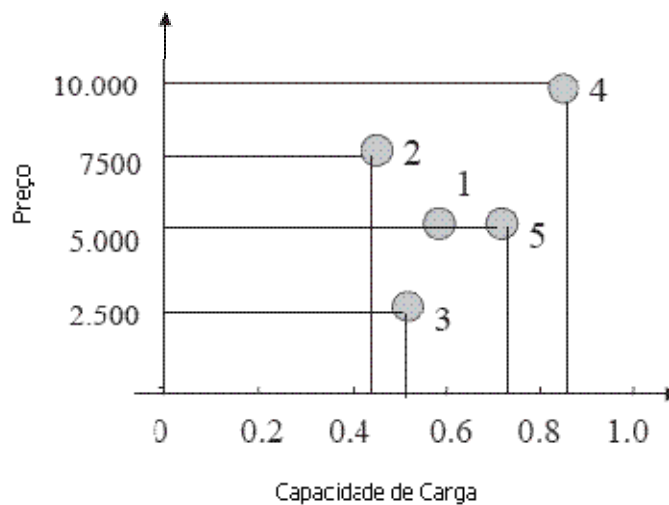
Partindo da formulação do problema de otimização mono-objetivo apresentada na seção 2.1.3, pode-se formular o problema multiobjetivo modificando a equação 2.2 incorporando o índice  $k$  para representar as múltiplas funções objetivo:

$$\begin{aligned} f_k(X) &= f_k(x_1, x_2, x_3, x_4, \dots, x_n) \\ \text{para } k &= (1, 2, \dots, m) \end{aligned} \tag{2.5}$$

### **2.1.7.2 Soluções Pareto-ótimas**

Tomar decisões implica um processo que consiste de vários fatores, cujo objetivo é encontrar a melhor solução. Em alguns casos, podem existir várias soluções boas, das quais nenhuma é absolutamente melhor que a outra.

A melhor forma de entender esse cenário é através de um exemplo: Suponha-se que na hora de comprar um caminhão considera-se o preço e a capacidade de carga. Na figura 2.2 são mostradas várias alternativas de escolha disponíveis.



**Figura 2.2.** Exemplo de várias alternativas para a compra de um caminhão.

O objetivo é minimizar o custo e maximizar a capacidade de carga. Neste caso há cinco possíveis opções de compra. Intuitivamente, descarta-se a solução 1, já que a solução 5 fornece maior capacidade de carga por igual preço. A solução 2 é descartada pela mesma razão. Resultam, então, as soluções: 3, 4, 5, que são boas alternativas de compra. Em termos quantitativos nenhuma é melhor que a outra, pois uma maior capacidade de carga implica maior custo. Como já discutido, existe um compromisso entre os objetivos, pois, quanto maior a capacidade de carga, maior o preço e, vice-versa.

O leitor precipitado pode imaginar que dividindo-se a capacidade de carga pelo preço de cada caminhão, pode-se obter um índice que permite determinar qual compra é a mais vantajosa. Porém é fácil verificar que existem infinitas combinações de preço e carga que resultam em índices de mesmo valor, sendo portanto, igualmente boas.<sup>6</sup>

Diz-se que uma solução domina uma outra se seus valores são melhores em todos os objetivos. Por exemplo, a solução 5 domina a solução 1. Então a solução 5 é não dominada por nenhuma outra. O mesmo acontece com as soluções 3 e 4. Se não se conhece antecipadamente a importância relativa de cada objetivo, pode-se dizer que as soluções 3, 4, e 5 são igualmente boas. Portanto, existe um conjunto de soluções ótimas;

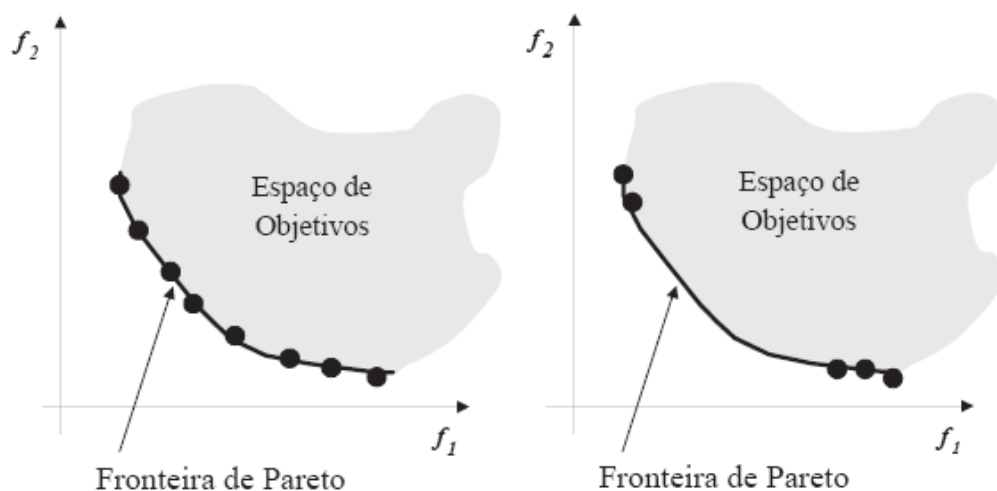
<sup>6</sup> Para verificar isso considere uma das alternativas de compra qualquer. Duplicando-se tanto o preço quanto a capacidade de carga do caminhão, o quociente entre estas características não se altera e, portanto, as duas soluções são igualmente boas.

esse conjunto é chamado de conjunto não dominado. As outras soluções (1 e 2) formam o conjunto dominado.

Se os pontos não dominados estão em um espaço contínuo, pode-se desenhar uma curva. Todos os pontos contidos na curva formam a chamada Fronteira de Pareto. Quando a informação adicional sobre a importância dos objetivos é desconhecida, todas as soluções Pareto-ótimas são igualmente importantes. **Ticona (2003)** apud **Kalyanmoy (2001)** destaca duas importantes metas na otimização de problemas multiobjetivo:

- um conjunto de soluções o mais próximo possível da Fronteira de Pareto.
- um conjunto de soluções com a maior diversidade possível.

A primeira meta é comum para qualquer processo de otimização, pois soluções muito distantes da Fronteira de Pareto não são desejáveis. Porém, encontrar a maior diversidade dentro das soluções é meta específica para a otimização multiobjetivo. Na figura 2.3a mostra-se uma boa distribuição de soluções ao longo da Fronteira de Pareto, enquanto que na figura 2.3b as soluções estão distribuídas apenas em algumas regiões. É necessário assegurar a maior cobertura possível da fronteira a fim de garantir que todas as funções objetivo foram igualmente consideradas.



**Figura 2.3.** Exemplos de distribuição das soluções ao longo da Fronteira de Pareto.

### 2.1.7.3 A otimização de problemas multiobjetivo

Os métodos clássicos de otimização iniciam-se com um único candidato, chamado de solução básica, e pelo cálculo de derivadas se determina para qual direção se deve caminhar na busca do próximo candidato.



Essas técnicas foram, originalmente, formuladas para trabalhar com uma única função objetivo e encontrar uma solução ótima; sendo inadequadas para a resolução de problemas multiobjetivo. Para superar essa limitação existem 3 abordagens clássicas:

A primeira abordagem consiste em combinar as diferentes funções objetivo normalizadas em uma única função, utilizando pesos de acordo com a importância de cada função objetivo (equação 2.6). Assim, os vários objetivos são combinados em uma única função, e os métodos clássicos podem ser utilizados.

$$F_p(X) = \sum_k (\omega_k * f_k(X)) = \omega_1 * f_1(X) + \omega_2 * f_2(X) + \dots + \omega_m * f_m(X) \quad (2.6)$$

Considerando o exemplo da compra do caminhão apresentado na seção 2.1.7.2, o problema poderia ser novamente proposto usando a abordagem de combinação de pesos. A primeira possibilidade seria atribuir a mesma importância ao custo e à capacidade de carga do caminhão. Neste caso a equação 2.6 seria escrita na forma:

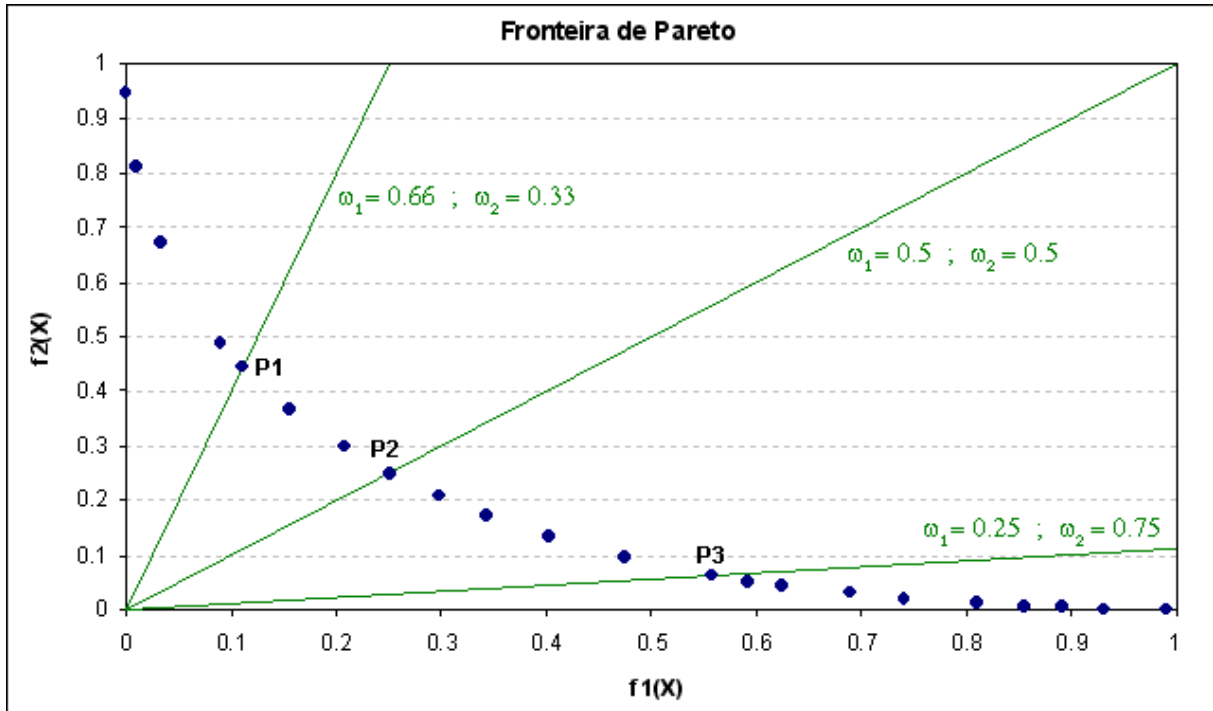
$$F_p(X) = \sum_k (\omega_k * f_k(X)) = 0.5 * f_1(X) + 0.5 * f_2(X) \quad (2.7)$$

Podem-se definir, no entanto, graus diferentes de importância para os objetivos modificando-se os valores de  $\omega_1$  e  $\omega_2$ .

É importante ressaltar que a otimização da função de ponderação, definida na equação 2.6, determina um único ponto da Fronteira de Pareto, dado pela intersecção da Fronteira de Pareto com a reta (ou superfícies, no caso de problemas com mais de duas funções objetivo) definida pelos pesos da função de ponderação.

A Fronteira de Pareto normalizada construída no processo de otimização do exemplo discutido na seção 5.5.1 é mostrada na figura 2.4, juntamente com 3 exemplos de funções de ponderação com diferentes valores para os pesos  $\omega_i$ .

Convém ressaltar que a inclinação das retas definidas pelos pesos da função de ponderação depende da relação entre os pesos e do comportamento das funções objetivo. Neste exemplo, as funções objetivo são quadráticas e, portanto, a inclinação das retas é definida pelo quadrado da relação entre os pesos  $\omega_1$  e  $\omega_2$ .



**Figura 2.4.** Intersecção da Fronteira de Pareto com a função de ponderação.

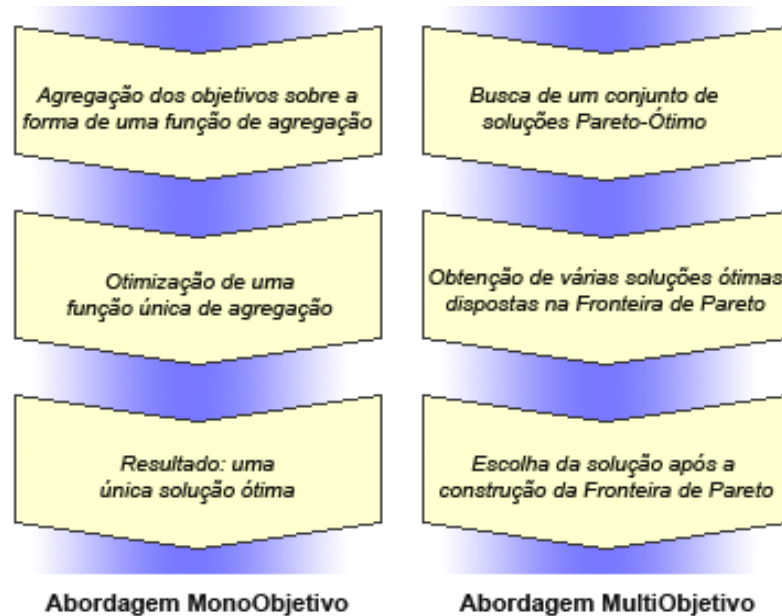
Uma abordagem pouco eficiente para a construção da Fronteira de Pareto consiste em definir diversas funções de ponderação; cuja otimização resultará em um conjunto de pontos que aproximarão a Fronteira de Pareto. Esse método costuma ser denominado “busca exploratória” e certamente foi utilizado nos primeiros trabalhos envolvendo otimização de problemas multiobjetivo.

Uma segunda abordagem consiste em otimizar um dos objetivos estabelecendo valores limites para os outros. Neste caso parte das funções objetivo é transformada em funções de restrições, e o problema passa a ser mono-objetivo:

$$\begin{aligned} F_p(X) &= f_1(X) \\ G_j(X) : f_j(X) &< \text{valor máximo} \quad j \neq 1 \end{aligned} \quad (2.8)$$

No exemplo da compra do caminhão, pode-se reformular o problema de acordo com essa abordagem. Considere-se que o problema passe a ser encontrar o caminhão com a maior capacidade de carga e com custo inferior a X, ou então encontrar o caminhão com o menor custo capaz de levar uma quantidade de carga mínima de Y.

Em ambas as abordagens o problema multiobjetivo passa a ser um problema mono-objetivo, e os métodos tradicionais podem ser facilmente empregados. Existem, porém, diferenças fundamentais entre a otimização multiobjetivo e mono-objetivo, que são detalhadas na figura 2.5.



**Figura 2.5.** Comparação entre a otimização mono-objetivo e multiobjetivo. Rabeau (2006).

Por fim, existem as abordagens que tratam esse tipo de problema com um enfoque realmente multiobjetivo, dispensando os artifícios e técnicas tradicionais. Entre as técnicas para a resolução de problemas multiobjetivo mais promissoras estão os algoritmos genéticos, cujos princípios serão descritos a seguir.

#### **2.1.7.4 Os algoritmos genéticos**

Segundo **Goldberg (1989)** os algoritmos genéticos são técnicas de busca inspiradas em mecanismos de genética e seleção natural. Dada uma população inicial de soluções, esta evolui até os indivíduos convergirem para uma solução, por meio da aplicação de operadores genéticos de seleção, cruzamento e mutação.

Considerando um problema de otimização qualquer, os algoritmos genéticos iniciam a busca da melhor solução a partir de um conjunto inicial de soluções aleatórias. Cada elemento do conjunto inicial de soluções é denominado indivíduo ou cromossomo. Um indivíduo é representado por uma cadeia de símbolos, geralmente, uma cadeia binária. É importante que cada indivíduo da população seja capaz de representar completamente uma possível solução do problema tratado.

Em seguida, uma nova população ou geração é construída a partir da população inicial. Para criar os indivíduos da nova população, são utilizados operadores genéticos de cruzamento e mutação. Uma população é obtida a partir da anterior aplicando-se o cruzamento para aqueles indivíduos com maior valor de aptidão, simulando o processo de seleção natural.

O valor de aptidão é calculado para cada indivíduo por meio de uma função chamada função de aptidão que considera o conceito de dominância de Pareto conforme discutido na seção 2.1.7.2.

O processo de geração de novas populações é repetido iterativamente até que o algoritmo genético chegue a uma solução aceitável, ou satisfaça alguma condição de parada. A representação do algoritmo genético na forma de fluxograma está ilustrada na figura 2.6.



**Figura 2.6.** Representação do algoritmo genético na forma de fluxograma.

### 2.1.8 Operadores genéticos

Nesta seção serão abordados os principais operadores genéticos. É importante ressaltar que os conceitos aqui descritos representam as linhas gerais do processo vistas por exemplo em **Kalyanmoy (2001)**, pois cada algoritmo evolucionário pode apresentar variações específicas nestes procedimentos.

### 2.1.8.1 Seleção para reprodução

A seleção é o processo pelo qual os indivíduos com melhor valor de aptidão têm a maior probabilidade de gerar um ou mais descendentes para a geração seguinte. Esse operador é a versão artificial da seleção das espécies do Darwinismo, que estabelece que os seres mais aptos têm maiores chances de sobreviver, ou seja, os mais fortes e menos vulneráveis aos predadores e doenças.

O objetivo principal do operador de seleção é copiar boas soluções, eliminando soluções de baixa aptidão, mantendo o tamanho da população constante. As melhores soluções são guardadas em uma lista de soluções escolhidas e utilizadas para realizar as operações de cruzamento e mutação.

Existe um grande número de estratégias de seleção. As mais comuns são: seleção por torneio, seleção proporcional, seleção por *ranking* e seleção por sorteio.

Na seleção por torneio, são realizados várias competições entre duas soluções, e a melhor solução é selecionada. Esse processo é repetido até preencher a lista. Segundo **Kalyanmoy (2001)**, esse método possui uma convergência igual às outras estratégias de seleção, ou até mesmo melhor que elas, além de exigir menor complexidade computacional.

Na estratégia de seleção proporcional, o número de cópias de uma solução na lista de soluções escolhidas é proporcional ao seu valor de aptidão. Ou seja, as soluções com melhor valor de aptidão terão mais cópias na lista de soluções.

A seleção proporcional apresenta o problema de escalonamento, pois quando existe uma solução com um valor de aptidão muito maior comparado com o resto da população, esta super-solução terá uma probabilidade de escolha perto de 1 e terá muitas cópias na lista de soluções. Por outro lado, caso todas as soluções possuam valores similares de aptidão, terão a mesma probabilidade de serem escolhidas, e cada uma delas será copiada na lista de soluções. Isto é equivalente a não realizar operação de seleção.

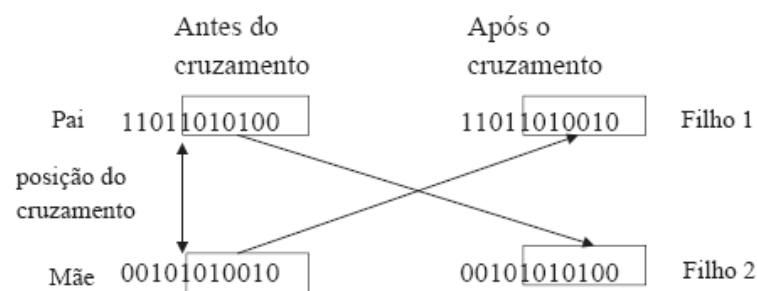
Como destaca **Ribeiro (2006)**, outra estratégia comumente utilizada é a seleção por sorteio. Nesse caso o intervalo de sorteio é dividido em frações cujos tamanhos são proporcionais ao valor de aptidão de cada solução da população. Dessa forma, soluções melhores têm maior chance de serem sorteadas e, portanto, selecionadas.

Por fim, a seleção pelo *ranking* ordena as soluções da população conforme seu valor de aptidão, desde a pior solução (*ranking* 1) até a melhor (*ranking* N). Depois, o número de cópias das soluções é proporcional ao valor de *ranking*.

### 2.1.8.2 Cruzamento

Este operador gera novas soluções (filhas) a partir de soluções escolhidas da lista de soluções (pais) produzida no processo de seleção. O operador de cruzamento possui diferentes variações, muitas delas específicas de um determinado problema. A forma mais simples de cruzamento é conhecida como cruzamento em um ponto, ilustrada na figura 2.7.

Este procedimento consiste em escolher arbitrariamente dois indivíduos na lista de soluções, selecionar dentro da cadeia do indivíduo uma posição  $k$  chamada posição de cruzamento e, por fim, criar novos descendentes trocando as cadeias parciais de cada um dos indivíduos.



**Figura 2.7.** Operador de cruzamento em um ponto.

*“Apesar das operações de seleção e cruzamento parecerem muito simples, é nelas que se concentra o poder dos algoritmos genéticos.”* **Goldberg (1989).**

### 2.1.8.3 Mutação

A mutação é um operador que produz uma alteração aleatória em uma posição de um pequeno número de indivíduos. A mutação é a segunda maneira de os algoritmos genéticos explorarem o espaço de busca. Essa pequena alteração impede que o algoritmo tenha convergência muito rápida, evitando sua estabilização em regiões de mínimos locais.

### 2.1.8.4 Elitismo

O operador de elitismo transfere as melhores soluções encontradas para a geração seguinte, evitando que a melhor solução se deteriore ao longo das gerações.

Uma forma de implementar elitismo é copiar diretamente as melhores soluções da população atual para a população seguinte. As demais soluções são geradas usando os operadores genéticos de cruzamento e mutação sobre a população atual. Desta forma, as melhores soluções passam diretamente para a população seguinte, além de participar da criação do restante das soluções da população seguinte.

Outra forma de elitismo consiste em criar a população seguinte a partir da população atual utilizando os operadores genéticos usuais e escolher as melhores soluções de ambas as populações para compor a geração seguinte.

### 2.1.9 O desenvolvimento histórico dos algoritmos evolucionários

Nesta seção serão descritos os principais algoritmos evolucionários desenvolvidos para a otimização com múltiplos objetivos. Para manter o escopo do trabalho, serão comentados apenas os aspectos mais relevantes de cada algoritmo. Informações adicionais podem ser encontradas em **Ticona (2003)**, **Castro (2001)** e **Rabeau (20006)**.

A primeira implementação de um algoritmo evolucionário para a otimização multiobjetivo foi proposta por **Schaffer (1984)**. O modelo sugerido foi denominado VEGA (*Vector Evaluated Genetic Algorithm*) e foi desenvolvido a partir de uma modificação no algoritmo genético clássico para avaliar cada objetivo separadamente. Como observado por **Castro (2001)**, um dos problemas do VEGA, é que esse algoritmo não obtém boa diversidade nas soluções da Fronteira de Pareto, pois privilegia uma função objetivo em detrimento das outras.

Conforme visto em **Ticona (2003)**, **Goldberg (1989)** criou um procedimento para ordenação de soluções baseado no conceito de dominância. Esse método fornece um valor de aptidão para uma solução  $i$  proporcional ao número de soluções que  $i$  domina. Desta forma, as soluções não dominadas são enfatizadas e terão maior quantidade de cópias na lista de soluções. Para manter a diversidade das soluções, Goldberg sugeriu o emprego de um método de compartilhamento para valorizar soluções em regiões distintas da Fronteira de Pareto. As idéias propostas por Goldberg foram a base para o desenvolvimento de vários modelos de algoritmos evolucionários para problemas multiobjetivo.

Dois anos após o trabalho de Goldberg, **Kursawe (1991)** publicou um método de otimização multiobjetivo denominado VOES (*Vector Optimized Evolution Strategy*). Embora contivesse grande parte das idéias de Goldberg, foi aplicado somente a problemas simples, e sua aplicabilidade a problemas complexos foi questionada, como visto em **Rabeau (2006)**.

O trabalho apresentado por **Hajela e Lin (1992)** propõe um algoritmo baseado na agregação dos objetivos que recebeu o nome HLGA em homenagem a seus autores. Neste procedimento é usado o método da soma dos produtos entre os objetivos e seus correspondentes pesos para o cálculo da função de aptidão:  $(f = \sum w_i \cdot f_i)$ . Normalmente os valores dos objetivos devem ser normalizados para o caso de magnitudes muito diferentes,

embora em seu estudo original a normalização não tenha sido empregada devido à natureza dos problemas testes utilizados.

Os pesos variam no intervalo  $]0, 1[$  tal que o somatório de todos resulte na unidade. A exclusão dos limites do intervalo evita que o problema multiobjetivo seja convertido em um problema mono-objetivo. Para buscar por múltiplas soluções em paralelo, os pesos não são fixos e sim codificados no próprio genótipo. Por esse motivo este método ficou conhecido também por WBGA (*Weight Based Genetic Algorithm*).

O trabalho de **Fonseca (1993)** implementa as sugestões de Goldberg de um modo diferente, o MOGA (*Multi-objective Optimization Genetic Algorithm*) utiliza um procedimento de ordenamento não-dominado. Toda a população é verificada, e todos os indivíduos não-dominados recebem uma posição ou ordem 1. Outros indivíduos são posicionados segundo a não dominância deles em relação ao restante da população do seguinte modo: para cada indivíduo, o número de soluções que o dominam é determinado, logo, a posição deste indivíduo no ordenamento será este número mais 1. Desta forma, no final desse procedimento poderão existir muitos indivíduos compartilhando a mesma posição no ordenamento. A rotina de seleção utiliza esse ordenamento para selecionar ou remover blocos de pontos até escolher os indivíduos para reprodução. Esse tipo de atribuição de aptidão por blocos exerce grande pressão de seleção que pode causar convergência prematura do problema.

A implementação proposta por Fonseca também utiliza o método de formação de nichos para distribuir a população através da região ótima de Pareto, além de compartilhar os valores da função de aptidão. Como ressaltado por **Castro (2002)**, embora isto mantenha a diversidade nos valores da função de aptidão, pode não manter a diversidade no conjunto das variáveis, assim, o MOGA pode não estar apto a achar as múltiplas soluções em problemas em que diferentes pontos ótimos de Pareto correspondam aos mesmos valores de aptidões.

Desenvolvido no mesmo ano que o trabalho de Fonseca, a implementação de **Horn (1993)** propõe um algoritmo genético em que nem todos os indivíduos são substituídos de uma geração para outra. Esse algoritmo foi denominado NPGA (*Niched Pareto Genetic Algorithm*) e faz uso de um procedimento de ordenamento na etapa de seleção baseado no conceito de não dominância. Esse método se caracteriza por não utilizar a função de aptidão. Ao contrário, utiliza um método de seleção por torneio denominado Torneio de Pareto, baseado unicamente no conceito de dominância.



Um conjunto de comparação compreendido de um número específico de indivíduos é tomado randomicamente da população no início de cada processo de seleção. Em seguida, dois indivíduos são retirados da população para selecionar um vencedor conforme o seguinte procedimento: ambas as soluções são comparadas com os membros desse conjunto de comparação para determinar a dominância segundo as funções objetivo. Se um deles é não-dominado, e o outro é dominado, então o ponto não-dominado é selecionado, mas caso ambos sejam não-dominados ou dominados, um contador de nicho é criado para cada indivíduo na população inteira. O contador é baseado no número de soluções da população com uma distância do indivíduo inferior a um valor pré-definido. Assim, a solução que apresentar o menor contador de nicho é selecionada.

Como discutido por **Ticona (2003)**, o conceito de formação de nichos entre os pontos não-dominados é o aspecto mais relevante desse algoritmo, que tem como desvantagem a introdução de dois parâmetros (o tamanho do subconjunto e a distância de referência) que devem ser corretamente configurados.

**Srinivas (1994)** desenvolveu mais uma implementação baseada nas sugestões de Goldberg denominada NSGA (*Nondominated Sorting Genetic Algorithm*). A idéia por trás do algoritmo é a utilização de um procedimento de seleção por ordenamento, juntamente com um método para a criação de nichos (destinado a manter a diversidade da população).

A diferença desta implementação em relação a um algoritmo genético simples está apenas no modo como os indivíduos são selecionados. Antes de o procedimento de seleção ser aplicado, a população é ordenada com base no nível de não-dominância dos indivíduos, isto é, todas as soluções não-dominadas da população corrente recebem valores altos de aptidão. Essa aptidão é a mesma para todos os indivíduos não-dominados, garantindo assim que todos possuam mesmo potencial reprodutivo.

Para manter a diversidade na população, as soluções não-dominadas compartilham os seus valores de aptidão segundo suas distâncias euclidianas. Finalmente, divide-se o valor da aptidão de cada indivíduo pelo contador de nichos (proporcional ao número de vizinhos ao seu redor).

Esse procedimento proporciona a coexistência de pontos ótimos múltiplos na população. O pior valor de aptidão compartilhada da primeira fronteira não-dominada é então guardado para uso posterior. Depois que o compartilhamento é executado, e que as aptidões são modificadas, os indivíduos não-dominados são ignorados temporariamente no processamento do restante da população.

O procedimento para determinar novas soluções não-dominadas (segundo nível) é novamente executado, sendo que agora eles recebem um valor de aptidão um pouco menor que o pior valor de aptidão compartilhada no nível anterior.

Uma vez mais, o procedimento de compartilhamento é executado entre as soluções não-dominadas do segundo nível, e as novas aptidões são calculadas como antes. Esse processo é continuado até que todos os membros da população tenham um valor de aptidão compartilhada.

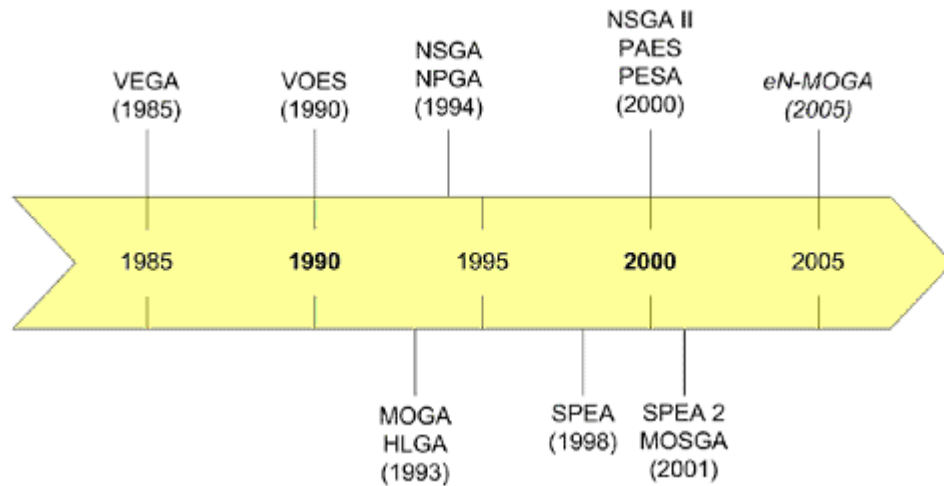
A reprodução da população é efetuada utilizando-se a aptidão compartilhada, ou seja, como o primeiro nível de soluções não-dominadas possui as mais altas aptidões, um número maior de cópias dos seus indivíduos será realizado e conduzirá a busca para a Fronteira de Pareto. Porém, assim como o NPGA, o NSGA apresenta a desvantagem de introduzir novos parâmetros a serem determinados.

Recentemente foi sugerido por **Zitzler (1998)** o SPEA (*Strength Pareto Evolutionary Algorithm*), um algoritmo evolucionário multiobjetivo elitista com conceitos de não dominância. Este algoritmo funciona com a manutenção de uma população externa a cada geração a qual armazena um conjunto de soluções não-dominadas desde a população inicial, além de participar das operações genéticas.

A aptidão de cada indivíduo na população corrente e na população externa é decidida com base no número de soluções dominadas pelo procedimento a seguir descrito. Inicialmente, uma população combinada pela população corrente e a externa é construída. A seguir, todas as soluções não-dominadas nesta população recebem um valor de aptidão baseado no número de soluções que elas dominam e na menor distância euclidiana até outra solução não dominada, mantendo assim, a diversidade. Toma-se o cuidado de não atribuir para as soluções não-dominadas uma aptidão pior que a das melhores soluções dominadas, para que a busca caminhe na direção da fronteira não-dominada e garanta a diversidade entre os indivíduos dominados e não-dominados.

Como destaque do método cita-se a ausência de quaisquer parâmetros de distância, tais como raio de nicho, e o fato de que a aptidão dos indivíduos é determinada apenas pelas soluções armazenadas no conjunto de Pareto externo.

Uma cronologia sobre o desenvolvimento dos principais algoritmos evolucionários para a otimização de problemas multiobjetivo pode ser vista na figura 2.8.



**Figura 2.8.** Cronologia de aparecimento dos diferentes algoritmos evolucionários. Rabeau (2006).

A diferença fundamental dos algoritmos genéticos multiobjetivo em relação aos algoritmos evolutivos tradicionais reside, portanto, basicamente no operador de seleção, dado que a comparação entre duas soluções deve realizar-se conforme o conceito de dominância de Pareto. Em alguns métodos, como o MOGA e o SPEA, o valor de aptidão é proporcional à dominância da solução. Outros métodos, como NPGA, utilizam apenas a dominância de Pareto e não calculam um valor de aptidão.

#### 2.1.9.1 *Elitismo*

Os modelos de algoritmos genéticos multiobjetivo são classificados por **Kalyanmoy (2001)** em dois tipos:

- Não elitistas: são aqueles modelos que, como indica o próprio nome, não utilizam nenhuma forma de elitismo nas suas iterações.
- Elitistas: modelos que utilizam alguma forma o elitismo. Alguns modelos, como o SPEA e o PESA, utilizam uma população externa em que são guardadas as soluções não dominadas encontradas até o momento. Outros métodos, como o NSGA-II, combinam a população atual com a população posterior para preservar as melhores soluções de ambas.

O estudo realizado por **Zitzler (2000)** conclui que o elitismo melhora as soluções encontradas por um modelo de algoritmo genético multiobjetivo. A partir deste trabalho, os métodos desenvolvidos, em sua grande maioria, passaram a incorporar sempre alguma estratégia de elitismo.

### 2.1.9.2 Comparação entre algoritmos genéticos multiobjetivo

A otimização de problemas multiobjetivo é um assunto em pleno desenvolvimento nos fóruns internacionais, resultando constantemente na publicação de novos métodos ou variações dos métodos existentes.

Os recentes trabalhos de **Zitzler (2000)** e **Großan (2002)** descrevem esse cenário e estabelecem procedimentos para comparação e classificação dos algoritmos genéticos multiobjetivo.

A seguir os dados relativos aos diferentes algoritmos genéticos multiobjetivo apresentados até o momento serão resumidos na tabela 2.1.

Tabela 2.1. Comparação entre os diferentes algoritmos evolucionários multiobjetivo.\*

Algoritmo	Cálculo de aptidão	Mecanismo para diversidade	Elitismo	População externa	Vantagens	Desvantagens
VEGA VOES HLGA	Cada população é avaliada em direção a um objetivo diferente	Nenhum	Não	Não	Primeiro algoritmo evolucionário multiobjetivo, fácil implementação	Tendência a convergir sobre os extremos de cada objetivo
MOGA	Dominância de Pareto	Cálculo de aptidão por nichos	Não	Não	Simples extensão de um algoritmo genético mono-objetivo	Convergência geralmente lenta, problemas relacionados com o tamanho dos nichos
NSGA	<i>Ranking</i> que não considera a dominância de Pareto	Cálculo de aptidão por nichos	Não	Não	Rápida convergência	Problemas relacionados com o tamanho dos nichos
NPGA SPEA	Sem cálculo de aptidão, seleção por torneio	Nichos utilizados como desempate na seleção por torneio	Não	Não	Processo de seleção simples	Problemas relacionados com o tamanho dos nichos
SPEA-2 NSGA-II PAES	Força de dominância	Densidade baseada sobre o vizinho mais próximo	Sim	Sim	SPEA melhorado, assegura a conservação dos extremos	Cálculo de aptidão e densidade da população lentos
PESA	Nenhum cálculo de aptidão	Densidade das células	Sim	Sim	Facilidade de implementação, eficiência	A performance depende do tamanho das células, problemas relacionados com o tamanho dos nichos

\* Traduzido de Rabeau (2006).

### 2.1.10 Problemas multidisciplinares

Durante o projeto de estruturas navais e oceânicas (como, por exemplo, grandes plataformas de exploração de petróleo) é comum se deparar com relações de *trade-off* envolvidas no projeto. Em geral, a busca pela melhor solução passa pelo compromisso entre

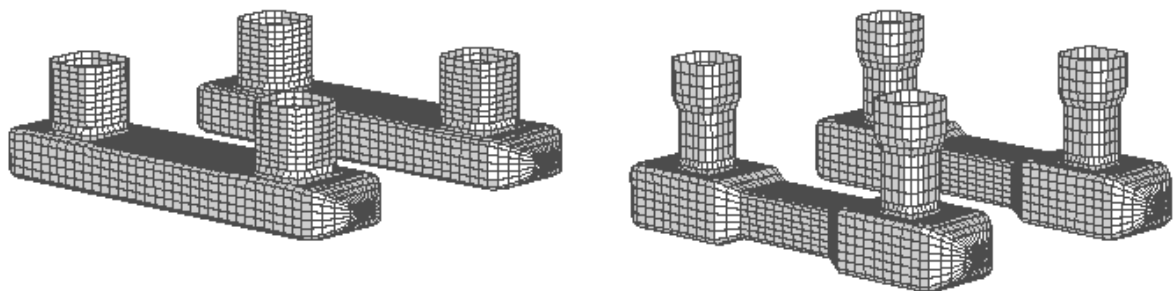
diferentes critérios de qualificação. Esses critérios são influenciados por diversos parâmetros, e a obtenção da melhor solução depende de ciclos sucessivos de projeto. Nesse caso, como em **Silva (2007)**, um algoritmo de otimização deve ser capaz de realizar mudanças no projeto, direcionando-o para a solução otimizada.

Como visto no capítulo anterior, a estratégia de otimização multidisciplinar vem ganhando corpo entre os escritórios de projeto e os laboratórios de pesquisa, agregando múltiplos objetivos pertencentes a diferentes campos do conhecimento num mesmo projeto.

Um dos primeiros trabalhos interessante sobre o assunto pode ser visto em **Birk (2001)** e trata da otimização de formas de embarcações e plataformas *offshore*, agregando os objetivos hidrodinâmicos aos objetivos estruturais.

O trabalho de Birk propõe o projeto automatizado de linhas de casco utilizando um algoritmo de otimização para obter a melhor solução. Embora esse algoritmo dependa fortemente de ferramentas de análise de performance, o algoritmo por si só, quando bem utilizado, pode representar importante ferramenta não apenas para acelerar a convergência do processo cíclico de projeto, mas também para permitir que se explorem arranjos inovadores para o projeto.

Trabalhando com o *downtime*<sup>7</sup> como critério qualificador, os resultados obtidos por Birk são bastante animadores e representam uma diminuição de até 30% do *downtime* de uma plataforma, em comparação com a performance sem a otimização multidisciplinar. A geometria inovadora determinada por Birk pode ser vista na figura 2.9.



**Figura 2.9.** Otimização multidisciplinar com objetivos estruturais e hidrodinâmicos. Birk (2001).

Muitos outros trabalhos têm sido publicados sobre o emprego da otimização multidisciplinar em problemas de engenharia. Ainda na área da engenharia naval e oceânica pode-se destacar o trabalho de **Belegundu (2000)** e o recente trabalho de **Campana**

<sup>7</sup> O termo *downtime* pode ser traduzido como “tempo morto”. Nesse caso se refere ao período de tempo que a plataforma de petróleo deixa de produzir em virtude do comportamento no mar.

(2007). Já na indústria mecânica vale citar os trabalhos de **Batill (1999)** e, mais recentemente, de **Peri (2008)**.

Essas aplicações bem sucedidas incentivam a aplicação da otimização multidisciplinar em projetos de engenharia, ainda que, mesmo internacionalmente, a aplicação de otimização multidisciplinar em projetos de engenharia seja muito restrita e afeita a círculos acadêmicos menos tradicionalistas.

#### **2.1.10.1 A terminologia do problema multidisciplinar**

Nesta seção serão apresentadas as principais definições envolvidas na otimização de problemas multidisciplinares.

- **Disciplina:** sub conjunto do problema que compõe ao menos uma função objetivo. Pode incluir variáveis e restrições específicas da disciplina.
- **Variáveis Comuns:** vetor de variáveis que interferem em todas as disciplinas envolvidas no problema.
- **Acoplamentos:** cada acoplamento representa uma igualdade entre uma variável específica de uma disciplina e um parâmetro (função de restrição ou função objetivo) de outra disciplina. Podem-se tratar os acoplamentos como restrições globais do problema.

É importante observar que alguns autores assinalam a possibilidade de que os acoplamentos sejam definidos por uma igualdade entre uma variável e um parâmetro qualquer de outra disciplina, não necessariamente uma restrição ou objetivo. Entretanto, pode-se transformar qualquer parâmetro em uma restrição, estipulando um valor arbitrariamente grande de tal forma que tal restrição nunca venha a ser violada, nem afete a otimização do problema.

#### **2.1.10.2 Formulação do problema multidisciplinar**

Pode-se formular o problema de otimização multidisciplinar separando o problema multiobjetivo em disciplinas acopladas. O primeiro passo é separar o vetor de variáveis, visto na equação 2.1, em um vetor de variáveis comuns (a todas as disciplinas) e  $n_d$  vetores, cada um contendo as variáveis específicas de cada disciplina, como mostrado na equação 2.9.

$$X = X_C + X_I + X_{II} + X_{III} \dots X_{n_d} \quad (2.9)$$

Os vetores disciplinares devem contém também variáveis artificiais, que correspondem a parâmetros de acoplamentos oriundos de outras disciplinas.

O passo seguinte é agrupar as funções de restrição e as funções objetivo de acordo com as disciplinas a que se referem:

*Disciplina I*

$$\begin{aligned} \min f_{I,i}(X_C + X_I) &= \min f_{I,i}(x_{c,1}, x_{c,2}, \dots, x_{I,1}, x_{I,2}, \dots) \\ g_{I,j}(X_C + X_I) &\leq 0 \end{aligned}$$

(2.10)

*Disciplina II*

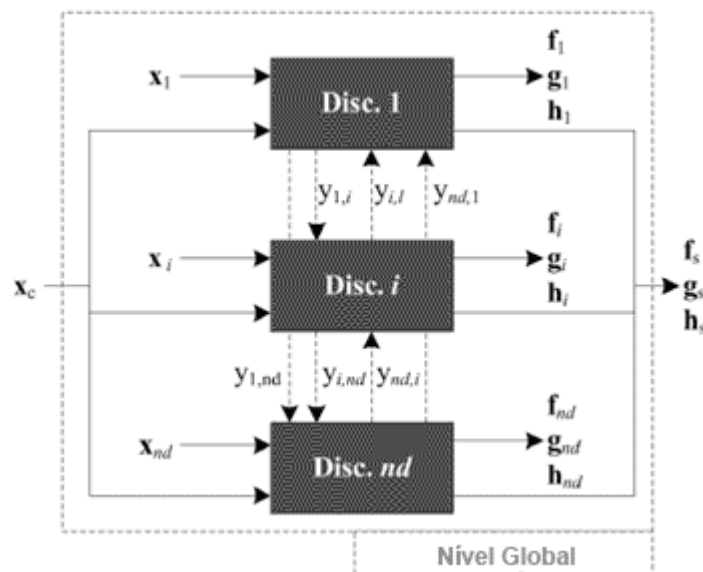
$$\begin{aligned} \min f_{II,i}(X_C + X_{II}) &= \min f_{II,i}(x_{c,1}, x_{c,2}, \dots, x_{II,1}, x_{II,2}, \dots) \\ g_{II,j}(X_C + X_{II}) &\leq 0 \end{aligned}$$

Por fim, descrevem-se os acoplamentos indicando a que disciplina e função correspondem cada variável artificial introduzida:

*Acoplamentos*

$$A_k : x_{nd,nv} = (f_{nd,nf} ; g_{nd,ng}) \quad (2.11)$$

A figura 2.10 permite visualizar melhor a formulação do problema multidisciplinar.



**Figura 2.10.** Fluxograma do problema multidisciplinar.

Pode-se observar que esta é a formulação completa de um problema multidisciplinar, incluindo variáveis comuns, acoplamentos e funções globais de restrição e objetivo. Porém alguns problemas podem apresentar apenas alguns destes elementos.

Embora funções globais de restrição e objetivo possam ser definidas, em alguns casos, pode-se convertê-las em restrições e objetivos locais para melhor se adequarem a estratégia de otimização multidisciplinar utilizada.

É importante observar que a otimização multidisciplinar é um conceito muito mais próximo da engenharia do que da matemática, visto que a separação do problema em disciplinas é artificial e destina-se a facilitar o processo de projeto e análise. Por esse motivo alguns autores, como **Rabeau (2006)**, preferem denominar os problemas multidisciplinares como problemas multiobjetivo do tipo multidisciplinares.

### **2.1.11 O desenvolvimento histórico das estratégias multidisciplinares**

Entre todas as indústrias, a indústria aeroespacial talvez seja aquela submetida aos mais rigorosos critérios e restrições. Por isso, não espanta que os primeiros conceitos de otimização multidisciplinar tenham surgido nesse contexto. O trabalho de **Kodiyalam (1998)** é um dos muitos produzidos pela *National Aeronautics and Space Administration* (NASA) no ano de 1998 e ainda é uma das referências mais importantes sobre o assunto.

De maneira geral, os métodos multidisciplinares podem ser classificados em duas categorias de acordo com a abordagem do problema de otimização: mono-nível e multi-nível.

A abordagem mono-nível é assim chamada por utilizar uma única rotina de otimização localizada no nível global do problema multidisciplinar. Tal abordagem implica uma análise por decomposição, embora a concepção e a otimização continuem centralizadas.

Como visto em **Cramer (1993)** e **Allison (2004)**, a abordagem mono-nível reúne 3 métodos multidisciplinares: MDF (*Multidisciplinary Feasible*), IDF (*Individual Disciplinary Feasible*) e AAO (*All-at-Once*).

Os três maiores inconvenientes dos métodos mono-nível que favoreceram o desenvolvimento das abordagens multi-nível são:

- a centralização completa do projeto não se adéqua ao novo paradigma de projeto distribuído descrito na introdução deste relatório;
- a expressiva quantidade de mensagens trocadas, em função do processo centralizado;
- a exclusão da otimização das disciplinas individualmente.

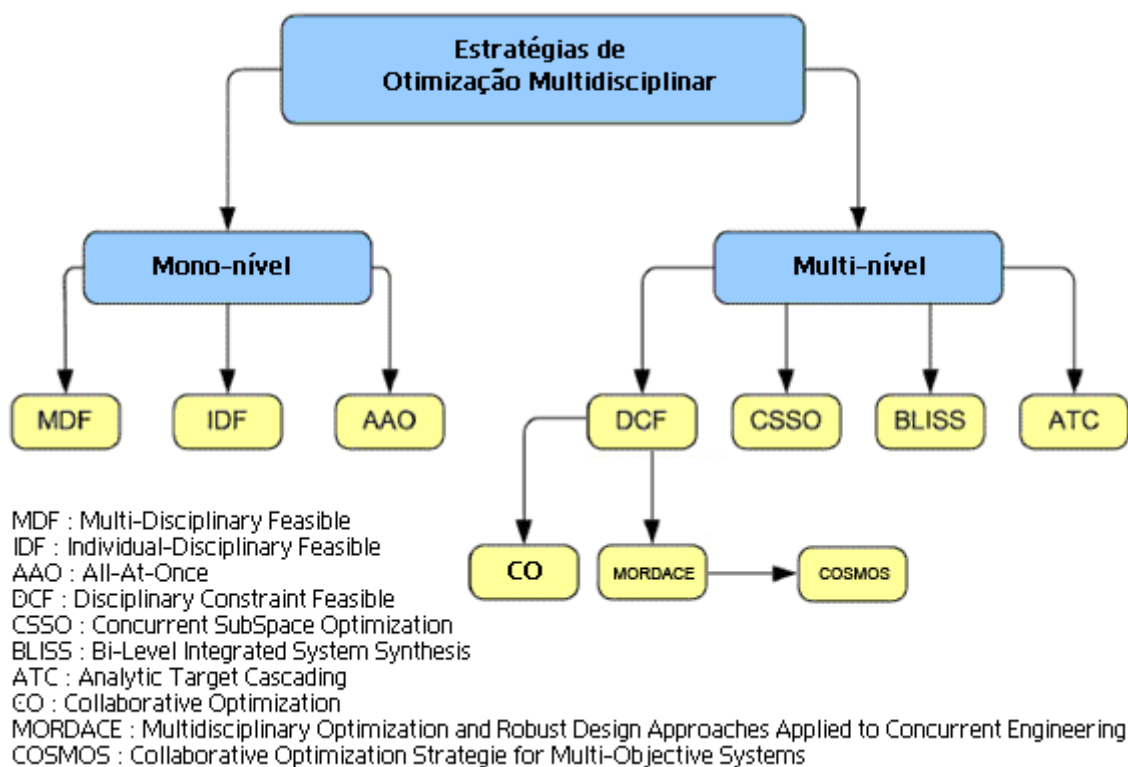


Em oposição aos métodos mono-nível estão os métodos multi-nível que realizam uma otimização descentralizada, conduzindo, em algum momento, processos de otimização mono-disciplinares independentes.

Entre os métodos mais relevantes a utilizar a abordagem multi-nível estão os seguintes: DCF (*Disciplinary Constraint Feasible*), CSSO (*Concurrent Sub-Space Optimization*), BLISS (*Bi-Level Integrated System Synthesis*), ATC (*Analytic Target Cascading*), etc.

No grupo da família de métodos denominadas DCF encontram-se as abordagens CO (*Collaborative Optimization*), MORDACE (*Multidisciplinary Optimisation and Robust Design Approaches Applied to Concurrent Engineering*) e COSMOS (*Collaborative Optimization Stratégie for Multi-Objective Systems*).

Na figura 2.11 são apresentadas, sob a forma de fluxograma, as diferentes estratégias de otimização multidisciplinar.



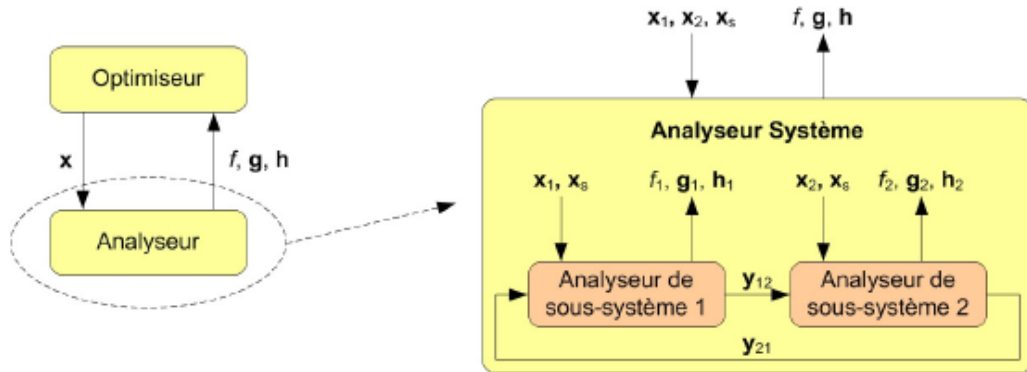
**Figura 2.11.** Principais estratégias de otimização multidisciplinar. Adaptado de Rabeau (2006).

### 2.1.12 Métodos de otimização multidisciplinar mono-nível.

Nesta seção serão apresentadas as três principais estratégias de otimização multidisciplinar mono-nível, enfatizando-se as principais diferenças entre elas.

### 2.1.12.1 Multi-Disciplinary Feasible Design (MDF)

Segundo **Cramer (1993)**, o método MDF possui a formulação mais simples entre todos os métodos existentes para a resolução de problemas multidisciplinares. Esta formulação une um módulo de análise multidisciplinar (MDA) com um módulo de otimização, mostrados na figura 2.12.



**Figura 2.12.** Fluxograma do método MDF.

O processo determina o vetor ótimo global  $z$  e as variáveis locais  $x$  para a função objetivo sujeito a restrições. A solução multidisciplinar encontrada é viável para todo o conjunto de disciplinas. No módulo de análise multidisciplinar, as variáveis  $y$  são determinadas por iterações *Gauss-Seidel* entre várias análises disciplinares (realizadas com base em um conjunto de parâmetros  $x$ ,  $z$  e valores de acoplamento estimados). A abordagem MDF pode ser descrita pela equação 2.12:

$$\begin{aligned} \min \quad & f(z, y_i(x, y_j, z), x) \quad i, j = 1, \dots, n \quad j \neq i \\ & g(z, y_i(x, y_j, z)) \leq 0 \end{aligned} \quad (2.12)$$

### 2.1.12.2 Individual Discipline Feasible (IDF)

O método IDF fornece uma abordagem que permite evitar a otimização unicamente através do módulo de análise multidisciplinar presente no método MDF.

Como visto em **Dennis (1994)**, este método desacopla as análises disciplinares, todavia mantém um módulo de otimização unificado, como mostrado na figura 2.13. Isto permite ao módulo de otimização conduzir processos para cada disciplina individualmente, porém impondo restrições através de variáveis extras de acoplamento  $y'$  que são introduzidas na formulação, conforme discutido por **Haftka (1992)** e **Balling (1994)**.

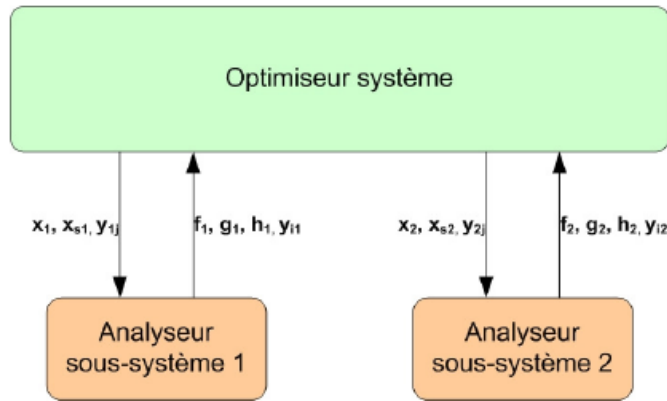


Figura 2.13. Fluxograma do método IDF.

Cada disciplina local pode ser viável, porém o sistema completo pode não ser viável até que o processo convirja. A abordagem IDF pode ser descrita pela equação 2.13:

$$\begin{aligned}
 \min \quad & f(z, y_i(x, y_j', z), x) \quad i, j = 1, \dots, n \quad j \neq i \\
 & g(z, y_i(x, y_j', z), x) \leq 0 \\
 & y_i' - y_i(x, y_j', z) = 0
 \end{aligned} \tag{2.13}$$

onde  $y'$  é um vetor de variáveis extras de acoplamento, criado para desacoplar as análises disciplinares.

### 2.1.12.3 All-at-Once (AAO)

Conforme apresentado por **Cramer (1993)**, o método AAO se caracteriza por uma formulação cuja otimização é altamente centralizada e lembra a formulação do método IDF, porém com a diferença de incluir uma restrição auxiliar que visa anular os resíduos para a convergência do problema. O fluxograma do AAO pode ser visto na figura 2.14.

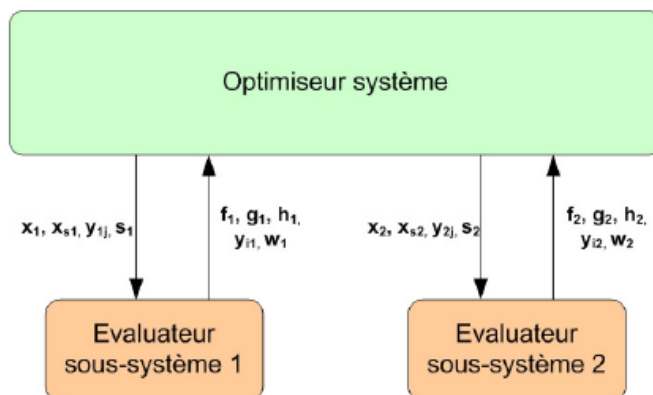


Figura 2.14. Fluxograma do método AAO.

Neste método a análise do sistema e as análises dos subsistemas são realizadas simultaneamente (por isso o nome *All-at-Once*), enquanto a viabilidade da solução é garantida em segundo plano com as restrições residuais.

Segundo **Rabeau (2006)**, os algoritmos genéticos de otimização se adequam muito bem a este método, pois os indivíduos (cromossomos) gerados podem representar o conjunto completo das variáveis de concepção. Ainda segundo Rabeau, esse método, no entanto, só apresenta boa performance para problemas pouco complexos; do contrário, uma simplificação do problema pode ser necessária.

#### 2.1.12.4 Comparação entre as estratégias multidisciplinares mono-nível

Na tabela 2.2 são reunidas as principais características das estratégias de otimização multidisciplinar apresentadas nessa seção.

Tabela 2.2. Comparação entre as principais estratégias multidisciplinares mono-nível.\*

	AAO	IDF	MDF
Utilização de processos locais para cada uma das disciplinas	Não	Sim	Sim, necessita de um acoplamento
Satisfação das equações que regem o sistema	Somente na convergência	A cada iteração de otimização	A cada iteração de otimização
Coerência do sistema	Somente na convergência	Somente na convergência	A cada iteração de otimização
Variáveis de decisão da otimização	$X = (X_i, X_c), Y, S$ $i \in [1..n \text{ disciplinas}]$	$X = (X_i, X_c), Y$ $i \in [1..n \text{ disciplinas}]$	$X = (X_i, X_c)$ $i \in [1..n \text{ disciplinas}]$
Velocidade de convergência	Rápida	Média	Fraca
Robustez	Desconhecida	Alta	Média

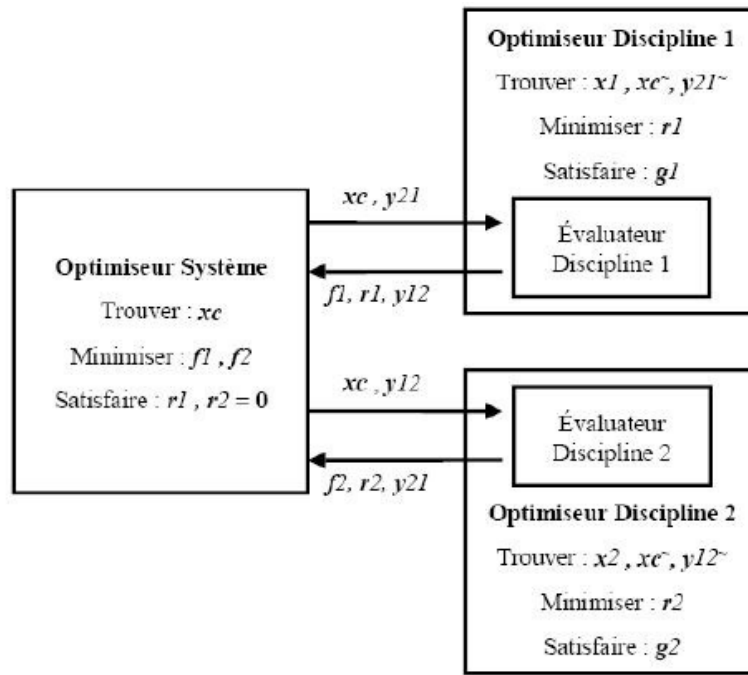
\* Traduzido de Rabeau (2006).

#### 2.1.13 Métodos de otimização multidisciplinar multi-nível.

Nesta seção serão apresentadas as principais estratégias de otimização multidisciplinar multi-nível, enfatizando-se as principais diferenças entre elas. Essas estratégias são importantes por utilizarem uma abordagem descentralizada que permite a utilização de superfícies de resposta, bem como o projeto e processamento distribuído; sendo, por isso, apropriadas para o ambiente desenvolvido.

##### 2.1.13.1 Collaborative Optimization (CO)

A Otimização Colaborativa (CO) introduz ao processo de otimização a decomposição e a descentralização em dois níveis, como descrito por **Braun (1996)** e como pode ser visto na figura 2.15.



**Figura 2.15.** Fluxograma do método CO.

A otimização no nível global é responsável por fornecer para o projeto global as variáveis  $z$  e as respostas do sistema  $y$ . Enquanto que a otimização no nível das disciplinas assegura que as discrepâncias entre as disciplinas sejam eliminadas (para assegurar a viabilidade multidisciplinar), aplicando restrições de compatibilidade.

Esta modelagem visa minimizar as discrepâncias interdisciplinares enquanto satisfaz as restrições locais. A formulação do método CO pode ser descrita pela equação 2.14:

$$\begin{aligned} \min \quad & f(z_{SL}, y_{SL}) \\ J_i(z_{SL}, z_i^*, y_{SL}, y_i^*(x_i^*, y_i, z_i^*)) &= 0 \quad i, j = 1, \dots, n \quad j \neq i \end{aligned} \quad (2.14)$$

onde  $J$  representa as restrições de compatibilidade, uma para cada disciplina. Os parâmetros  $z^*$ ,  $y^*$  e  $x^*$  são os resultados obtidos na otimização no nível disciplinar. A otimização no nível disciplinar de número  $i$  pode ser formulado pela equação 2.15:

$$\begin{aligned} \min \quad & J_i = \sum (z_{SL} - z_i)^2 + \sum (y_{SL} - y_i)^2 \\ g_i(x_i, z_i, y_i(x_i, y_i, z_i)) &\leq 0 \end{aligned} \quad (2.15)$$

### 2.1.13.2 Concurrent Subspace Optimization (CSSO)

O método CSSO utiliza uma estratégia de decomposição a qual permite a otimização simultânea. A vantagem desse método, segundo **Sellar (1996)**, é o fato de as aproximações

dos estados locais <sup>8</sup> ajudarem na compreensão de como as variáveis locais influenciam as restrições e objetivos globais.

Uma condição específica é aproximada a cada otimização disciplinar para simular as variáveis de estado de outras disciplinas. De forma análoga, a otimização global utiliza modelos de aproximação para substituir as análises disciplinares. Então, os modelos disciplinares locais são atualizados com base nas soluções disciplinares otimizadas. A otimização local da disciplina  $i$  pode ser representada pela equação 2.16:

$$\begin{aligned} \min \quad & f(z, y(x_i, y_j^{app}, z_i), y_i^{app}) \quad i, j = 1, \dots, n \quad j \neq i \\ & g_i(x_i, z, y_i(x_i, y_j^{app}, z_i), y_j^{app}) \leq 0 \end{aligned} \quad (2.16)$$

onde  $y_j^{app} = y_j^{app}(z, x_j)$  representa a resposta aproximada de outra disciplina.

Uma análise multidisciplinar completa é realizada para cada solução no nível global de projeto para gerar uma solução multidisciplinar viável utilizada para atualizar o modelo global aproximado. A formulação da otimização global do problema pode ser vista na equação 2.17:


$$\begin{aligned} \min \quad & f(z, y^{app}) \\ & g(z, y^{app}) \leq 0 \end{aligned} \quad (2.17)$$


Para completar o processo, análises multidisciplinares são realizadas com as soluções disciplinares locais para aprimorar os modelos.

### 2.1.13.3 Bi-Level Integrated System Synthesis Method

O método BLISS, proposto por **Sobieszczanski-Sobieski (1998)**, é uma decomposição extensa das equações de sensibilidade global.<sup>9</sup> Este método calcula a derivada total dos valores de acoplamento  $y$  com relação ao local de sensibilidade.

Cada disciplina é otimizada por meio da modificação das variáveis locais  $x$ , mantendo as variáveis globais  $z$  constantes e minimizando o objetivo das disciplinas sobre as restrições locais. As variáveis globais são utilizadas apenas para a otimização global. As

<sup>8</sup>  Originalmente: “approximations of non-local disciplinary states”

<sup>9</sup>  As equações de sensibilidade global (GSE) foram propostas 10 anos antes por Sobieszczanski-Sobieski (1988) para a resolução de sistemas acoplados, e, de forma simplificada, pode-se dizer que representam a derivada do vetor das funções globais em relação ao vetor de variáveis globais.

derivadas totais, obtidas a partir das equações de sensibilidade global, são usadas para prever os efeitos de cada conjunto de variáveis da função objetivo. A otimização da disciplina  $i$  é obtida por meio da equação 2.18.

$$\begin{aligned} \min \quad & d(f, x_i)^T \Delta x_i \\ & g_i(x_i) \leq 0 \end{aligned} \quad (2.18)$$

onde  $d(f, x_i)^T$  é a derivada da função objetivo em relação as variáveis locais e inclui os efeitos indiretos destas variáveis em outras disciplinas. O termo  $d(f, x_i)^T \Delta x_i$  corresponde à previsão de primeira ordem da mudança da função objetivo em relação à mudança de  $x_i$ .

O objetivo global na formulação BLISS é fortemente relacionado com as funções objetivo das disciplinas, as quais são expressas por meio de termos de primeira ordem da série de Taylor:


$$\begin{aligned} \min \quad & \Phi = d(y_{1,i}, x_1)^T \Delta x_1 + d(y_{2,i}, x_2)^T \Delta x_2 + \dots \\ & g_i(z, y(x, z), x) \end{aligned} \quad (2.19)$$

#### 2.1.14 As estratégias MORDACE e COSMOS

Pressionado pelo paradigma de projeto distribuído e inspirado por trabalhos como o de **Alexandrov (1999)**, que apresenta um comparativo entre a estratégia de Otimização Colaborativa e as demais estratégias multidisciplinares, **Giassi (2004)** propôs uma nova estratégia denominada MORDACE (*Multidisciplinary Optimisation and Robust Design Approaches Applied to Concurrent Engineering*).

O método MORDACE foi desenvolvido a partir da colaboração entre a Ecole Centrale de Nantes, o Institut de Recherche en Communications et Cybernétique de Nantes e a Sociedade Sirehna.<sup>10</sup>

O desenvolvimento do MORDACE levou em consideração as vantagens e limitações das estratégias de otimização multidisciplinares existentes: redução do tempo de concepção, flexibilidade e a garantia da implementação de soluções satisfatórias e exeqüíveis

<sup>10</sup>  Criada em 1986, nas vizinhanças do Laboratório de Mecânica dos Fluidos da Escola Centrale de Nantes (ECN/LMF), a sociedade Sirehna destinou-se a transferir as pesquisas acadêmicas para o desenvolvimento de aplicações industriais.

A estratégia MORDACE é baseada em uma abordagem de concepção robusta: o objetivo é encontrar as soluções robustas levando em consideração as diferenças nas variáveis entre as diferentes disciplinas. O método permite a otimização paralela de cada disciplina como representado na figura 2.16.

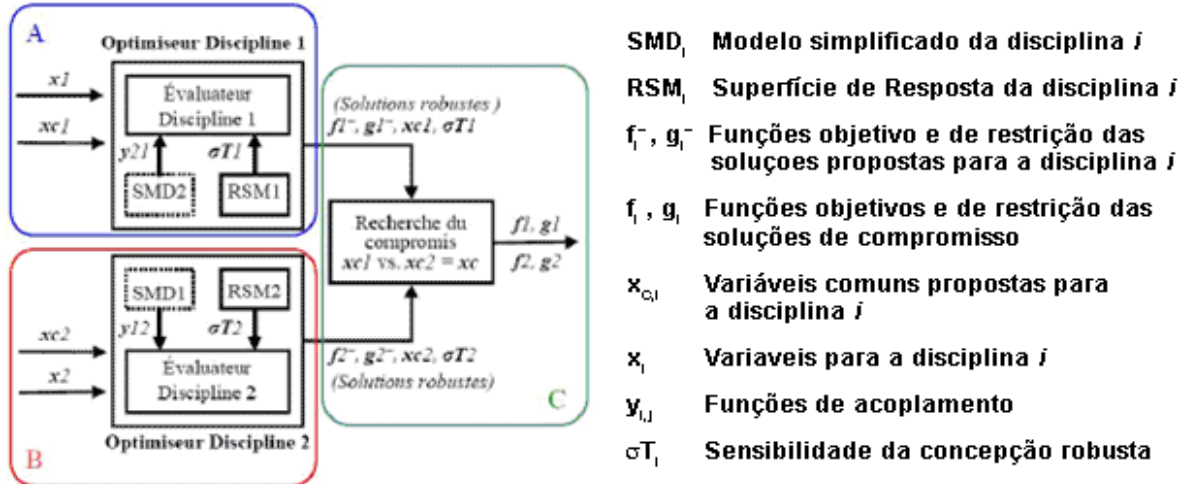


Figura 2.16. Fluxograma do método MORDACE.

Prosseguindo o trabalho de Giassi, **Rabeau (2006)** modificou o MORDACE e desenvolveu sua própria estratégia de otimização multidisciplinar que foi denominada COSMOS (*Collaborative Optimization Stratégie for Multi-Objective Systems*). O fluxograma do método COSMOS pode ser visto na figura 2.17.

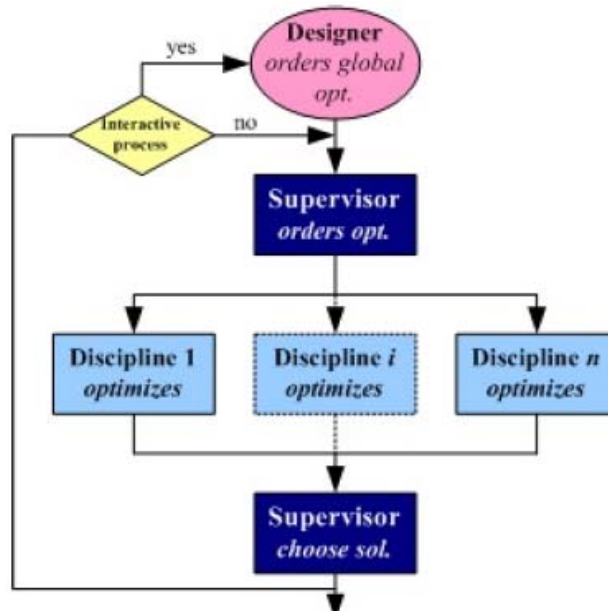


Figura 2.17. Fluxograma do método COSMOS.



### 2.1.14.1 A comparação entre estratégias de otimização multidisciplinares

A correta resolução de um problema de otimização multidisciplinar não depende somente da eficiência do algoritmo, mas também da efetividade da formulação da arquitetura. Portanto, é necessário levar ambos os aspectos em consideração quando se comparam as estratégias de otimização multidisciplinar. **Behdinin (2005)** propõe um conjunto de parâmetros para avaliar e comparar diferentes estratégias de otimização multidisciplinar. Os parâmetros propostos por Behdinin são:

- Simplicidade – facilidade de implementação e modificação. Métodos simples requerem menos tempo para serem modificados e são mais facilmente adaptados a diferentes problemas. A simplicidade é medida em termos do número total de processos de otimização e variáveis requeridas.
- Transparência – a capacidade de se entender o modelo matemático do qual o método deriva, e estendê-lo a outros problemas.
- Portabilidade – a viabilidade de integrar o método em uma estrutura organizacional. Esta métrica leva em consideração a habilidade de uma arquitetura em aproveitar a divisão do trabalho e a autonomia das disciplinas.
- Eficiência – o custo computacional requerido para obter uma solução multidisciplinar ótima viável. Normalmente é medida com base no número total de avaliações disciplinares.
- Precisão – a capacidade de obter a solução multidisciplinar ótima viável exata dentro dos limites de projeto.

Com base nestes parâmetros, Behdinin comparou as estratégias de otimização multidisciplinar mais utilizadas; os resultados obtidos são mostrados na tabela 2.3.

Tabela 2.3. Comparação entre as diferentes estratégias multidisciplinares.\*

Classificação*	Precisão	Eficiência	Transparência	Simplicidade	Portabilidade
1	MDF	IDF	MDF	MDF	CO
2	IDF	BLISS	IDF	IDF	CSSO
3	BLISS	CSSO	CO	CO	BLISS
4	CO	CO	CSSO	CSSO	IDF
5	CSSO	MDF	BLISS	BLISS	MDF

\* Quanto menor a classificação, melhor é a performance da estratégia segundo Behdinin.

Como pode ser observado na tabela 2.3 os métodos mono-nível são mais precisos (visto que não utilizam aproximações para as funções de acoplamento) porém são os de menor eficiência e portabilidade, pois utilizam uma formulação centralizada.

Por essa razão, os métodos multi-nível, em especial aqueles baseados em Otimização Colaborativa, são os mais adequados ao ambiente de otimização desenvolvido, pois valorizam os conceitos de projeto e processamento distribuído.

## 2.2 Superfícies de resposta

Nesta seção serão apresentados os conceitos fundamentais sobre superfícies de resposta, incluindo as principais abordagens existentes. Serão apresentados também, os conceitos fundamentais associados à utilização de redes neurais artificiais na representação de superfícies de respostas.

### 2.2.1 Principais aplicações de superfícies de resposta

O avanço dos recursos computacionais, bem como a popularização dos algoritmos de otimização, determinaram um novo paradigma para projetos de engenharia. Como mostram **Shyy (2001)** e **Birk (2001)**, o objetivo passou a ser não apenas atender aos requisitos propostos, mas ao mesmo tempo encontrar uma vantagem competitiva que explore o melhor resultado entre vários objetivos avaliados (muitas vezes imersos em disciplinas diferentes), privilegiando uma visão global do sistema.

Projetos de grandes sistemas multidisciplinares, como as plataformas de petróleo e os navios, são objetos ideais para serem abordados utilizando técnicas de otimização multidisciplinar. Como discutido em **Giassi (2004)**, a colaboração entre empresas (ou centros de pesquisa) tem sido uma das características mais marcantes na nova tendência mundial.

Os engenheiros inseridos nesse contexto de projeto têm encontrado dois problemas para desenvolver seus trabalhos. Por um lado, projetos de engenharia, especialmente no contexto da indústria naval, envolvem a utilização de parâmetros que são tabelados ou descritos na forma de gráficos e diagramas fornecidos por regras de sociedades classificadoras ou estudos específicos sobre particularidades do assunto. A utilização de tais gráficos e diagramas inviabiliza a automatização do processo e, portanto, a otimização, tornando a análise um processo lento e manual.

Classicamente, existem duas abordagens para automatizar o processo. A primeira consiste em tabelar uma quantidade significativa de pontos e, ao se solicitar um dado, utilizar uma interpolação linear entre os dados próximos. Esta abordagem, embora demande grande trabalho inicial, funciona com relativa precisão para funções e superfícies regulares com até duas variáveis.

Por outro lado, pode-se tentar encontrar, por meio de regressão ou outra abordagem estatística, uma descrição matemática para a função ou superfície utilizada. Essa abordagem tem o claro inconveniente de que muitos diagramas e gráficos não são facilmente representados analiticamente. **Nielsen (2002)** segue nesta linha propondo um complexo algoritmo de interpolação de superfícies utilizando funções de base radial.

O outro problema encontrado é o fato de que classicamente, como descrito por **Augusto (1996)**, sistemas complexos de engenharia consomem longo tempo de trabalho e de recursos computacionais para a análise de uma única configuração, o que inviabiliza processos de otimização que demandem grande número de iterações, de modo que se adota, muitas vezes, a primeira solução a satisfazer as restrições do projeto, como a solução de projeto.

A utilização de superfícies de respostas é um recurso conhecido para a redução do tempo gasto para a análise de ciclos sucessivos de otimização. **Zeboudi (2005)**, **Edwards (1997)**, **Anjum (1997)**, **Agatonovic-Kustrin (1998)**, por exemplo, demonstram que o uso de superfícies de resposta para aumentar a eficiência do processo de otimização é bem difundido, sendo empregado nos mais diferentes tipos de problema.

### **2.2.2    *Abordagens para a utilização de superfícies de resposta***

As superfícies de resposta podem ser representadas por redes neurais artificiais, como visto em **Takahara (1997)** ou por uma abordagem analítica, como proposto por **Shuaeib (2007)**. Classicamente existem duas abordagens para o uso dessas superfícies em processos de otimização.

A primeira, e mais popular, utiliza superfícies de resposta estáticas, construídas a partir de um conjunto inicial de avaliações. Tal abordagem, no entanto, normalmente falha ao utilizar uma superfície de resposta inicial que ao longo dos ciclos de projeto se mostra distante da superfície real do problema associado, o que conduz o processo de otimização por caminhos equivocados e, muitas vezes, acaba por eliminar a vantagem teórica de se utilizar uma superfície de resposta.

**Takahara (1997)** propõe duas alternativas para viabilizar o uso de superfícies de resposta estáticas. A primeira alternativa consiste em utilizar um grande número de avaliações iniciais para assegurar a correta convergência do problema, enquanto a segunda alternativa consiste em realizar parte das avaliações sobre a superfície de resposta e parte das avaliações sem a superfície de resposta.

Ambas as alternativas, no entanto, possuem a mesma fraqueza: um alto número de avaliações do problema, reduzindo a eficiência do processo para problemas nos quais cada análise requer um alto tempo computacional.

O recente trabalho de **Zarei (2006)** propõe uma segunda abordagem para a utilização de superfícies de resposta, construir a cada ciclo de otimização uma nova superfície de resposta incluindo, além dos pontos existentes no histórico, as soluções encontradas a cada ciclo. O trabalho desenvolvido por Zarei, no entanto, não inclui a representação das superfícies de resposta por meio de redes neurais e, portanto, a construção de cada nova superfície ignora as superfícies preexistentes, desprezando a memória do processo.

### 2.2.3 Conceitos fundamentais sobre redes neurais artificiais

Redes neurais artificiais<sup>11</sup> são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

O sistema nervoso, essencial para o raciocínio, é constituído por um conjunto extremamente complexo de células, os neurônios. Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central, e pelos axônios que são longos terminais de saída, como visto na figura 2.18.

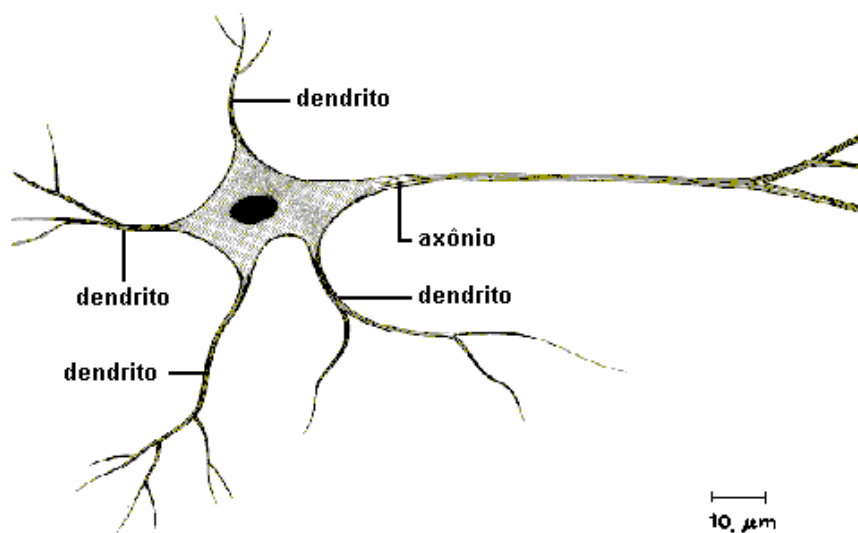

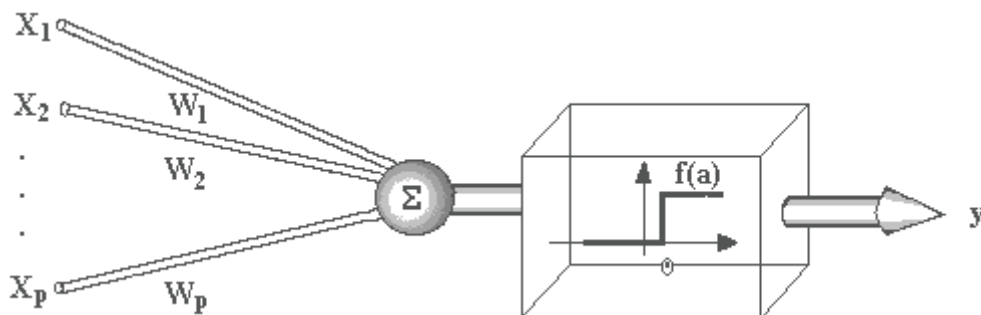


Figura 2.18. Representação de um neurônio.

<sup>11</sup>  O termo “artificiais” visa diferenciar as técnicas computacionais que simulam os processos neurais, das recentes pesquisas que utilizam estruturas neurais biológicas. Porém, em alguns casos, para simplificar o texto, o termo será omitido.

Os neurônios se comunicam através de sinapses. Sinapse é a região onde dois neurônios entram em contato e através da qual os impulsos nervosos são transmitidos entre eles. Em média, cada neurônio forma entre mil e dez mil sinapses. O cérebro humano possui cerca de  $10^{11}$  neurônios, e o número de sinapses é de mais de  $10^{14}$ , possibilitando a formação de redes muito complexa.

As primeiras pesquisas sobre o assunto foram publicadas por **McCulloch (1943)**, que propôs um modelo de unidade de processamento simples para simular o funcionamento de um neurônio, representado na figura 2.19.




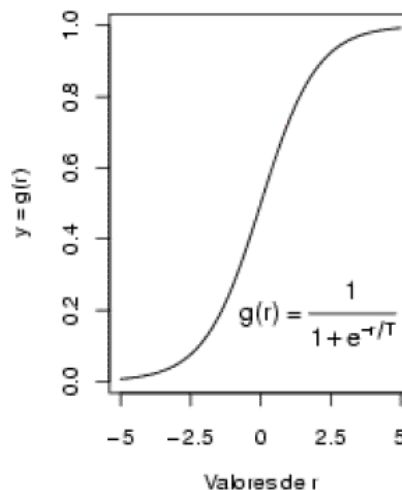
**Figura 2.19.** Modelagem de um neurônio proposta por McCulloch (1943).

O funcionamento da unidade de processamento, proposta por McCulloch, pode ser resumida da seguinte maneira:

- Sinais  $x_i$  são apresentados à entrada;
- Cada sinal é multiplicado por um número (peso)  $w_i$  que indica a sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de excitação;
- Se este nível de excitação exceder um certo limite (*threshold*)<sup>12</sup> a unidade produz uma determinada resposta de saída  $y$ .

As respostas geradas pelas unidades são calculadas através de uma função de ativação. Como visto em **Murray (1995)**, existem vários tipos de funções de ativação, as mais comuns são: *Hard Limiter*, *Threshold Logic* e *Sigmoid*. Para grande parte das aplicações, a função *Sigmoid*, apresentada na figura 2.20, obtém bons resultados. O parâmetro  $T$  indica a taxa de inclinação da função.

<sup>12</sup>  Embora o termo em inglês seja muito utilizado por autores brasileiros, pode ser traduzido como “limite de ativação”.



**Figura 2.20.** Função de ativação do tipo *Sigmoid*.

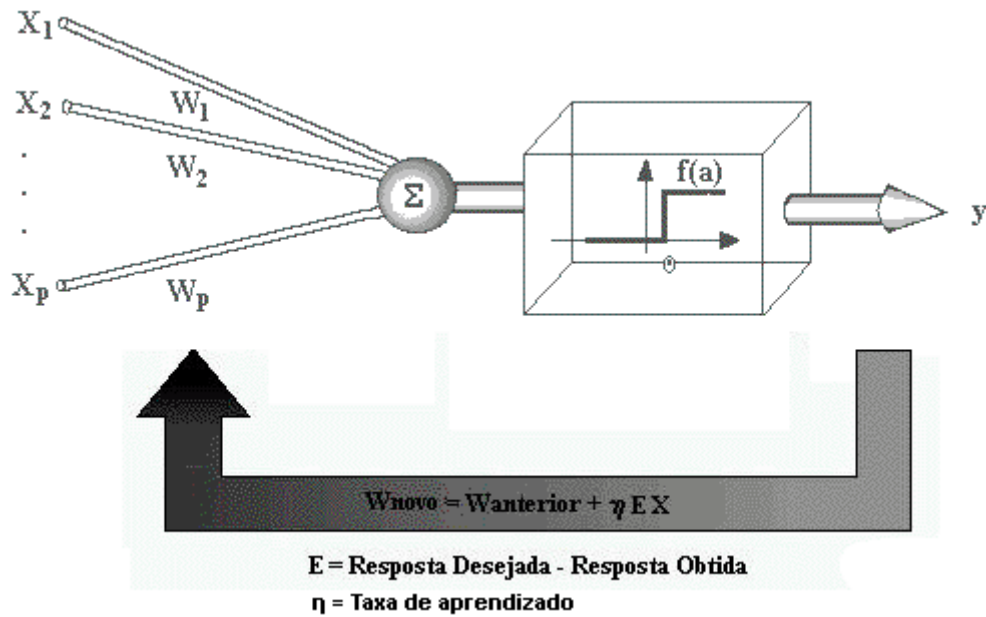
Embora o sistema nervoso humano possua milhões de neurônios, uma rede neural artificial é composta por algumas unidades de processamento, conectadas por canais de comunicação, aos quais estão associados diferentes valores de pesos.

Cada unidade de processamento realiza operações apenas sobre seus dados locais (as entradas recebidas por suas conexões). O comportamento inteligente de uma rede neural artificial vem das interações entre as diferentes unidades de processamento da rede. Este modelo foi proposto pela primeira vez por **Hebb (1949)**, e até hoje é uma das referências mais importantes sobre o assunto.

A propriedade mais importante das redes neurais é a habilidade de aprender a partir de experimentos e, dessa maneira, melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes dos pesos e dos limites de ativação denominado treinamento.

Existem muitos tipos de algoritmos de aprendizado, específicos para determinados modelos de redes neurais, que diferem entre si principalmente pelo modo como os pesos são modificados.

Uma das contribuições mais importantes sobre o aprendizado de redes neurais foi proposta por **Roseblatt (1958)**. O modelo *Perceptron* de aprendizado supervisionado, proposto por Roseblatt e apresentado na figura 2.21, consiste em, para cada experimento (conjunto de dados de entrada e saída), modificar os pesos de cada conexão de maneira proporcional ao erro apresentado pela rede.



**Figura 2.21.** Modelo *Perceptron* de aprendizado supervisionado. Rosembblatt (1958).

A taxa de aprendizado  $\eta$  é uma constante positiva, que corresponde à velocidade do aprendizado. Idealmente, o processo de treinamento requer que sejam tomados passos infinitesimais. Assim quanto maior for essa constante, maior será a mudança nos pesos, aumentando a velocidade do aprendizado, o que pode levar a uma oscilação do modelo na superfície de erro em torno de um ponto de mínimo. O ideal seria utilizar a maior taxa de aprendizado possível que não levasse à uma oscilação, resultando em um aprendizado mais rápido.

Durante o processo de aprendizado, denomina-se ciclo de treinamento uma apresentação de todos os  $N$  pares (entrada e saída) do conjunto de treinamento. A correção dos pesos em um ciclo de treinamento pode ser executado de dois modos:

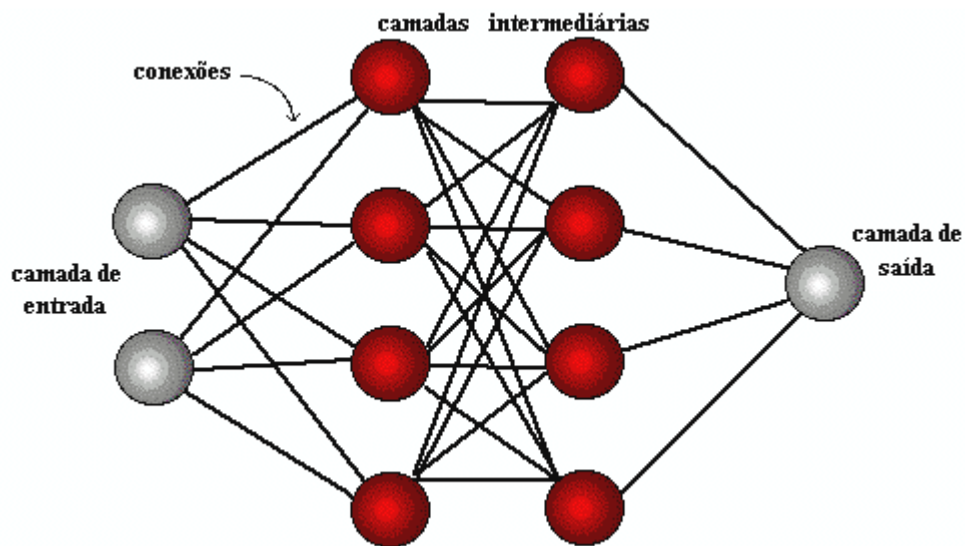
- **Modo Padrão:** A correção dos pesos acontece a cada par do conjunto de treinamento apresentado à rede. Cada correção de pesos baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, em cada ciclo ocorrem  $N$  correções.
- **Modo *Batch*:** Apenas uma correção é feita por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede, seu erro médio é calculado e a partir deste erro fazem-se as correções dos pesos.

Quando redes neurais artificiais de uma só camada são utilizadas, não é possível a formação de uma representação interna. Tal restrição implica que padrões de entrada similares resultem em padrões de saída similares, o que leva o sistema à incapacidade de

aprender importantes mapeamentos, nos quais entradas similares levem a saídas diferentes.

Esse fato limitou a aplicação das redes neurais por muitos anos até a publicação dos trabalhos de **Hopfield (1982)**, relatando a utilização de redes simétricas para otimização, e de **Rumelhart (1986)**, que introduziu o poderoso método *back-propagation* para o treinamento de redes neurais com camadas intermediárias.<sup>13</sup>

O desenvolvimento do algoritmo de treinamento *back-propagation* resultou no modelo de redes neurais artificiais mais utilizado atualmente, as redes *Multilayer Perceptrons* (MLP), treinadas com o algoritmo *back-propagation*, cujo exemplo pode ser visto na figura 2.22.



**Figura 2.22.** Rede neural artificial do tipo *Multilayer Perceptrons*.

Conforme visto em **Silva (2000)**, cada camada de uma rede *Multilayer Perceptrons* tem uma função específica.

*“Na camada de entrada os padrões são apresentados à rede. A camada de saída recebe os estímulos da camada intermediária e constrói o padrão que será a resposta. As camadas intermediárias funcionam como extratoras de características, seus pesos são uma codificação de características apresentadas nos padrões de entrada e permitem que a rede crie sua própria representação, mais rica e complexa, do problema.”*

<sup>13</sup> Embora a autoria do algoritmo de treinamento *back-propagation* (ou treinamento com retropropagação) seja creditado ao trabalho de Rumelhart (1986), tentativas similares foram registradas em anos anteriores.



O treinamento supervisionado da rede MLP utilizando *back-propagation* consiste em apresentar um padrão às unidades da camada de entrada e, a partir desta camada, calcular a resposta produzida na camada de saída. O erro obtido entre a resposta calculada e a resposta original (do padrão apresentado) é propagado, a partir da camada de saída, até a camada de entrada, modificando-se os pesos das conexões por meio da regra delta generalizada. Dessa maneira, o erro vai sendo progressivamente diminuído a cada padrão apresentado.<sup>14</sup>

Como provou **Cybenko (1989)**, a partir de extensões do Teorema de Kolmogoroff, são necessárias no máximo duas camadas intermediárias, com um número suficiente de unidades por camada, para se produzir quaisquer mapeamentos. Também foi provado que apenas uma camada intermediária é suficiente para aproximar qualquer função contínua.

#### **2.2.4 O uso de redes neurais na representação de superfícies de resposta**


O conceito de redes neurais artificiais vem se popularizando como uma ferramenta interessante na representação de padrões e predição de resultados para problemas de difícil formulação, mas de histórico conhecido.

Outra característica importante das redes neurais é a capacidade destas em se adaptarem a cada novo evento, incorporando novas experiências e preservando a memória do processo.

A utilização desse potencial para representar superfícies de respostas capazes de prever resultados de forma rápida, sem necessidade de se recorrer a métodos robustos que dependam de malhas e consumam excessivo tempo de processamento é assunto em pleno desenvolvimento nos meios acadêmicos, como pode ser visto nos trabalhos de **Takahara (1997)**, **Baş (2007)** e **Zhu (1998)**.

Essas superfícies de respostas seriam calibradas por um pequeno histórico construído por análises clássicas feitas com *softwares* de análise numérica, como os baseados no método dos elementos finitos, e recalibradas a cada ciclo de otimização.

---

<sup>14</sup>  A regra delta generalizada, como o próprio nome indica, é uma generalização do treinamento supervisionado de Roseblatt, apresentado na figura 2.21, para a correção dos pesos e limites de ativação em redes neurais com múltiplas camadas.

## 2.3 Processamento paralelo e distribuído

Nesta seção serão apresentados os principais conceitos do processamento paralelo, além de uma revisão bibliográfica sobre o assunto. Serão apresentados os diferentes métodos de processamento paralelo, a medida de qualidade de um algoritmo paralelo, o protocolo MPI para programação distribuída e o conceito de escalonamento.

### 2.3.1 Breve histórico sobre processamento paralelo e distribuído

Entre os assuntos que fervilham na comunidade científica, (**Andrews (1999)** e **Duncan (1990)**, por exemplo), certamente está o desenvolvimento do paralelismo em longos algoritmos computacionais que, mais velozes, permitem aos engenheiros explorarem um número maior de possibilidades; potenciando a descoberta de arranjos ousados nunca antes imaginados.

O aumento da quantidade de dados a serem processados, em conjunto com a necessidade de obter resultados em prazos cada vez menores, provocou aumento significativo na capacidade computacional requerida. Algoritmos seqüenciais mostraram-se computacionalmente ineficazes diante destes novos paradigmas.


Tendo em vista esta ineficácia, a alternativa encontrada foi a programação paralela e distribuída, na qual diversos processadores podem trabalhar simultaneamente em um mesmo problema, tornando sua implementação viável.

Conforme relatado em **Almasi (1994)**, o aumento de desempenho é a maior razão para o uso do processamento paralelo, tendo em vista que esse tipo de processamento pode reduzir consideravelmente o tempo de resolução de um problema.

A história da programação paralela seguiu os mesmos estágios de outras áreas experimentais da Ciência da Computação. Segundo **Andrews (1999)**, a programação paralela se originou nos anos 60, sendo motivada pela invenção de unidades de *hardware* denominadas controladoras de dispositivos.<sup>15</sup>

Pouco após a invenção das controladoras de dispositivo, engenheiros de *hardware* criaram o projeto de máquinas multiprocessadas. Porém, como apontado por **Quinn (1994)** foi necessária uma espera de mais de vinte anos para que os computadores paralelos

---

<sup>15</sup>  Sem se aprofundar na arquitetura de computadores, pode-se dizer de forma simplificada que estes dispositivos permitem que as operações de entrada e saída sejam realizadas simultaneamente com as instruções executadas pelo processador central.

fizessem a transição dos laboratórios para o mercado de trabalho. **Andrews (1999)** justifica esta demora pelo alto preço dos computadores multiprocessados até então.

Em meados dos anos 80, deu-se o advento dos computadores paralelos microprocessados comerciais. Atualmente todos os computadores de grande porte são multiprocessados, sendo que os maiores possuem centenas de processadores e são chamados de MPP's (*Massively Parallel Processors* ou Processadores Paralelos Massivos).

Além do aumento de desempenho, descrito por **Almasi (1994)**, outra importante razão que faz jus ao uso da programação paralela e distribuída é o fato de que, como apontado por **Monte-Mor (2002)**, alguns problemas são de natureza paralela, e sua implementação seqüencial não produz resultados satisfatórios.

Como se verá adiante, a otimização de problemas multidisciplinares utilizando superfícies de resposta pode, dependendo do algoritmo de otimização utilizado, ser um exemplo de problema de natureza paralela.

### 2.3.2 A computação de grande porte

Segundo **Brasil (2006)**, a computação de grande porte é definida como um sistema de alta capacidade de computação, também conhecida como "Alta Plataforma", e é caracterizada pela utilização de *mainframes* e supercomputadores.<sup>16</sup>

Trata-se de computadores que geralmente ocupam grande espaço físico e necessitam de ambiente especial para funcionarem. Os *mainframes* nasceram em 1946 e foram constantemente aperfeiçoados, sendo o ano de 1964 considerado como um dos marcos mais importantes em sua evolução, visto que, nesse ano, a IBM apresentou o "System/360", *mainframe* que, na época, representou um salto considerável de performance em relação a seus predecessores.

A década de 60 também marcou o surgimento dos primeiros supercomputadores criados por Seymour Cray que dominou o mercado da supercomputação durante 25 anos (1965-1990).

---

<sup>16</sup> A distinção entre supercomputadores e *mainframes* não é clara, nem direta, mas, como visto em Brasil (2006), eles podem ser genericamente diferenciados pelas tarefas a que são submetidos: os supercomputadores são utilizados na solução de problemas em que o tempo de cálculo é um fator preponderante, enquanto os *mainframes* são utilizados em tarefas que exigem alta disponibilidade envolvendo alta taxa de transferência de dados e acessos simultâneos. Dentro desta definição, os supercomputadores são equipamentos mais adequados para a resolução dos problemas de otimização.

Além dos altos custos de implantação, os supercomputadores apresentam o problema de que quando o limite de processamento do equipamento é alcançado, torna-se necessário realizar um *upgrade*, geralmente caracterizado pela substituição do equipamento original por um novo, ou pela incorporação de novo *hardware* ao equipamento original. Qualquer uma das alternativas provocará elevado custo financeiro.

Para contornar esse problema foram desenvolvidos os *clusters*, um tipo de supercomputador criado a partir da cooperação de vários computadores convencionais. Por utilizar componentes físicos comuns em sua arquitetura, um ambiente *cluster* apresenta facilidade de dimensionamento da capacidade de processamento, pois à medida que a carga aumenta, novos componentes físicos podem ser facilmente alocados para suprir a necessidade de processamento.

Os *clusters*, entretanto, ainda exigiam que os componentes físicos estivessem alocados em um mesmo ambiente. Em meados dos anos 90, surgiram os *Grids* computacionais com a promessa de viabilizar a execução de aplicações paralelas em recursos geograficamente dispersos e pertencentes a múltiplas organizações. Tal proposta tinha o objetivo de fornecer uma plataforma muito mais barata para execução de aplicações distribuídas (em comparação com os supercomputadores paralelos).<sup>17</sup>

Uma comparação das vantagens e desvantagens entre os ambientes de computação de grande porte é apresentada na tabela 2.4.

Tabela 2.4. Comparação entre as soluções para a computação de grande porte.\*

<i>Mainframes</i>	<i>Clusters e Grids</i>
- Alto custo de implementação	- Baixo custo de implementação
- Dependência de fornecedor único	- Independência de fornecedores; facilidade de negociação
- Utilização de <i>hardware</i> específico	- Utilização de <i>hardware</i> comum, padrão PC
- Alto custo de manutenção	- Baixo custo de manutenção
- Dificuldade de redimensionamento do ambiente	- Facilidade de redimensionamento do ambiente
- Utilização parcial da capacidade de processamento	- Maximização da capacidade de processamento
- Grande custo total de propriedade	- Baixo custo total de propriedade
- Tecnologia estabelecida no mercado	- Tecnologia inovadora

\* Adaptado de Brasil (2006).

Apesar de recentes, a evolução dos *Clusters* e *Grids* criou uma nova alternativa para os ambientes computacionais de grande porte. A utilização desses ambientes para

<sup>17</sup> Brasil (2006) destaca ainda o enorme potencial que os *Grids* possuem como plataforma de serviços sob demanda utilizando a Internet como meio de conexão. No entanto, o serviço de execução remota amplia a preocupação com questões como segurança e escalonamento.

computação de alta performance vem crescendo rapidamente e já é quase predominante para as grandes máquinas utilizadas nos dias de hoje.

Para esclarecer um pouco mais a diferença entre os *Grids* e outras plataformas de computação de grande porte, podemos citar algumas características intrínsecas aos *Grids*:

- Heterogeneidade: Os componentes que formam a infra-estrutura tendem a ser extremamente heterogêneos. Ou seja, é importante ter em mente que qualquer solução para *Grids* computacionais deverá lidar com recursos de várias gerações, *softwares* de várias versões e sistemas operacionais diversos.
- Alta dispersão geográfica: Essa característica se refere à escala que um *Grid* pode atingir. Nesse sentido, *Grids* podem ter escala global, agregando serviços localizados em várias partes do planeta.
- Múltiplos domínios administrativos: *Grids* reúnem recursos de várias instituições. Sendo assim, além da heterogeneidade mencionada anteriormente, é possível também a existência de várias políticas de acesso e uso dos serviços.

Os aspectos apresentados nesta seção justificam a escolha do desenvolvimento de um ambiente para a otimização de complexos projetos de engenharia baseados no paradigma do *Grid* computacional.


### **2.3.3 Principais conceitos do processamento paralelo e distribuído**

Nesta seção serão apresentados os principais conceitos sobre a utilização de processamento paralelo e distribuído em processos de otimização, que como demonstram os trabalhos de **Junior (2002)**, **Andrews (1999)**, **Ziviani (1993)** e **Almasi (1994)**, representa uma solução eficiente para aumentar a velocidade desse tipo de processo.

#### **2.3.3.1 O paralelismo real e o paralelismo lógico**

A maioria dos computadores atuais de grande desempenho permite realizar tarefas concorrentes.<sup>18</sup> Porém, segundo **Quinn (1994)**, não se deve chamar todos os computadores modernos de computadores paralelos. O tratamento de concorrência em muitas máquinas é simulado e transparente ao usuário. Portanto, a seguinte definição é utilizada:

---

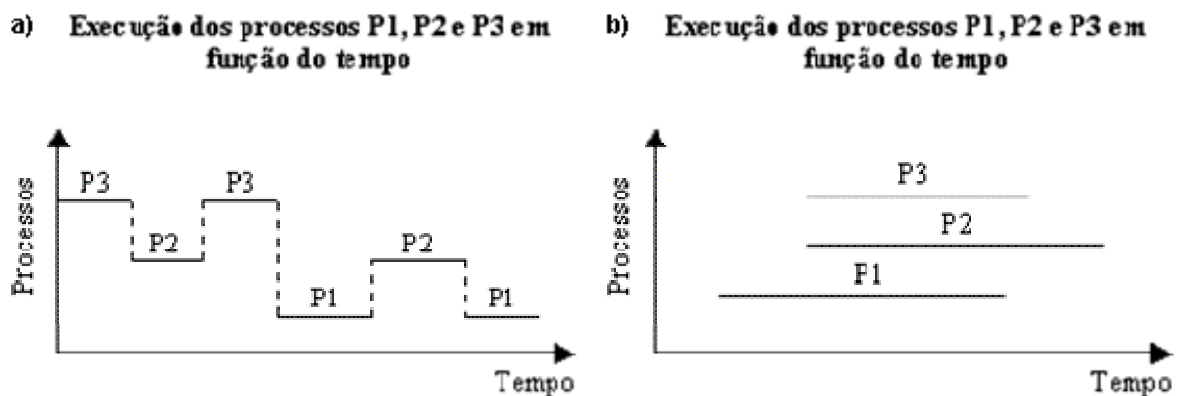
<sup>18</sup>  Embora o dicionário não registre “concorrente” como sinônimo de simultâneo, Quinn utiliza esse termo em seu livro, quando o correto seria “concomitante”.

*“Processamento Paralelo é o processamento de informação que enfatiza a manipulação concorrente de dados relativos a um ou mais processos pertencentes a um único programa”.* **Quinn (1994)**<sup>19</sup>

Uma arquitetura paralela fornece uma estrutura de alto nível para o desenvolvimento de soluções através do processamento paralelo, utilizando múltiplos processadores, simples ou complexos, que cooperam para resolver problemas por meio da execução concorrente.

Em computadores sequenciais existe o que é chamado de concorrência lógica, ou seja, o tempo de processamento é compartilhado entre os processos concorrentes, de modo que apenas um processo é executado por vez. O usuário, por sua vez, tem a impressão de um processamento paralelo, visto que o escalonamento dos processos não é visível, como mostrado na figura 2.23a.

Em computadores paralelos, todavia, existe uma concorrência real entre os processos, pois cada um dos processadores executa um processo distinto. É importante salientar que, ao dividir-se as tarefas entre os diferentes processadores, obtém-se uma redução no tempo de processamento, e a divisão dessas tarefas entre seus respectivos processadores fica a cargo do programador. A execução de processos com o paralelismo real é mostrada na figura 2.23b.



**Figura 2.23.** Diferença entre o paralelismo lógico e o paralelismo real. Junior (2002).

### 2.3.3.2 O modelo PRAM de processamento paralelo

Conforme visto em **Quinn (1994)**, o modelo PRAM (*Parallel Random Access Machine*) permite aos programadores tratar o poder computacional como um recurso

<sup>19</sup> Deve-se aqui, explicitar a diferença entre processo e programa: Programa pode ser entendido como sendo o código fonte, de natureza estática, enquanto que o processo representa os programas em execução, possuindo natureza dinâmica, sendo que o mesmo programa pode disparar vários processos independentes.

ilimitado. Apesar de ser o mais conhecido modelo de processamento paralelo ele é não é realista, pois ignora a complexidade de comunicação entre os processadores.

*“Como a complexidade de comunicação é ignorada, os criadores de algoritmos paralelos podem focar suas atenções no desenvolvimento do paralelismo inerente à computação”. Quinn (1994).*

Segundo **Fortune (1978)**, o modelo PRAM consiste de uma unidade de controle, uma memória global e um número ilimitado de processadores, cada qual com sua memória local.

Uma computação, no modelo PRAM, inicia com a entrada armazenada na memória global e apenas um processador ativo. Durante cada ciclo da computação um processador ativo pode ler um valor da memória local ou global, realizar uma operação sobre este valor e escrevê-lo na memória local ou global. Alternativamente, ele pode ativar outro processador. Cada processador ativo deve executar a mesma instrução em diferentes locais da memória. A computação chega ao fim quando o último processador termina a execução.

O modelo PRAM de programação paralela é importante, pois representa a base de qualquer aplicação de programação paralela ou distribuída.

Um dos obstáculos à paralelização de um método qualquer são os problemas denominados conflitos de escrita e leitura que ocorrem quando dois ou mais processadores tentam ler ou escrever na mesma posição de memória simultaneamente. Qualquer aplicação de programação paralela deve, portanto, se preocupar com a transmissão e sincronização dos dados.

### **2.3.3.3 Medidas de desempenho para algoritmos paralelos**

Em se tratando de algoritmos paralelos, as medidas de qualidade ficam a cargo de dois parâmetros: *speedup* e eficiência. O ganho de desempenho obtido com a paralelização do algoritmo é chamado de *speedup*, que é a razão entre o tempo de execução do algoritmo sequencial mais eficiente e o tempo de execução do algoritmo paralelo correspondente utilizando  $p$  processadores.<sup>20</sup>

---

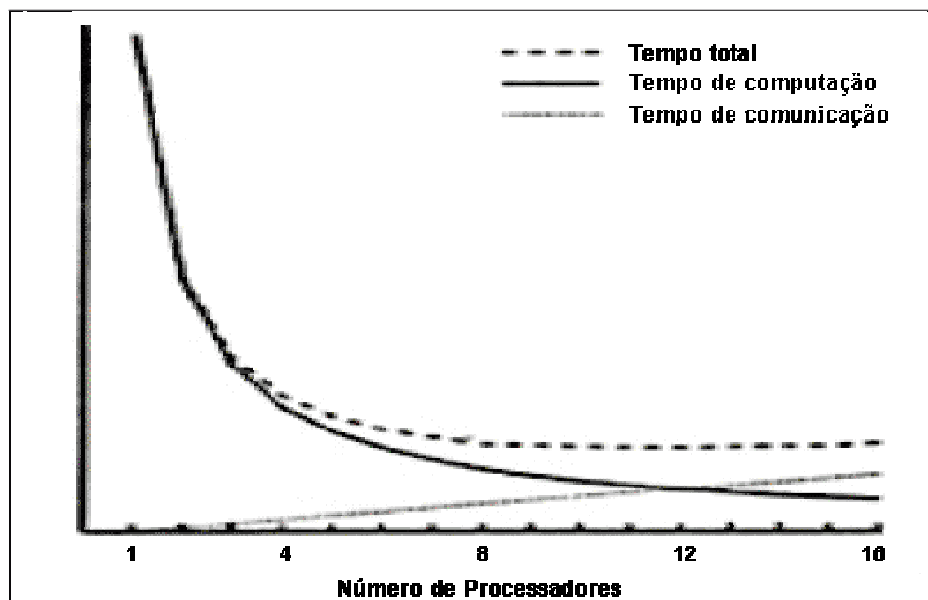
<sup>20</sup> É importante ressaltar que nunca se deve comparar o tempo de execução do algoritmo paralelo em um ambiente sequencial com o tempo de execução do mesmo algoritmo em um ambiente paralelo, pois neste caso os custos de paralelização e sincronização estão sendo incorretamente incorporados ao problema sequencial. A comparação deve ser feita sempre entre algoritmos específicos para os ambientes sequencial e paralelo.

Embora teoricamente o *speedup* ótimo seja diretamente proporcional ao número de processadores utilizados, na prática esta hipótese provou-se errada, visto que o *speedup* não se mostrou proporcional ao número de processadores. Apesar disso, um *speedup* acima do linear é possível em alguns problemas particulares.<sup>21</sup>

Por outro lado, **Quinn (1994)** explica que a queda no *speedup* ao se aumentar muito o número de processadores pode ser explicada pela sobrecarga (*overhead*) de tempo de comunicação entre os processadores – fato ignorado pelo modelo PRAM – e a existência de tarefas inerentemente sequenciais, mesmo em algoritmos paralelos.

Com o aumento do número de processadores o tempo de computação torna-se cada vez menor, ao passo que o excesso de tempo de comunicação pode tornar o processamento final mais lento. O tempo de computação é inversamente proporcional ao número de processadores utilizados, enquanto que o tempo de comunicação entre os processadores cresce linearmente com o aumento do número dos processadores.

Conclui-se que, para cada problema, existe um número de processadores que maximiza o *speedup*, a partir do qual o aumento do tempo de comunicação é maior que a diminuição do tempo de computação, e o tempo total começa a aumentar. Na figura 2.24 é mostrado um gráfico dessa relação.



**Figura 2.24.** Tempo de comunicação x Tempo de computação. Quinn (1994).

<sup>21</sup> Um exemplo clássico é o da busca de um elemento em um vetor de tamanho  $n$ . Enquanto o algoritmo sequencial, a cada elemento analisado, tem  $1/n$  chances de encontrar o elemento e encerrar a busca, o algoritmo paralelo com  $k$  processadores tem  $k/n$  chances e, portanto, estatisticamente pode apresentar *speedup* acima do valor linear de  $k$  vezes.



O conceito de máximo *speedup* foi proposto originalmente por **Amdahl (1967)** e ficou conhecido como lei de Amdahl.

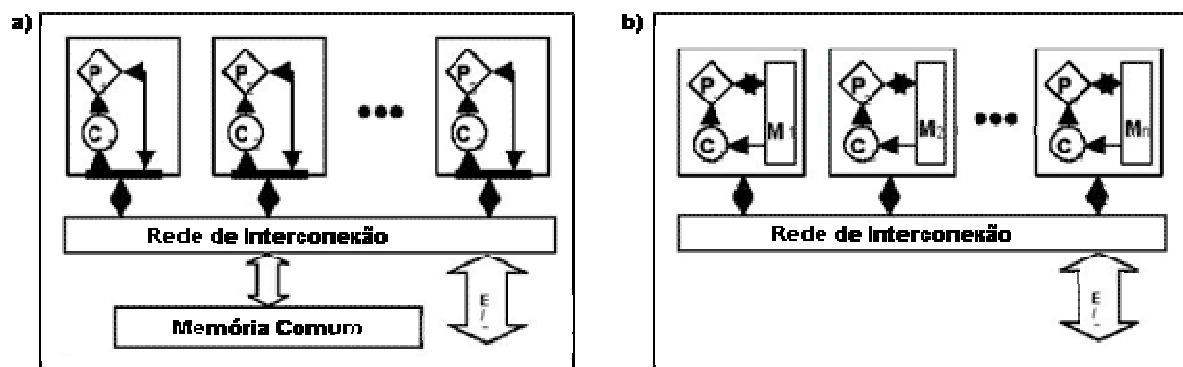
Pode-se definir a eficiência de um algoritmo paralelo executado em uma máquina com  $p$  processadores como sendo a razão entre o *speedup* alcançado e o número de processadores utilizados.

Como se verá, a otimização multidisciplinar sobre neuro superfícies de resposta oferece grande potencial para *speedup*, sendo possível determinar o número de computadores que maximiza o *speedup* do processo.

#### 2.3.3.4 A programação paralela com memória compartilhada e não compartilhada

Desde o surgimento da programação paralela e distribuída, diversos mecanismos foram desenvolvidos para realizar a comunicação entre os processadores responsáveis pela execução do programa paralelo.

Estes mecanismos podem ser separados em dois grupos: os que atuam sobre memória compartilhada e os que atuam sobre memória não compartilhada. Na arquitetura de memória compartilhada (figura 2.25a) os processadores estão conectados por uma rede de comunicação, e os processos dividem a memória principal. Por outro lado, na arquitetura de memória distribuída (figura 2.25b) os processadores também estão conectados por uma rede de comunicação, possuindo, porém, uma memória própria.



**Figura 2.25.** Programação com memória compartilhada e não compartilhada. Brasil (2006).

De maneira geral, as aplicações que utilizam memória compartilhada são denominadas de programação paralela, ao passo que as aplicações que utilizam memória não compartilhada são denominadas programação distribuída.

No entanto, existem mecanismos para emular uma memória compartilhada mesmo em arquiteturas não compartilhadas. Portanto, na prática, uma aplicação paralela é

caracterizada por ser iniciada em um único computador, a partir do qual são iniciados os demais processos paralelos. Já na aplicação distribuída é necessário iniciar um processo em cada computador do ambiente, que são sincronizados por meio da passagem de mensagem.

Entre os mecanismos que atuam sobre memória não compartilhada estão os mecanismos de trocas de mensagens, cujo expoente é o MPI (*Message Passing Interface*), amplamente utilizado atualmente.

#### **2.3.3.5 Biblioteca disponível para o processamento distribuído: MPI**

A interface de troca de mensagens MPI é uma biblioteca de rotinas de envio e recebimento de mensagens. Quando o MPI é usado, os processos de um programa distribuído são escritos sequencialmente em *C* ou em *Fortran*. Esses processos se comunicam e são sincronizados por meio de chamadas às funções MPI.

A API (*Application Programmer's Interface*) do MPI foi definida em meados dos anos 90 por um grande conjunto de representantes de faculdades, governos e indústrias [www.lam-mpi.org]. A interface reflete as experiências passadas dos representantes com as bibliotecas de trocas de mensagens anteriores (como o PVM (*Parallel Virtual Machine*)). O objetivo do grupo era desenvolver uma biblioteca única que poderia ser facilmente implementada em uma diversidade de máquinas multiprocessadas. Segundo **Andrews (1999)**, o MPI vem se tornando um padrão cada vez mais usado, e diversas implementações estão disponíveis.

Infelizmente, conforme se verá no capítulo seguinte, a biblioteca MPI está limitada a programação utilizando *C* ou *Fortran*, o que impossibilita seu uso direto no ambiente *Delphi*. No entanto, a aplicação desenvolvida utiliza o paradigma de memória distribuída, onde cada processador possui apenas acesso a sua própria memória, e a sincronização é feita por passagem de mensagem, no caso, pela Internet através do protocolo TCP/IP utilizando os padrões estabelecidos no MPI.

#### **2.3.3.6 O paralelismo de dados e de controle**

Pode-se classificar a programação paralela entre dois paradigmas:

- Paralelismo de dados: representa o uso de múltiplas unidades para se aplicar a mesma operação simultaneamente em um dado conjunto de elementos. Segundo **Quinn (1994)**,  $K$  unidades de processamento adicionais geram um aumento de  $K$  vezes no número de respostas por segundo. A

execução deste tipo de algoritmo pode ser verificada, por exemplo, em algoritmos paralelos de multiplicação de matrizes.

- Paralelismo de controle: diferentemente do paralelismo de dados em que o paralelismo é atingido por meio de diversas unidades de processamento executando uma única instrução, atinge-se o paralelismo com a aplicação de diferentes operações a diferentes conjuntos de dados simultaneamente. Conforme **Quinn (1994)**, o fluxo de dados sobre este processo pode ser arbitrariamente complexo. No paralelismo de controle, a computação é dividida em passos, chamados segmentos ou estágios, que são distribuídos entre os processadores. Cada segmento realiza uma parte do processamento e pode ser possível que a entrada de um segmento seja a solução gerada na saída do segmento anterior.

#### **2.3.3.7 Escalabilidade**

*“um algoritmo é escalável se o nível de paralelismo aumenta pelo menos linearmente com o tamanho do problema; uma arquitetura é escalável se o desempenho de cada processador não se altera quando se aumenta o número de processadores”.* **Quinn (1994)**.

Como regra geral, pode-se dizer que tanto a escalabilidade algorítmica, quanto a arquitetural são importantes, pois podem-se resolver problemas maiores por meio da aquisição de um computador paralelo com mais processadores.

Os algoritmos baseados no paralelismo de dados são mais escaláveis que os baseados no paralelismo de controle, visto que, no caso do paralelismo de controle o nível de paralelismo é geralmente uma constante, enquanto no paralelismo de dados, o nível de paralelismo é uma função crescente com o aumento do tamanho do problema.

## 3 Desenvolvimento

---

3.1	Abordagem conceitual	76
3.2	Implementação	78
3.2.1	Programação orientada a objetos	79
3.3	O cliente de otimização	79
3.3.1	Interface	80
3.3.2	Objeto para geração de gráficos	81
3.3.3	Objeto para edição do problema	82
3.3.4	Objeto para resolução vetorial de expressões strings	83
3.3.5	Regiões tabus	85
3.3.6	Algoritmos	86
3.3.7	A otimização em regiões viáveis normalizadas	87
3.3.8	A otimização em regiões viáveis discretas	88
3.3.9	Superfícies de Resposta	89
3.3.10	Módulos de análises	90
3.3.11	Exportação de Dados	91
3.4	O ambiente de programação distribuída desenvolvido	92
3.4.1	Requisitos do sistema	93
3.4.2	Implementação da programação distribuída	95
3.4.3	Iniciando o ambiente de programação distribuída.	95
3.4.4	A otimização utilizando ambiente distribuído	98

*"The first 90 percent of the code accounts for the first  
90 percent of the development time..."*

*The remaining 10 percent of the code accounts for  
the other 90 percent of the development time."*

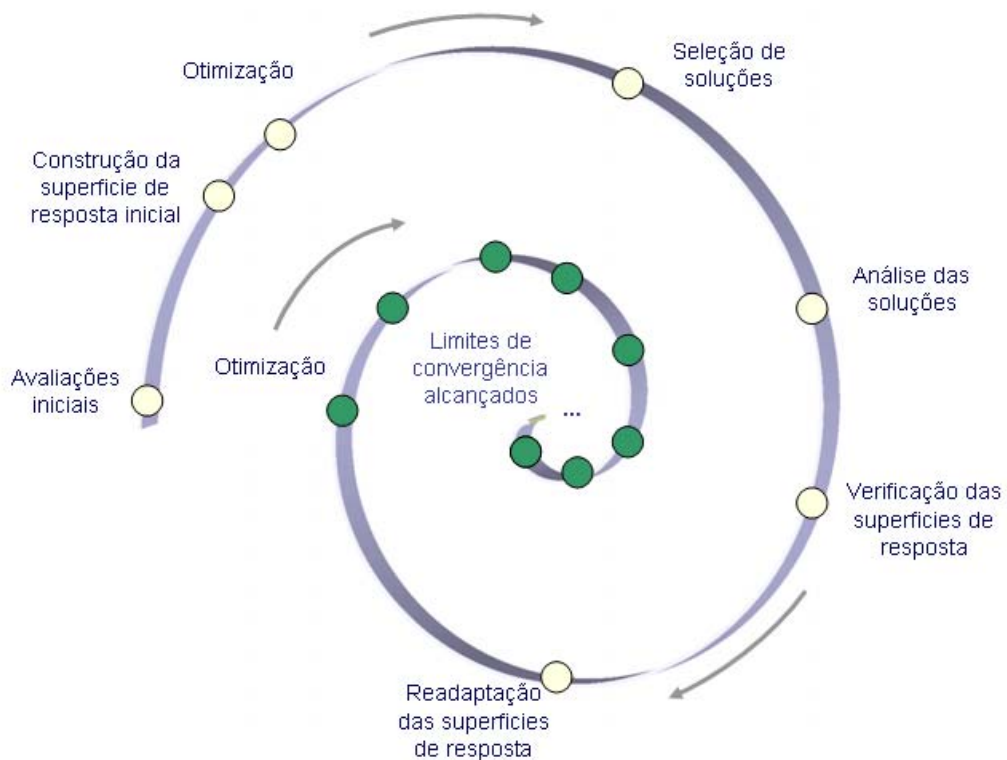
Tom Cargill

Neste capítulo será descrita a fase de implementação do ambiente para a otimização multidisciplinar distribuída de complexos projetos de engenharia. Embora muito do que será descrito possa parecer um problema mais de implementação computacional do que de engenharia, foram absolutamente vitais no desenvolvimento das contribuições descritas no capítulo seguinte.

### 3.1 Abordagem conceitual

O conceito utilizado no ambiente desenvolvido tem como inspiração a metodologia de projeto em espiral, muito empregada na engenharia de sistemas navais e oceânicos.

Esta abordagem consiste em realizar ciclos sucessivos de análise, otimização e readaptação das superfícies de resposta até a convergência do processo, como mostrado na figura 3.1.



**Figura 3.1.** Abordagem em espiral do ambiente desenvolvido.

Antes de iniciar o projeto, o engenheiro deve modelar o problema em questão, determinando as variáveis, restrições e objetivos que constituirão o modelo a ser analisado. Para os problemas que requerem análises utilizando aplicativos computacionais, deve-se ainda parametrizar as macros utilizadas nas análises, conforme exigido por cada aplicativo.

O processo inicia com a avaliação de um conjunto de pontos selecionados para a construção de um histórico inicial, a partir do qual são treinadas as superfícies de resposta exigidas pelo problema.

A seguir, a estratégia de otimização adequada ao problema é utilizada, realizando a maior parte das análises por meio das superfícies de respostas iniciais. O número de soluções encontradas é determinado pelo tipo de problema analisado. Como visto na seção 2.1.7.2, problemas multiobjetivo resultam em um conjunto de soluções, enquanto problemas mono-objetivo resultam em apenas uma única solução.

Com base nas soluções encontradas, verifica-se se as superfícies de resposta não são anômalas ou viciadas. Nestes casos, uma região tabu é criada para redirecionar o próximo ciclo de otimização.

As soluções encontradas são analisadas sem o uso das superfícies de resposta e os resultados obtidos são comparados com aqueles calculados com o uso das superfícies de

respostas. Caso os resultados sejam diferentes, as superfícies de respostas são readaptadas e o processo reinicia utilizando as novas superfícies de resposta.

O processo continua até que as soluções encontradas estejam dentro do limite de convergência e todas as regiões tabus tenham sido removidas. Caso o problema não utilize superfícies de resposta, ou caso o usuário desative-as, apenas 1 ciclo de otimização é necessário.

Cada uma das etapas pode ser paralelizada para acelerar o processo de convergência, conforme visto na seção 2.3.

### 3.2 Implementação

Para o desenvolvimento e testes foi utilizado um *laptop Aspire 3003LCI* equipado com um processador *Sempron 3000+ 512 mb DDR* com o sistema operacional *Windows XP Home Edition*.


Todo o desenvolvimento computacional foi feito utilizando a linguagem *Delphi* de programação. A escolha pela linguagem *Delphi* ocorreu de forma natural, dada a experiência pré-existente e a adequação do ambiente *Delphi* para o rápido desenvolvimento de projetos complexos de programação.<sup>22</sup>

Embora natural, tal escolha está longe de ser uma unanimidade. O debate sobre ambientes de programação representa uma discussão que beira a filosofia ou religião, sendo inadequado até mesmo citar referências sobre o assunto.

“Puritanos” defendem fervorosamente a utilização de linguagens que classicamente são utilizadas na programação de ferramentas numéricas, tais como *Fortran* ou *C*. De outro lado estão os admiradores dos ambientes de desenvolvimento rápido como *Visual Basic* e *Delphi*. No centro dessa discussão estão os apaixonados por *Java* ou tecnologias *.net*.

Não convém a este trabalho entrar nessa discussão, mas apenas apontar que o *Delphi* reúne condições suficientes para a execução deste projeto: programação orientada a objetos e programação estruturada simultâneas, fácil portabilidade para *Linux*, ambiente de desenvolvimento rápido, farta documentação e independência de bibliotecas externas.

---

<sup>22</sup>  Nota pessoal: Durante sua experiência internacional o autor observou que pesquisadores europeus têm a tendência de desenvolver seus projetos utilizando o *MatLab*. A utilização de *MatLab* reflete um desenvolvimento mais acadêmico e fortemente baseado na abordagem matemática do problema, pois reduz parte das dificuldades na implementação computacional, porém em contrapartida, restringe outros aspectos do projeto.

### **3.2.1 Programação orientada a objetos**

Inicialmente o projeto foi desenvolvido utilizando a programação estruturada. O decorrer do trabalho, porém, levou à constatação de que, mesmo utilizando módulos isolados, a complexidade das rotinas impossibilitava essa abordagem.

A partir desse momento praticamente todas as rotinas relevantes foram implementadas na forma de objetos. Embora mais difícil na fase inicial, a implementação da programação orientada a objetos simplificou enormemente a manutenção e o desacoplamento dos diferentes módulos do sistema.

O sistema desenvolvido possui dezenas de objetos que podem ser modificados livremente sem afetar o restante do sistema. Por exemplo: uma modificação no algoritmo de otimização pode ser feita modificando-se unicamente o objeto de otimização, sem afetar outros objetos. Pode-se substituir um método de otimização por outro ou mesmo criar vários processos de otimização simultâneos.

Embora seja um problema voltado ao campo da computação, essa estrutura de programação representa um dos objetivos desta pesquisa, o de desenvolver um ambiente de projeto que possa unir diferentes técnicas existentes e permitir a rápida implementação de novas técnicas ou modificações.

Por outro lado, a programação orientada a objeto requer controle rigoroso do gerenciamento de memória durante a execução do sistema. É comum que programadores inexperientes permitam que objetos não utilizados ocupem espaço na memória. A esta ocorrência é dado o nome de “vazamento de memória”. Um programa com vazamento de memória costuma ocupar mais memória a cada iteração do processo, o que pode se tornar extremamente crítico em algoritmos de otimização. O sistema desenvolvido passou por rigoroso controle para eliminar os vazamentos de memória e permitir que o programa não sofresse queda de desempenho em função da quantidade de memória alocada ao longo do processo.

## **3.3 O cliente de otimização**

Nessa seção será descrito de forma detalhada o principal módulo do ambiente desenvolvido, o cliente de otimização.

### 3.3.1 Interface

A interface visa promover a interação entre o usuário (ou usuários) e a ferramenta computacional propriamente dita. A interface desenvolvida para o cliente de otimização segue o modelo clássico dos programas para *Windows*, com 3 linhas horizontais de controles:

- Na primeira linha está o menu de opções. Todas as configurações e comandos do problema são realizados por meio desse menu.
- A segunda linha é composta por 6 campos gráficos e 3 campos de texto que apresentam todos os dados durante o processo de otimização. Em problemas multidisciplinares pode-se trocar a disciplina em exibição por meio das abas localizadas sobre os campos gráficos.
- Por fim, a barra de *status* apresenta informações sobre o estágio do processo, a quantidade de memória utilizada, o *status* da conexão com a Internet e o idioma utilizado no programa. Embora o programa detecte o idioma mais apropriado ao usuário (com base na configuração do *Windows*), pode-se trocar o idioma a qualquer momento.

A interface do módulo de otimização desenvolvido pode ser vista na figura 3.2.

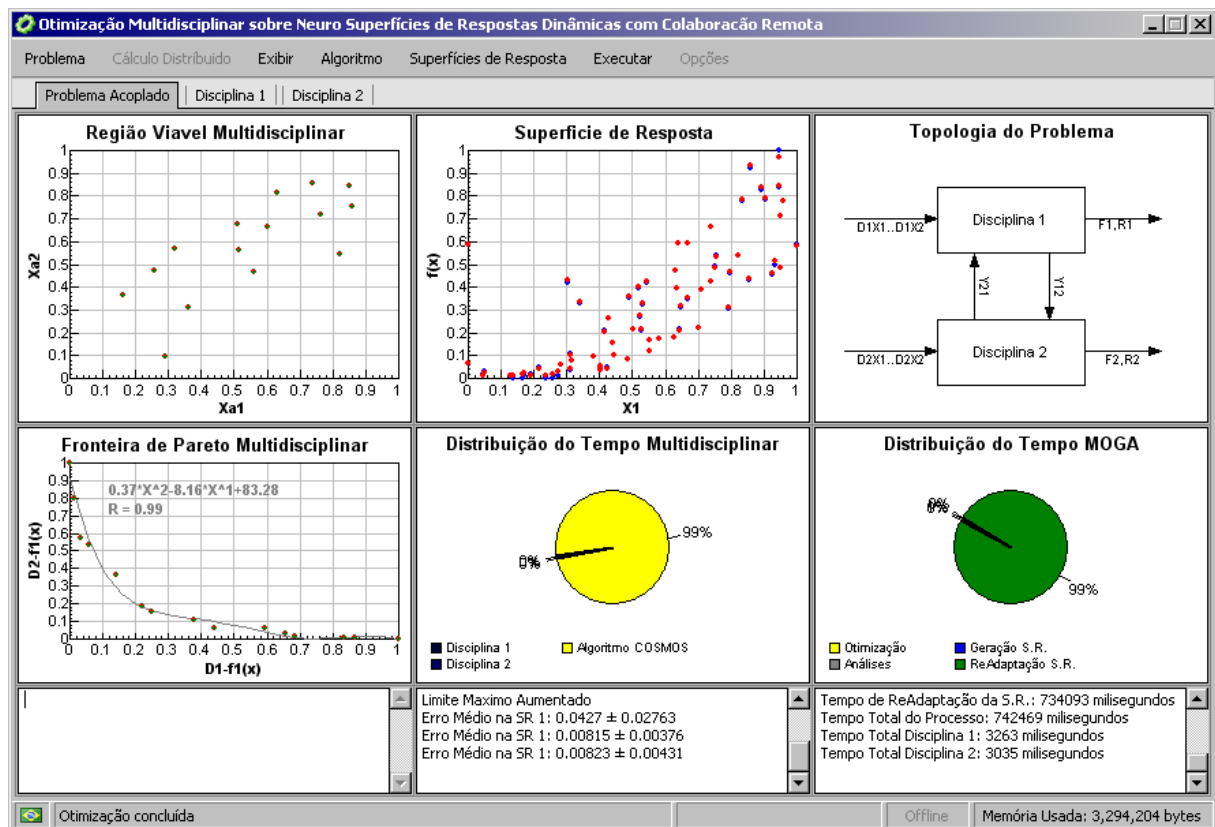


Figura 3.2. Interface multidisciplinar do módulo de otimização.

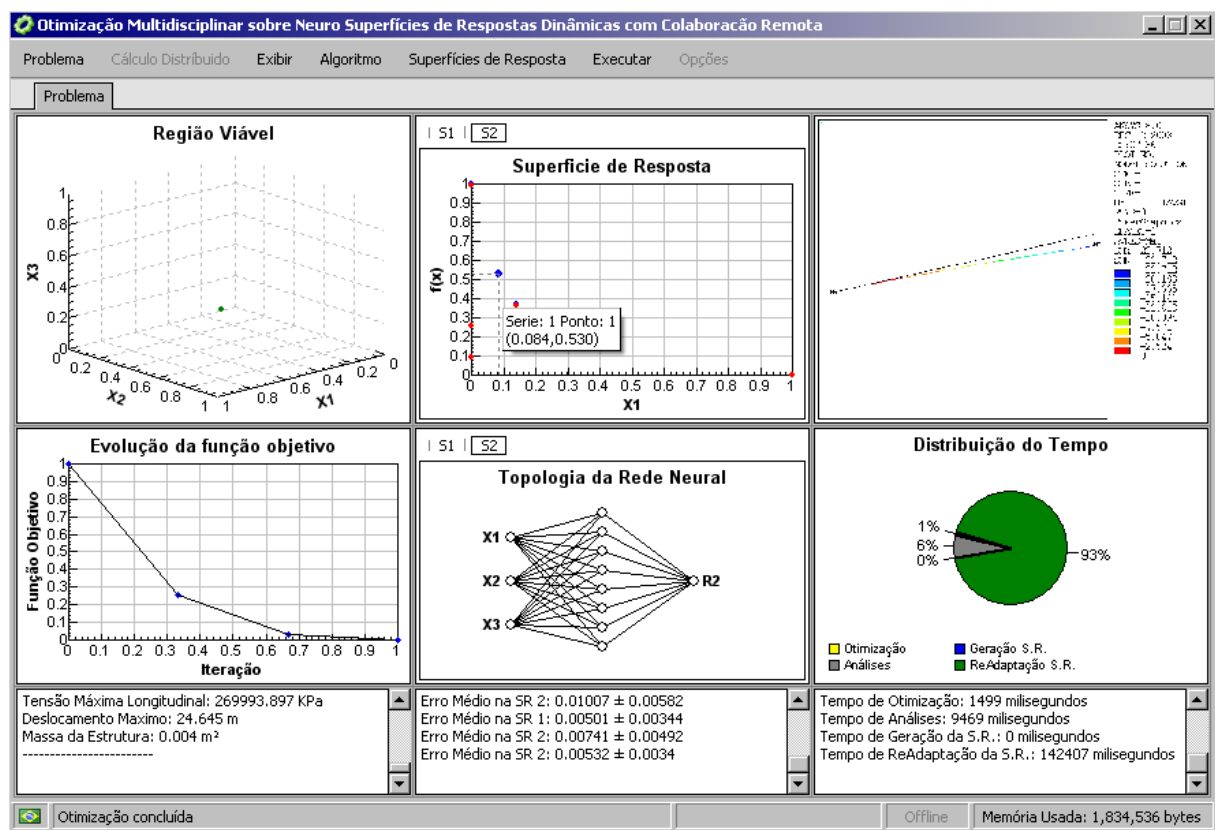


### 3.3.2 Objeto para geração de gráficos

Um dos módulos mais importantes para a interface do ambiente desenvolvido é o módulo de plotagem de gráficos.

Desenvolvido na forma de um objeto, podem ser implementados diversos gráficos simultâneos tendo como parâmetro de entrada um formulário padrão do *Delphi* e as coordenadas que definem a posição e o tamanho do gráfico utilizado.

Na figura 3.3 são exibidos 6 exemplos de gráficos gerados usando o objeto desenvolvido que pode ser utilizado para *plotar* gráficos de séries com 1, 2 (como o gráfico do canto inferior esquerdo) ou 3 parâmetros (como o gráfico do canto superior esquerdo), gráficos circulares (como o gráfico do canto inferior direito), figuras (como o gráfico do canto superior direito) e fluxogramas (como o gráfico do centro inferior). Podem-se incluir múltiplas séries além de uma linha de tendência polinomial.



**Figura 3.3.** Exemplos de gráficos construídos utilizando o objeto de gráficos desenvolvido.

O objeto permite configurar ainda os títulos dos eixos, a presença de legenda, o tipo de marca, as cores dos pontos das séries e da linha. Por fim, é possível criar, em um mesmo objeto, diferentes gráficos que são acessados por meio de uma barra de abas

superior. No caso de gráficos de séries pode-se configurar o objeto para destacar um ponto sobre o *mouse* e apresentar os valores do ponto em destaque.

O objeto é construído sobre um componente padrão do tipo *TPanel* configurado para utilizar *buffer* duplo, de forma a evitar o *flick* e apresentar uma atualização suave durante o processo. Outra característica importante do objeto é apresentar a capacidade de zoom. Ao sofrer um *click* do *mouse*, a imagem no gráfico é reproduzida em uma janela ampliada.

### 3.3.3 Objeto para edição do problema

Por definição, uma interface deve permitir a interação do usuário com a ferramenta computacional. Normalmente, em programas desse tipo, a entrada de dados é feita na forma de um arquivo de texto ou, em casos particulares, diretamente no código fonte.

Para resolver esse problema o programa desenvolvido possui um objeto de edição que permite editar todos os parâmetros do problema, como visto na figura 3.4.

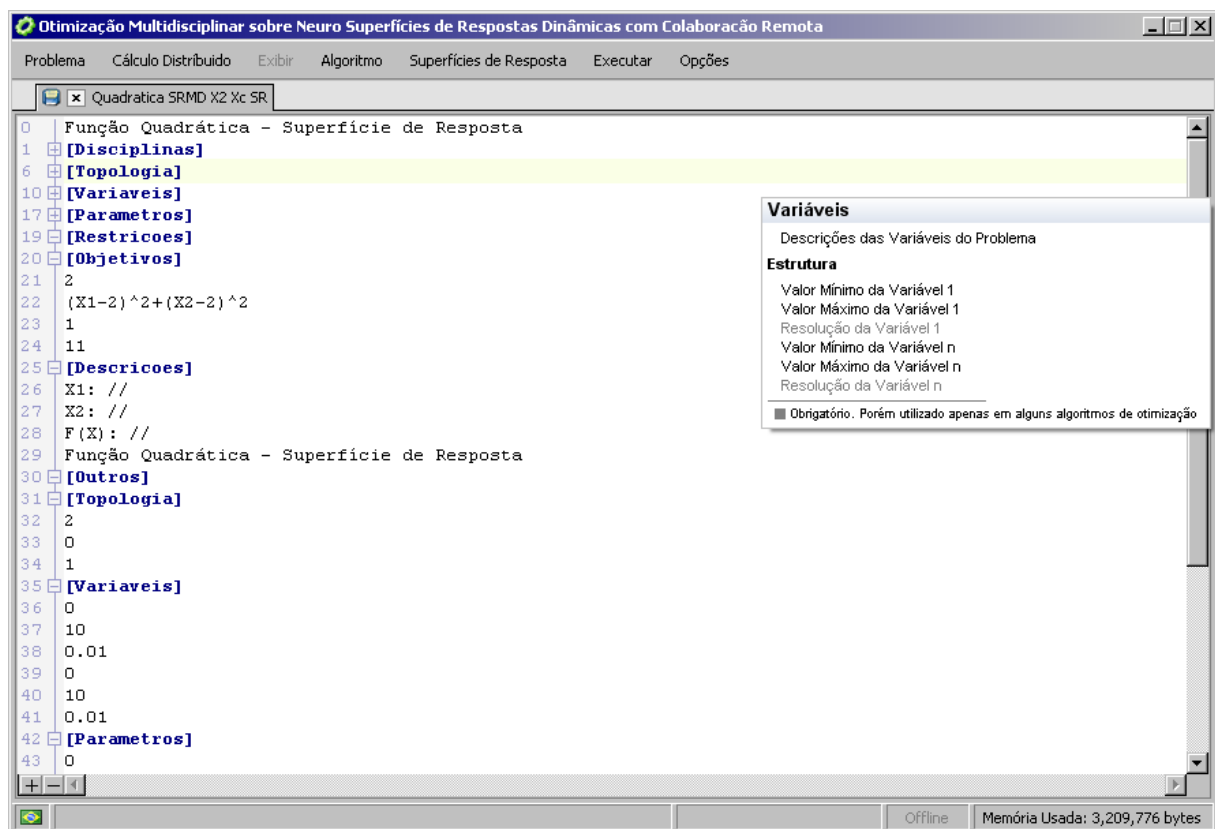


Figura 3.4. Objeto para edição do problema por blocos.

Esse objeto permite a edição por blocos, além de aceitar a presença de comentários que auxiliam a compreensão do arquivo de entradas. Além disso, a contagem de linhas, o destaque da linha em edição, a formatação por cores e a caixa de ajuda dinâmica, auxiliam o usuário a modelar de forma rápida o problema de otimização.

### 3.3.4 Objeto para resolução vetorial de expressões strings

Conforme visto na seção anterior, a entrada de dados para o problema de otimização é feita por meio de um arquivo de texto contendo, entre outras coisas, equações que representam parâmetros, funções objetivo ou de restrições.

Ao contrário dos ambientes de desenvolvimento científico (tais como *MatLab*, *Excel* e *Mathematica*), as linguagens de programação não possuem rotinas nativas e otimizadas para a resolução de expressões escritas na forma de texto.

Esse problema é facilmente percebido por pesquisadores que se empenham em desenvolver sistemas científicos complexos e que, portanto, exigem interfaces de entradas de dados simples, normalmente alimentadas por arquivos texto.

Normalmente a saída utilizada por pesquisadores no desenvolvimento de sistemas científicos é utilizar as rotinas de resolução de expressões que utilizam buscas de padrões e substituição de variáveis para calcular as respostas. A origem dessa técnica encontra-se nas interfaces de sistemas CAD (*Computer Aided Design*). Idealizados para serem programas complexos de desenho, grande parte dos sistemas CAD incorporaram rotinas de resolução de expressões texto baseadas em substituição de padrões. Com essas rotinas, o usuário podia inserir coordenadas relativas ou resultados de construções matemáticas diretamente na forma de expressões, enquanto o programa se responsabilizava por resolver a expressão e encontrar as coordenadas corretas de desenho. Essa técnica se popularizou e se espalhou por outros tipos de programas. Nesses casos a expressão é resolvida uma única vez, e a diferença entre milissegundos e segundos não afeta o trabalho do usuário.

O procedimento é extremamente lento e, quando repetido grande número de vezes, reduz significativamente a eficiência de algoritmos complexos. Para resolver esse problema foi desenvolvido um objeto para a resolução de expressões em formato de texto de maneira eficiente.

O princípio utilizado foi um processo em duas etapas. A primeira etapa ocorre logo quando o arquivo é aberto. Nesse momento são construídas as expressões na forma vetorial, utilizando as rotinas para a busca de padrões e substituição de variáveis. Cada valor presente é armazenado em um vetor de constantes, e cada operação existente é transformada em uma operação vetorial composta de 3 números. O primeiro número indica a posição do primeiro valor no vetor de constantes; o segundo número representa o tipo de operação a ser realizada, e o último número indica a posição do segundo valor no vetor de constantes. Completando o vetor de constantes com os valores intermediários encontrados,

o processo continua até que se converta totalmente a expressão da forma de texto para a expressão vetorial.

A segunda etapa ocorre todas as vezes em que é necessário resolver a expressão com um conjunto diferente de dados. Os valores das variáveis são substituídos no vetor de constantes, e a solução é obtida sem as lentas operações que utilizam *strings*.

Para verificar a eficiência do objeto de resolução de expressão utilizando o método vetorial realizou-se um teste comparativo, cujos resultados são mostrados na tabela 3.1.

Tabela 3.1. Comparação entre métodos de resolução de expressões na forma de texto.\*

Expressão resolvida 10 mil vezes	N de operações	Método String (milissegundos)	Método Vetorial (milissegundos)
3	0	0	0
3+5	1	312.5	1.6
(3+5)+(4-2.5)	3	1089.1	1.6
(3+5)*6+(4-2.5)*4	5	1776.5	3.0
(((3+5)*6+(4-2.5)*4))/3+33*1.2-4)	7	3346.9	4.7
(((3+5)*6+(4-2.5)*4))/3+33*(1.2-4)/(2+0.5)	10	4701.5	6.3

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

Para melhor visualização, os resultados desta comparação são mostrados na figura 3.5 como uma função que correlaciona o tempo de resolução em função do número de operadores da expressão.

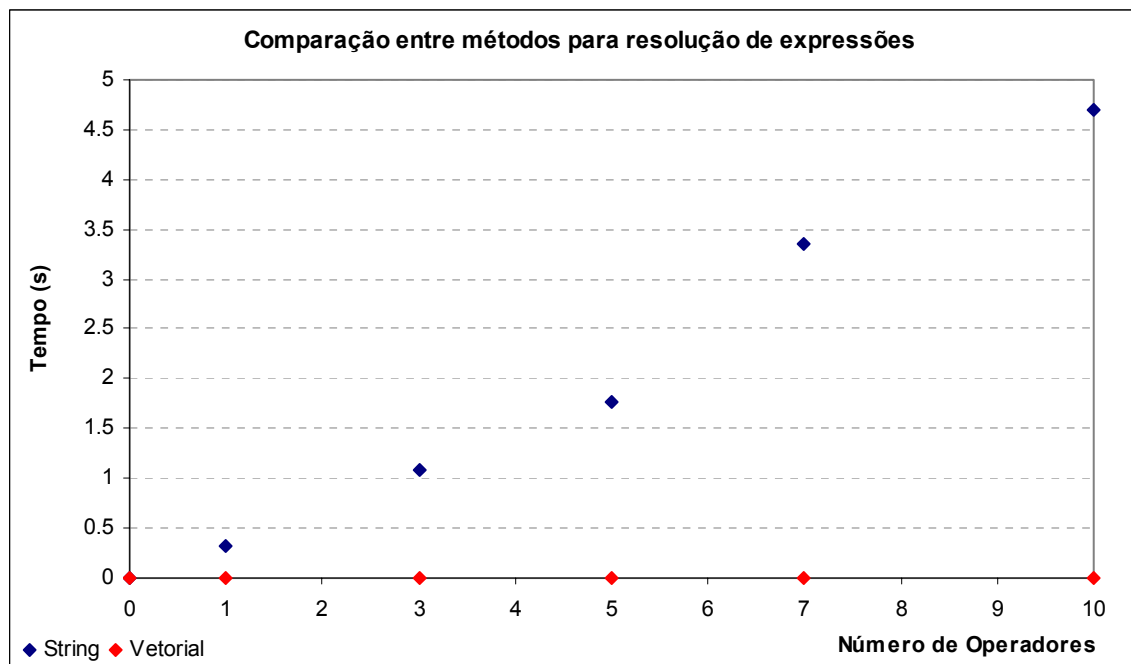


Figura 3.5. Comparação entre métodos de resolução de expressões na forma de texto.

### 3.3.5 Regiões tabus

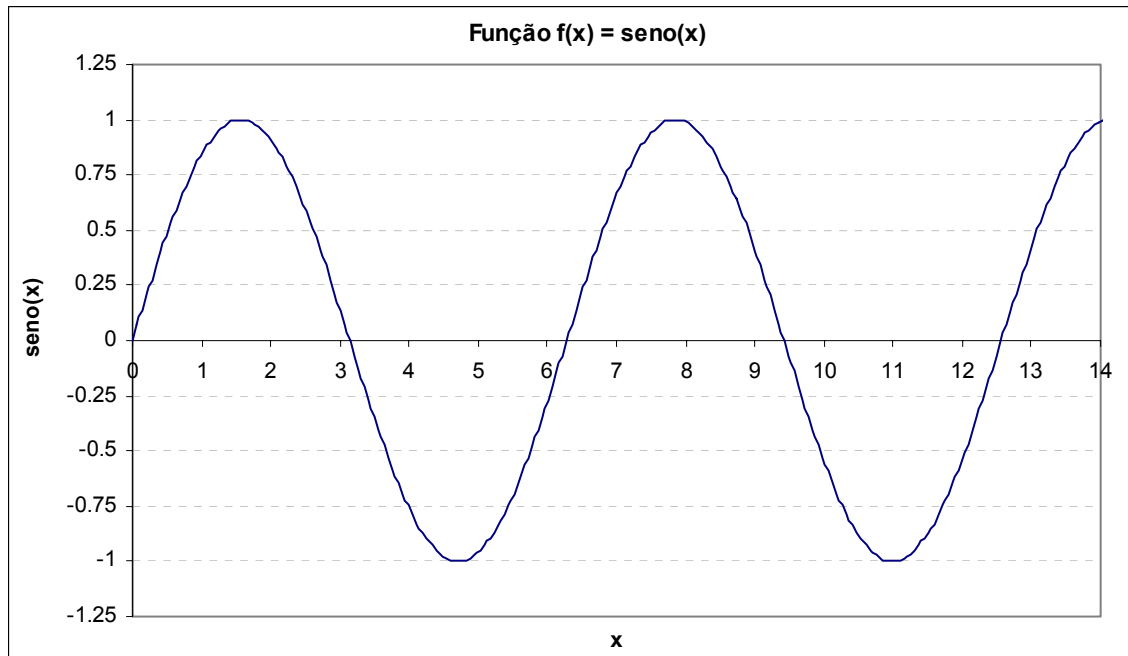
A criação de regiões tabus tem como inspiração o método *Tabu Search*, originalmente proposto por **Glover (1989)** e **Glover (1990)**. Neste caso algumas sub-regiões da região viável são bloqueadas de forma a limitar a movimentação do algoritmo de otimização.

No programa desenvolvido é possível criar dois tipos de regiões tabus:

- Esféricas: onde qualquer ponto a uma distância cartesiana menor que um valor definido torna-se restrito.
- Circular: onde apenas uma das variáveis é considerada, e qualquer ponto cujo valor desta variável seja arbitrariamente próximo torna-se restrito.

As regiões esféricas são utilizadas para a busca de soluções alternativas e, como será visto no capítulo seguinte, para tratar superfícies anômalas e viciadas.

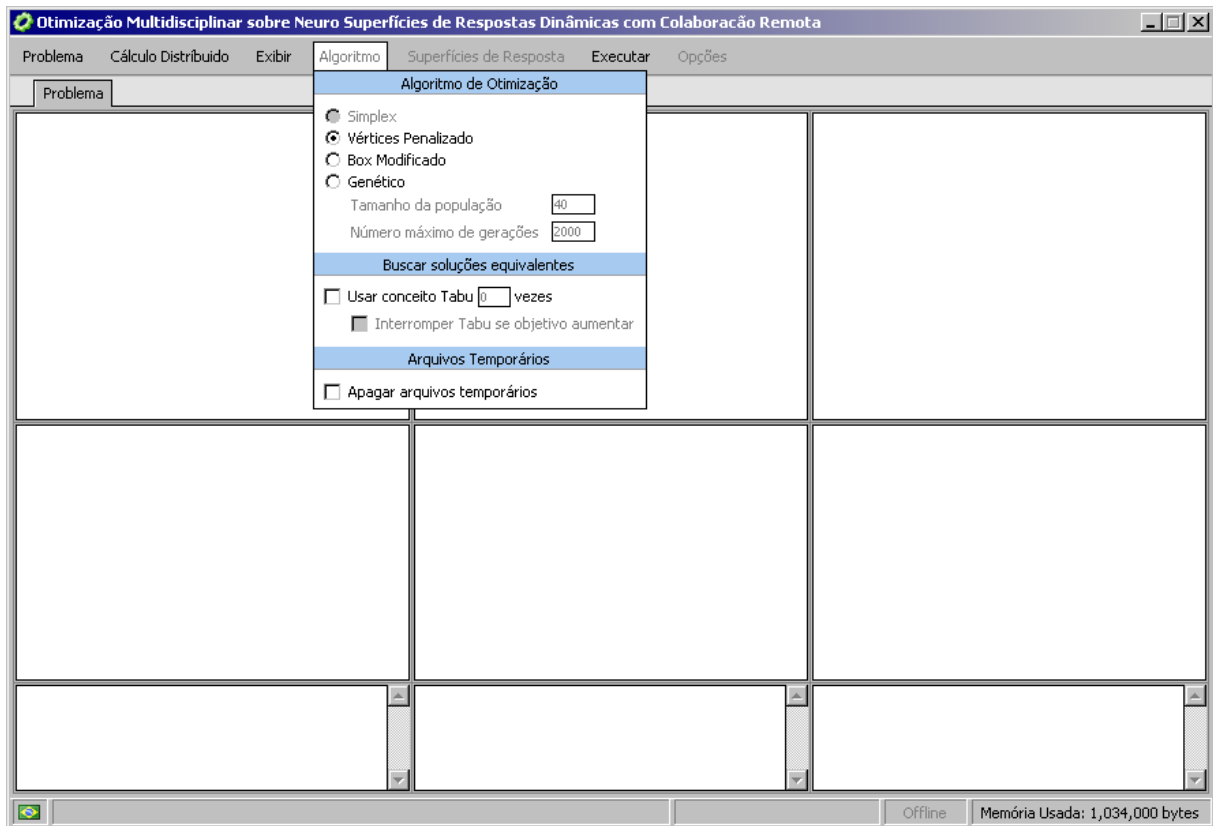
Um exemplo clássico em que se podem procurar soluções alternativas ocorre na função  $f(x) = \text{seno}(x)$  definida em um intervalo maior que  $[0, 2\pi]$ . Neste caso a função apresenta múltiplos pontos igualmente ótimos, como pode ser visto na figura 3.6.



**Figura 3.6.** Função  $f(x) = \text{seno}(x)$ .

Um algoritmo de otimização pode facilmente encontrar uma das soluções em detrimento da outra. Para evitar isso, pode-se configurar o programa (figura 3.7) para, após

ser encontrada a primeira solução, criar uma região tabu esférica no ponto ótimo e repetir a otimização, encontrando assim múltiplas soluções ótimas.



**Figura 3.7.** A pesquisa por soluções equivalentes usando regiões tabus.

Em alguns casos as soluções não são idênticas, mas próximas o suficiente para que o usuário possa escolher entre uma ou outra utilizando critérios não incluídos na modelagem do problema, como estética ou facilidade de fabricação.

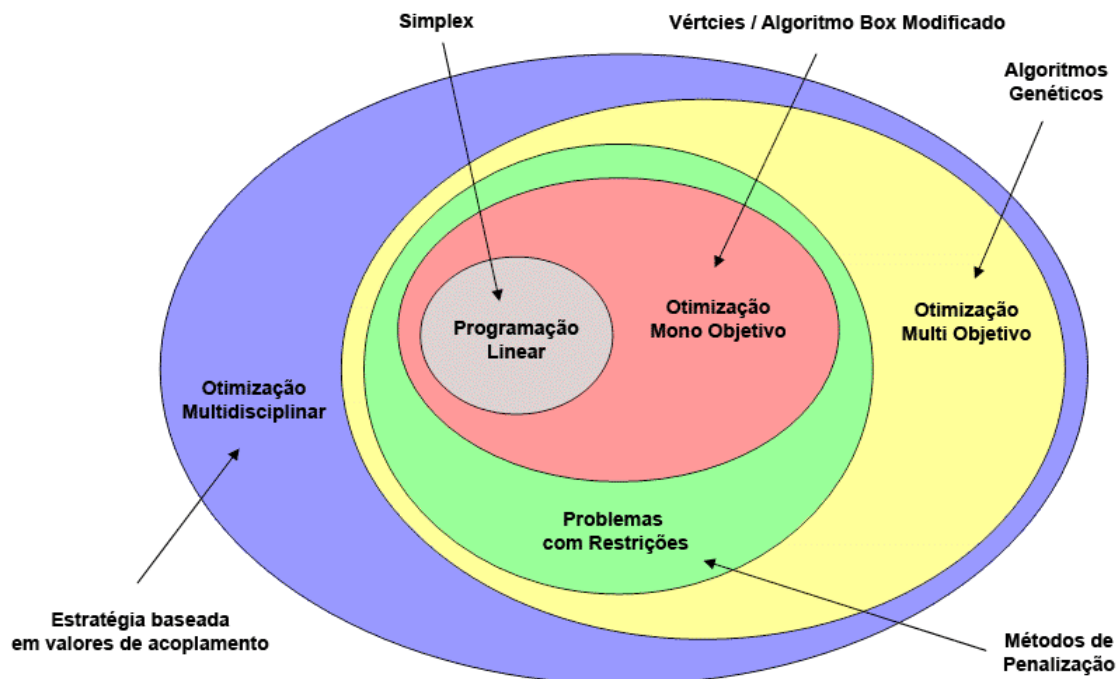
### 3.3.6 Algoritmos

Como foi visto na revisão bibliográfica, existem muitos tipos de problema de otimização e um igual número de técnicas para a resolução desses problemas.

Ao iniciar um problema, o programa detecta o tipo de problema e seleciona o algoritmo de otimização mais adequado a ele. Contudo, em alguns casos, o usuário pode escolher outro algoritmo de otimização para a resolução do problema.

Graças à estrutura do programa, baseada em objetos, pode-se facilmente incluir, substituir ou modificar um algoritmo de otimização para incluir novas técnicas ou modificações.

O esquema de algoritmos utilizados pelo programa está representada na figura 3.8. Conforme visto na figura, pode-se utilizar um algoritmo mono-objetivo (MBA ou Vértices<sup>23</sup>) para resolver problemas de programação linear, embora o oposto não seja possível.<sup>24</sup>



**Figura 3.8.** Abordagens utilizadas pelo programa de otimização.

É importante observar que internamente cada região da figura 3.8 representa um *loop* de programação. Isto faz com que o algoritmo seja naturalmente mais lento do que algoritmos específicos para cada um dos tipos de problema. Porém, é importante destacar que não se deseja disputar milissegundos com as soluções já existentes, mas sim desenvolver um ambiente robusto, capaz de resolver o maior número de problemas, em especial os problemas mais complexos de engenharia.

### 3.3.7 A otimização em regiões viáveis normalizadas

Conforme visto no capítulo anterior, a normalização das variáveis e das funções a serem representadas é uma parte fundamental para a geração das superfícies de resposta. Porém, para a normalização das variáveis é necessário estabelecer limites para todas as variáveis, o que implica uma região viável limitada.

<sup>23</sup> ✎ O algoritmo Vértices foi inspirado no método da Dicotomia. Embora não utilize a razão áurea para a contração da região viável, o algoritmo Vértices utiliza o deslocamento da região viável nas iterações nas quais o vértice ótimo é diferente da etapa anterior.

<sup>24</sup> ✎ Inicialmente planejava-se utilizar a estratégia COSMOS para a resolução de problemas multidisciplinares. No entanto, o decorrer do trabalho mostrou que uma nova abordagem, baseada em valores de acoplamento, se adequava melhor ao ambiente desenvolvido. Essa nova abordagem será detalhada no capítulo seguinte.

Analisar pontos fora do domínio pode resultar em erros numéricos, ao passo que truncar os pontos pode produzir pontos colineares que distorcem os resultados. A solução encontrada foi não analisar os pontos fora do domínio, atribuindo automaticamente valores elevados para as funções objetivo, de maneira que estes sejam descartados naturalmente ao longo do processo de otimização.

### **3.3.8 A otimização em regiões viáveis discretas**

Variáveis discretas sempre representaram um desafio para os algoritmos de otimização. Normalmente realiza-se a otimização supondo as variáveis contínuas, arredondando-se as soluções para os valores discretos permitidos. Porém, como alerta **Silva (2007)**, no entanto, que em alguns casos o ótimo para um conjunto discreto está distante do ótimo calculado para as variáveis contínuas e depois arredondado.

Para resolver esse problema, pode-se utilizar nos algoritmos genéticos uma resolução de tal sorte que os valores utilizados no processo estejam sempre nos intervalos discretos do problema. Essa abordagem resolve os problemas que possuem variáveis discretas em intervalos regulares. Porém, podem-se resolver os problemas com variáveis discretas não regulares ajustando a resolução das variáveis e atribuindo restrições tabus do tipo cilíndricas para os valores de variáveis não permitidos.

Outra alternativa é transformar as variáveis discretas não regulares em variáveis discretas regulares. Essa transformação é feita incluindo-se um parâmetro definido por uma função artificial que relaciona um índice inteiro e regular aos valores discretos reais do problema.

O uso de algoritmos genéticos implica, todavia, um problema existente em todos os algoritmos que dependam do particionamento da região viável. A definição da resolução das variáveis implica um particionamento da região viável que, em alguns casos, pode deixar escapar o valor ótimo. Pode-se observar, por exemplo, que a função com singularidade representada na figura 3.9 possui um ponto ótimo localizado em uma singularidade. Se a resolução da variável for maior do que o espaço ocupado pela singularidade, o algoritmo não será capaz de detectá-la.



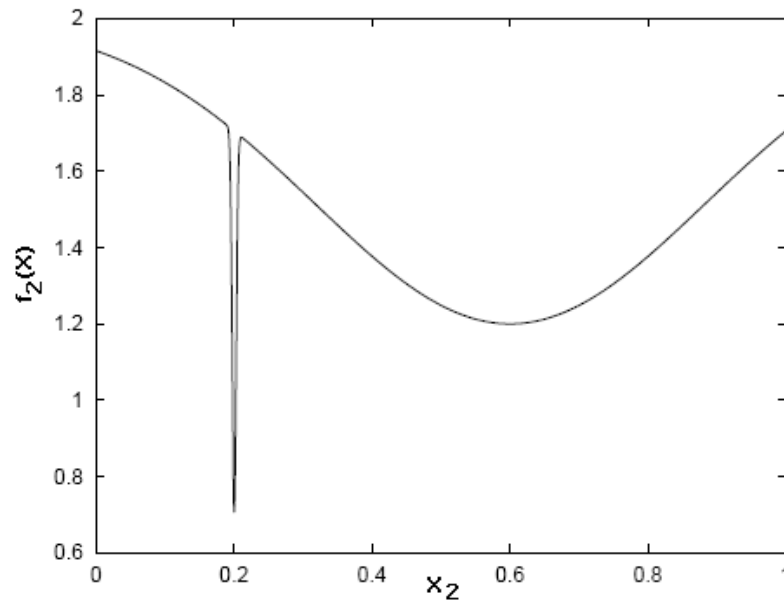


Figura 3.9. Função com singularidade.

### 3.3.9 Superfícies de Resposta

No capítulo anterior, foi apresentada a dificuldade associada ao longo tempo gasto nas análises computacionais ao longo de ciclos sucessivos de projeto e otimização. Como solução para acelerar esse processo podem ser utilizadas superfícies de resposta cujo uso permitirá ao algoritmo de otimização evitar muitas das longas análises baseadas em métodos numéricos que requerem alto tempo computacional, de modo a aumentar a eficiência do ambiente.

Embora, como apresentado, o uso de superfícies de respostas e redes neurais artificiais seja por várias vezes discutido e utilizado nos mais diferentes tipos de problemas de otimização, este trabalho se diferencia em dois aspectos: por utilizar superfícies de resposta representadas por redes neurais readaptadas dinamicamente na otimização de problemas de engenharia e por propor uma metodologia simples e eficiente para ser utilizada na automatização de qualquer projeto que demande a consulta de tabelas, gráficos e diagramas em qualquer fase do projeto.

Na figura 3.10 são apresentadas as diversas opções de configurações que o ambiente desenvolvido possui para a utilização de superfícies de respostas neurais no processo de otimização.

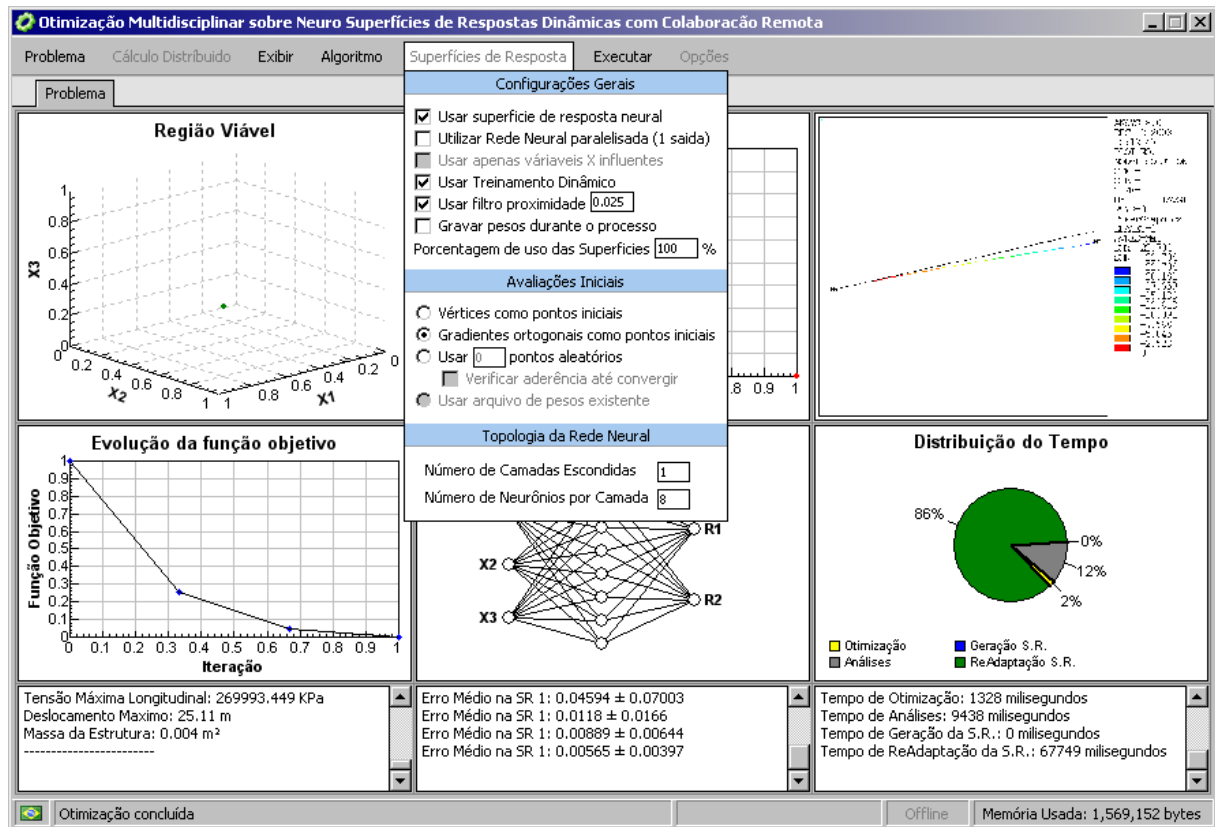


Figura 3.10. Menu de configuração das superfícies de resposta.

Por definição o programa sugere utilizar redes neurais com 1 camada escondida e 8 neurônios em cada camada escondida. Conforme visto na seção 2.2.3, esta topologia é suficiente para aproximar qualquer superfície não caótica. Para superfícies com grande número de discontinuidades sugere-se aumentar o número de camadas escondidas para 2.

Pode-se ainda optar por paralelizar as superfícies de resposta, modificar o conjunto de pontos de geração ou desabilitar o treinamento dinâmico e o filtro de proximidade.

### 3.3.10 Módulos de análises

Uma das características mais importantes do ambiente desenvolvido é a capacidade de utilizar diversos programas de análise em um mesmo projeto otimizado de engenharia, pois tais ferramentas são imprescindíveis aos projetos de engenharia, e normalmente representam grande limitação à otimização.

Podem ser utilizadas, simultaneamente, análises com equações e aplicativos como ANSYS, SAP, Excel ou ainda interpolações a partir de tabelas de dados. É importante, ressaltar que a estrutura modulada do programa permite incluir de forma fácil qualquer programa de análise que se deseje, desde que este permita a importação e exportação de dados utilizando arquivos de texto. Entre os exemplos já testados estão: Arena, MatLab,

*Wamit*, entre outros. As análises podem ser feitas em um ambiente de programação distribuída ou seqüencial.

Graças à estrutura de programação distribuída baseada no conceito de *Grid*, é possível utilizar recursos geograficamente separados, como discutido no capítulo anterior.

### 3.3.11 Exportação de Dados

Qualquer ferramenta computacional, para ser realmente útil, necessita de uma forma de armazenar e compartilhar os resultados. Por isso o ambiente desenvolvido possui duas formas diferentes para exportar os dados, presentes no menu executar, conforme visto na figura 3.11.

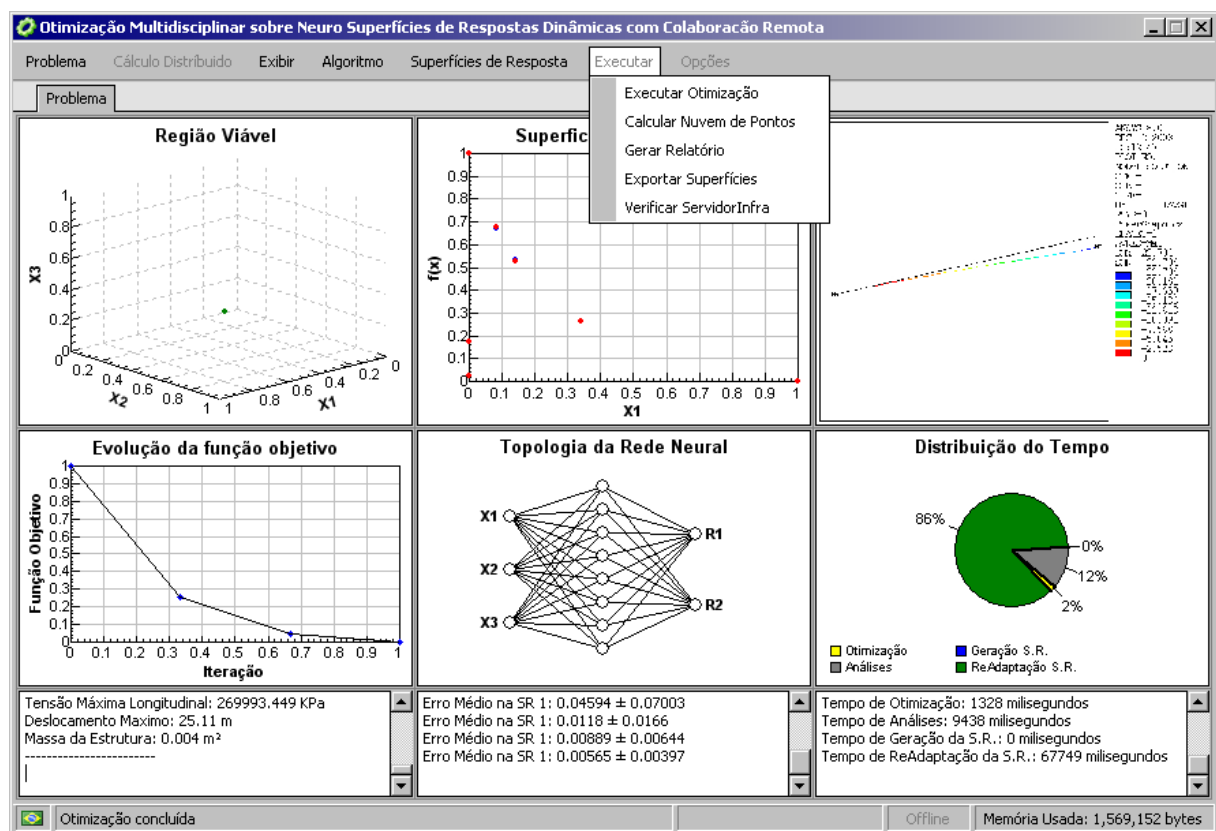


Figura 3.11. Opções para a exportação de dados.

A primeira forma é exportar as superfícies de resposta em arquivos compatíveis com o *Excel*. A planilha gerada permite calcular o valor das funções a partir de qualquer combinação dos valores das variáveis, dentro da região viável, como pode ser visto na figura 3.12.

Como será tratado no capítulo seguinte, a geração da planilha com a rede neural implementada é, por si só, uma ferramenta valiosa no projeto de sistemas complexos de engenharia

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Entradas		Saídas											
2	X1	0.00	R1	0.9030										
3	X2	0.00												
4	X3	0.00												
5														
6	0.0000	0.0000	13.91	0.0000	0.0000	2.09	0.8901	0.8901	-18.51	#####	-7.8010	10.03	0.9030	
7	0.0000	0.0000	1.46	0.0000			0.5620	0.5620	8.14	4.5737				
8	0.0000	0.0000	1.55	0.0000			0.5261	0.5261	1.97	1.0353				
9							0.4882	0.4882	-0.39	-0.1911				
10		0.0000	-1.95	0.0000	0.0000	0.25	0.9220	0.9220	-1.89	-1.7394				
11		0.0000	-0.95	0.0000			0.5408	0.5408	2.68	1.4482				
12		0.0000	-1.24	0.0000			0.7784	0.7784	1.66	1.2888				
13							0.5748	0.5748	3.93	2.2604				
14		0.0000	0.28	0.0000	0.0000	0.10								
15		0.0000	0.38	0.0000										
16		0.0000	0.08	0.0000										
17														
18		0.0000	1.19	0.0000	0.0000	-0.05								
19		0.0000	0.48	0.0000										
20		0.0000	1.32	0.0000										
21														
22		0.0000	3.88	0.0000	0.0000	2.47								
23		0.0000	1.09	0.0000										
24		0.0000	1.49	0.0000										
25														
26		0.0000	0.71	0.0000	0.0000	0.16								
27		0.0000	0.18	0.0000										
28		0.0000	-0.13	0.0000										
29														
30		0.0000	2.36	0.0000	0.0000	1.26								
31		0.0000	1.05	0.0000										
32		0.0000	0.99	0.0000										
33														
34		0.0000	0.30	0.0000	0.0000	0.30								
35		0.0000	-0.38	0.0000										
36		0.0000	-0.19	0.0000										
37														

Figura 3.12. Superfície de resposta exportada para o Excel.

A segunda forma de exportar os resultados é gerar um relatório de otimização. O relatório pode ser gerado no formato PDF ou TXT e contém todas as informações do processo de otimização: dados do problema, algoritmos utilizados, configurações, gráficos, soluções, superfícies de resposta, tempo do processo, entre outras.

### 3.4 O ambiente de programação distribuída desenvolvido

No capítulo anterior demonstrou-se o potencial da programação paralela e distribuída para o aumento da eficiência e redução do tempo computacional gasto na resolução de um problema complexo.

O trabalho desenvolvido aproveita esse potencial ainda latente, em um ambiente para a elaboração de projeto de engenharia com processamento distribuído pela Internet.

O objetivo é desenvolver um ambiente de projeto, em que os ciclos de análise sejam distribuídos ao longo de diferentes nós de uma arquitetura de programação distribuída com memória não compartilhada.

Cada nó realizará tarefas específicas com um conjunto de dados pertencentes ao domínio de otimização, podendo apresentar grande heterogeneidade entre os nós. Os processos serão sincronizados e balanceados pelo servidor de infra-estrutura, o qual alimentará o módulo central de otimização propriamente dito.

A sincronização será feita por passagem de mensagem pela Internet utilizando o protocolo TCP/IP. A comunicação entre os módulos seguirá o modelo clássico MPI, discutido no capítulo anterior.

A arquitetura de programação distribuída não impedirá que cada nó possua uma máquina de processamento paralelo para resolver os complexos algoritmos baseados no Método dos Elementos Finitos.

Portanto, o ambiente proposto não será homogêneo e terá nós formados por computadores com diferentes capacidades computacionais. Tal estrutura, como visto no capítulo anterior, claramente classifica o modelo utilizado como *Grid* computacional.

### **3.4.1 Requisitos do sistema**

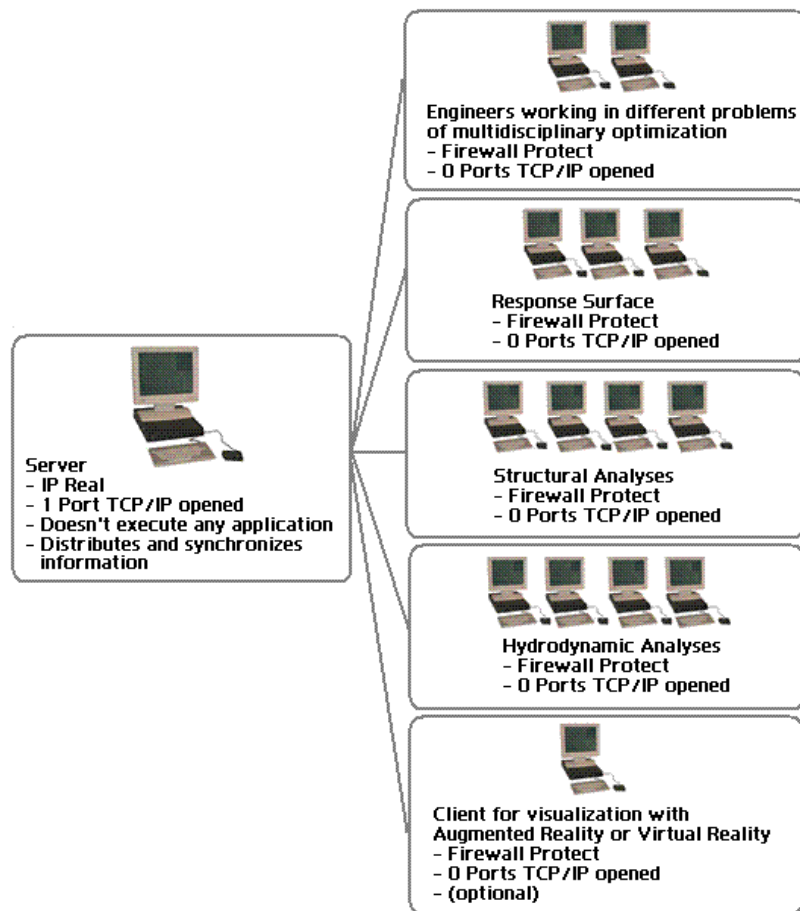
Os requisitos do sistema proposto incluem o desenvolvimento de:

- um cliente que pode reunir a interface de modelagem e visualização, bem como o algoritmo de otimização propriamente dito;
- um servidor de infra-estrutura que distribuirá as tarefas e sincronizará os resultados, fazendo o balanço de carga entre os computadores conectados ao ambiente;
- diferentes clientes, construídos sobre uma base padrão, capazes de realizar, autonomamente, as análises solicitadas pelo servidor de infra-estrutura. Pode-se destacar em especial o cliente capaz de realizar análises utilizando o Método dos Elementos Finitos;
- um cliente de redes neurais capaz de treinar constantemente redes que representem as superfícies de resposta do problema analisado;

- um cliente opcional de visualização em realidade aumentada.<sup>25</sup> O sistema desenvolvido pode incluir um cliente especial para visualização de dados remotamente.

Todos os módulos devem ser sincronizados utilizando passagem de mensagem pela Internet. O cliente de otimização deve poder ser executado dentro do ambiente de programação distribuída ou isoladamente, trabalhando de forma seqüencial.

O ambiente desenvolvido pode ser visto na figura 3.13.



**Figura 3.13.** Representação do ambiente desenvolvido.

<sup>25</sup> Os diversos trabalhos publicados sobre Realidade Aumentada discutidos, por exemplo, por Azuma (2001), sinalizam o potencial que essa área oferece para a visualização científica e modelagem de complexas estruturas tridimensionais. Entre os trabalhos mais recentes destaca-se Belcher (2003) que propõem a utilização da Realidade Aumentada para visualização de gráficos e superfícies tridimensionais, destacando-se por estabelecer um estudo comparativo entre o desempenho de usuários que utilizam o atual paradigma de visualização em monitor e usuários que utilizam interfaces baseadas em Realidade Aumentada. Belcher conclui que tais interfaces podem ser até 30% mais eficientes, embora ressalte a necessidade de novos estudos para estabelecer métricas para uma melhor avaliação dessas interfaces.

No decorrer deste trabalho um cliente de visualização baseado na biblioteca *ArToolKit* (voltada para o desenvolvimento de aplicações em Realidade Aumentada) chegou a ser desenvolvido. Porém, esta biblioteca está limitada à linguagem de programação C++, o que dificulta sobremaneira sua integração ao ambiente desenvolvido. Por limitações de tempo e para manter o foco nos elementos centrais deste trabalho, o desenvolvimento desse módulo foi interrompido.

Um requisito importante é que o servidor de infra-estrutura deve ser capaz de detectar a entrada ou saída de computadores do ambiente de trabalho e assim rebalancear as cargas, realocando as tarefas conforme a disponibilidade existente.

### **3.4.2 Implementação da programação distribuída**

O problema existente é que a biblioteca MPI foi desenvolvida para programas escritos em linguagem C, o que inviabiliza a sua utilização nas aplicações desenvolvidas em *Delphi*. Portanto, desenvolveu-se a aplicação baseada nos paradigmas do MPI, implementando o mesmo conjunto básico de funções e princípios de sincronização existentes.

A simplificação realizada é que só é possível trocar dados com a aplicação servidora (no caso o servidor de infra-estrutura), não sendo possível a troca de informações direta entre os nós do ambiente.

As mensagens trocadas para a sincronização do processo podem ser blocantes<sup>26</sup> ou não blocantes. Isto é, em alguns casos o programa que enviou a mensagem só pode continuar o processo se receber a confirmação de que a mensagem enviada foi corretamente recebida e processada.


### **3.4.3 Iniciando o ambiente de programação distribuída.**

O ambiente é composto por, no mínimo, quatro módulos que podem ser iniciados em um mesmo computador ou em computadores diferentes.

O primeiro, e mais importante módulo, é o servidor de infra-estrutura. Este módulo é iniciado em primeiro lugar e não pode ser encerrado até que o ambiente de projeto seja desfeito.

Um servidor de infra-estrutura pode distribuir e balancear a carga, além de sincronizar mais de um ambiente de projeto (ou cliente de otimização) simultaneamente, aceitando até 255 conexões remotas. O computador utilizado pelo servidor de infra-estrutura é o único computador do ambiente a executar uma rotina de servidor, aguardando de forma passiva que outros computadores se conectem a ele através de uma porta TCP/IP designada.

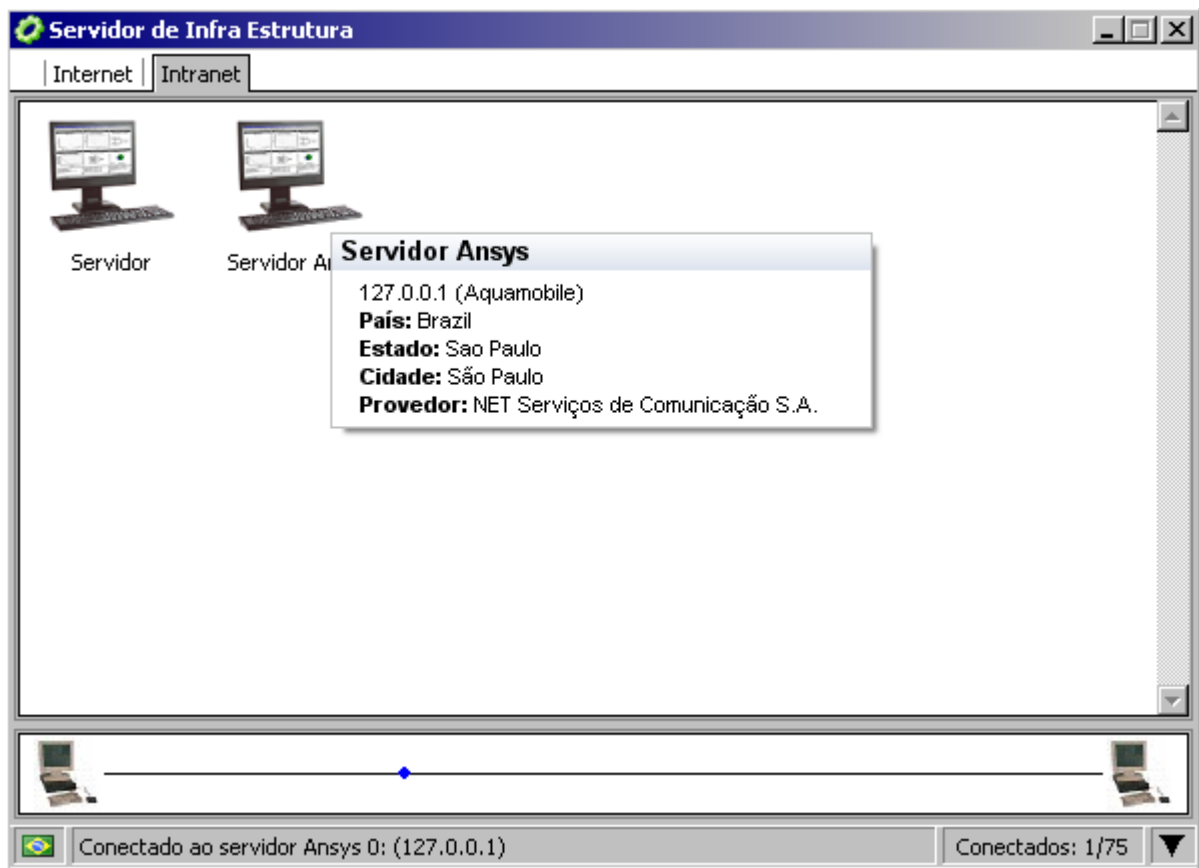
---

<sup>26</sup>  Procura-se ainda uma formalização da língua portuguesa para muitos termos utilizados em programação. Há quem prefira o termo “bloqueante”.

Ao ser iniciado, o servidor de infra-estrutura executa um diagnóstico verificando a presença de roteadores, *firewalls* ou outras barreiras, indicando se está pronto ou não para iniciar a otimização distribuída.

A seguir, cada um dos computadores que servirá de módulo de análise deve ser iniciado. Toda vez que um módulo (cliente) de análise é iniciado, ele se comunica com o servidor de infra-estrutura sinalizando que está disponível para realizar um tipo de análise. Podem-se iniciar clientes de análise em qualquer instante do processo até o limite de conexões do servidor de infra-estrutura.

Ao receber uma conexão, o servidor de infra-estrutura identifica o cliente, determinando sua localização (Intranet ou Internet) e o tipo de tarefa a que se destina, exibindo essas informações na interface, conforme visto na figura 3.14.



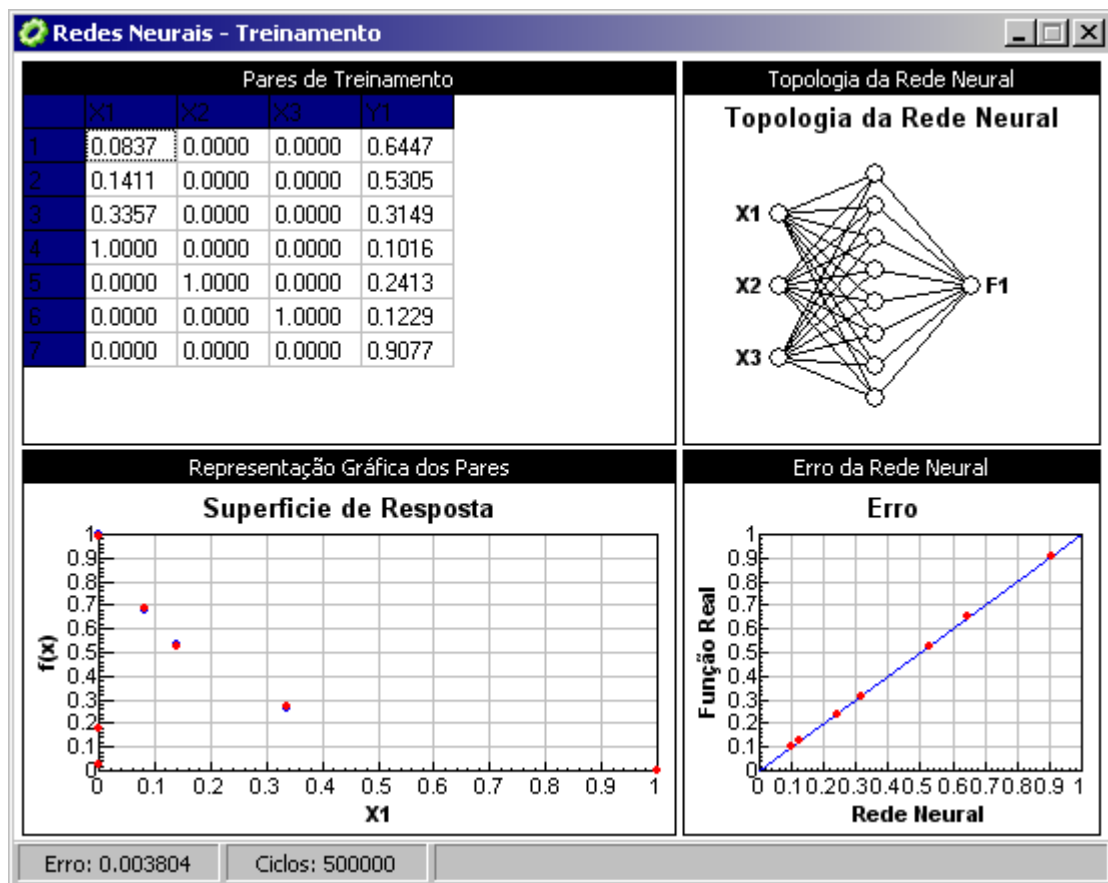
**Figura 3.14.** Interface do servidor de infra-estrutura.

Uma observação interessante é que os clientes de análise podem ser computadores ou arquiteturas paralelas com grande poder de processamento e que, concluída a análise, retornam à posição inicial e aguardam uma nova solicitação do servidor de infra-estrutura.



Os clientes de redes neurais assim que iniciados também se conectam ao servidor de infra-estrutura e se colocam à disposição para realizar treinamento de redes ou fornecer resultados baseados nas redes já treinadas. Pode-se configurar esse módulo para exibir as informações durante o treinamento, conforme mostrado na figura 3.15.

O módulo de treinamento neural compreende 4 campos visuais que permitem ao usuário acompanhar o progresso do treinamento. No campo superior esquerdo são armazenados os pares normalizados utilizados no treinamento. No campo superior direito é mostrada a topologia da rede neural construída. No campo inferior esquerdo é mostrado um gráfico contendo os pares utilizados no treinamento e os pares calculados no último ciclo de treinamento, permitindo acompanhar a convergência do treino. Por fim, no campo inferior direito, são mostrados os erros existentes para cada um dos pares de treinamento.



**Figura 3.15.** Interface do cliente de redes neurais executando um treinamento remoto.

Por fim, o cliente de otimização, que pode servir como interface de modelagem e visualização do projeto, se conecta ao servidor de infra-estrutura. Ao ser iniciado, o cliente de otimização tenta se conectar ao servidor de infra-estrutura. Quando consegue se conectar, o cliente de otimização libera a possibilidade de o usuário utilizar a otimização com cálculo distribuído; caso contrário, restringe o processo à otimização seqüencial.

Uma vez que todos os módulos tenham sido iniciados e conectados ao servidor de infra-estrutura, o ambiente de projeto está construído.

Como descrito anteriormente, outros módulos podem ser iniciados a qualquer instante, e diferentes processos de otimização podem ocorrer de forma simultânea, compartilhando os clientes de análise e clientes neurais disponíveis no ambiente.

Para reduzir o número de computadores utilizados, o mesmo computador pode realizar diferentes funções. A rigor, todos os módulos poderiam ser iniciados em um mesmo computador, embora isso reduziria significativamente a eficiência do processo. Porém, para ambientes com apenas um cliente de otimização, o mesmo computador pode atuar como cliente de redes neurais e como cliente de análise, sem perda de eficiência, pois ambos os processos não ocorrem simultaneamente.

#### **3.4.4 A otimização utilizando ambiente distribuído**

O processo de otimização precisa realizar inúmeros ciclos de análise para então conduzir o projeto a uma solução ótima que represente algum ganho de eficiência, desempenho ou custo.

Os parâmetros do projeto (variáveis, funções objetivo e de restrições) e os modelos (macros) de análise são desenvolvidos no cliente de otimização.

A cada iteração do processo, o cliente de otimização solicita ao servidor de infra-estrutura análises numéricas associadas ao ponto em questão.

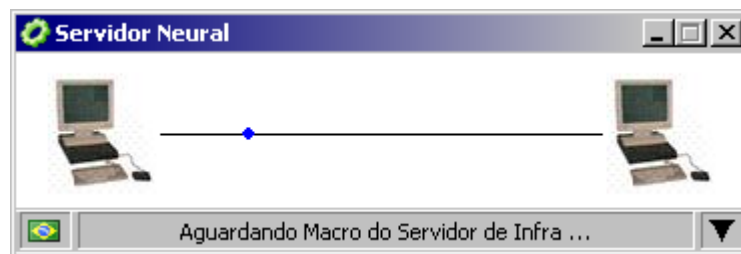
O servidor de infra-estrutura verifica quais computadores do ambiente têm capacidade no momento para realizar essas análises. Caso não existam computadores disponíveis, o servidor de infra-estrutura armazena a solicitação em uma fila de processos. Ao determinar que um computador no ambiente pode realizar a análise, o servidor de infra-estrutura transmite ao cliente de análise apenas os parâmetros do ponto relativo àquela análise.

O cliente de análise então processa a análise e, ao concluí-la, envia os resultados ao servidor de infra-estrutura, que por sua vez, retransmite os resultados ao cliente de otimização. Esse processo se repete continuamente até que o algoritmo de otimização encontre a solução ótima para o projeto em desenvolvimento.

#### 3.4.4.1 Sincronização

Como foi descrito, existe grande fluxo de dados entre os módulos de análise, os módulos de treinamento neural, o módulo de otimização e o servidor de infra-estrutura. Esse fluxo é intensificado quando se utilizam múltiplos clientes de otimização trabalhando diferentes problemas simultaneamente.

Um problema importante é a sincronização correta das iterações e seus respectivos conjuntos de dados e resultados. Essa sincronização é feita mantendo-se um “pipe” exclusivo entre o servidor de infra-estrutura e cada um dos módulos. Dessa forma, os dados ficam associados ao número de cada conexão. Um exemplo de *pipe* de conexão é mostrado na figura 3.16.



**Figura 3.16.** Comunicação entre o cliente de redes neurais e o servidor de infra-estrutura.

Todas as mensagens são passadas de forma não bloqueante, permitindo que se continuem os processos, porém a cada mensagem é exigida uma confirmação de recebimento e, assim, cada módulo só pode enviar uma nova mensagem se a anterior tiver sido processada corretamente.

#### 3.4.4.2 Paralelismo

A aplicação desenvolvida utiliza tanto o paralelismo de dados, (visto que diferentes processos similares trabalham com partes de um conjunto de dados) quanto o paralelismo de controle, pois processos distintos utilizam o mesmo conjunto de dados.

Ao avaliar vários indivíduos de uma população, o sistema pode dividi-los em múltiplos computadores, de forma a realizar o mesmo procedimento aplicado a diferentes valores de variáveis. Neste caso é empregado o conceito de paralelismo de dados. Por outro lado, uma análise de um único indivíduo pode incluir, por exemplo, uma subanálise estrutural e uma subanálise hidrodinâmica. Neste caso o sistema pode dividir as subanálises entre computadores especializados empregando o conceito de paralelismo de controle.

#### **3.4.4.3 Segurança**

Em função das possibilidades apresentadas até o momento, algumas considerações sobre a segurança do ambiente desenvolvido fazem-se necessárias.

Todos os clientes trabalham com uma conexão direta com o servidor de infra-estrutura. Portanto, nenhum dos computadores nos quais esses módulos são iniciados possuem portas TCP/IP abertas. Além disso, todas as mensagens trocadas com o servidor de infra-estrutura podem ser criptografadas e um filtro verifica as macros recebidas em busca de comandos prejudiciais ao sistema, evitando possíveis ataques externos.

O servidor de infra-estrutura é o único computador “vulnerável” do sistema, pois mantém portas TCP/IP abertas. Entretanto, o servidor de infra-estrutura tem a função de distribuição, balanceamento e sincronização dos dados, não executando nenhum processo externo ao módulo, o que garante a segurança do sistema.

## 4 Contribuições Relevantes

---

4.1	O uso de redes neurais artificiais com treinamento dinâmico	102
4.1.1	<i>O modelo e topologia da rede neural utilizada</i>	103
4.1.2	<i>A geração de superfícies de resposta inicial</i>	104
4.1.3	<i>Otimização e adaptação dinâmica</i>	107
4.1.4	<i>Readaptação dinâmica da superfície de resposta</i>	108
4.1.5	<i>Avaliação do uso de superfícies de resposta</i>	109
4.1.6	<i>O problema de interpolação de gráficos e diagramas</i>	111
4.1.7	<i>Conclusão</i>	111
4.2	Otimização utilizando ambiente distribuído pela Internet	112
4.2.1	<i>O ambiente desenvolvido</i>	112
4.2.2	<i>Otimização e sincronização</i>	113
4.2.3	<i>Considerações sobre a eficiência do ambiente distribuído</i>	113
4.2.4	<i>Estimativa do tempo total de processo</i>	118
4.2.5	<i>Conclusão</i>	119
4.3	Superfícies anômalas e viciadas, tratamento com regiões tabus	120
4.3.1	<i>Superfícies anômalas de resposta</i>	120
4.3.2	<i>Superfícies viciadas de resposta</i>	122
4.3.3	<i>Abordagens tradicionais para superfícies anômalas e viciadas</i>	123
4.3.4	<i>Abordagem baseada em regiões tabus</i>	124
4.3.5	<i>Avaliação da utilização de regiões tabus com superfícies de resposta</i>	125
4.3.6	<i>Conclusão</i>	127
4.4	Otimização multidisciplinar baseada nos valores de acoplamento	127
4.4.1	<i>Diferença entre acoplamentos fortes e fracos</i>	128
4.4.2	<i>A estratégia multidisciplinar baseada em valores de acoplamento</i>	128
4.4.3	<i>A utilização de superfícies de resposta</i>	131
4.4.4	<i>Conclusão</i>	132

*"Genius is one percent inspiration,  
ninety-nine percent perspiration."*

Thomas Alva Edison

Neste capítulo serão apresentadas as principais contribuições relevantes desenvolvidas nessa pesquisa. Essas contribuições distanciam esta pesquisa dos trabalhos apresentados na revisão bibliográfica e, juntamente com a ferramenta computacional desenvolvida, representam o principal fruto deste trabalho. O capítulo está estruturado em 4 seções, correspondendo a cada uma das contribuições relevantes desenvolvidas.

#### 4.1 O uso de redes neurais artificiais com treinamento dinâmico<sup>27</sup>

Como visto na seção 2.2.1, os engenheiros inseridos no moderno contexto de otimização multidisciplinar de sistemas complexos de engenharia em ambientes distribuídos têm encontrado dois problemas para desenvolver seus trabalhos.

Por um lado, projetos de engenharia, especialmente no contexto da indústria naval, envolvem a utilização de parâmetros que são tabelados ou descritos na forma de gráficos e diagramas, fornecidos por regras de sociedades classificadoras ou estudos específicos sobre particularidades do assunto. A utilização de tais gráficos e diagramas tornam a análise um procedimento lento e manual, inviabilizando a automatização do processo e, portanto, a otimização.


Por outro lado, classicamente sistemas complexos de engenharia consomem longo tempo de trabalho e recursos computacionais para a análise de uma única configuração, o que inviabiliza processos de otimização que demandem grande número de iterações, de modo que, muitas vezes, se adota a primeira solução a satisfazer as restrições.

O conceito geral proposto, conforme apresentado na seção 2.2.2, é automatizar a representação de funções normalmente descritas em gráficos e diagramas e, ao mesmo tempo, viabilizar a otimização de complexos sistemas de engenharia, utilizando uma superfície de resposta sobre a qual o processo de otimização é conduzido de maneira acelerada, evitando as lentas análises numéricas normalmente associadas a estes tipos de projetos.

Como visto na seção 2.2.4, a escolha do uso de superfícies de resposta representadas por redes neurais artificiais deve-se a simplicidade desta técnica de inteligência computacional que permite criar sinapses que podem intuir resultados baseando-se em um conjunto de experiências. Desta forma, o uso de redes neurais permite a adaptação a novos eventos, bem como armazena de forma nativa a experiência do engenheiro ou da empresa ao longo de anos de projetos ou análises similares.

Porém, a metodologia proposta por este trabalho se distancia do atual modelo de otimização com superfícies de resposta estáticas (utilizado, por exemplo, em **Edwards**

---

<sup>27</sup>  Os conceitos discutidos nesta seção foram apresentados na *Conferência Nacional em Inteligência Computacional Aplicada à Indústria de Petróleo (ICAPetro)*, Rio de Janeiro, Brasil, 11-14 Junho, 2007.

(1997)), pois as superfícies são dinamicamente adaptadas durante as iterações do processo de otimização. A adaptação dinâmica das superfícies de resposta garante a rápida e coerente convergência do processo de otimização.

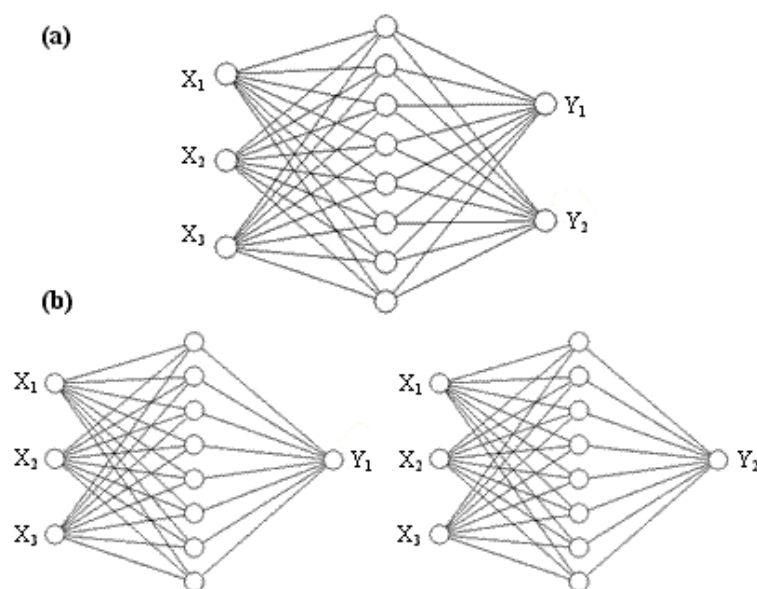
#### 4.1.1 O modelo e topologia da rede neural utilizada

Entre os vários tipos de redes neurais artificiais existentes, as redes do tipo *Multilayer Perceptrons* (MLP), discutidas na seção 2.2.3, foram aquelas que se mostraram mais adequadas para a representação de superfícies de resposta.<sup>28</sup>

Para a função de ativação dos neurônios, optou-se pela clássica função *Sigmoid*, discutida na seção 2.2.3, que apresentou bons resultados de convergência. Embora alguns autores, como **Nielsen (2002)**, usem funções de base radial, essa opção não modifica sensivelmente a eficiência do método.

A primeira questão importante que se coloca é o número de redes neurais que serão utilizadas. Existem basicamente duas abordagens diferentes para o mesmo problema:

- 1 rede neural com  $n$  saídas, onde  $n$  representa o número de parâmetros do problema a serem representadas; (figura 4.1a)
- $n$  redes neurais cada uma com 1 saída, onde  $n$  representa o número de parâmetros do problema a serem representadas; (figura 4.1b)



**Figura 4.1.** Determinação da Topologia da Rede Neural.

<sup>28</sup> Aos leitores interessados, o trabalho de Murray (1995) apresenta uma boa discussão sobre esse e outros tipos de redes neurais existentes, bem como suas principais aplicações.

Embora, a favor da simplicidade (todos os parâmetros do problema estão agrupados), o impulso inicial seja utilizar uma única rede neural com  $n$  saídas, a experiência mostrou que o treinamento de uma rede complexa é mais demorado do que o treinamento de  $n$  redes simples com 1 única saída, especialmente quando aos  $n$  parâmetros estão associadas disciplinas diferentes. Além disso, a existência de  $n$  redes neurais permite que se utilizem conceitos de processamento distribuído, descritos na seção 2.3, para aumentar a eficiência total do processo de otimização, dado que os treinamentos podem ocorrer em paralelo com custo praticamente zero de sincronização.

O número de entradas da rede corresponde ao número de variáveis associadas ao modelo. Em caso de múltiplas redes neurais, podem ser utilizadas apenas as variáveis que influenciem o parâmetro representado.


Por fim, o número de camadas escondidas e de neurônios em cada camada não distanciam dos valores usuais para esse tipo de aplicação. Conforme demonstrado por **Cybenko (1989)**, para superfícies contínuas, sugere-se uma camada escondida com 8 neurônios, enquanto superfícies complexas e com alto índice de descontinuidade podem requerer até 2 camadas escondidas e um número maior de neurônios por camada.

Como regra geral, quanto mais complexa for a topologia da rede, mais lento é o treinamento (não apenas o tempo de cada ciclo de treino é maior, mas também a taxa de convergência é menor, exigindo, em geral, mais ciclos para o mesmo erro de aproximação). No entanto, podem-se alcançar erros menores para redes mais complexas, em virtude da melhor representação do problema. Por fim, é importante destacar que, como alerta **Anjum (1997)**, para cada ligação neural estabelecida na rede é necessário um número maior de pares para o treinamento.

#### **4.1.2 A geração de superfícies de resposta inicial**

Após escolher a topologia da rede neural, deve-se determinar o número de pares iniciais que comporão o conjunto de treinamento que adaptará a superfície de resposta ao problema em questão. Esse processo, conhecido como “planejamento de experimentos”, tem como base os estudos existentes para a determinação dos ensaios realizados em pesquisas experimentais.<sup>29</sup>

---

<sup>29</sup>  Embora o termo em português seja muito utilizado, algumas publicações ainda registram a expressão em inglês “design of experiments”.



Diversos autores exploram o assunto, entre eles estão **Shyy (2001)** e **Baş (2007)**. O recente trabalho de **Shuaeib (2007)** discute este assunto no contexto da otimização.

Escolhendo-se um número muito elevado de pontos, (abordagem utilizada, por exemplo, por **Agatonovic-Kustrin (1998)**) a superfície apresentará grande aderência ao problema proposto, porém demandará um longo tempo de treinamento e, principalmente, um conjunto inicial muito grande de análises, de tal sorte que podem ser mais demoradas do que o próprio processo de otimização.

Por outro lado, um número pequeno de pontos iniciais, apesar de rápido inicialmente, pode resultar em uma superfície de resposta com baixa aderência ao problema real, potencialmente conduzindo o processo de otimização para direções equivocadas, exigindo muitas readaptações dinâmicas da superfície para a obtenção da solução ótima.

A experiência dos testes realizados sugere que devem ser utilizadas ao menos duas variações para cada uma das variáveis do problema.

Alguns autores acreditam que, à exceção das restrições do problema, devem-se deixar as variáveis livres, permitindo quaisquer valores possíveis, deixando a cargo do algoritmo de otimização o direcionamento desses valores aos valores ótimos.

No entanto, a experiência mostrou que, para a grande maioria dos problemas práticos, é seguro e útil estabelecer uma faixa de valores possíveis para cada uma das variáveis do problema.

Embora estabelecer uma região viável fechada signifique aumentar o número de restrições e, portanto, a complexidade do problema, tal medida auxilia questões práticas (como, por exemplo, a normalização das variáveis, indispensável no treinamento das redes neurais) e facilita a convergência do problema em situações onde os parâmetros apresentam ordens de grandeza diferentes. Nesses casos, deve-se ter a prudência de, ao se encontrar o valor ótimo sobre um vértice da região viável, avaliar se o subsequente alargamento do domínio do problema não tende a conduzir o processo a uma nova solução.

Admitindo que o problema deve ter uma região viável fechada, sugere-se a seguinte regra prática para a determinação dos pontos que comporão o conjunto inicial para o treinamento da rede neural:

- normalizar todas as variáveis tal que  $X \in [0,1]$ ;
- para as variáveis cujo efeito na função objetivo é suposto monotônico, deve-se adotar dois valores iniciais possíveis: 0 e 1;
- para as variáveis cujo comportamento não é previsível, deve-se adotar três valores iniciais possíveis: 0, 0.5 e 1;
- o conjunto de pontos que comporão o treinamento inicial da rede neural deverá ser dado pela combinação de todos os possíveis valores de  $X$ , observando as possíveis variações acima descritas.


Evidentemente, o sucesso dessa heurística depende fortemente das características do problema e da experiência do engenheiro em antever quais são as variáveis cujo efeito na função objetivo seja monotônico.<sup>30</sup> De maneira geral, essa regra prática garante um conjunto mínimo de pontos que permitem que uma superfície de resposta tenha, inicialmente, a aderência necessária para garantir a rápida convergência, ao mesmo tempo em que permite a rápida readaptação em direção à solução ótima.

O processo de treinamento utilizado é o clássico método *back-propagation* com apresentação padrão dos pares, apresentado na seção 2.2.3, e cujo detalhes pode ser visto em diversas referências, entre elas, **Caudill (1994)**.

É interessante observar que, como visto na figura 4.2, mesmo com poucos ciclos de treinamento, a rede neural já representa uma superfície que possui claramente a “tendência” da função real, permitindo o correto direcionamento do algoritmo de otimização.

Outra observação importante, vista na figura 4.2, é que a rede neural tende a “suavizar” comportamentos locais, facilitando o trabalho do algoritmo de otimização, evitando que ele fique aprisionado em mínimos locais.

---

<sup>30</sup>  Um exemplo clássico de problemas onde as variáveis têm claro efeito monotônico são os problemas de otimização estrutural. Em geral, nesses problemas, o aumento da quantidade de material conduz à redução das tensões e dos deslocamentos, mas implica no aumento da massa e do custo. Ao passo que o aumento dos vãos livres e cargas resultam no aumento das tensões e dos deslocamentos. Nestes casos, basta conhecer a influência associada a cada variável para que a rede neural possa antever o comportamento estrutural, com relativa precisão, para qualquer combinação de variáveis.

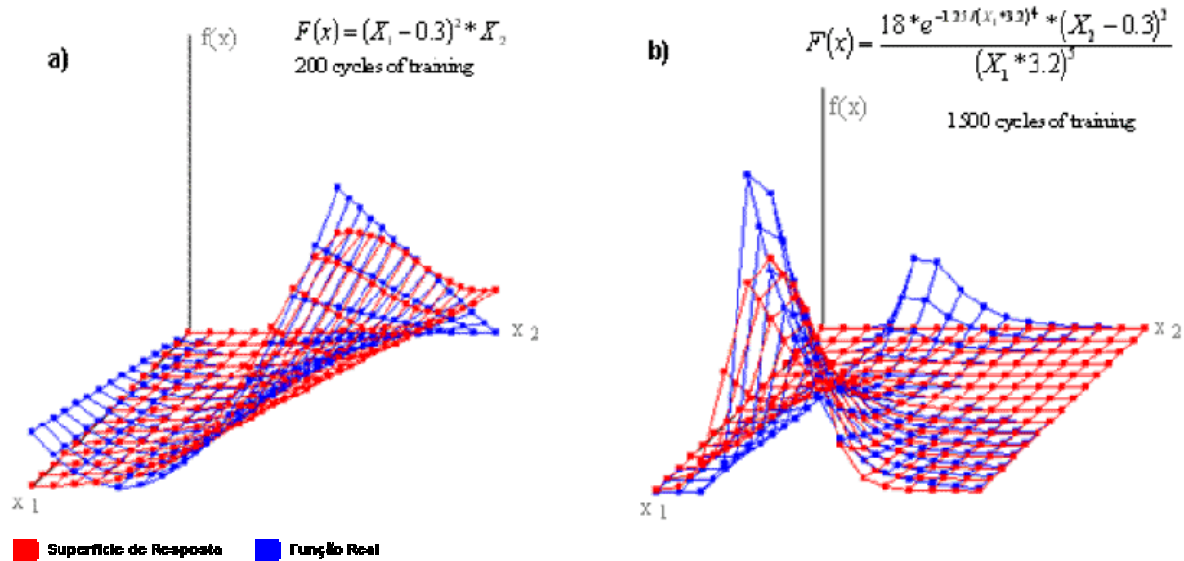


Figura 4.2. Exemplos de superfícies de resposta.

#### 4.1.3 Otimização e adaptação dinâmica

Uma vez construída a superfície de resposta inicial, pode-se iniciar o processo de otimização sobre esta superfície.

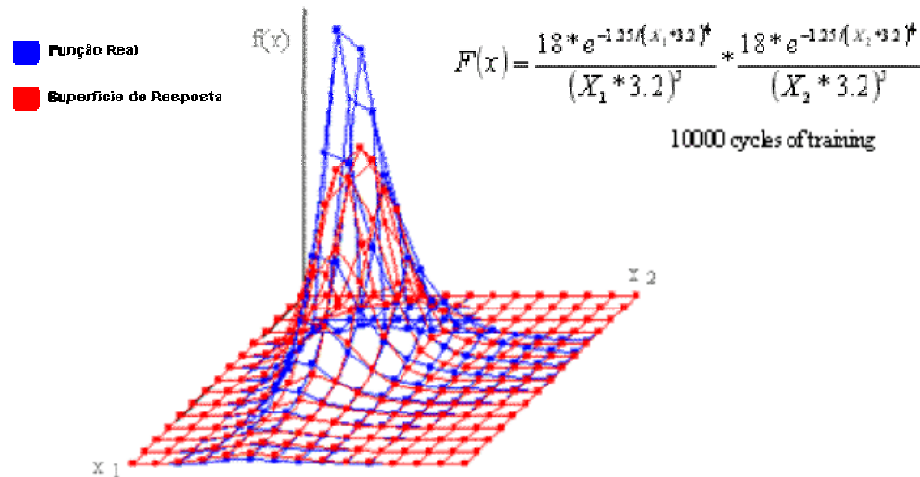
Como descrito por **Hafka (1993)**, o algoritmo utilizado depende das características do problema. Porém, é importante destacar que, diferentemente do processo clássico de otimização, no qual se defini um único conjunto de critérios de convergência, na otimização com adaptação dinâmica, é necessário definir dois conjuntos de critérios de convergência.

O primeiro conjunto deve possuir tolerâncias maiores, suficientes para serem alcançadas rapidamente com poucas iterações do processo. Nesse ponto, o algoritmo de otimização deve parar o processo e analisar as soluções encontradas até o presente momento, comparando os resultados com aqueles obtidos com o auxílio da superfície de resposta.

Caso a diferença entre os resultados seja superior ao limite de tolerância (também conhecido como limite de aderência), a superfície neural é então readaptada e a otimização reiniciada. No entanto, se a diferença for menor que o limite de tolerância, o processo de otimização prossegue até que o segundo limite de convergência seja encontrado, definindo então a solução ótima.

A solução ótima também deve ser analisada e comparada ao resultado obtido sobre a superfície de resposta. Caso o limite de tolerância seja ultrapassado, a superfície de resposta deve ser readaptada e o processo de otimização reiniciado.

Vale destacar que muitas vezes o vetor da solução ótima é corretamente determinado, porém o valor numérico da função objetivo é diferente. Isso porque, como visto na figura 4.3, a superfície neural consegue, com poucos ciclos de treinamento, alcançar a mesma “forma” da superfície real, porém, com valores de máximos e mínimos diferentes. Normalmente o valor encontrado na superfície de resposta é menor (em módulo) do que o valor real.



**Figura 4.3.** Redução do módulo dos valores ótimos na superfície de resposta.

Portanto, nos casos onde a superfície de resposta é utilizada para aproximar a função objetivo, pode-se encontrar o vetor ótimo mesmo que a superfície não apresente o valor correto da função objetivo. Nesses casos os limites de aderência podem ser maiores. Já para os casos onde a superfície de resposta é usada na aproximação de uma restrição, é necessário um limite de aderência muito menor para assegurar a correta determinação do vetor ótimo.

#### 4.1.4 Readaptação dinâmica da superfície de resposta

A readaptação dinâmica da superfície de resposta é a chave para um processo de otimização rápido e ao mesmo tempo coerente.

Como já foi mencionado, o algoritmo de otimização, seja ele qual for, deve possuir um limite de convergência “grosseiro” no qual a aderência da superfície neural deve ser avaliada, readaptando, caso seja necessário, a superfície de resposta.

Porém, diferente do treinamento clássico (visto, por exemplo, no trabalho de **Zhu (1998)**), algumas observações devem ser respeitadas durante a readaptação da superfície:

- os pesos iniciais e limites de ativação não devem ser valores randômicos, mas sim os últimos valores obtidos no treinamento da rede neural, mantendo-se a memória do processo;
- deve-se fixar não apenas um critério de convergência baseado no conjunto dos pares, mas também um critério associado ao par (ou pares) que representa a possível solução ótima. Ou seja, tanto a média dos erros deve estar abaixo do limite de convergência, quanto o erro na região da possível solução ótima também deve ser menor do que o limite de convergência.
- deve-se utilizar durante o treinamento, além do conjunto original de pares, uma repetição do par que representa o vetor ótimo encontrado no processo de otimização. Essa repetição deve ter uma intensidade que não comprometa a aderência da superfície, mas aproxime a superfície neural da superfície real, especialmente na região de interesse (onde possivelmente se encontra a solução ótima).

Uma estratégia interessante é atribuir a cada par um índice de significância, proporcional à distância do par em relação a possível solução ótima. Esse índice é utilizado na determinação da frequência de repetição dos pares durante o treinamento dirigido.

O índice de significância pode ser utilizado no cálculo da média ponderada do erro, a qual pode substituir a utilização de dois critérios de convergência no treinamento dinâmico, visto que os pesos enfatizam a região da possível solução (ou soluções) ótima.

#### 4.1.5 Avaliação do uso de superfícies de resposta

Nesta seção será apresentada uma avaliação do desempenho do uso de superfícies de resposta, as quais, potencialmente, podem acelerar a otimização de problemas.

Tabela 4.1. Avaliação do uso de Superfícies de Resposta

Problema	Elementos Finitos sem Superfície de Resposta (ms)	Elementos Finitos com Superfície de Resposta (ms)	Diferença na Função Objetivo (%)
Viga em Balanço (seção 5.4.1)	2555766	68219	0.0025
Grelha (seção 5.4.2)	21474484	21500	0.0173
Vaso de Pressão (seção 5.4.3)	2309969	40563	0.0105

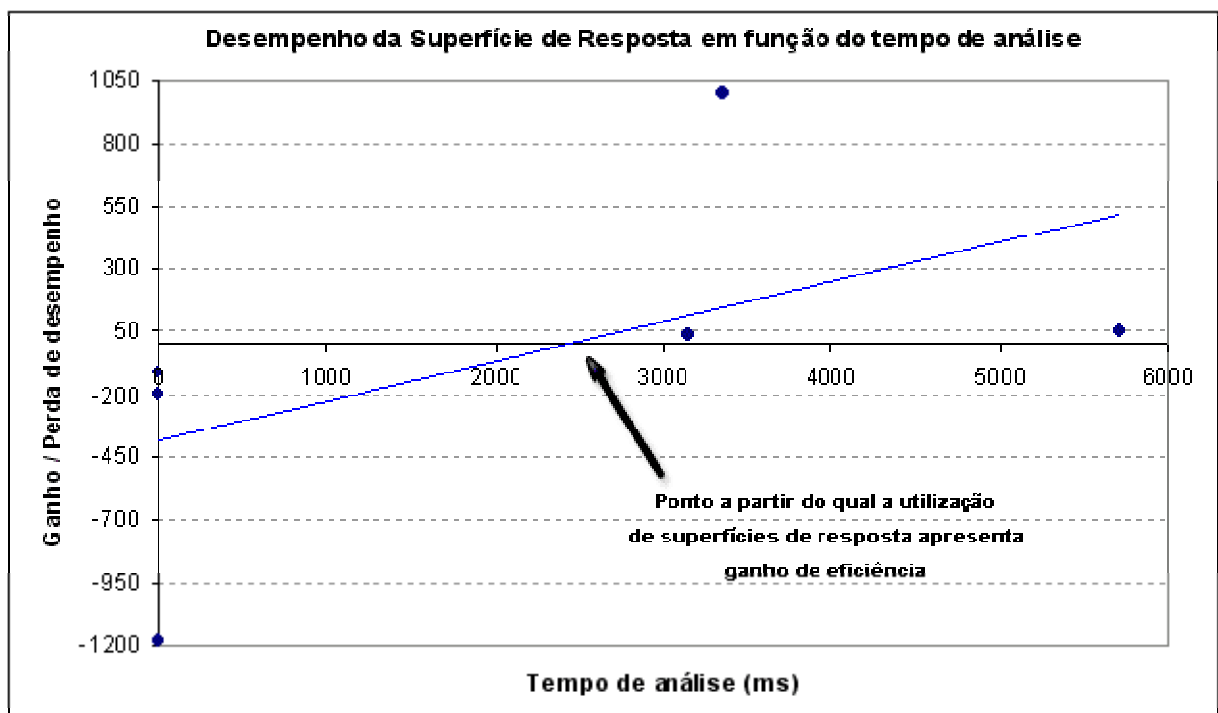
  

Problema	Analítico sem Superfície de Resposta (ms)	Analítico com Superfície de Resposta (ms)	Diferença na Função Objetivo (%)
Viga em Balanço (seção 5.4.1)	344	66906	0.015
Grelha Simples (seção 5.2.3)	391	41656	0.002
Rosenbrock (seção 5.2.1)	328	387125	0.013

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

Os resultados apresentados permitem concluir que a eficiência da utilização de superfícies de resposta é fortemente influenciada pelo tempo de cada análise do problema. Dessa forma, problemas que utilizam análises numéricas, apresentam alto ganho de eficiência com o uso de superfícies de resposta. Por outro lado, problemas compostos apenas de expressões analíticas apresentam redução considerável de performance com o uso de superfícies de resposta.

Na figura 4.4 é apresentada uma correlação entre o tempo de cada análise do problema, em milissegundos, e a eficiência da utilização de superfícies de resposta. Nesta correlação, valores negativos indicam que o tempo de processo com superfícies de resposta é superior ao tempo de processo sem superfícies de resposta.



**Figura 4.4.** Desempenho da superfície de resposta em função do tempo de cada análise.

A alta dispersão dos pontos existentes na figura 4.4 é justificada pela ocorrência de comportamentos anômalos e viciados (os quais serão discutidos na seção 4.3) e pela aleatoriedade do procedimento de treinamento e otimização. Por essa razão, é impossível determinar precisamente o tempo de cada análise do problema a partir do qual a utilização de superfícies de resposta representa ganho de desempenho. No entanto, pode-se estabelecer como regra prática, que problemas nos quais cada análise demore mais do que 3 segundos, devem ser otimizados com o auxílio de superfícies de resposta.

#### **4.1.6 O problema de interpolação de gráficos e diagramas**

Para a representação de gráficos e diagramas, inicialmente tabela-se uma quantidade significativa de dados. A quantidade de pontos iniciais depende do comportamento da função a ser representada. Funções com curvaturas suaves exigem uma quantidade pequena de pontos, enquanto funções caóticas demandam um número muito alto de pontos para serem bem descritas.

Os pontos tabelados são chamados pares de treinamento e devem ser normalizados para que todos os valores estejam entre 0 e 1. Podem-se determinar limites globais ou então normalizar cada parâmetro ou atributo. Usualmente, normalizar cada variável resulta em uma precisão maior, principalmente quando elas possuem ordens de grandeza diferentes.

Alcançado o limite de erro (ou o número de ciclos de treinamento) arbitrado pode-se escrever a “expressão neural” utilizando os limites de ativação e os pesos das ligações neurais. Esta expressão pode ser utilizada em qualquer ambiente de projeto e permitirá a rápida avaliação de qualquer combinação de variáveis para a obtenção dos valores de saída. Deve-se ter o cuidado de transpor também os limites de normalização, pois a “expressão neural” só poderá ser utilizada com os valores de entrada normalizados e só terá significado físico ao converter a saída normalizada da rede (entre 0 e 1) para o valor real.

#### **4.1.7 Conclusão**

O presente trabalho apresenta uma metodologia simples para a representação de superfícies de resposta utilizando redes neurais artificiais, que podem facilmente aproximar superfícies e funções descritas em gráficos e diagramas, e cuja capacidade nativa de se adaptarem a cada novo evento dirigido, se mostra ideal como superfície base sobre a qual se realiza o processo de otimização.

A principal contribuição, que distância esta pesquisa dos demais trabalhos publicados sobre o assunto, é o procedimento de readaptação dinâmica das superfícies de resposta. Esse procedimento potencialmente acelera a convergência do processo de otimização, pois concentra o treinamento em uma região próxima da localização da solução (ou soluções) ótima, garantindo a convergência coerente do processo e preservando a memória dos ciclos anteriores de otimização.

Os resultados apresentados na seção 4.1.5 demonstram que a utilização de superfícies de resposta aumenta a eficiência dos problemas nos quais a avaliação de uma solução demora mais do que alguns segundos. Como regra prática tem-se que problemas

que utilizem apenas equações e expressões analíticas não devem ser otimizados com o auxílio de superfícies de resposta, ao passo que, para todos os outros tipos de problemas, o uso de superfícies de resposta representa ganho significativo de performance.

É importante destacar que o treinamento dinâmico não deve ser utilizado na otimização de problemas multidisciplinares, pois, nesses casos, toda a região viável é igualmente importante.

## 4.2 Otimização utilizando ambiente distribuído pela Internet <sup>31</sup>

Em 2003, os pesquisadores do IRCCYN publicaram um importante artigo (**Giassi (2003)**) sobre as vantagens da utilização de ambientes distribuídos em projetos de engenharia. Como discutido na seção 2.3.3, esse artigo é um dos muitos trabalhos que, motivados pelo desenvolvimento dos recursos computacionais e a redução de preço apresentada por esses dispositivos nos últimos anos, propõem o desenvolvimento de projetos em ambientes colaborativos e a utilização de programação distribuída para aumentar a eficiência de processos de otimização.

O ambiente para a otimização de problemas multidisciplinares com processamento distribuído por passagem de mensagem pela Internet foi apresentado na seção 3.4. Nesta seção serão discutidas as principais contribuições deste trabalho


O conceito proposto é viabilizar a otimização de sistemas complexos de engenharia, nos quais, o tempo demandado em cada iteração do processo, inviabiliza a abordagem linear do problema.

### 4.2.1 O ambiente desenvolvido

Conforme discutido na seção 3.4, o ambiente desenvolvido é composto de módulos que se comunicam por passagem de mensagem pela Internet, permitindo assim, a utilização de recursos geograficamente dispersos.

A primeira contribuição importante é que, diferentemente da abordagem proposta por **Giassi (2004)**, o ambiente desenvolvido atua em tempo real, pois, enquanto o trabalho de Giassi realiza a troca de mensagens utilizando o protocolo *POP3/SMTP* para o envio de *emails*, o ambiente desenvolvido utiliza conexões *TCP/IP* tendo como base o protocolo *MPI*.

---

<sup>31</sup>  Os conceitos discutidos nesta seção foram aceitos na *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'07)*, Las Vegas, Nevada, USA, June 25-28, 2007. Porém devido a impossibilidade do autor em obter um visto de entrada nos EUA antes da conferência, deixaram de ser apresentados.



A abordagem de Giassi provavelmente deve-se as questões de segurança discutidas na seção 3.4.4.3. Por outro lado, o ambiente desenvolvido utiliza um servidor de infraestrutura independente, que permite a conexão dos módulos sem que esses se comuniquem entre si, evitando assim possíveis restrições de segurança.

A segunda contribuição importante é o fato de que, embora o trabalho de Giassi utilize recursos geograficamente dispersos, não o faz de maneira distribuída. No ambiente desenvolvido, novos módulos de análise podem ser introduzidos ao ambiente a qualquer momento do processo, permitindo inclusive, incluir novos clientes de otimização e assim executar mais de um processo de otimização simultaneamente.

Para que o ambiente de processamento distribuído funcione corretamente, o servidor de infra-estrutura deve sincronizar as solicitações recebidas e realizar o balanceamento das tarefas entre os módulos disponíveis. O balanceamento de *Grids* computacionais é um assunto amplamente discutido em fóruns internacionais e sua implementação representa um diferencial deste trabalho.

#### **4.2.2 Otimização e sincronização**

Embora o funcionamento padrão do ambiente desenvolvido tenha sido discutido na seção 3.4.4, pode-se optar por um procedimento diferente, no qual o ambiente distribuído é melhor aproveitado.

No procedimento convencional, um limite de erro (ou número máximo de ciclos) é definido para o treinamento das redes neurais, a partir do qual o treinamento é interrompido e as informações são retornadas ao cliente de otimização.

Para aproveitar melhor o potencial da programação distribuída, pode-se continuar o treinamento das redes neurais em paralelo com o algoritmo de otimização. Nesse caso, o treinamento constante tende a produzir redes neurais com uma melhor representação do problema. Por outro lado, o treinamento constante impede que mais de um processo de otimização ocorra simultaneamente, pois ocupa constantemente os módulos destinados ao treinamento de redes neurais.

#### **4.2.3 Considerações sobre a eficiência do ambiente distribuído**

Embora, como visto na seção 2.3.3.3, um *speedup* acima do linear seja possível em alguns problemas particulares, isto não ocorre em problemas de otimização, pois não existe possibilidade do processo ser encerrado prematuramente.

Um dos aspectos mais relevantes na implementação da programação distribuída é que, conforme discutido na seção 2.3.3.3, existe um limite para a paralelização do processo, a partir do qual o custo de paralelização se torna superior ao ganho de eficiência no processo.

O ambiente desenvolvido tem um alto custo de sincronização, pois a conexão pela Internet é extremamente lenta, se comparada a conexão física existente em *clusters* e supercomputadores. Em função disso, a experiência mostrou que o ambiente desenvolvido apresenta ganho de eficiência apenas nos problemas em que o tempo de processamento é superior ao tempo de comunicação.

A eficiência da programação distribuída em problemas de otimização depende fortemente do tipo de algoritmo utilizado no processo. Algoritmos que utilizam processos de busca (MBA e Newton, por exemplo) apresentam baixo ganho de desempenho quando comparados aos algoritmos do tipo populacional (Genéticos e Vértices, por exemplo), pois nesse caso existem diversas análises independentes que podem ser realizadas em paralelo.

O número de processadores que maximiza a eficiência do processo é determinado pelas características do problema e do algoritmo de otimização utilizado. Um número maior de processadores é inútil, pois permanecem ociosos durante o processo.

Para problemas com superfícies de resposta, o número ideal de processadores para o treinamento neural corresponde ao número de superfícies definidas no problema. Por outro lado, para a construção das superfícies é necessário um conjunto de avaliações iniciais, conforme discutido na seção 4.1.2. Nesse caso, o número ideal de processadores corresponde ao número de pares do conjunto de avaliações iniciais.

Por fim, para a avaliação de soluções, o número ideal de processadores corresponde ao número de análises feitas a cada iteração do processo de otimização. Para algoritmos genéticos, esse número é igual ao tamanho estabelecido para a população, enquanto para o algoritmo Vértices, corresponde ao número de vértices da região viável. Algoritmos de busca, como o MBA, realizam apenas uma avaliação por ciclo e por isso não devem ser paralelizados.

#### **4.2.3.1 Avaliação da utilização do ambiente distribuído**

Nesta seção será apresentada uma avaliação do desempenho do ambiente distribuído que potencialmente pode acelerar a otimização de problemas. Em todos os exemplos apresentados, o ambiente será constituído de 2 computadores destinados a

realização das análises, bem como 1 computador que servirá de servidor de infra-estrutura e 1 computador no qual o módulo de otimização é executado.

Na tabela 4.2 são apresentados os resultados do processo de otimização do exemplo discutido na seção 5.4.1 utilizando o Método dos Elementos Finitos para a estimativa das tensões e deformações da estrutura.

Tabela 4.2. Avaliação do uso do ambiente distribuído, abordagem com M.E.F.\*

Algoritmo	Vértices				MBA			
	Não		Sim		Não		Sim	
Superfície de resposta								
Ambiente distribuído	Não	Sim	Não	Sim	Não	Sim	Não	Sim
Avaliações para geração das superfícies	0	0	4	4	0	0	4	4
Avaliações com M.E.F.	809	809	3	3	1745	2134	3	3
Avaliações com superfície de resposta	0	0	3233	3233	0	0	3100	2576
Treinamento de redes neurais	0	0	6	6	0	0	7	7
Tempo total do processo (ms)	2555766	1785437	68219	2790229	5512746	9406099	84534	3318248

\* Todos os resultados foram obtidos sobre o mesmo índice de performance e a mesma precisão.

Na tabela 4.3 são apresentados os resultados do processo de otimização do exemplo discutido na seção 5.4.1, porém utilizando expressões analíticas para a estimativa das tensões e deformações da estrutura.

Tabela 4.3. Avaliação do uso do ambiente distribuído, abordagem analítica.\*

Algoritmo	Vértices				MBA			
	Não		Sim		Não		Sim	
Superfície de resposta								
Ambiente distribuído	Não	Sim	Não	Sim	Não	Sim	Não	Sim
Avaliações para geração das superfícies	0		4	4	0		4	4
Avaliações com M.E.F.	809		3	3	1631		4	3
Avaliações com superfície de resposta	0	**	3240	3240	0	**	3042	3420
Treinamento de redes neurais	0		7	7	0		8	7
Tempo total do processo (ms)	343		66906	2777785	906		81312	4359329

\* Todos os resultados foram obtidos sobre o mesmo índice de performance e a mesma precisão.

\*\* O ambiente desenvolvido não permite distribuir a resolução de expressões analíticas.

A partir desses resultados, é possível estabelecer algumas conclusões relevantes, que serão apresentadas a seguir.

Tabela 4.4. Duração média de cada etapa do processo.

Etapa do processo	Tempo médio de duração (ms)
Avaliação utilizando o Método dos Elementos Finitos	3159
Avaliação utilizando expressões analíticas	0.42
Treinamento das redes neurais, independe do número de ciclos	66563
Envio das variáveis e retorno de soluções por meio de conexão da Internet	1255

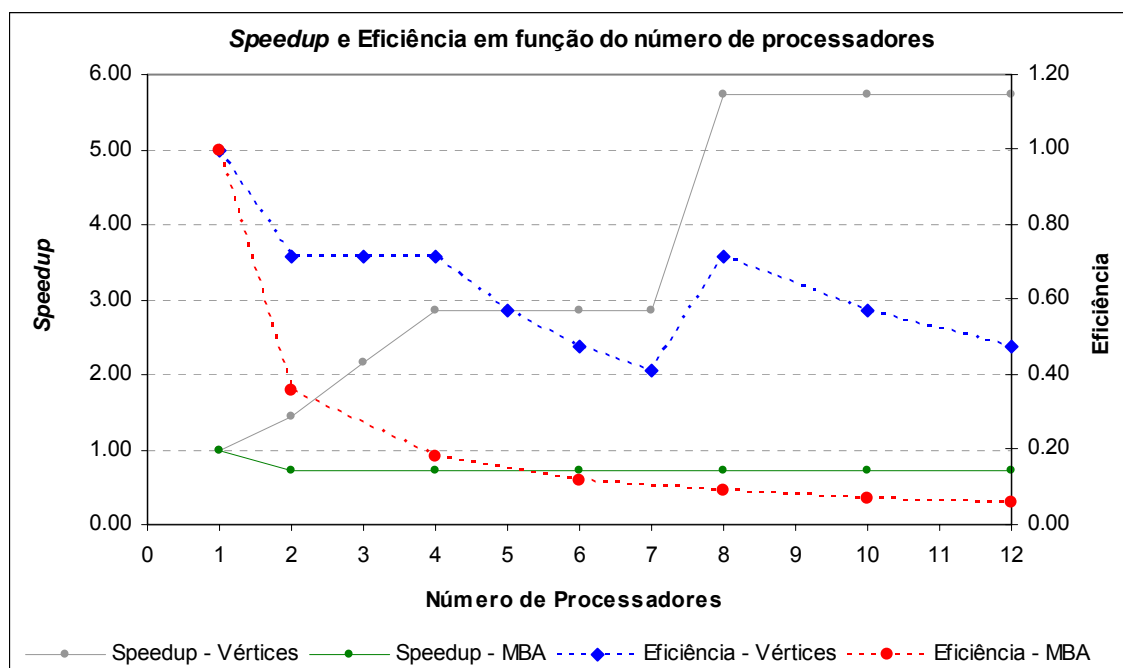
Tabela 4.5. *Speedup* e eficiência dos algoritmos de otimização.\*

Algoritmo	Vértices	MBA
Número de computadores utilizados **	2	2
Número de pontos avaliados a cada iteração do processo de otimização	8	1
<i>Speedup</i> do processo utilizando o Método dos Elementos Finitos	1.43	0.59
Eficiência do processo utilizando o Método dos Elementos Finitos	0.72	0.29
<i>Seedup</i> do processo utilizando análises neurais (Superfície de Resposta)	0.02	0.03
Eficiência do processo utilizando análises neurais (Superfície de Resposta)	0.01	0.01

\* Todos os resultados foram obtidos sobre o mesmo índice de performance e a mesma precisão.

\*\* Foram considerados apenas os computadores destinados às análises.

Para demonstrar a diferença de *speedup* alcançado entre os algoritmos de busca (como o MBA) e os algoritmos populacionais (como o Vértices e o Genético) utilizando ambientes distribuídos, um estudo da projeção de eficiência em função do número de processadores utilizados é apresentado na figura 4.5. Para este estudo foi escolhido o exemplo discutido na seção 5.4.1 com a abordagem baseada no Método dos Elementos Finitos.



**Figura 4.5.** Projeção do *Speedup* e eficiência dos algoritmos de otimização utilizando M.E.F.

O número de avaliação do algoritmo MBA é aleatório e depende do complexo de pontos escolhidos no início do processo. Por essa razão, utilizou-se um valor médio constante para o cálculo de *speedup* e eficiência, a fim de evitar que os dados fossem distorcidos por essa aleatoriedade.

Pode-se observar que, para ambientes homogêneos (todas as máquinas possuem a mesma capacidade computacional), o ganho de *speedup* não ocorre de forma constante,

mas sim em intervalos discretos, que são definidos em função das características do problema e do algoritmo de otimização utilizado.

No problema em questão, a cada iteração do algoritmo Vértices de otimização, 8 pontos precisam ser analisados. Neste caso, o ganho de eficiência ocorre para as seguintes quantidades de processadores: 1, 2, 3, 4 e 8. Ao se utilizar, por exemplo, 4 processadores, 4 análises podem ser feitas simultaneamente, restando outras 4 análises para um segundo lote completo. Essa situação é idêntica à utilização de 5 processadores, com a diferença de que, neste caso, durante o segundo lote de análises, 2 processadores permanecem ociosos. Pela mesma razão, não existe ganho de *speedup* ao se utilizar mais de 8 processadores.

Por outro lado, o algoritmo MBA executa apenas 1 análise por iteração do processo de otimização e, portanto, não existe ganho ao se utilizar mais de 1 processador, exceto na geração do complexo inicial.

Ao contrário do algoritmo MBA, o algoritmo Vértices não possui aleatoriedade, logo, para um limite de erro fixo, o número de iterações é constante, sendo ideal para a realização dos estudos de eficiência, como a importante relação apresentada na figura 4.6.

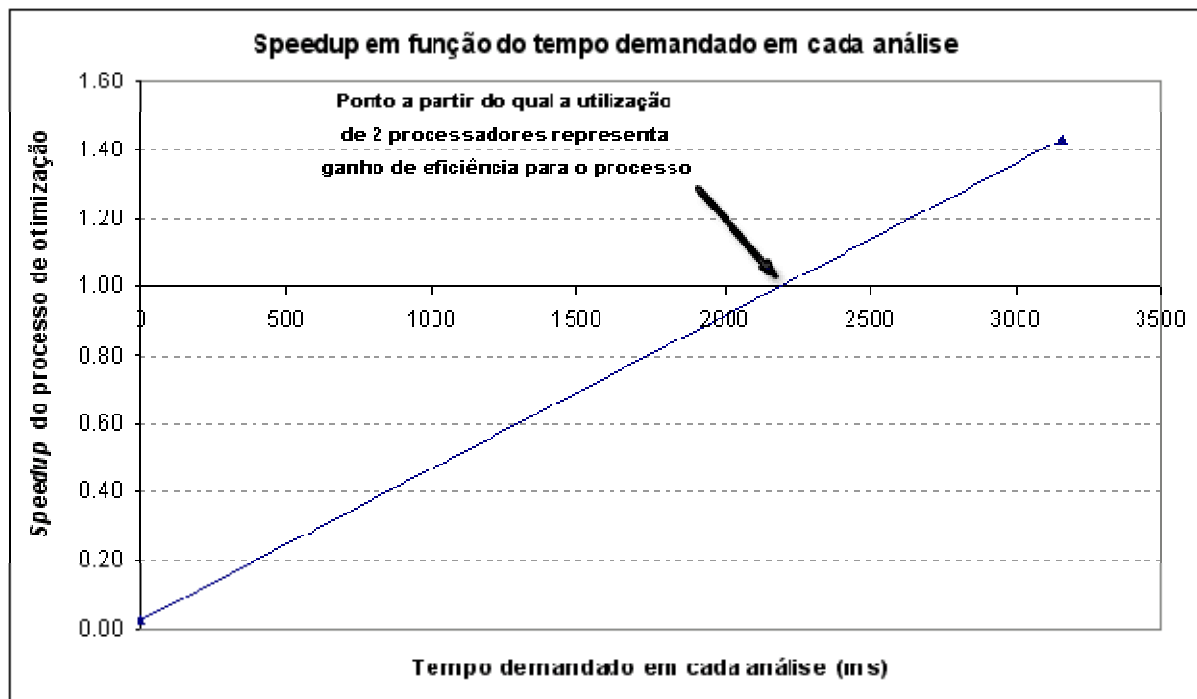


Figura 4.6. *Speedup* em função do tempo de análise. Algoritmo Vértices com 2 processadores.

Como regra prática pode-se dizer que problemas nos quais a análise de uma solução demore mais do que 3 segundos e cujo algoritmo de otimização realize mais do que 1 avaliação por iteração, apresentam ganho de eficiência ao se utilizar processamento distribuído.

#### 4.2.4 Estimativa do tempo total de processo

Os resultados e considerações apresentados nesta seção permitem estimar o tempo total de um processo de otimização em ambiente distribuído por meio da equação:

$$\begin{aligned}
 \text{Tempo total} &= T_{\text{avaliações iniciais}} + T_{\text{treinamento}} + T_{\text{avaliações verificação}} + T_{\text{análises de otimização}} \\
 T_{\text{avaliações iniciais}} &= \text{Arredonda para cima} \left( \frac{N_{\text{avaliações iniciais}}}{N_{\text{processadores}}} \right) * (T_{\text{cada análise}} + T_{\text{comunicação}}) \\
 T_{\text{análises de otimização}} &= \text{Arredonda para cima} \left( \frac{N_{\text{avaliações por iteração}}}{N_{\text{processadores}}} \right) * (T_{\text{cada análise}} + T_{\text{comunicação}}) \\
 T_{\text{avaliações verificação}} &= \text{Arredonda para cima} \left( \frac{N_{\text{soluções por ciclo otimização}}}{N_{\text{processadores}}} \right) * (T_{\text{cada análise}} + T_{\text{comunicação}})
 \end{aligned} \tag{4.1}$$

Onde:

- $T_{\text{cada análise}}$  corresponde ao tempo demandado para a avaliação de uma única solução do problema;
- $T_{\text{comunicação}}$  corresponde ao tempo demandado para estabelecer uma conexão pela Internet. Pode variar em função do tráfego da rede e da quantidade de informações a serem transmitidas;
- $T_{\text{treinamento}}$  corresponde ao tempo de treinamento e readaptação das redes neurais que representam as superfícies de resposta. Varia conforme a topologia utilizada nas redes neurais e a complexidade do problema;
- $T_{\text{avaliações iniciais}}$  corresponde ao tempo demandado pelas avaliações responsáveis pela geração da superfície de resposta;
- $T_{\text{avaliações verificação}}$  corresponde ao tempo demandado pelas avaliações responsáveis pela verificação da aderência das superfícies de resposta. Em problemas mono-objetivo é gerada apenas uma única solução por ciclo de otimização. Por outro lado, em problemas multiobjetivo, o número de soluções depende do algoritmo utilizado. Para algoritmos genéticos, correspondem ao tamanho da população;
- $T_{\text{análises de otimização}}$  corresponde ao tempo demandado pelas avaliações responsáveis pela condução do processo de otimização. Em problemas sem a utilização de superfícies de resposta corresponde ao  $T_{\text{cada análise}}$ , enquanto para problemas com superfícies de resposta corresponde ao tempo de avaliação utilizando as superfícies de resposta.

Para problemas que não utilizem superfícies de resposta, a equação 4.1 por ser reduzida a:

$$Tempo\ total = Arredonda\ para\ cima\left(\frac{N_{avaliações\ por\ iteração}}{N_{processadores}}\right) * (T_{cada\ análise} + T_{comunicação}) \quad (4.2)$$

Onde:

- $T_{cada\ análise}$  corresponde ao tempo demandado para a avaliação de uma única solução do problema;
- $T_{comunicação}$  corresponde ao tempo demandado para estabelecer uma conexão pela Internet. Pode variar em função do tráfego da rede e da quantidade de informações a serem transmitidas.

#### 4.2.5 Conclusão

O principal resultado apresentado é um ambiente de otimização que incorpora diferentes conceitos como paralelismo, distribuição de processamento em tempo real pela Internet, passagem de mensagem utilizando o protocolo MPI, servidor de infra-estrutura com balanceamento de carga, memória não compartilhada e sincronização de processos simultâneos.

A experiência mostrou que o ganho de eficiência para processos de otimização utilizando o ambiente distribuído é proporcional ao número de processadores utilizados, desde que esse seja no máximo igual ao número de avaliações realizadas por iteração do processo de otimização.

Os processos de construção e treinamento de redes neurais apresentam alto ganho de eficiência quando paralelizados, desde de que, para o treinamento, o número de processadores seja no máximo igual ao numero de superfícies de resposta do problema, enquanto para a construção, o número de processadores deve ser no máximo igual ao número de pontos iniciais.

Outra importante conclusão é o fato de que a utilização do ambiente distribuído oferece ganho de eficiência nas situações em que o tempo de análise computacional é superior ao tempo de comunicação entre os módulos.

### 4.3 Superfícies anômalas e viciadas, tratamento com regiões tabus

Como visto na seção 2.2, a utilização de superfícies de resposta é uma solução conhecida para acelerar processos de otimização em que as análises dos pontos exigem grande esforço computacional.

Embora o uso de tal técnica resulte significativo ganho de desempenho em muitos tipos de problemas (como demonstrado na seção 4.1.5), a experiência mostrou que a utilização de superfícies de resposta pode, em alguns casos, resultar em superfícies anômalas ou viciadas, as quais conduziriam o algoritmo de otimização à resultados distantes da solução ótima.

Nesta seção este problema será descrito juntamente com as abordagens tradicionalmente utilizadas para o tratamento de superfícies de resposta anômalas e viciadas. Por fim, será apresentada uma abordagem inovadora para o problema, seguida de alguns exemplos que demonstrarão a eficiência de tal abordagem.

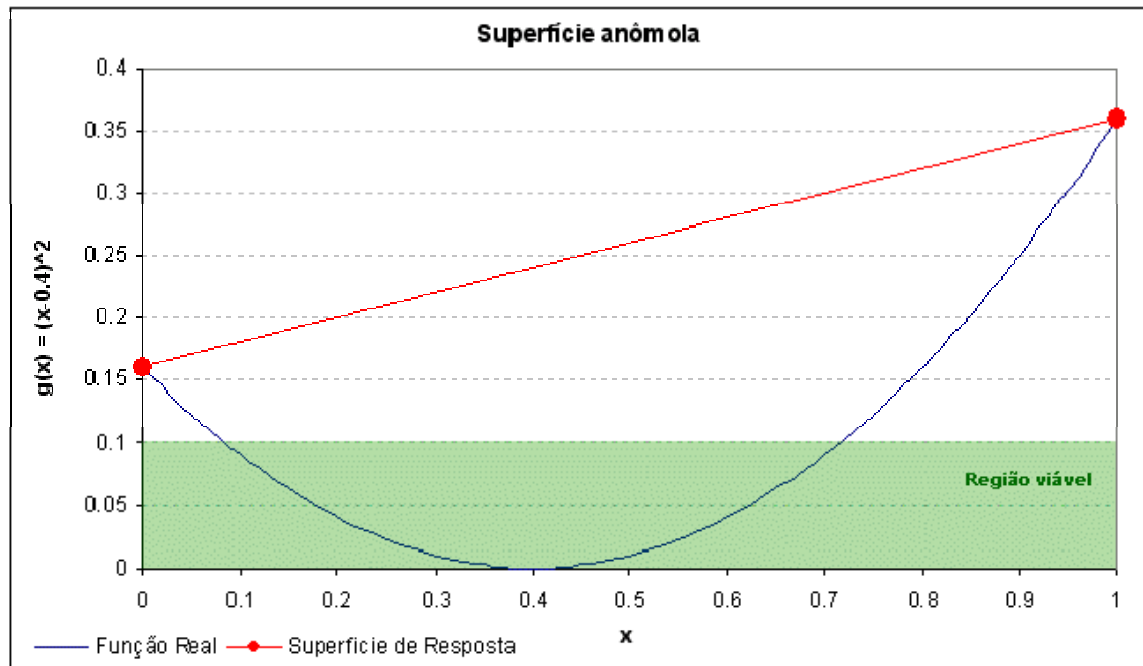
#### 4.3.1 Superfícies anômalas de resposta

O problema existente consiste no fato de que a escolha dos indivíduos que constituirão a superfície de resposta inicial pode determinar a representação de uma superfície anômala de resposta. As superfícies anômalas ocorrem quando, ao representarem restrições do problema, todos os pontos escolhidos são inviáveis.

Essa característica produz uma superfície de resposta totalmente localizada na região inviável do problema a qual, por causa do método de penalização (descrito na seção 2.1.6) utilizado em problemas com restrições, conduz o processo de otimização para uma região distante da solução ótima.

Para exemplificar esse comportamento, considere a função  $g(x) = (x-0.4)^2$  como uma restrição de um problema qualquer de otimização. Considere que esta função deve ser menor do que 0.1 para que a solução seja considerada viável. A função  $g(x)$  possui claramente uma região viável e outra região não viável, que podem ser vistas na figura 4.7.





**Figura 4.7.** Superfície de resposta anômala.

Ao construir a superfície de resposta, representada em vermelho na figura 4.7, todas as avaliações feitas sobre a superfície de respostas resultam em um valor  $g(x)$  inviável. Não importa o valor da função objetivo, se todos os pontos forem inviáveis, a solução encontrada não será a solução ótima, mas sim aquela com o menor valor em módulo de  $g(x)-0.1$ , nesse caso o ponto  $x = 0$ .

O leitor precipitado pode sugerir que a inclusão de um ponto médio entre os valores iniciais seria suficiente para evitar a superfície anômala da função descrita na figura 4.7. No entanto, é fácil demonstrar que, não importa a quantidade de pontos utilizados no treinamento inicial da superfície de resposta, sempre existirá a possibilidade de que a superfície construída seja anômala.

Para exemplificar essa condição, considere a função  $g(x) = \cos(X)$  definida no intervalo  $[0, 4\pi]$ , mostrada na figura 4.8. Considere que esta função deve ser menor do que 0.5 para que a solução seja considerada viável. Ao se utilizar 3 pontos iniciais, a superfície de resposta, representada em vermelho, apresenta comportamento anômalo e, portanto, inviabilizará um processo de otimização coerente.

É interessante observar que, assim como o exemplo da figura 4.7, em função do método de penalização, a solução encontrada será aquela com o menor valor em módulo de  $g(x)-0.5$ . No entanto, devido a particularidade do problema, todos os pontos da superfície de resposta possuem mesmo valor e o processo será conduzido pelas equações restantes do problema.

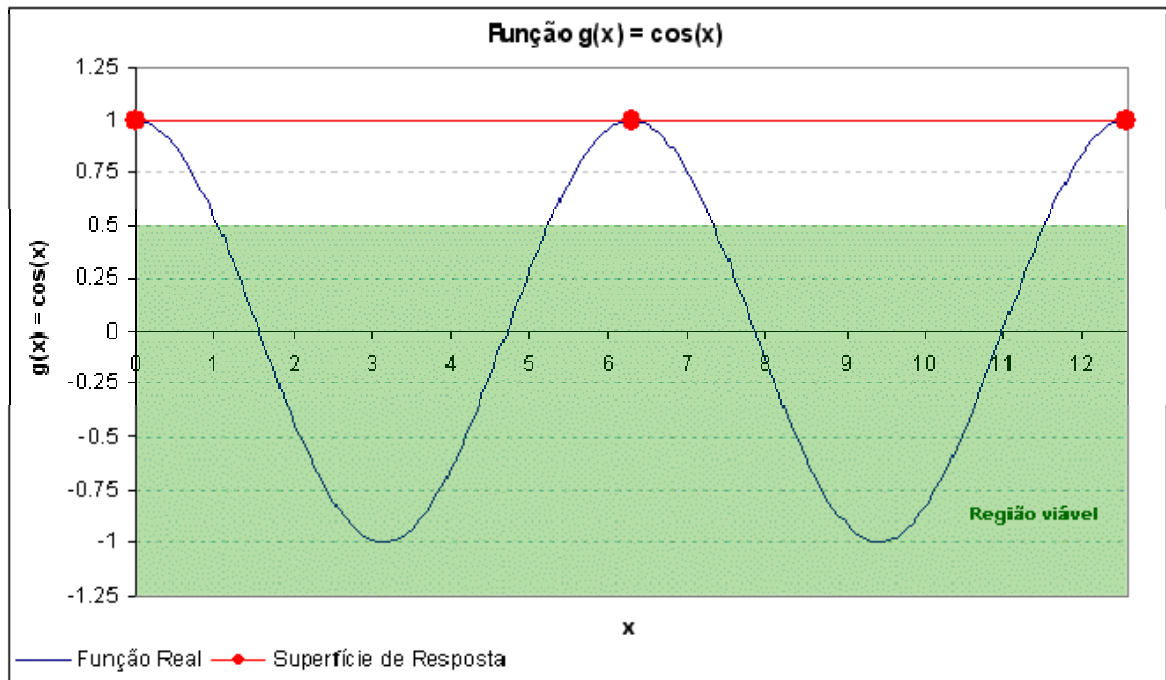


Figura 4.8. Superfície anômala com 3 pontos iniciais.

#### 4.3.2 Superfícies viciadas de resposta

As superfícies viciadas ocorrem quando, ao representarem funções objetivo do problema, ao final do processo de otimização, a solução encontrada corresponde a um dos pontos utilizados na geração da superfície de resposta. Para exemplificar esse comportamento, considere a função  $f(x) = (x-0.2)^2$ , mostrada na figura 4.9, como uma função objetivo de um problema de otimização.

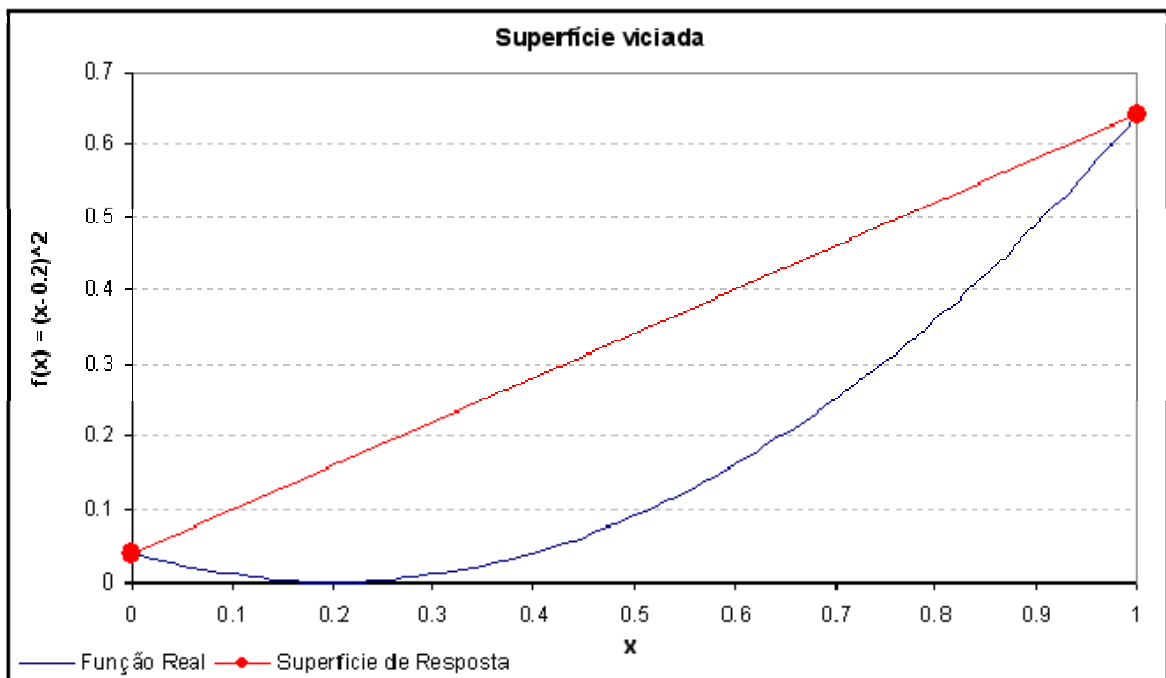


Figura 4.9. Superfície de resposta viciada.

Ao construir a superfície de resposta, representada em vermelho na figura 4.9, o processo de otimização é conduzido ao ponto  $x = 0$ , pois esse é o ponto de mínimo da superfície de resposta, que, no entanto, está distante do ponto  $x = 0.2$ , mínimo da função real do problema.

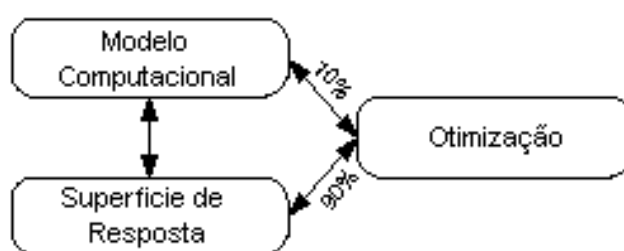
Após determinar a solução, o algoritmo de otimização deve verificar a aderência da superfície de resposta ao problema real. O ponto  $x = 0$  foi utilizado na geração da superfície de resposta e, por esta razão, o erro associado a esse ponto está abaixo do limite de aderência, não exigindo nenhuma readaptação da superfície de resposta.

A exemplo das superfícies anômalas de resposta, não importa a quantidade de pontos utilizados na geração da superfície, essa sempre poderá apresentar um comportamento viciado durante a otimização.

#### 4.3.3 Abordagens tradicionais para superfícies anômalas e viciadas

Existem duas abordagens para evitar que o processo de otimização seja governado por uma superfície de resposta anômala ou viciada.

A primeira abordagem, amplamente utilizada, é conduzir o processo de otimização realizando parte das análises com o auxílio da superfície de resposta e parte das análises sem a superfície de resposta. Dessa maneira, as análises realizadas sem a superfície de resposta permitem direcionar corretamente o processo de otimização. Na figura 4.10 é mostrado um esquema dessa abordagem, no qual 90% das análises durante o processo de otimização são realizadas com o uso de superfícies de resposta.



**Figura 4.10.** Abordagem percentual para a utilização de superfícies de resposta.

A segunda abordagem empregada, implica em selecionar aleatoriamente os pontos utilizados no treinamento da superfície inicial de resposta até que ao menos um dos pontos seja viável, evitando assim a geração de superfícies anômalas.

Porém, ambas as abordagens têm a mesma fraqueza: um grande número de avaliações do problema sem a utilização de superfícies de respostas, que, dependendo do

tempo demandado por cada uma das análises, podem reduzir consideravelmente a eficiência do processo de otimização.

#### 4.3.4 Abordagem baseada em regiões tabus

A abordagem proposta é utilizar regiões tabus para limitar a movimentação do algoritmo de otimização.

O procedimento proposto consiste em, transcorrido o processo tradicional de geração das superfícies iniciais de resposta e a otimização, verificar, a partir das soluções encontradas, se as superfícies utilizadas apresentam comportamento anômalo ou viciado.

A verificação do comportamento viciado é realizada comparando-se as respostas encontradas no processo de otimização com os pontos utilizados para a geração das superfícies de resposta. Enquanto o comportamento anômalo é verificado analisando-se se, dentre o conjunto de geração da superfície, existe algum ponto viável.

Se superfícies de resposta anômalas ou viciadas forem detectadas, uma região tabu esférica é construída em torno das soluções encontradas, conforme mostrado na figura 4.11, e o processo de otimização é reiniciado.

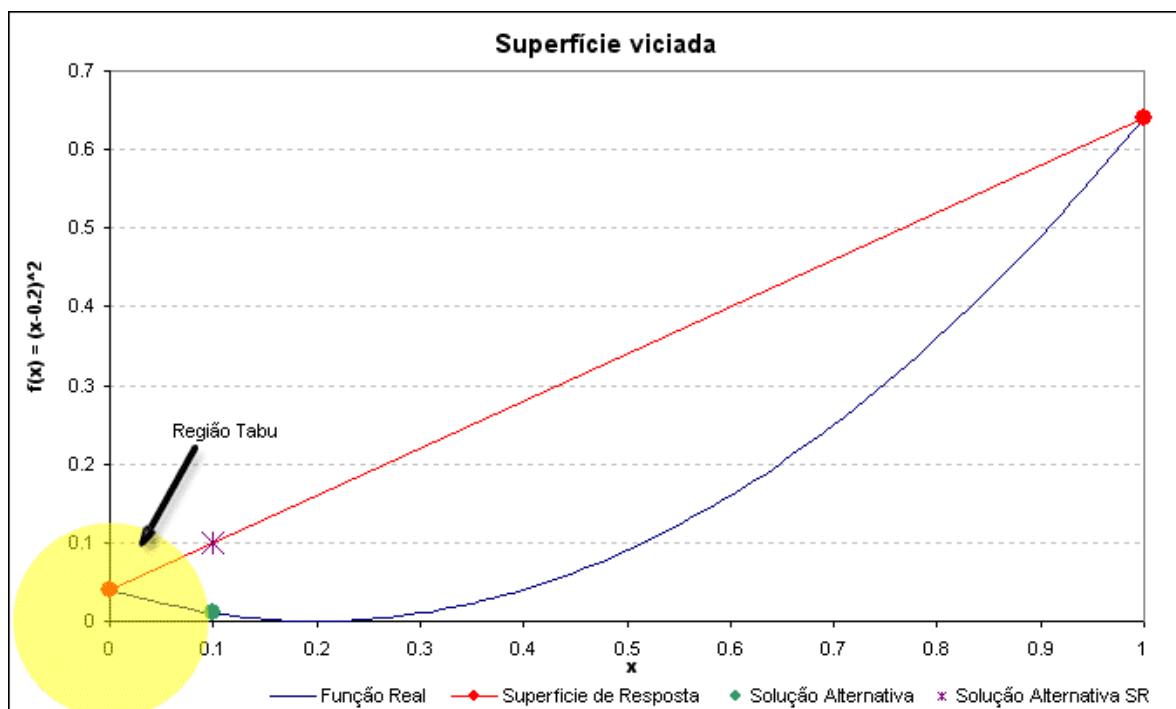


Figura 4.11. Região tabu utilizada no tratamento de superfície viciada.

As regiões tabus esféricas impedem o comportamento anômalo e viciado, conduzindo o processo a novas soluções. Essas novas soluções, conforme mostrado na

figura 4.11, apresentam erros superiores ao limite de aderência (pois não foram utilizadas durante o processo de geração) e por essa razão, conduzem a uma readaptação das superfícies, que se aproximam das superfícies reais do problema.

Após readaptarem as superfícies de resposta, as regiões tabus são removidas e o processo de otimização é reiniciado. As novas soluções são verificadas quanto ao comportamento anômalo e viciado das superfícies. Embora iterativo, esse processo converge rapidamente.

O raio da região tabu pode afetar a condução do processo de otimização. Um valor muito elevado de raio pode fazer o algoritmo oscilar em torno de um ponto ótimo; ao passo que um raio muito pequeno provoca sucessivas readaptações da superfície. A experiência mostrou que um valor normalizado de 0.1 (como aquele mostrado na figura 4.11) permite, para a maioria dos problemas, uma condução rápida e coerente do processo de otimização.

O comportamento anômalo, na verdade, poderia ser verificado antes do processo de otimização. No entanto, sem o processo de otimização, não seria possível posicionar as regiões tabus, que seriam introduzidas aleatoriamente, diminuindo a eficiência do processo. Neste caso, uma estratégia interessante é realizar o primeiro ciclo de otimização com um valor nulo para o fator de penalização, definido na equação 2.4, na esperança de que a solução encontrada esteja mais próxima da solução ótima do que aquela encontrada em um processo conduzido sobre uma superfície anômala de resposta.

#### 4.3.5 Avaliação da utilização de regiões tabus com superfícies de resposta

Para avaliar a eficiência da abordagem utilizando regiões tabus, a função objetivo apresentada na figura 4.11 foi otimizada utilizando-se diferentes abordagens. Os resultados obtidos são mostrados na tabela 4.6.

Tabela 4.6. Abordagens para a utilização de superfícies de resposta.\*

	Sem superfícies	SR-100	SR-95	SR-90	SR-85
Avaliações analíticas	205	6	68	150	278
Avaliações utilizando superfícies de resposta		1618	1647	1872	3034
Ciclos de treinamento das redes neurais		4	4	3	5
Análises para a geração das superfícies de resposta		2	2	2	2
Valor da variável X	2.00	2.13	2.14	2.15	1.86
Tempo total de processo (ms)	406	15313	14484	13843	68985

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

Nos exemplos apresentados na tabela 4.6, a otimização sem superfícies de resposta atua como controle. A estratégia designada SR-100 realiza 100% das análises utilizando a

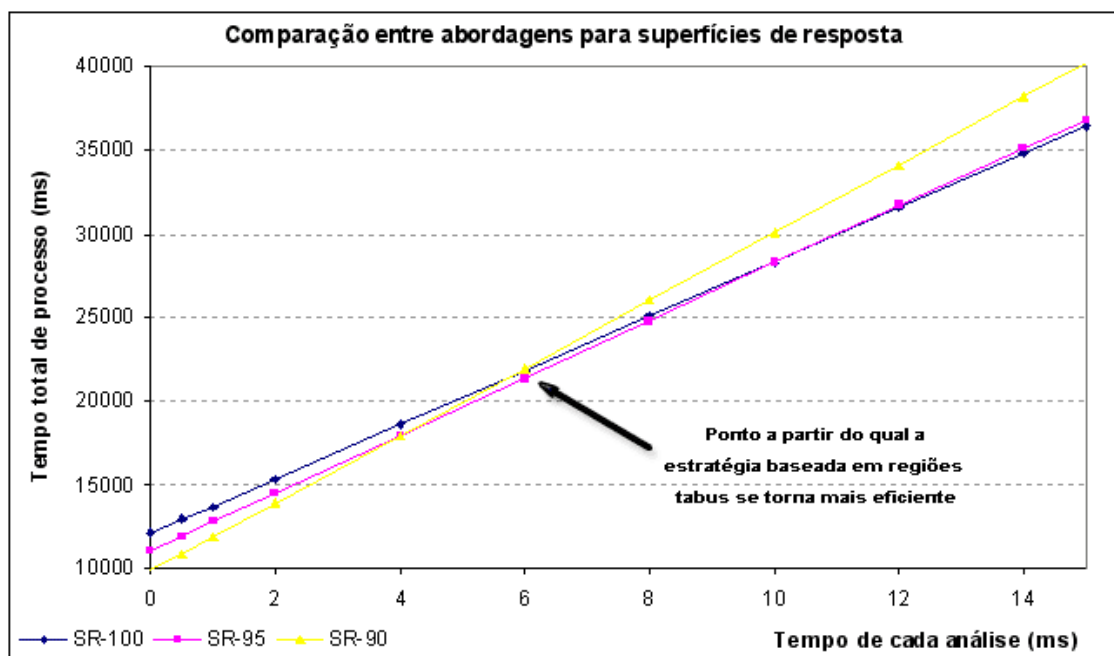
superfície de resposta, empregando regiões tabus para o tratamento da função viciada. Por fim, são empregadas três versões da abordagem tradicional, discutida na seção 4.3.3, cada uma delas com um percentual diferente de utilização da superfície de resposta.

As abordagens SR-95, SR-90 e SR-85 correspondem a processos de otimização que realizam 95%, 90% e 85%, respectivamente, das análises utilizando as superfícies de resposta.

A realização de parte das análises sem o emprego de superfícies de resposta elimina algumas readaptações da superfície, permitindo que o processo de otimização convirja mais rapidamente.

No entanto, ao realizar mais de 10% das análises sem a utilização de superfícies de resposta, a quantidade de pontos gerados reduz a velocidade do treinamento das redes neurais, diminuindo a eficiência do processo de otimização.

A eficiência de cada abordagem está diretamente associada ao tempo de análise de cada solução do problema. Conforme demonstrado na figura 4.12, pode-se determinar que problemas nos quais a análise de uma solução demora mais do que alguns poucos milissegundos apresentam ganho de eficiência ao se empregar a estratégia baseada em regiões tabus ao invés das abordagens tradicionais.



**Figura 4.12.** Comparação entre diferentes abordagens para a utilização de superfícies de resposta.

#### 4.3.6 Conclusão

Em contraponto as estratégias tradicionais discutidas na seção 4.3.3, o presente trabalho propõe uma abordagem inovadora para a utilização de superfícies de resposta, que utiliza regiões tabus para limitar o movimento do algoritmo de otimização ao longo da região viável.

A grande vantagem desta estratégia é reduzir consideravelmente o número de análises realizadas sem a utilização de superfícies de resposta. Assim, quanto maior for o tempo demandado em cada análise do problema, maior é o ganho de desempenho da abordagem utilizando regiões tabus em relação às abordagens tradicionais.

Por essa razão, problemas compostos unicamente de equações e expressões analíticas não apresentam ganho de desempenho ao se empregar tal abordagem. No entanto, como discutido na seção 4.1.7, nesses casos, a própria utilização de superfícies de resposta não é adequada.

#### 4.4 Otimização multidisciplinar baseada nos valores de acoplamento

Como visto na seção 2.1.11, existem várias estratégias para a otimização de problemas multidisciplinares que diferem em relação ao grau de centralização do processo e em relação aos mecanismos utilizados para estimar os valores de acoplamento.

Entre as diversas estratégias multidisciplinares, a estratégia de Otimização Colaborativa tem se destacado graças ao seu potencial para a otimização em ambiente distribuído, conforme discutido na seção 2.1.14.1. Os recentes trabalhos de **Giassi (2004)** e **Rabeau (2006)** utilizam essa abordagem, adaptando-a em duas das mais recentes estratégias multidisciplinares publicadas.

No entanto, ambos os trabalhos discutem a inviabilidade de se conduzir processos de otimização multidisciplinares em problemas que possuam forte acoplamento. Para esse tipo de problema a abordagem tradicionalmente proposta é encapsular o problema multidisciplinar em um problema multiobjetivo para então resolvê-lo.

A estratégia proposta neste trabalho difere dos outros trabalhos publicados sobre o assunto justamente por permitir a resolução de problemas com forte acoplamento.

#### 4.4.1 Diferença entre acoplamentos fortes e fracos

O conceito de acoplamento forte não pode ser definido de maneira rigorosa. Pode-se dizer, no entanto, que um problema apresenta acoplamento forte se uma pequena mudança no valor do acoplamento implicar em uma grande mudança em uma das funções objetivo do problema.

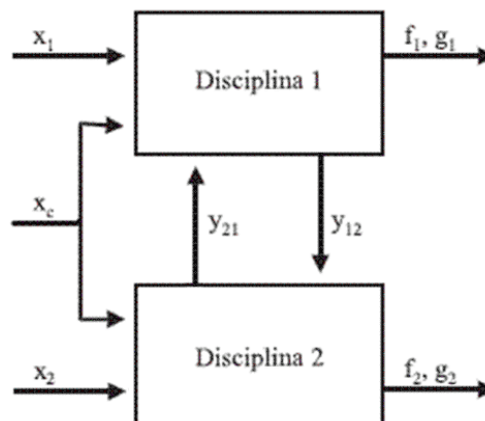
Como sempre, a melhor maneira de entender essa relação é considerar dois exemplos. No projeto do catamarã, apresentado na seção 5.8, o modelo de cálculo estrutural considera o regime de pequenas deformações, no qual a estrutura deformada se confunde com a estrutura não deformada. Nesse caso, pequenas variações no carregamento hidrodinâmico não implicam grandes variações no projeto estrutural. O acoplamento existe, porém é considerado fraco.

Por outro lado, o exemplo apresentado na seção 5.6.2 utiliza duas funções quadráticas como objetivos do problema. Nesse caso, uma modificação pequena nos valores de acoplamento tem efeito quadrático na função objetivo, sendo, portanto, considerado um acoplamento forte.

Embora a definição não seja precisa, quanto mais forte o acoplamento, maior a dificuldade das estratégias multidisciplinares tradicionais para encontrar a solução (ou soluções) ótima.

#### 4.4.2 A estratégia multidisciplinar baseada em valores de acoplamento

Na figura 4.13 é mostrada a formulação geral do problema multidisciplinar, conforme discutido na seção 2.1.10.2.



**Figura 4.13.** Formulação geral do problema multidisciplinar.

Existem duas categorias de acoplamento em um problema multidisciplinar: as variáveis comuns  $X_c$  e as funções de acoplamento propriamente ditas  $y_{i,j}$ .



A estratégia proposta consiste em definir um problema de otimização global a partir da formulação do problema original. A formulação do problema global pode ser vista na equação 4.3.

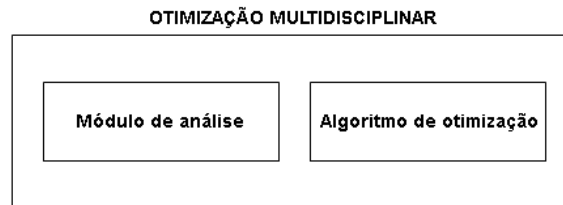
$$\begin{aligned} X_{global} &= (x_{c,1}, x_{c,2} \dots x_{c,n}, y_{1,2}, y_{2,2}) \\ F_{global} &= (f_{1,1}, f_{1,2} \dots f_{1,n}, f_{2,1}, f_{2,2} \dots f_{2,n}, f_{global,1}, f_{global,2} \dots f_{global,n}) \\ G_{global} &= (g_{global,1}, g_{global,2} \dots g_{global,n}) \end{aligned} \quad (4.3)$$

O vetor de variáveis globais  $X_{global}$  é definido pela união do vetor de variáveis comuns com as variáveis artificiais. Para cada acoplamento existente no problema, uma variável artificial é incluída ao vetor de variáveis globais do problema.

O vetor das funções objetivo  $F_{global}$  é definido pelos objetivos de cada disciplina além dos objetivos globais do problema original, se existirem.

O vetor das restrições globais  $G_{global}$  é definido pelas restrições globais do problema original, se existirem.

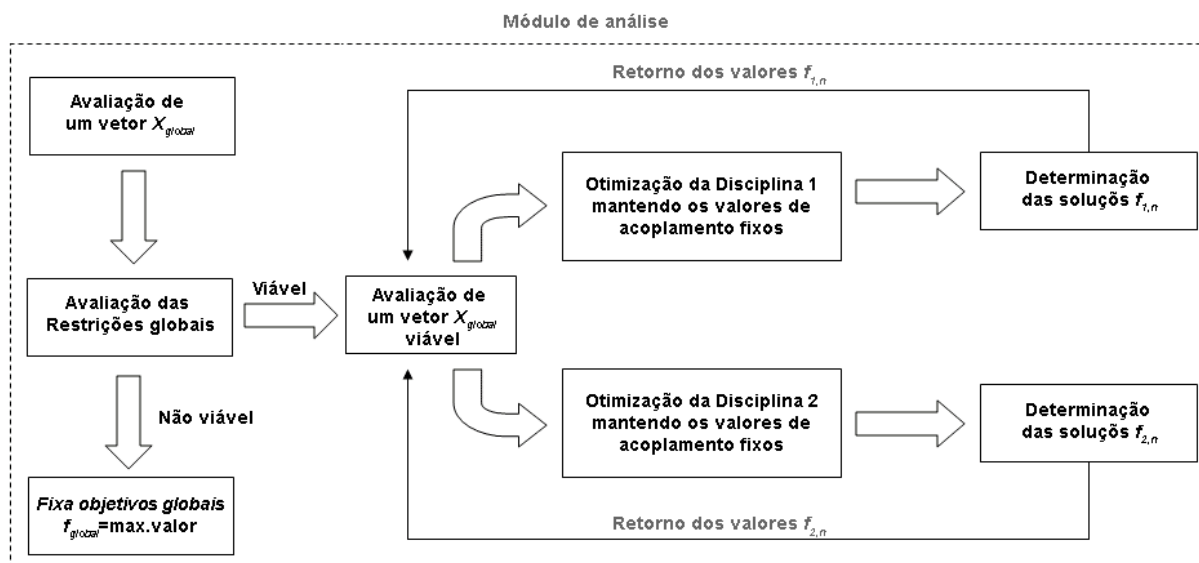
A estrutura de otimização multidisciplinar baseada em valores de acoplamento, mostrada na figura 4.14, é composta de dois módulos: um módulo de análise de soluções globais e um algoritmo de otimização.



**Figura 4.14.** Esquema geral da estratégia multidisciplinar.

O problema de otimização global é sempre um problema multiobjetivo e, portanto, deve ser resolvido utilizando-se um algoritmo de otimização multiobjetivo, como discutido na seção 2.1.7.3.

A essência da estratégia de otimização baseada em valores de acoplamento reside no módulo de análise do problema global, cujo esquema de funcionamento é mostrado na figura 4.15.



**Figura 4.15.** Módulo de análise para a estratégia multidisciplinar.

Para cada ponto avaliado no problema global, defini-se um vetor  $X_{global}$  único, que determina os valores para cada variável comum e cada acoplamento existentes no problema original.

Caso existam restrições globais no problema original, essas devem ser verificadas para o vetor  $X_{global}$ . Para vetores inviáveis, são atribuídos às funções objetivo o maior valor real possível, conforme discutido na seção 3.3.7, de tal sorte que esses pontos tem alta probabilidade de serem descartados no processo de otimização.

Os vetores viáveis, por outro lado, são analisados segundo o procedimento descrito a seguir. Para cada disciplina existente no problema original, é conduzido um processo de otimização local, fixando-se os valores das variáveis comuns e acoplamentos. Os valores das funções objetivo obtidos nas otimizações locais são transferidos para o problema global, completando a análise do vetor  $X_{global}$ .

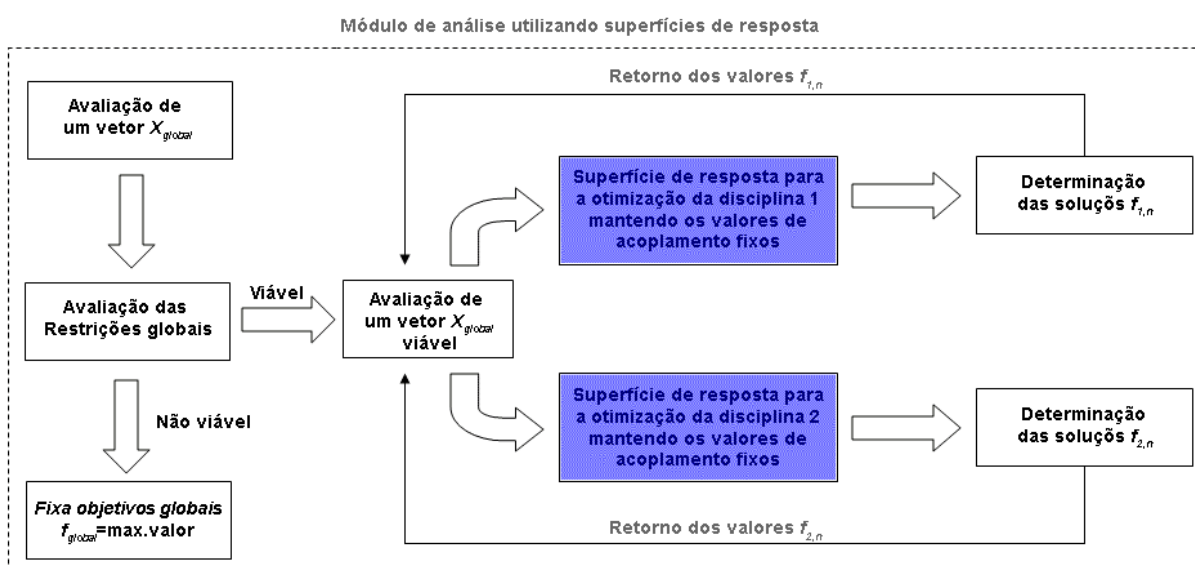
É importante destacar que uma única análise global pode resultar em diversos indivíduos adicionados à população global. Pois, ao se analisar disciplinas multiobjetivo, o processo de otimização local resulta em uma fronteira de soluções que compartilham os mesmo valores de acoplamento. Nesse caso, todas as soluções devem ser transferidas para a população global de indivíduos.

Definido o procedimento de análise, o algoritmo de otimização multiobjetivo pode conduzir o processo de otimização global até a convergência da solução.

#### 4.4.3 A utilização de superfícies de resposta

Como visto na seção anterior, para cada ponto do problema global analisado, deve-se conduzir uma otimização local para cada disciplina existente. Embora robusto, esse procedimento é extremamente ineficiente.

Para aumentar a eficiência e viabilizar o processo de otimização, pode-se, utilizando-se os conceitos discutidos nas seções 4.1 e 4.3, construir superfícies de resposta para representar os processos de otimização de cada disciplina do problema, acelerando a convergência do processo global. O módulo de análise utilizando superfícies de resposta é mostrado na figura 4.16.



**Figura 4.16.** Módulo de análise para a estratégia multidisciplinar com superfícies de resposta.

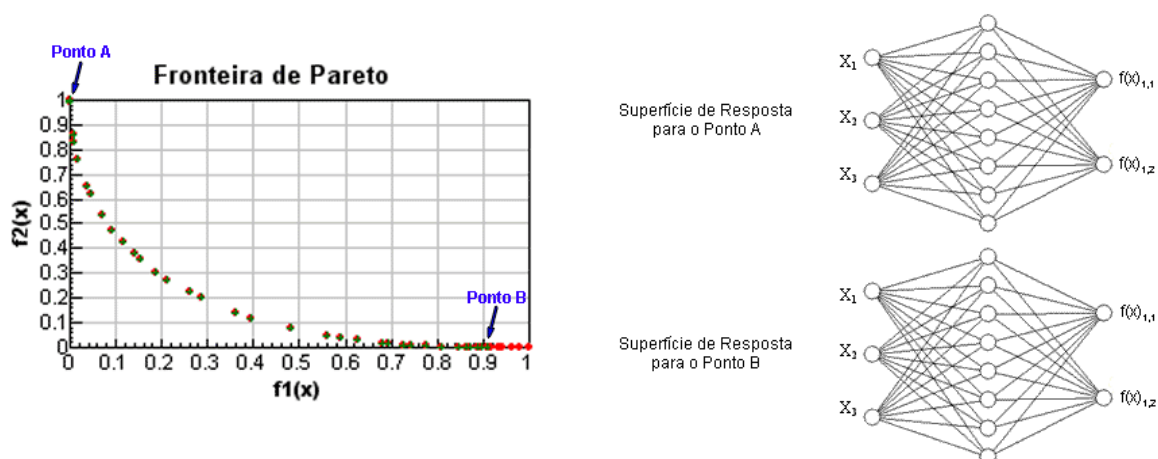
O emprego de superfícies de resposta na representação dos processos de otimização local aumenta a eficiência do processo global de otimização. No entanto, algumas considerações precisam ser feitas.

No caso de disciplinas mono-objetivo, para cada vetor  $X_{global}$ , a superfície de resposta fornecerá a estimativa de uma única solução. Nesse caso o procedimento é similar a todas as outras aplicações envolvendo superfície de resposta descritas neste trabalho.

Disciplinas multiobjetivo, por outro lado, tendem a fornecer um conjunto de soluções Pareto-Otimas para cada vetor  $X_{global}$ . Nesse caso, a utilização de superfícies de resposta não é capaz de fornecer diferentes soluções para um mesmo conjunto de valores de entrada.

Para resolver esse problema, pode-se utilizar  $n$  conjuntos de saídas (ou até mesmo  $n$  redes neurais), onde  $n$  representa o número de objetivos na disciplina.

Cada rede neural fornecerá os valores das funções objetivo para o ponto da fronteira de pareto que maximiza o valor de uma das funções objetivo, conforme exemplificado na figura 4.17. Dessa maneira as soluções locais são transferidas para a população global de modo paritário.



**Figura 4.17.** Exemplo de utilização de superfícies de resposta para uma disciplina multiobjetivo.

Embora a utilização dos extremos da Fronteira de Pareto limite a diversidade das soluções em problemas multiobjetivo, essa abordagem é suficiente para que o processo de otimização global se desenvolva de maneira eficiente e coerente. Ainda assim, se desejável, é possível incluir um número maior de superfícies de resposta para aumentar a diversidade das soluções transferidas dos problemas locais para o problema global.

Como discutido na seção 4.1.5, o ganho de eficiência obtido com a utilização de superfícies de resposta está condicionado ao tempo demandado por cada avaliação de cada uma das disciplinas do problema. Quanto maior esse tempo, mais superfícies podem ser utilizadas, e ainda assim o ganho de eficiência do processo global será significativo.

#### 4.4.4 Conclusão

A principal contribuição apresentada nesta seção é uma nova estratégia de otimização multidisciplinar baseada em valores de acoplamentos, que, como demonstram os exemplos apresentados na seção 5, permite resolver uma grande gama de problemas, incluindo os problemas com acoplamento forte.

No entanto, tal abordagem só se mostra eficiente ao serem utilizadas superfícies de resposta para a representação das disciplinas, pois do contrário, o número de avaliações em

cada disciplina, a cada iteração do problema global, reduz consideravelmente a eficiência do processo.

Outro recurso importante a ser destacado é o ganho de eficiência apresentado pela estratégia de otimização multidisciplinar baseada em valores de acoplamento em ambientes distribuídos. Pois, nesse caso, pode-se paralelizar tanto os processos de otimização de cada disciplina, quanto o treinamento das redes neurais que representam as superfícies de resposta que correspondem a esses processos.

## 5 Exemplos de Aplicações

---

5.1	Programação Linear	135
5.1.1	<i>Planejamento da Produção</i>	135
5.1.2	<i>Problema de Designação</i>	136
5.2	Otimização mono-objetiva	137
5.2.1	<i>Problema Rosenbrock</i>	137
5.2.2	<i>Tampa de Escotilha</i>	139
5.2.3	<i>Grelha Simples</i>	141
5.3	Interpolação de gráficos e diagramas	143
5.4	Otimização com análises numéricas	144
5.4.1	<i>Viga em balanço</i>	144
5.4.2	<i>Grelha</i>	146
5.4.3	<i>Vaso de Pressão</i>	148
5.5	Otimização multiobjetivo	150
5.5.1	<i>Multiobjetivo Simples</i>	150
5.5.2	<i>Multiobjetivo Complexo</i>	152
5.5.3	<i>Viga I</i>	153
5.6	Otimização multidisciplinar	155
5.6.1	<i>Multidisciplinar Simples</i>	156
5.6.2	<i>Multidisciplinar Acoplado</i>	157
5.6.3	<i>Múltiplas Funções Objetivo</i>	158
5.7	Redutor de Velocidade	160
5.8	Projeto do hidrofólio de uma embarcação catamarã	163
5.8.1	<i>Descrição do problema multidisciplinar</i>	165
5.8.2	<i>As características da embarcação projetada</i>	171
5.8.3	<i>A abordagem racional do problema</i>	171
5.8.4	<i>Resultados da otimização</i>	172

*"Look! It's moving. It's alive. It's alive..."*

Dr. Henry Frankenstein, Mary Shelley

Neste capítulo serão apresentados diferentes problemas matemáticos e de engenharia resolvidos utilizando o sistema desenvolvido e apresentado neste relatório. Embora todos os exemplos apresentem alguma contribuição relevante, destacam-se especialmente três: o projeto de um vaso de pressão utilizando o Método dos Elementos Finitos, o projeto multidisciplinar de um redutor de velocidade e o projeto do hidrofólio de uma embarcação do tipo catamarã.

## 5.1 Programação Linear

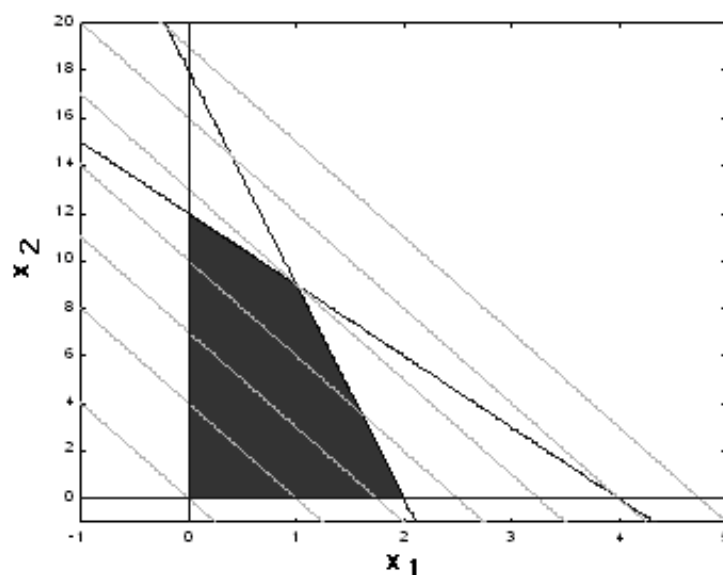
Nesta seção serão apresentados dois problemas de programação linear. Para representar esse grupo de problemas foram escolhidos dois dos mais clássicos exemplos: o problema de planejamento de produção e o problema de designação.

### 5.1.1 Planejamento da Produção (Planejamento.txt)

Uma empresa pode fabricar dois produtos (1 e 2). Na fabricação do produto 1 a empresa gasta nove horas-homem e três horas-máquina (a tecnologia utilizada é intensiva em mão-de-obra). Na fabricação do produto 2 a empresa gasta uma hora-homem e uma hora-máquina (a tecnologia é intensiva em capital). Sendo  $x_1$  e  $x_2$  as quantidades fabricadas dos produtos 1 e 2 e sabendo-se que a empresa dispõe de 18 horas-homem e 12 horas-máquina e ainda que os lucros dos produtos são \$4 e \$1 respectivamente, quanto deve a empresa fabricar de cada produto para obter o maior lucro possível?

$$\begin{aligned} \max F(X) &= 4 * x_1 + x_2 \rightarrow \min F = -4 * x_1 - x_2 \\ G_1(X) &: 9 * x_1 + x_2 \leq 18 \\ G_2(X) &: 3 * x_1 + x_2 \leq 12 \\ G_3(X) &: x_1 \geq 0 \\ G_4(X) &: x_2 \geq 0 \end{aligned} \quad (5.1)$$

Na figura 5.1 mostram-se as curvas de nível, para a função objetivo e as referentes às funções de restrição.



**Figura 5.1.** Curvas de nível para o problema Planejamento da Produção.

### 5.1.1.1 Solução do problema Planejamento da Produção

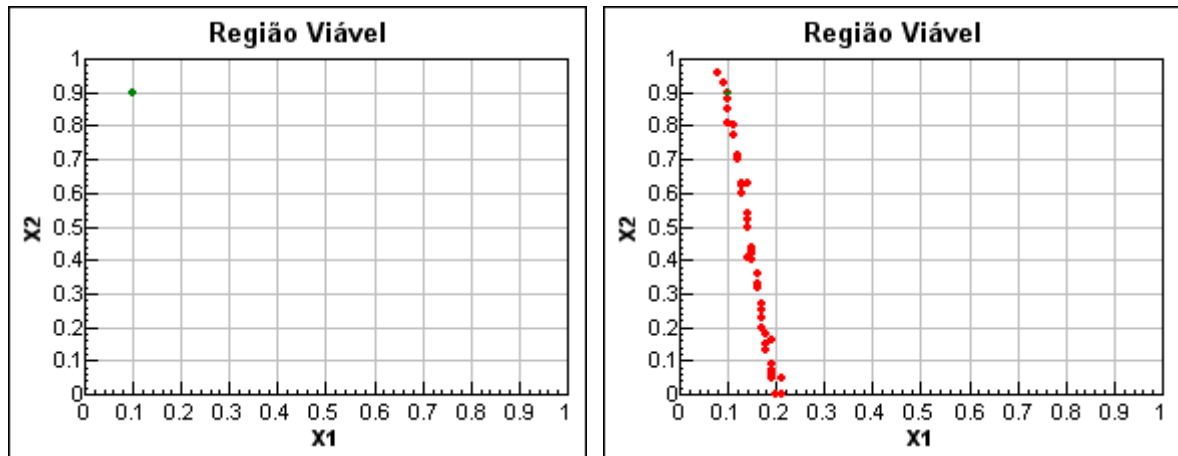


Figura 5.2. Solução do problema Planejamento da Produção. a) Vértices. b) Genético

Tabela 5.1. Solução do problema Planejamento da Produção.\*

Método	Referência	Simplex	Vértices	MBA	MOGA
$x_1$	1	1	1	1	1
$x_2$	9	9	9	9	9
F	-13	-13	-13	-13	-13
Tempo (ms)		78	609	485	938

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

### 5.1.2 Problema de Designação (Designacao..txt)

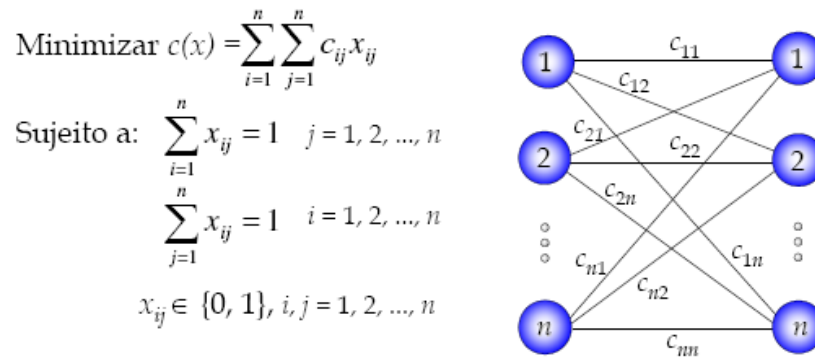
Uma indústria possui 3 funcionários (a,b,c) e 3 máquinas (A,B,C). Cada funcionário possui uma taxa de desperdício de matéria prima em cada uma das máquinas, mostradas na tabela 5.2. Cada funcionário deve trabalhar em uma única máquina, e todas as máquinas devem ser usadas. Determine qual funcionário deve trabalhar em cada máquina para minimizar o desperdício de matéria prima.

Tabela 5.2. Taxas de desperdício de cada funcionário em cada uma das máquinas.

	Máquina A	Máquina B	Máquina C
Funcionário a	0.3%	3.3%	8.7%
Funcionário b	4.0%	5.7%	2.5%
Funcionário c	1.2%	2.1%	4.4%

A formulação clássica do problema de designação é apresentada na figura 5.3. É importante observar que a característica mais importante desse tipo de problema é o fato de que as variáveis são discretas e binárias, sendo portanto apropriado o uso do Algoritmo Genético e do método Simplex.





**Figura 5.3.** Formulação do problema de Designação.

### 5.1.2.1 Solução do problema de Designação

Tabela 5.3. Solução do problema de Designação.\*

Método	Referência	Simplex	Vértices	MBA	MOGA
x <sub>1</sub>	1	1			1
x <sub>2</sub>	0	0			0
x <sub>3</sub>	0	0			0
x <sub>4</sub>	0	0			0
x <sub>5</sub>	0	0			0
x <sub>6</sub>	1	1			1
x <sub>7</sub>	0	0			0
x <sub>8</sub>	1	1			1
x <sub>9</sub>	0	0			0
F	4.9	4.9			4.9
Tempo (ms)		93			344

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

Interpretação: O funcionário *a* trabalhará na máquina *A*, o funcionário *b* trabalhará na máquina *C* e o funcionário *c* trabalhará na máquina *B*.

## 5.2 Otimização mono-objetiva

Nesta seção serão apresentados 3 problemas de otimização mono-objetivo que servirão para validar o ambiente desenvolvido. Os problemas apresentados foram originalmente propostos por **Augusto (1996)** e são inspirados em problemas de engenharia.

### 5.2.1 Problema Rosenbrock (Rosenbrock.txt)

Conhecido como problema de **Rosenbrock (1960)**, deseja-se encontrar o mínimo da função sujeito a quatro restrições. Trata-se de um problema puramente matemático, envolvendo não linearidade, tanto na função objetivo, quanto nas funções de restrição.

$$\min F(X) = (x_1^2 - x_2)^2 + (1 - x_1)^2$$

$$G_1(X) : 0.16 - x_1 * x_2 \geq 0$$

$$G_2(X) : 1 - x_1^2 - x_2^2 \geq 0$$

$$G_3(X) : x_1 \geq 0$$

$$G_4(X) : x_2 \geq 0$$

(5.2)

Na figura 5.4 mostram-se as curvas de nível, para a função objetivo e as referentes às funções de restrição.

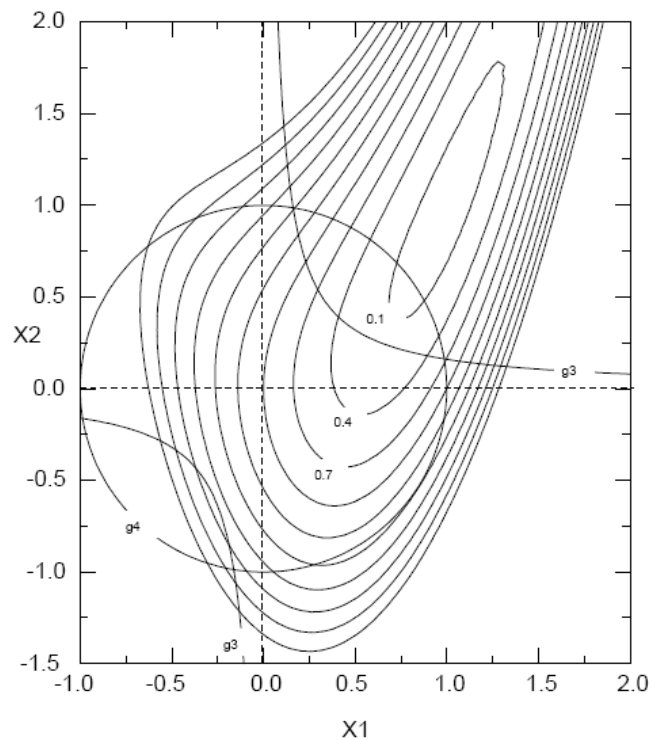


Figura 5.4. Curvas de nível para o problema Rosenbrock.

#### 5.2.1.1 Solução do problema Rosenbrock

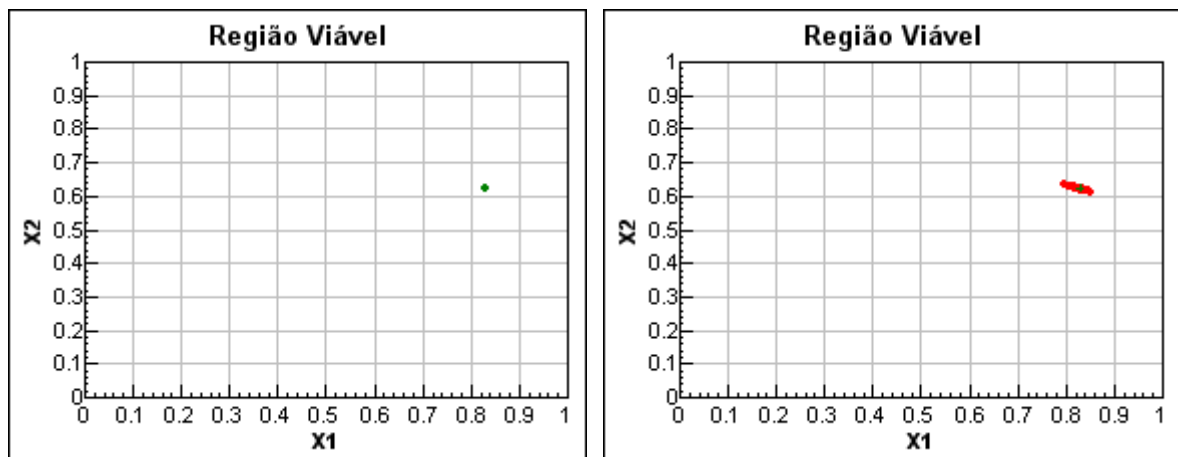


Figura 5.5. Solução do problema Rosenbrock. a) Vértices. b) Genético

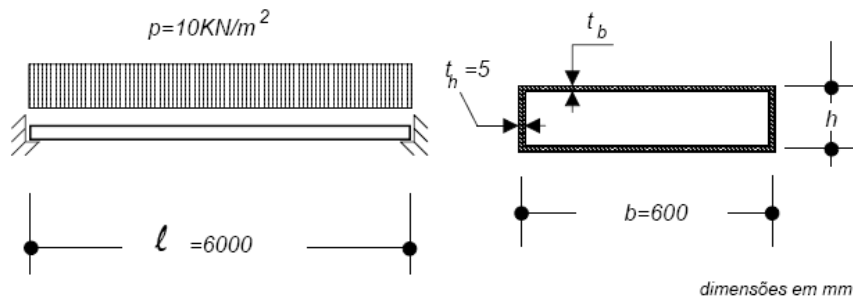
Tabela 5.4. Solução do problema Rosenbrock.\*

Método	Referência	Simplex	Vértices	MBA	MOGA
$x_1$	0.664		0.656	0.663	0.661
$x_2$	0.241		0.244	0.241	0.242
F	0.153		0.153	0.153	0.153
Tempo (ms)			360	1215	1604

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

### 5.2.2 Tampa de Escotilha (Tampa Escotilha.txt)

Objetivando o mínimo peso, deseja-se obter as dimensões  $h$  e  $t_b$ , mostradas na figura 5.6, para a tampa de escotilha de porão de carga de uma embarcação. A largura da escotilha é  $l_0 = 6.0$  m, e a tampa é formada por segmentos de viga caixão, fabricada em alumínio (módulo de elasticidade  $E = 70$  KN/mm<sup>2</sup>, tensão de escoamento,  $\sigma_e = 70$  N/mm<sup>2</sup>, coeficiente de Poisson,  $\mu = 0.3$ ) e largura de segmento de  $b = 600$  mm. A pressão, a ser suportada pela escotilha, é de  $p = 10$  KN/m<sup>2</sup>. Devem ser verificadas as restrições, a seguir, mencionadas.



#### Viga

- Máximo deslocamento,  $g_1(h, t_b)$ :

$$\delta = \frac{5(pb)\ell^4}{384EI} \leq \delta_{\text{lim}} = \frac{\ell}{500} = 12.0\text{mm}$$

- resistência nos pontos de máxima solicitação

flange, fibra externa,  $g_2(h, t_b)$ :

$$\sigma = \frac{(pb)\ell^2}{12W} \leq \sigma_{\text{max}} = 70\text{MPa}$$

alma, linha neutra,  $g_3(h, t_b)$ :

$$\tau = \frac{\left(\frac{pb\ell}{2}\right)m}{t_h I} \leq \tau_{\text{max}} = \frac{70}{\sqrt{3}}\text{MPa}$$

- não ocorrência de flambagem localizada dos flanges,  $g_4(h, t_b)$ :

$$\frac{b}{t_b} \leq \frac{4\pi}{\sqrt{12(1-\mu^2)}} \sqrt{\frac{E}{\sigma}}$$

#### Secção Transversal

Dados geométricos:

- Área:  $A = 2(ht_h + bt_b)$

- Inércia:

$$I = (t_h h + 3t_b b) \frac{h^2}{6}$$

- Módulo de resistência:

$$W = 2 \frac{I}{h}$$

- Momento estático na linha neutra:

$$m = \frac{1}{4}bt_b h + \frac{1}{8}h^2 t_h$$

Por simplicidade, o peso das anteparas transversais e dos reforçadores não foram considerados no problema, e, a função de mérito pode ser escrita em termos da área da secção transversal somente.

Figura 5.6. Ilustração do problema Tampa de Escotilha.

Na figura 5.7 mostram-se as curvas de nível para a função objetivo e as referentes às funções de restrição.

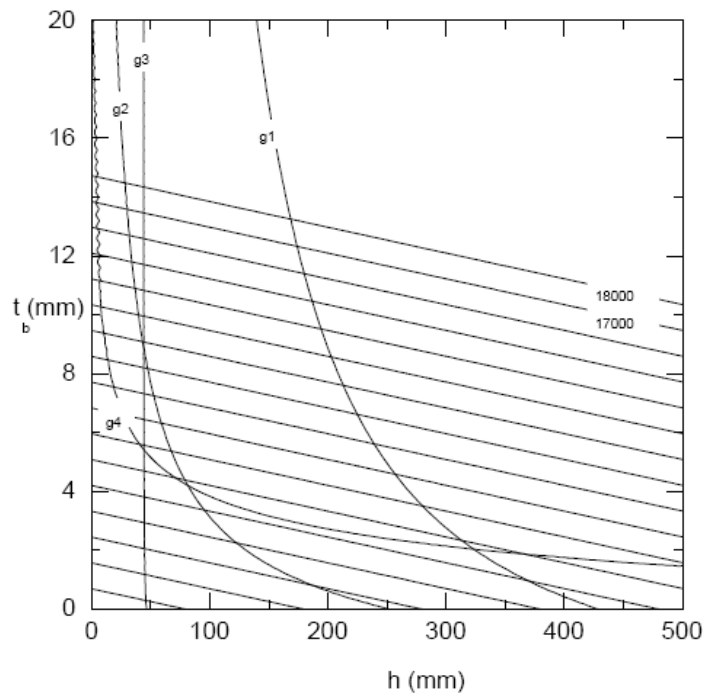


Figura 5.7. Curvas de nível para o problema Tampa de Escotilha.

#### 5.2.2.1 Solução do problema Tampa de Escotilha

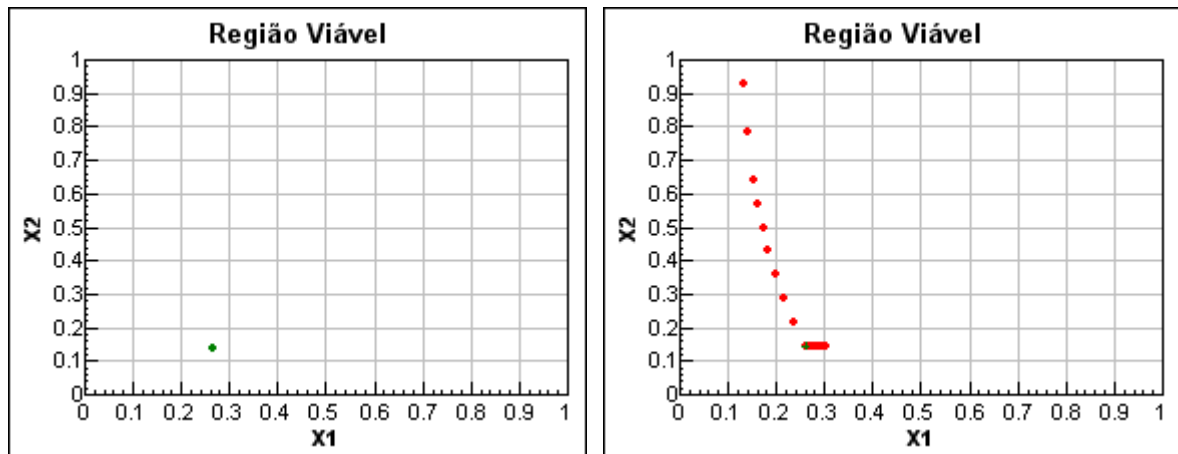


Figura 5.8. Solução do problema Tampa de Escotilha. a) Vértices. b) Genético

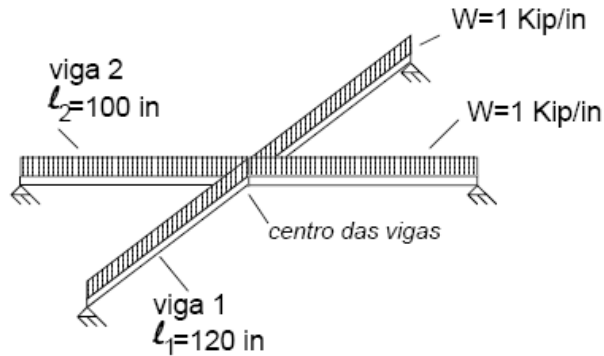
Tabela 5.5. Solução do problema Tampa de Escotilha.\*

Método	Referência	Simplex	Vértices	MBA	MOGA
$x_1$	323.89		328.45	323.81	324.35
$x_2$	2.93		2.91	2.93	2.93
F	6575		6781.05	6756.95	6759.50
Tempo (ms)			1219	766	7687

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

### 5.2.3 Grelha Simples (Grelha Simples.txt)

Deseja-se projetar a grelha da figura 5.9, definindo-se, como variáveis de projeto apenas as áreas das seções transversais das vigas que a compõem. Para tanto, utiliza-se uma relação empírica entre a inércia à flexão e módulo de resistência em função da área do perfil. A função objetivo é o volume total das duas vigas.



- Inércia:

$$I(in^4) = 1007 \left[ \frac{A(in^2)}{148} \right]^{2.65}$$

- Módulo de resistência:

$$Z(in^3) = \left[ \frac{A(in^2)}{148} \right]^{1.82}$$

- Tensão limite:

$$\sigma_{max} = 20\,000 \text{ Psi}$$

- Deslocamentos:

carga distribuída:

$$\delta = \frac{5w\ell^4}{384EI}$$

carga concentrada:

$$\delta = \frac{P\ell^3}{48EI}$$

- Momento fletor:

$$M(x) = Q_{END}x - w\frac{x^2}{2}$$

- Força Cortante:

$$Q(x) = Q_{END} - wx$$

- Localização do momento max.:

$$x_m = \frac{Q_{END}}{w} ; \quad [0 \leq x_m \leq \frac{\ell}{2}]$$

- Força Cortante na extremidade viga 1:

$$Q_{END} = \frac{w\ell - P}{2}$$

viga 2

$$Q_{END} = \frac{w\ell + P}{2}$$

- Incógnita Hiperestática:

$$P = \frac{5}{8} \left[ \frac{I_2\ell_1^4 - I_1\ell_2^4}{I_2\ell_1^3 + I_1\ell_2^3} \right] w$$

- Restrições de resistência:

$$g_j(A_1, A_2) = \sigma_{max} - \frac{M_{(j)max}}{Z_j}$$

**Figura 5.9.** Ilustração do problema Grelha Simples.

Na figura 5.10 mostram-se as curvas de nível para a função de mérito e as referentes às funções de restrição. Pode-se observar que o problema apresenta um mínimo global e dois mínimos locais que ficam evidenciados pelas concentrações de indivíduos após a otimização utilizando o algoritmo genético.

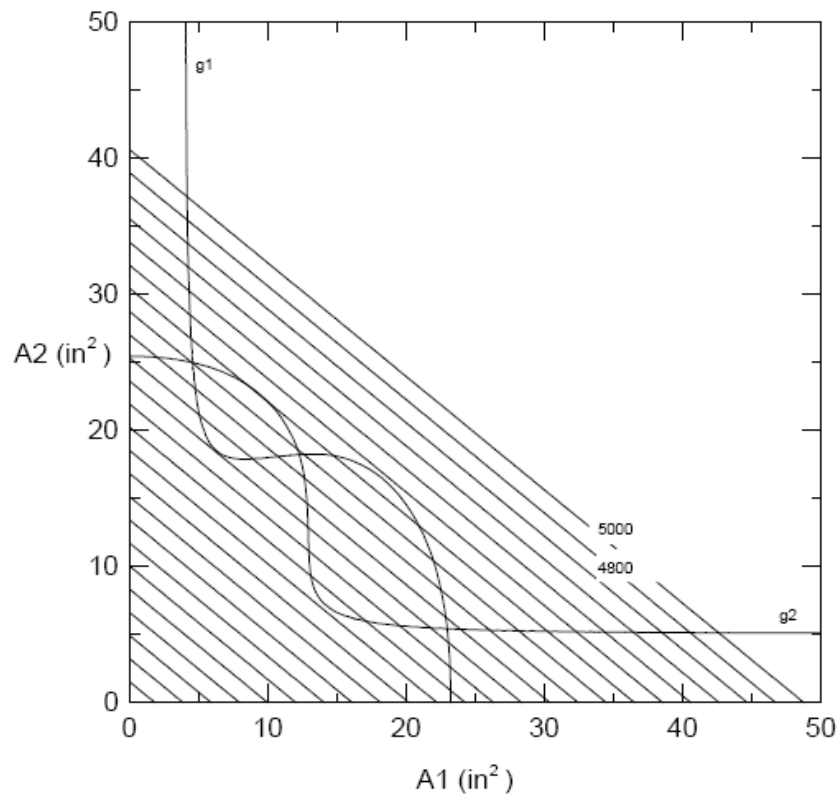


Figura 5.10. Curvas de nível para o problema Grelha Simples.

### 5.2.3.1 Solução do problema Grelha Simples

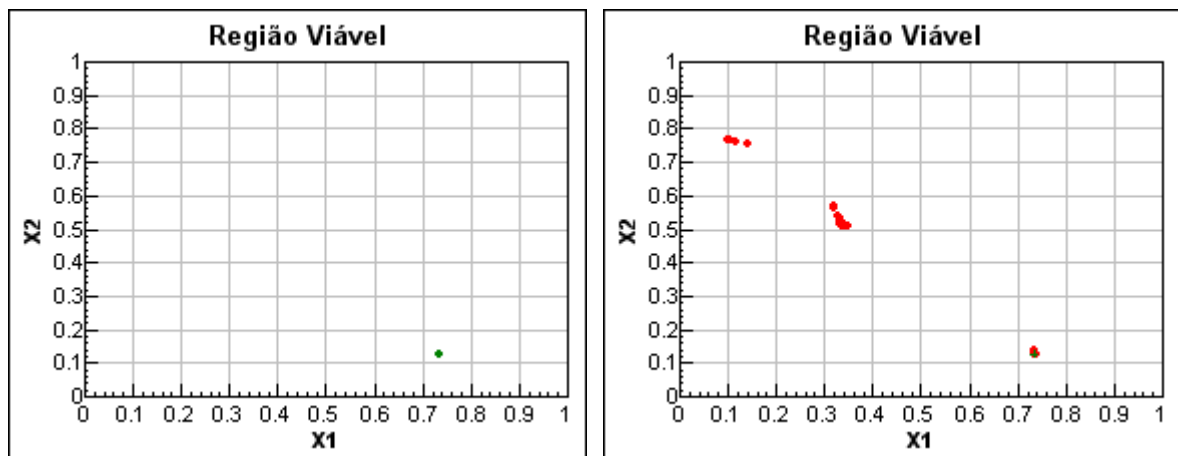


Figura 5.11. Solução do problema Grelha Simples. a) Vértices. b) Genético

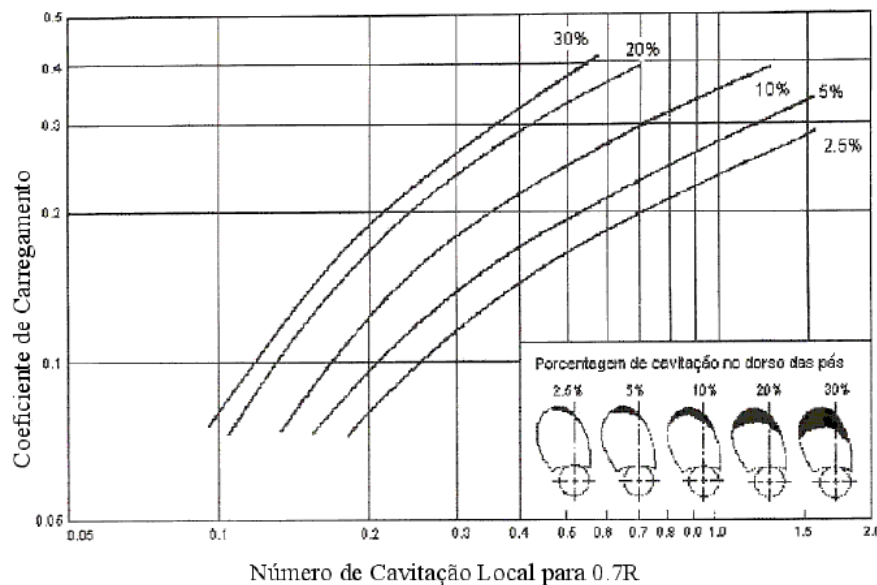
Tabela 5.6. Solução do problema Grelha Simples.\*

Método	Referência	Simplex	Vértices	MBA	MOGA
$x_1$	23.50		26.26	26.26	26.27
$x_2$	6.32		6.22	6.22	6.23
F	3108		3372.96	3372.93	3374.60
Tempo (ms)			969	735	6766

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

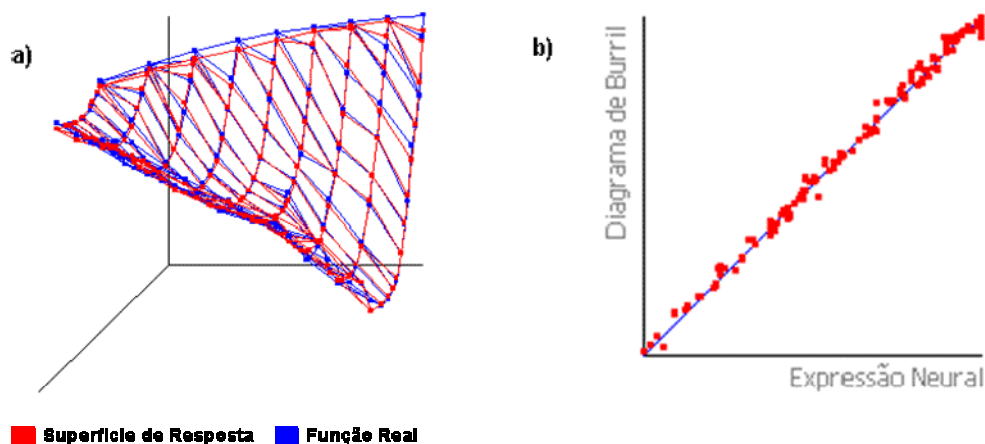
### 5.3 Interpolação de gráficos e diagramas

Ao tentar automatizar a análise para otimizar o projeto de navios pesqueiros, **Ugarte (2007)** utilizou referências com bastante aderência ao tipo de navio que estudava. Porém, os coeficientes para a estimativa da estabilidade e da cavitação no propulsor eram descritos na forma de gráficos e diagramas, exemplificados na figura 5.12. Para contornar este problema, Ugarte recorreu ao ambiente desenvolvido nesta pesquisa e conseguiu obter “expressões neurais” com perfeita aderência ao problema real e que mais tarde foram incorporadas ao algoritmo de otimização, permitindo a conclusão de seu estudo.



**Figura 5.12.** Diagrama de cavitação simples de Burril.

A superfície representada por rede neural treinada por Ugarte pode ser vista na figura 5.13a, bem como o erro associado aos pontos analisados (na figura 5.13b).



**Figura 5.13.** A superfície neural para o diagrama de cavitação de Burril, apud Ugarte.

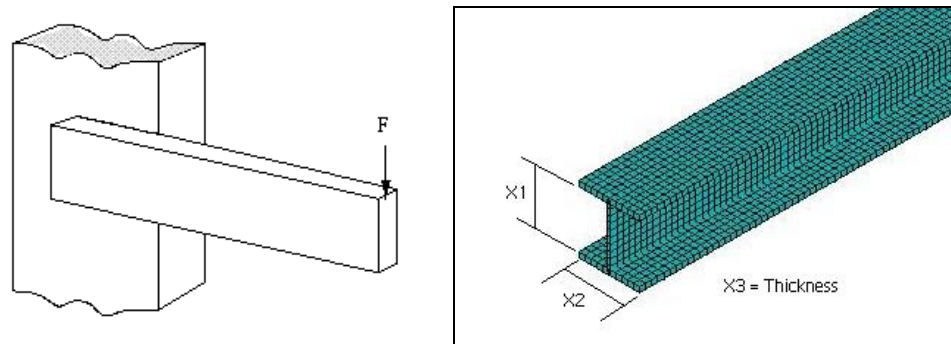
## 5.4 Otimização com análises numéricas

Nesta seção serão apresentados 3 exemplos de otimizações estruturais simples, cujas as avaliações utilizam o Método dos Elementos Finitos. Os problemas serão resolvidos com e sem o uso de superfícies de resposta, a fim de verificar a eficiência de tal técnica.

### 5.4.1 *Viga em balanço* (Viga em Balanco.txt)

Para exemplificar o efeito do uso das superfícies de resposta neurais na otimização de problemas de engenharia dependentes de análises numéricas, optou-se pelo mais simples dos problemas estruturais: o projeto de uma viga em balanço.

Uma viga de seção I, cujas dimensões são representadas na figura 5.14, está engastada em uma das extremidades, tendo na extremidade livre uma carga concentrada aplicada verticalmente no plano de simetria. Deseja-se minimizar a massa da viga limitando-se a máxima tensão atuante e o máximo deslocamento da extremidade livre.



**Figura 5.14.** Representação do problema Viga em balanço.

**Tabela 5.7.** Variáveis e constantes do problema Viga em balanço.

Variável	Mínimo (m)	Máximo (m)	Resolução (m)
Altura da alma da Viga	0.200	1.000	0.001
Largura do Flange da Viga	0.200	0.800	0.001
Espessura da Viga	0.006	0.040	0.001

Constantes	Valor
Comprimento da Viga	100 m
Máxima tensão admissível do material	270 MPa
Máximo deslocamento da extremidade livre	100 mm
Coefficiente de Poisson do material	0.3
Carga vertical no extremo livre da viga	1 N

O problema foi parametrizado para ser resolvido com o Método dos Elementos Finitos, utilizando-se o programa *Ansys*. A macro parametrizada é mostrada no apêndice A.



### 5.4.1.1 Solução do problema Viga I

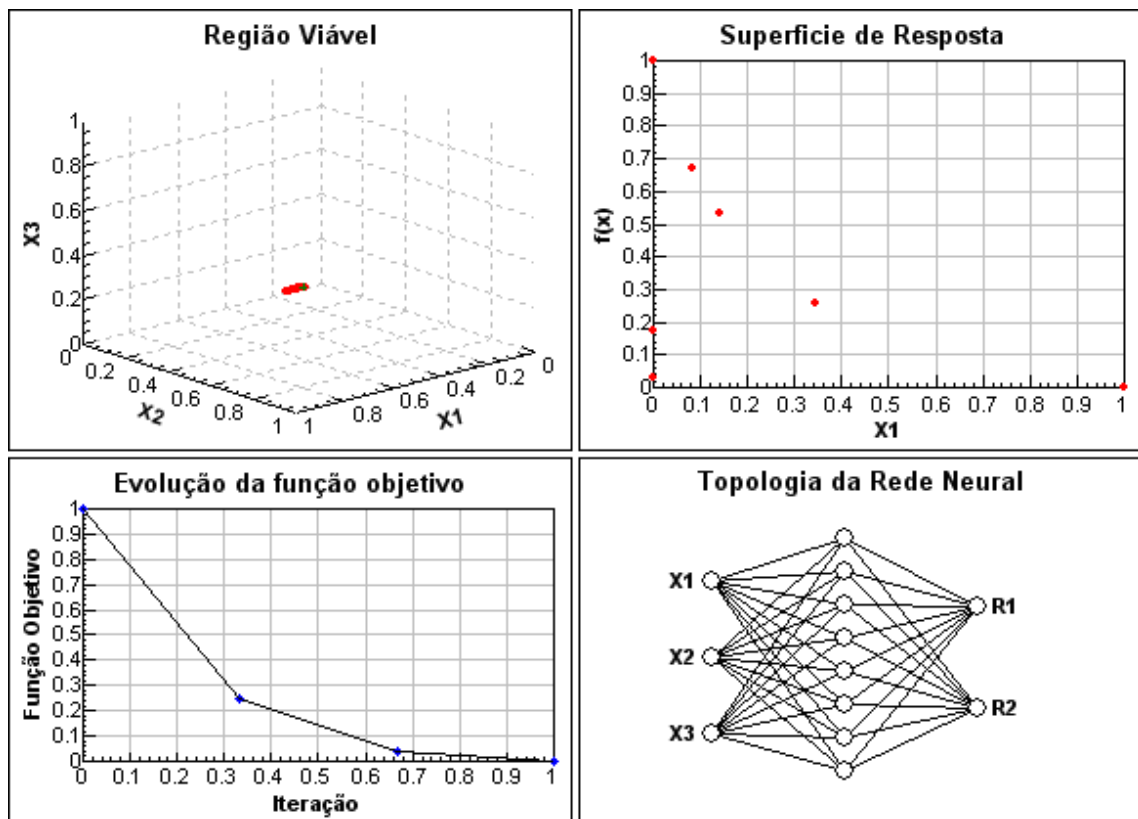


Figura 5.15. Solução do problema Viga em balanço.

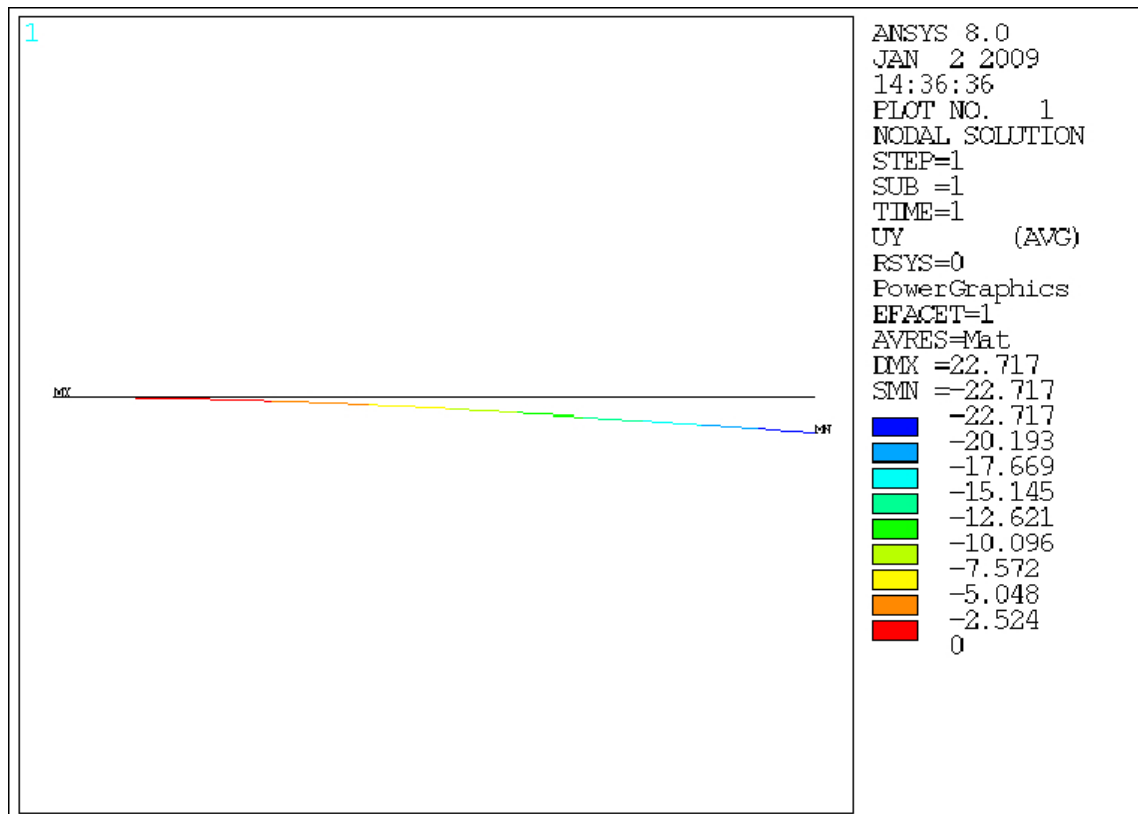


Figura 5.16. Distribuição da tensão máxima na solução final.

Tabela 5.8. Solução do problema Viga em Balanço.\*

	Analítico sem Superfície de Resposta			Analítico com Superfície de Resposta		
	Vértices	MBA	MOGA	Vértices	MBA	MOGA
$x_1$	0.256	0.258	0.257	0.258	0.258	0.259
$x_2$	0.2	0.2	0.2	0.2	0.2	0.2
$x_3$	0.006	0.006	0.006	0.006	0.006	0.006
Área da Seção	0.004	0.004	0.004	0.004	0.004	0.004
Tempo (ms)	344	2533	6828	66906	127953	103406

	Elementos Finitos sem Superfície de Resposta			Elementos Finitos com Superfície de Resposta		
	Vértices	MBA	MOGA	Vértices	MBA	MOGA
$x_1$	0.258	0.258	0.258	0.258	0.258	0.258
$x_2$	0.2	0.2	0.2	0.2	0.2	0.2
$x_3$	0.006	0.006	0.006	0.006	0.006	0.006
Área da Seção	0.004	0.004	0.004	0.004	0.004	0.004
Tempo (ms)	2555766	18500000	21600000	68219	158797	141031

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

#### 5.4.2 Grelha (Grelha-FEM.txt)

O projeto de grelhas e painéis reforçados representa a base de grande parte dos projetos estruturais. Normalmente esse tipo de projeto é feito de forma conservadora, utilizando modelos de chapa e viga, e desacoplando os elementos longitudinais e transversais. Neste exemplo busca-se otimizar a grelha ortogonal composta por 2 vigas de seção I descrita na figura 5.17, de maneira similar a realizada no item 5.2.3.

O problema possui 6 variáveis (3 para cada uma das vigas) e deseja-se minimizar a massa da estrutura sujeita a máxima tensão admissível do material.

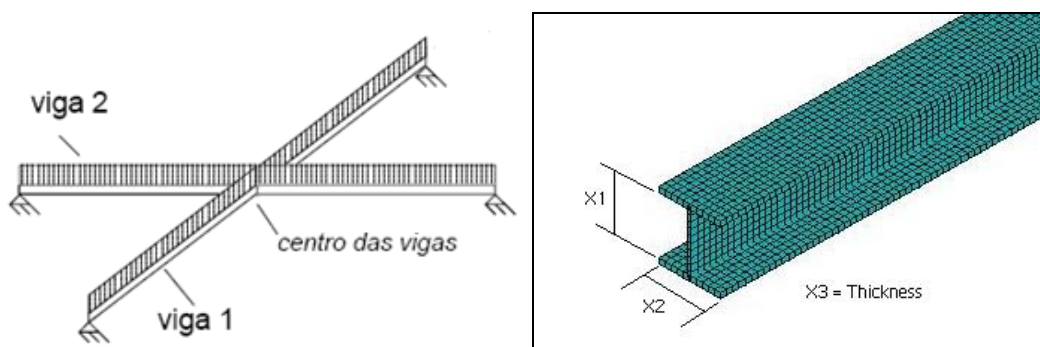


Figura 5.17. Representação do problema Grelha ortogonal com vigas de seção I.

Tabela 5.9. Variáveis e constantes do problema Grelha ortogonal.

Variável	Mínimo (m)	Máximo (m)	Resolução (m)
Altura da alma da Viga 1	0.200	1.000	0.001
Largura do Flange da Viga 1	0.200	0.800	0.001
Espessura da Viga 1	0.006	0.040	0.001

Altura da alma da Viga 2	0.200	1.000	0.001
Largura do Flange da Viga 2	0.200	0.800	0.001
Espessura da Viga 2	0.006	0.040	0.001

Constantes	Resolução
Comprimento da Viga 1	100 m
Comprimento da Viga 2	100 m
Máxima tensão admissível do material	270 MPa
Pressão sobre a grelha	5 Pa

#### 5.4.2.1 Solução do problema Grelha

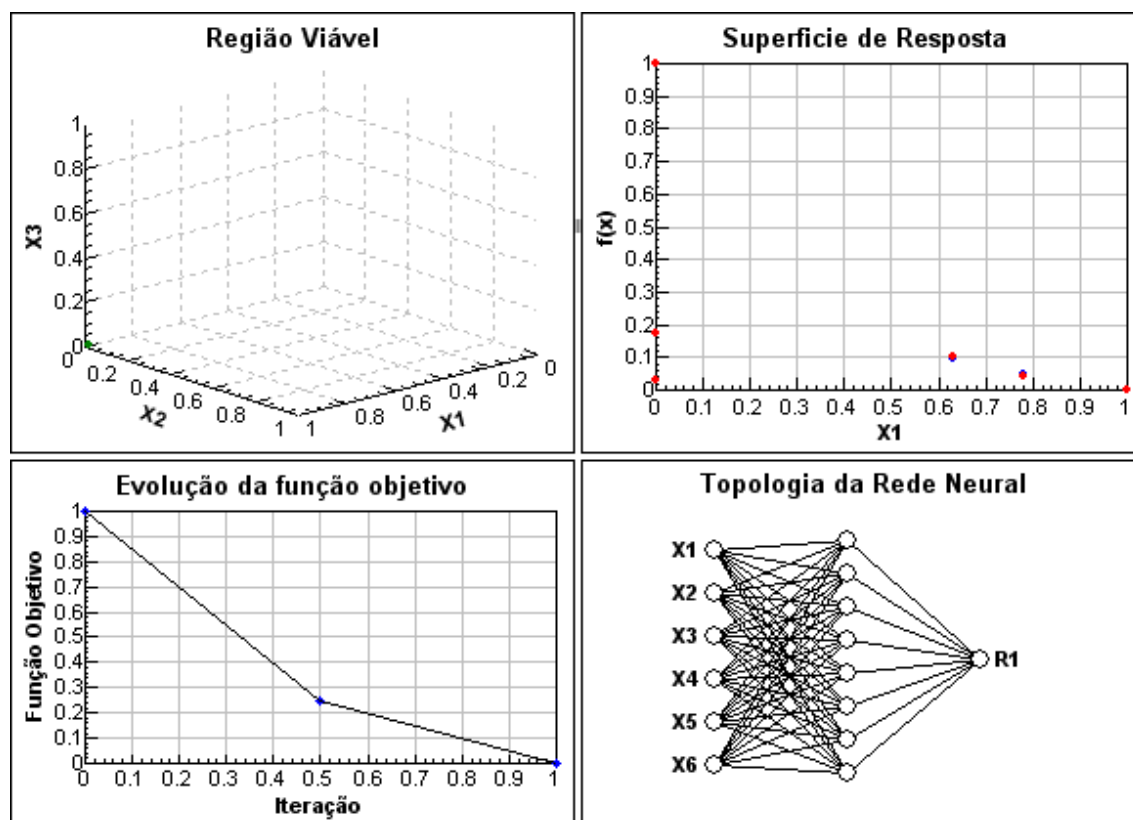


Figura 5.18. Solução do problema Grelha.

Tabela 5.10. Solução do problema Grelha.\*

	Elementos Finitos sem Superfície de Resposta	Elementos Finitos com Superfície de Resposta
X <sub>1</sub>	0.635	0.664
X <sub>2</sub>	0.200	0.200
X <sub>3</sub>	0.006	0.006
X <sub>4</sub>	0.200	0.200
X <sub>5</sub>	0.200	0.200
X <sub>6</sub>	0.006	0.006
Área da Seção Viga I + Área da Seção Viga II	0.00981	0.00998
Tempo (ms)	21474484	21500

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

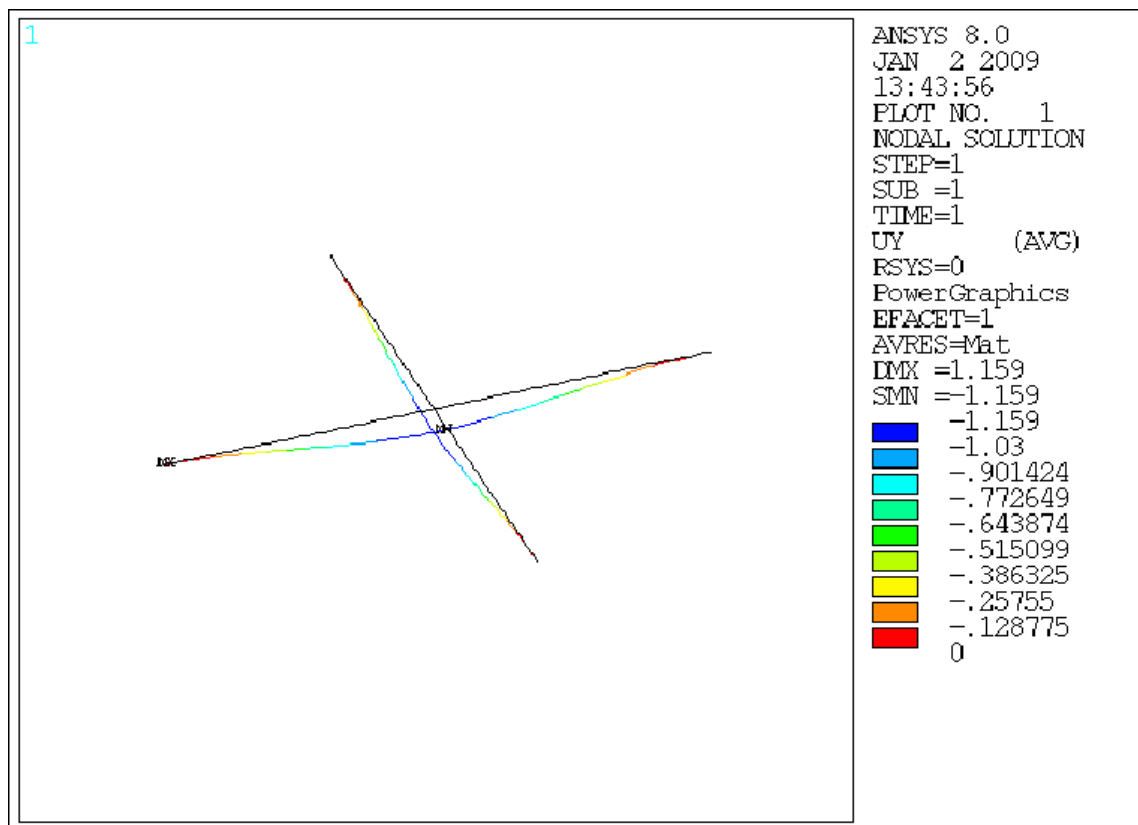


Figura 5.19. Estrutura deformada na configuração otimizada do problema Grelha.

#### 5.4.3 Vaso de Pressão (VasoPressao.txt)

O vaso de pressão mostrado na figura 5.20 é feito de ferro fundido (módulo de elasticidade  $E = 14.5 \text{ Msi}$ , tensão de escoamento,  $\sigma_e = 20 \text{ ksi}$ , coeficiente de Poisson,  $\mu = 0.3$ ) e contém uma pressão interna  $p = 1700 \text{ psi}$ . O vaso cilíndrico tem o diâmetro interno de 8 in com calotas esféricas em ambas as extremidades. Além disso existem 2 ranhuras com raio de  $1/8 \text{ in}$  ao longo da superfície interna e uma ranhura central de 2 in de largura por 0.25 in de profundidade ao longo da superfície externa.

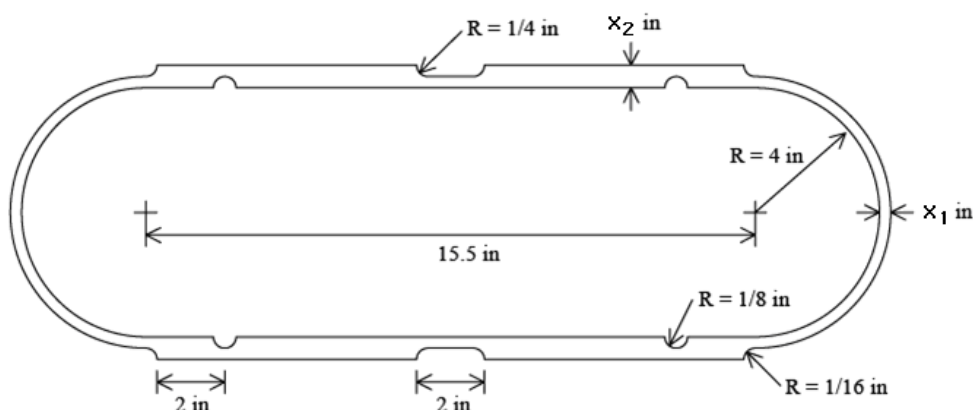


Figura 5.20. Representação do problema Vaso de Pressão.

### 5.4.3.1 Solução do problema Vaso de Pressão

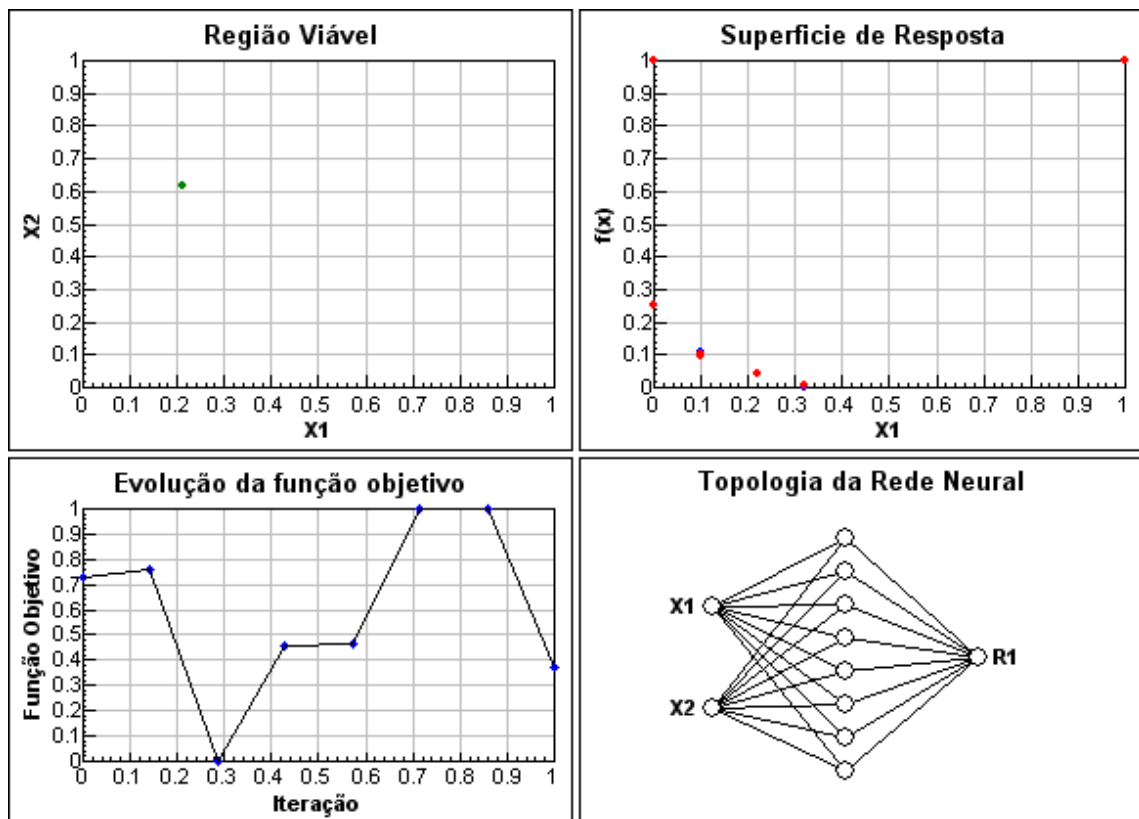


Figura 5.21. Solução do problema Vaso de Pressão.

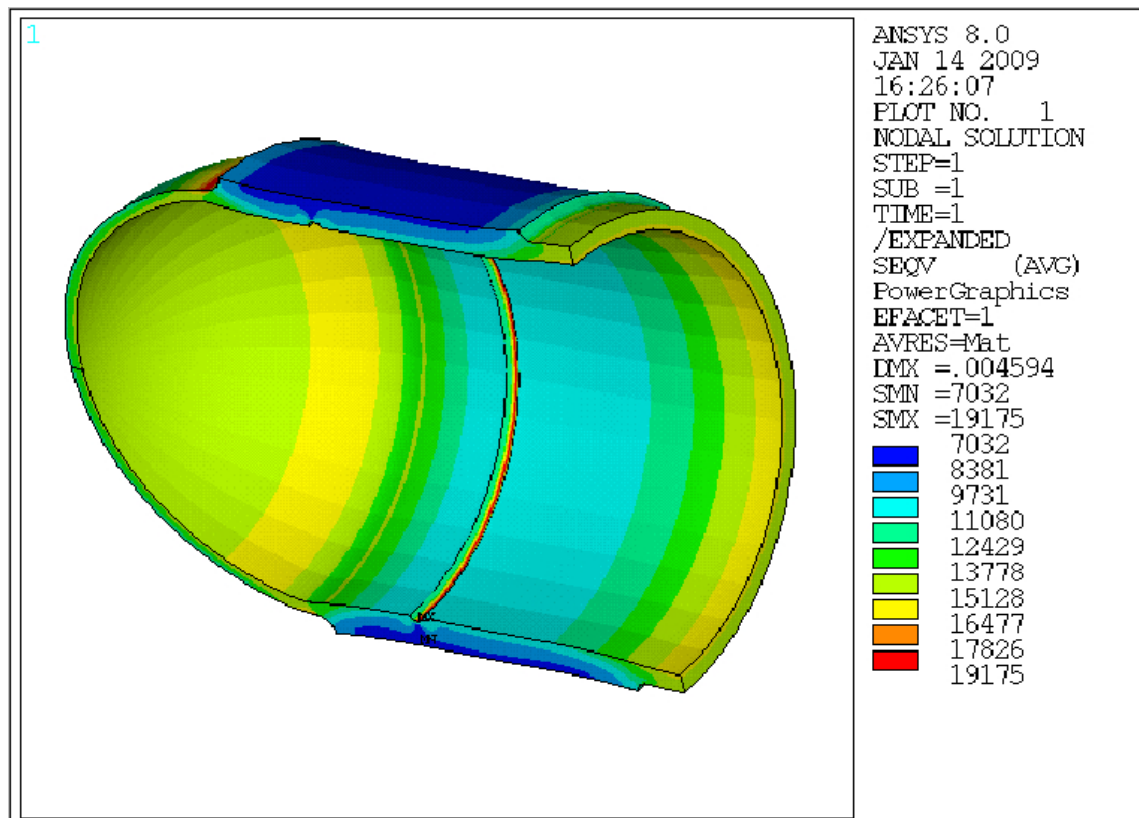
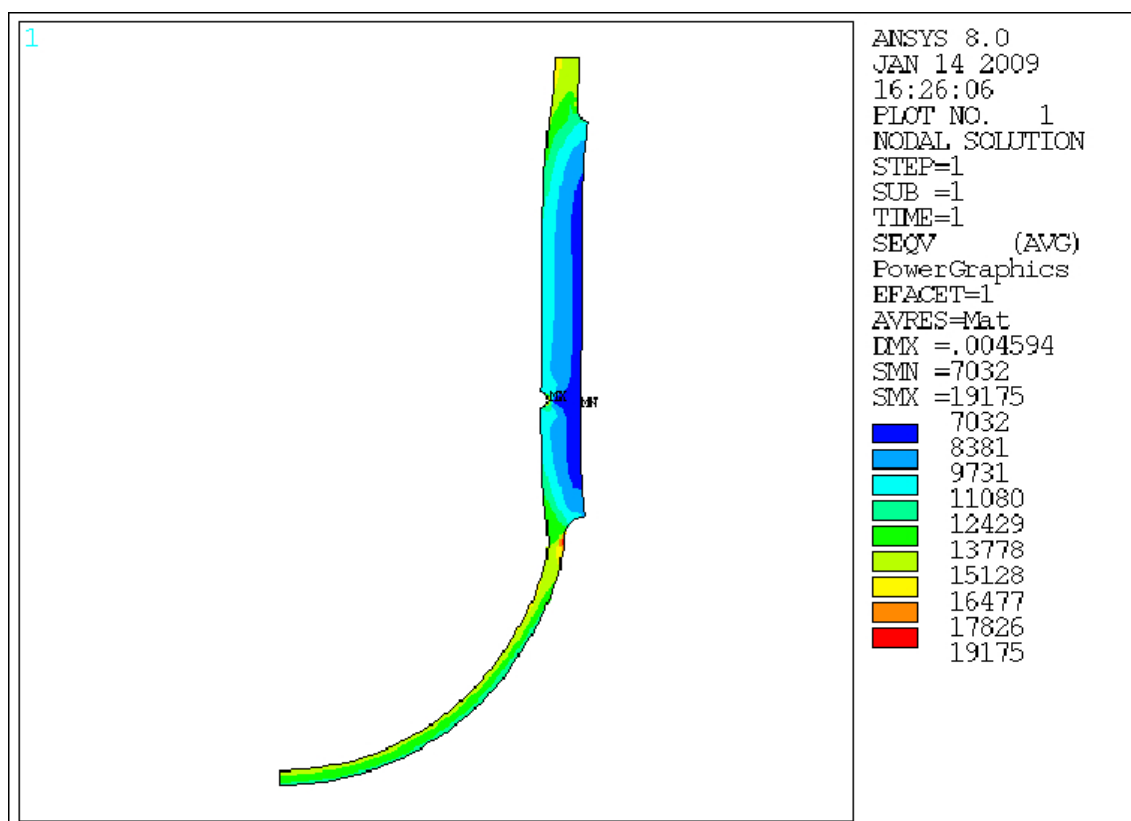


Figura 5.22. Distribuição da tensão máxima na solução otimizada.



**Figura 5.23.** Distribuição da tensão máxima ao longo de uma seção otimizada do vaso.

**Tabela 5.11.** Solução do problema Vaso de Pressão.\*

	Elementos Finitos sem Superfície de Resposta	Elementos Finitos com Superfície de Resposta
$x_1$	0.255	0.269
$x_2$	0.619	0.616
Área da Seção Viga I + Área da Seção Viga II	12.354	12.484
Tempo (ms)	2309969	40563

\* Todos os resultados foram obtidos sobre o mesmo índice de performance.

## 5.5 Otimização multiobjetivo

Nesta seção serão apresentados 3 problemas de otimização multiobjetivo. Os dois primeiros problemas são aplicações matemáticas do conceito de otimização multiobjetivo, enquanto o último problema mostra a aplicação da otimização multiobjetivo ao projeto de uma viga de aço.

### 5.5.1 Multiobjetivo Simples (Multiobjetivo Simples.txt)

Um problema bastante simples e largamente utilizado para testar otimizadores multi-objetivos é o problema f2 formulado por **Schaffer (1984)**. O problema consiste na minimização simultânea de duas funções objetivo. Embora seja um problema fácil para um algoritmo evolucionário, ele é comumente resolvido pela maioria dos autores como um ponto

inicial de referência, principalmente para a investigação da distribuição da população ao longo da Fronteira de Pareto. O problema é definido pelas seguintes funções:

$$\begin{aligned} f_1 &= x^2 \\ f_2 &= (x-2)^2 \end{aligned} \quad (5.3)$$

Na figura 5.24 mostram-se as curvas de nível, para ambas as funções objetivo.

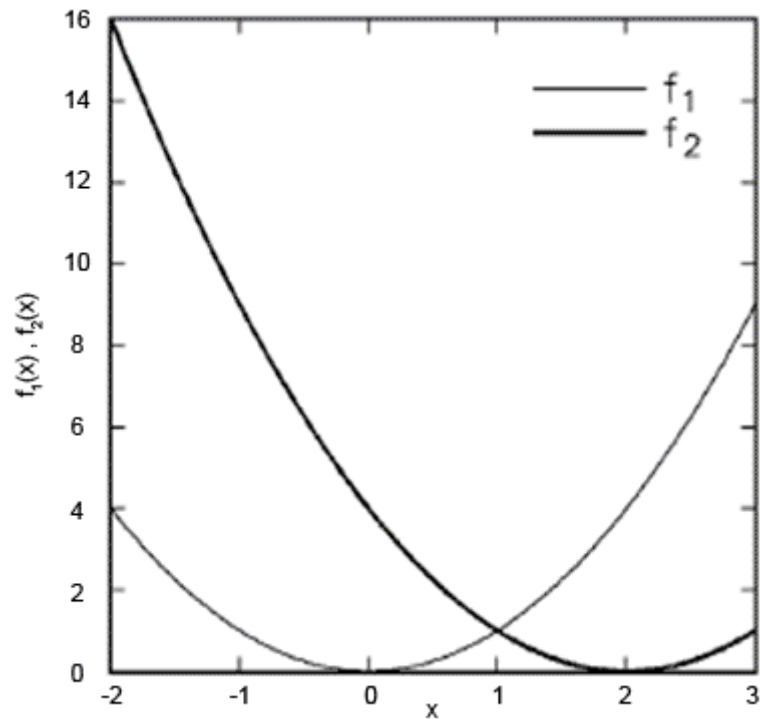


Figura 5.24. Representação gráfica do problema Multiobjetivo Simples.

#### 5.5.1.1 Solução do problema Multiobjetivo Simples

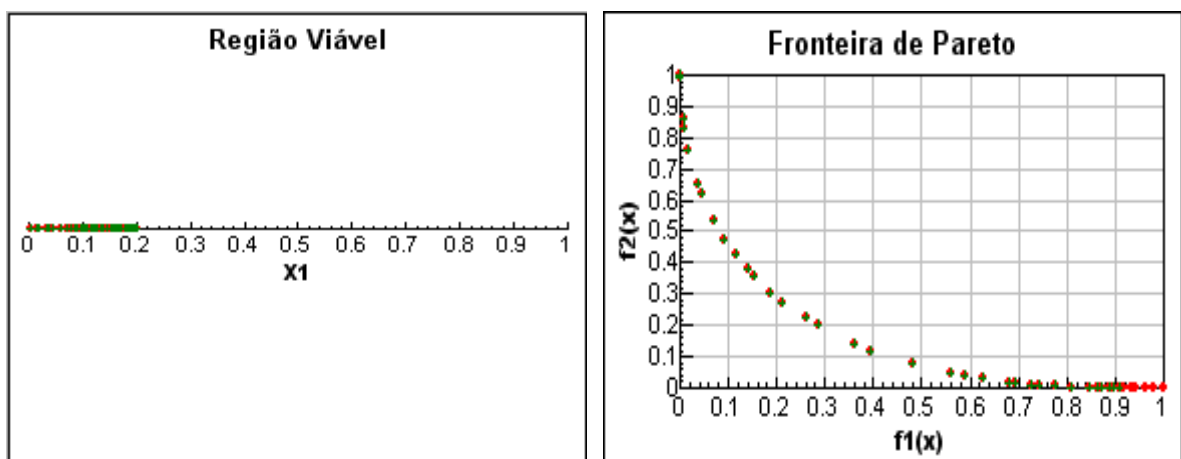


Figura 5.25. Solução do problema Multiobjetivo Simples.

Tabela 5.12. Solução do problema Multiobjetivo Simples.

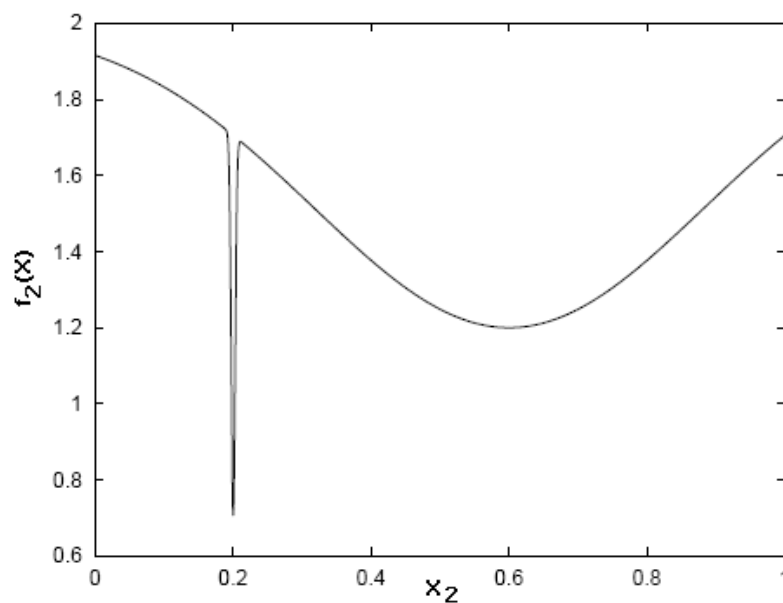
X	F <sub>1</sub>	F <sub>2</sub>	X	F <sub>1</sub>	F <sub>2</sub>
0.05	0.00	3.80	1.38	1.90	0.38
0.20	0.04	3.24	1.54	2.37	0.21
0.36	0.13	2.69	1.58	2.50	0.18
0.60	0.36	1.96	1.66	2.76	0.12
0.79	0.62	1.46	1.72	2.96	0.08
0.91	0.83	1.19	1.80	3.24	0.04
1.00	1.00	1.00	1.85	3.42	0.02
1.09	1.19	0.83	1.89	3.57	0.01
1.19	1.42	0.66	1.93	3.72	0.00
1.27	1.61	0.53	1.99	3.96	0.00

### 5.5.2 Multiobjetivo Complexo (Multiobjetivo Complexo.txt)

O problema proposto originalmente por **Kalyanmoy (1999)** e estudado por muitos outros pesquisadores, como por exemplo **Rabeau (2005)**, propõe a otimização de duas funções matemáticas descritas na equação 5.4.

$$\begin{aligned}
 f_1 &= x_1 \\
 f_2 &= \frac{2 - e^{-\left(\frac{(x_2-0.2)^2}{0.004}\right)} - 0.8 * e^{-\left(\frac{(x_2-0.6)^2}{0.4}\right)}}{x_1}
 \end{aligned} \tag{5.4}$$

A singularidade presente na função  $f_2$  é o principal desafio deste problema, como pode ser visto na figura 5.26.



**Figura 5.26.** Representação da função  $f_2(X)$  supondo  $x_1$  constante. Kalyanmoy (1999).



### 5.5.2.1 Solução do problema Multiobjetivo Complexo

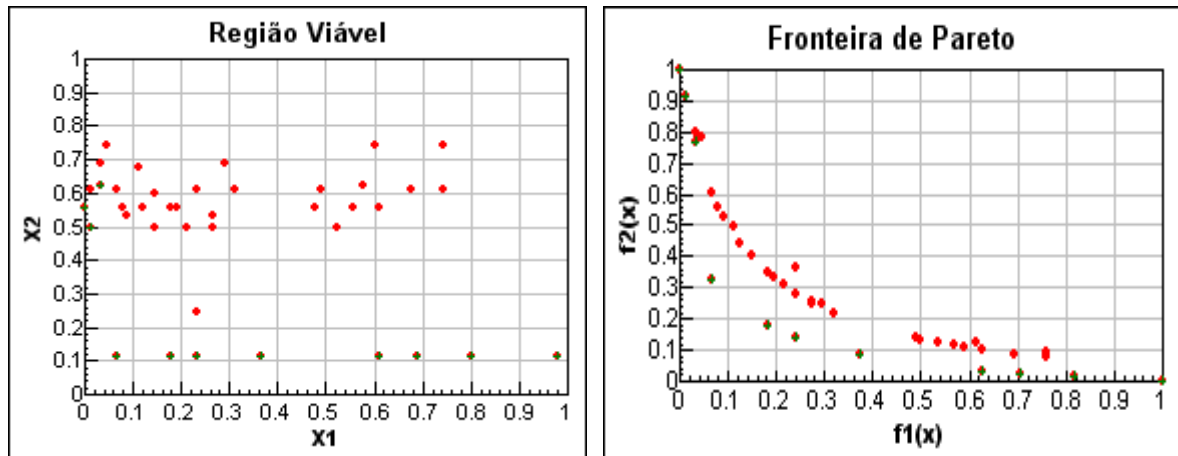


Figura 5.27. Solução do problema Multiobjetivo Complexo.

Tabela 5.13. Solução do problema Multiobjetivo Complexo.

$X_1$	$X_2$	$F_1$	$F_2$	$X_1$	$X_2$	$F_1$	$F_2$
0.11	0.55	0.11	11.02	0.65	0.20	0.65	1.09
0.16	0.20	0.16	4.41	0.43	0.20	0.43	1.64
0.98	0.20	0.98	0.72	0.31	0.20	0.31	2.28
0.82	0.20	0.82	0.86	0.72	0.20	0.72	0.98
0.10	0.60	0.10	12.00	0.13	0.66	0.13	9.37
0.26	0.20	0.26	2.71				

### 5.5.3 Viga I (Vigal.txt)

O objetivo do problema originalmente proposto por **Castro (2001)** é determinar o conjunto ótimo de Pareto decorrente da otimização multiobjetivo da viga em seção I apresentada na figura 28. As propriedades da viga e os valores das cargas são:

- $E = 2 \times 10^4$  kN/cm<sup>2</sup> (módulo de Young).
- $kg = 16$  kN/cm<sup>2</sup> (tensão de projeto da viga).
- $P = 600$  kN (carga vertical em  $L/2$ ).
- $Q = 50$  kN (carga horizontal em  $L/2$ ).

O vetor das variáveis de projeto são as dimensões da viga apresentadas em centímetros, conforme mostrado na figura 5.28. O objetivo é minimizar a área da seção transversal (equação 5.5) e o deslocamento estático máximo em  $L/2$  (equação 5.6), sujeita a restrição de tensão (equação 5.7).

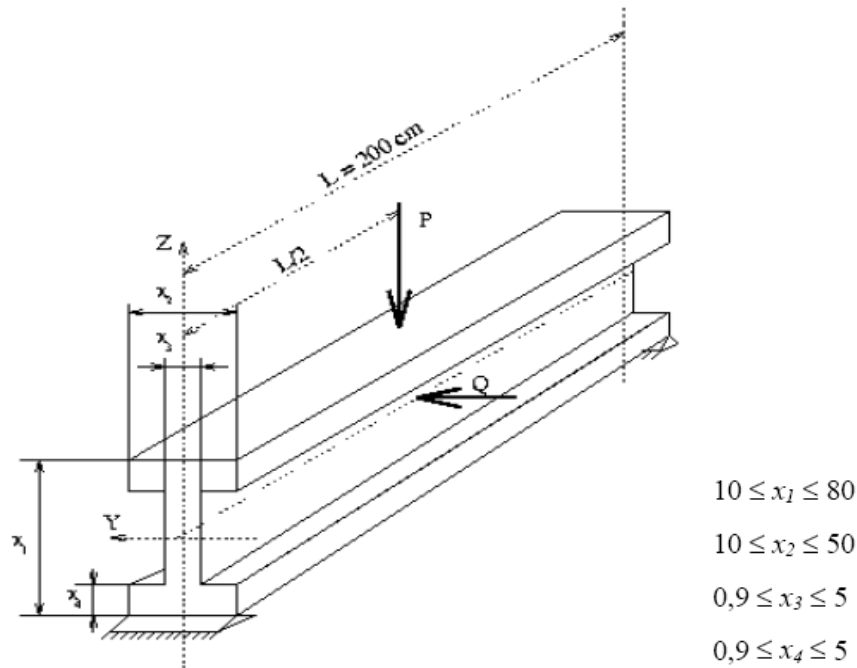


Figura 5.28. Ilustração do problema Viga I.

$$f_1(X) = 2x_2x_4 + x_3(x_1 - 2x_4) \quad (5.5)$$

$$f_2(X) = \frac{60000}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} \quad (5.6)$$

$$G_1(X) : 16 - \frac{180000}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} - \frac{15000}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3} \geq 0 \quad (5.7)$$

O desenvolvimento completo das equações do problema, considerando o carregamento bi-axial, pode ser visto em **Castro (2001)**.

#### 5.5.3.1 Solução do problema Viga I

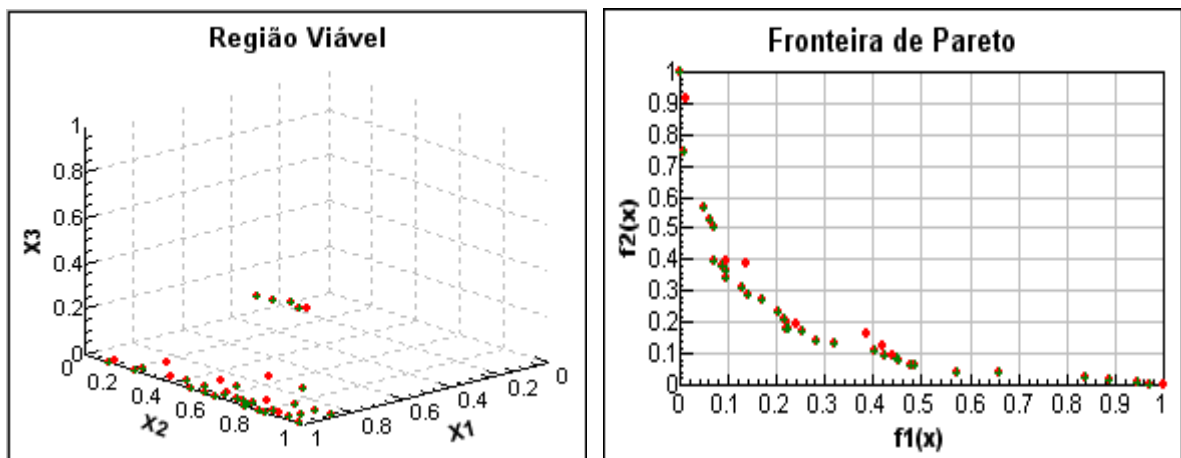


Figura 5.29. Solução do problema Viga I.

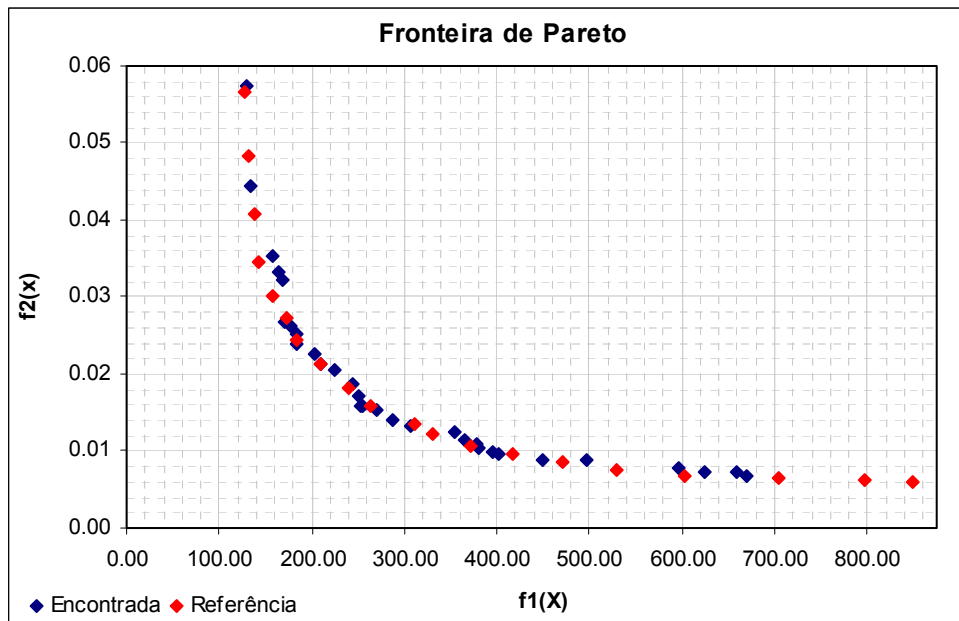


Figura 5.30. Comparação entre a solução obtida e solução de referência.

Tabela 5.14. Solução do problema Viga I.

$x_1$	$x_2$	$x_3$	$x_4$	$F_1$	$F_2$	$x_1$	$x_2$	$x_3$	$x_4$	$F_1$	$F_2$
78.84	38.86	0.90	2.63	270.63	0.02	79.93	49.22	0.90	3.34	394.71	0.01
71.18	49.21	0.90	1.08	168.41	0.03	79.93	14.21	0.90	4.90	202.38	0.02
71.18	42.96	0.90	1.11	157.44	0.04	79.93	19.16	0.90	4.89	250.52	0.02
79.90	39.21	0.90	2.39	255.03	0.02	78.84	46.37	0.90	1.09	170.08	0.03
79.93	44.22	2.95	4.71	624.56	0.01	77.71	34.17	0.90	1.62	177.73	0.03
77.74	42.96	0.90	3.67	378.69	0.01	79.93	29.17	0.90	4.17	307.71	0.01
77.74	39.20	0.91	1.81	209.35	0.02	79.93	33.86	0.90	4.69	381.10	0.01
75.56	46.71	0.90	4.69	497.70	0.01	78.81	46.70	0.91	4.12	449.03	0.01
79.93	31.99	0.90	1.80	183.86	0.02	79.93	41.37	2.95	4.69	596.17	0.01
79.93	41.71	0.90	2.63	286.60	0.01	77.74	36.37	0.90	4.17	365.79	0.01
79.90	49.21	2.95	4.70	670.55	0.01	77.74	37.33	0.90	1.55	182.90	0.03
76.65	29.84	0.90	4.94	354.91	0.01	77.74	19.21	0.90	4.24	225.23	0.02
77.74	46.37	2.95	4.94	658.32	0.01	71.18	46.36	0.90	1.10	164.07	0.03
79.93	42.92	0.90	3.92	401.37	0.01	76.62	26.39	0.90	3.42	243.31	0.02
79.93	49.17	0.90	1.87	252.47	0.02	62.24	38.66	0.91	0.97	129.87	0.06
71.18	32.33	0.90	1.11	133.84	0.04						

## 5.6 Otimização multidisciplinar

Nesta seção serão apresentados 4 problemas multidisciplinares. Os dois primeiros problemas são aplicações matemáticas do conceito de otimização multidisciplinar, enquanto os dois últimos problemas mostram a aplicação da otimização multidisciplinar ao projeto de uma viga de aço e de um redutor de automóvel.

### 5.6.1 Multidisciplinar Simples (Multidisciplinar Simples.txt)

Um problema bastante simples e largamente utilizado para testar otimizadores multi-objetivos é o problema f2 formulado por **Schafer (1984)**. Esse problema foi resolvido no item 5.5.1 deste relatório com o emprego de uma abordagem multiobjetivo. Nesta seção o mesmo problema será resolvido utilizando uma abordagem multidisciplinar. O problema pode ser formulado modificando-se a equação 5.3 para uma abordagem multidisciplinar.

*Disciplina 1*

$$f_{1,1} = x_c^2$$

*Disciplina 2*

$$f_{2,1} = (x_c - 2)^2$$

(5.8)

A topologia do problema f2 formulado de forma multidisciplinar é mostrada na figura 5.31.

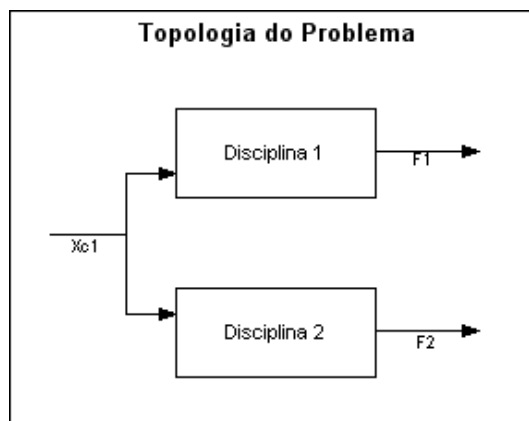


Figura 5.31. Topologia do problema Multidisciplinar Simples.

#### 5.6.1.1 Solução do problema Multidisciplinar Simples

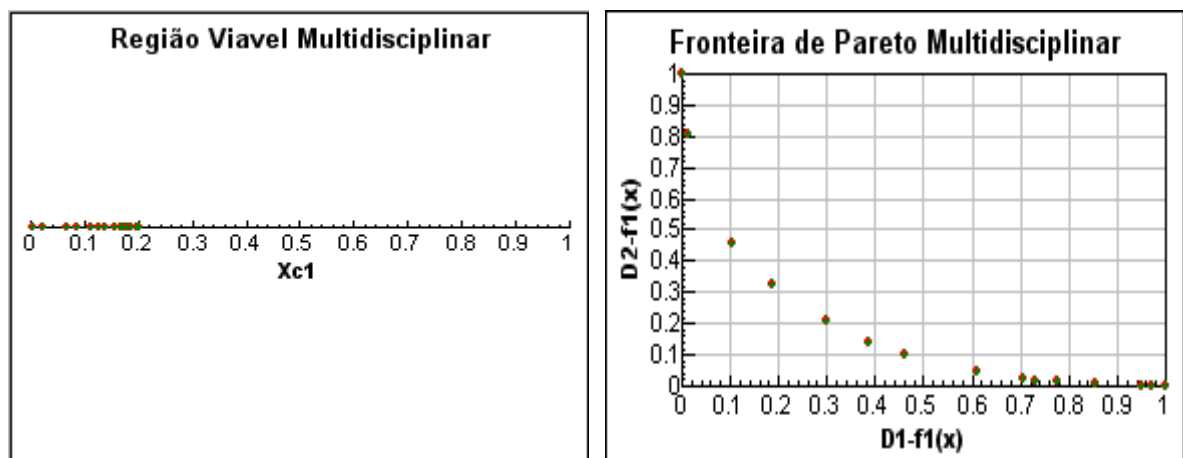


Figura 5.32. Solução do problema Multidisciplinar Simples.

Tabela 5.15. Solução do problema Multidisciplinar Simples.

Acoplado	Disciplina I		Disciplina II	
$X_c$	$X_{c1,1}$	$f_{1,1}$	$X_{c2,1}$	$f_{2,1}$
0.87	0.86	0.74	0.88	1.25
1.77	1.76	3.10	1.78	0.05
0.66	0.65	0.42	0.67	1.77
1.25	1.24	1.54	1.26	0.55
1.69	1.68	2.82	1.70	0.09
0.23	0.22	0.05	0.24	3.10
1.57	1.56	2.43	1.58	0.18
1.86	1.85	3.42	1.87	0.02
1.37	1.36	1.85	1.38	0.38
1.98	1.97	3.88	1.99	0.00
1.96	1.95	3.80	1.97	0.00
1.72	1.71	2.92	1.73	0.07
1.10	1.09	1.19	1.11	0.79
0.03	0.02	0.00	0.04	3.84
2.01	2.00	4.00	2.00	0.00

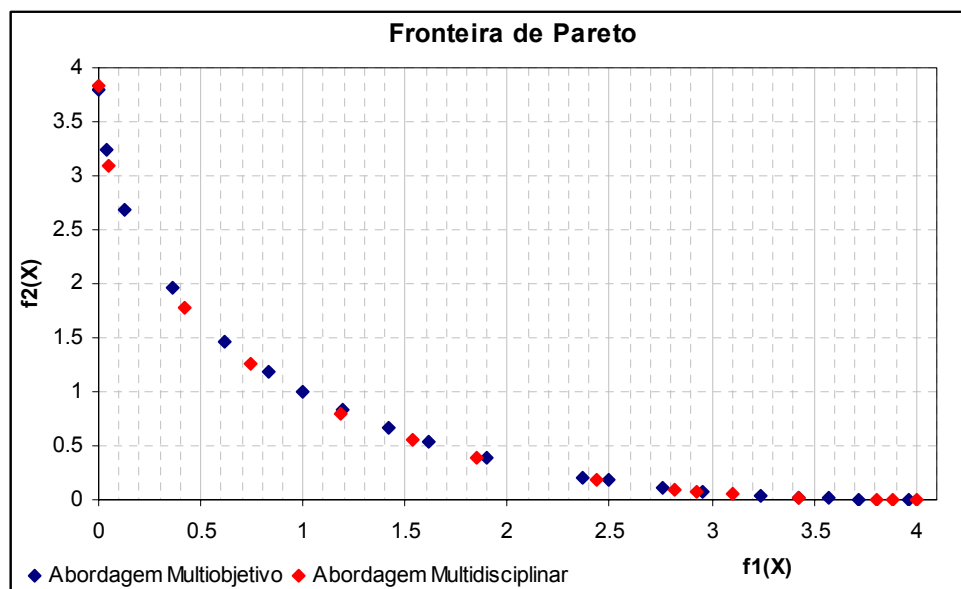


Figura 5.33. Comparação entre a abordagem Multiobjetivo e Multidisciplinar.

### 5.6.2 Multidisciplinar Acoplado (Multidisciplinar Acoplado.txt)

Grande parte das estratégias de otimização multidisciplinar possuem a limitação de não resolver problemas com acoplamento forte. O problema proposto a seguir tem como característica principal justamente a existência de acoplamentos fortes.

Deseja minimizar as funções objetivo  $f_1 = f_1(x_1, f_2)$  e  $f_2 = f_2(x_2, f_1)$  como mostrado na equação 5.9.

*Disciplina 1*

$$f_{1,1} = (x_{1,1} - 2)^2 + (x_{1,2} - 2)^2$$

*Disciplina 2*

$$f_{2,1} = (x_{2,1} - 8)^2 + (x_{2,2} - 8)^2 \quad (5.9)$$

*Acoplamentos*

$$A_1 : x_{1,2} = f_{2,1}$$

$$A_2 : x_{2,2} = f_{1,1}$$

A topologia do problema Multidisciplinar Acoplado formulado de forma multidisciplinar é mostrada na figura 5.34.

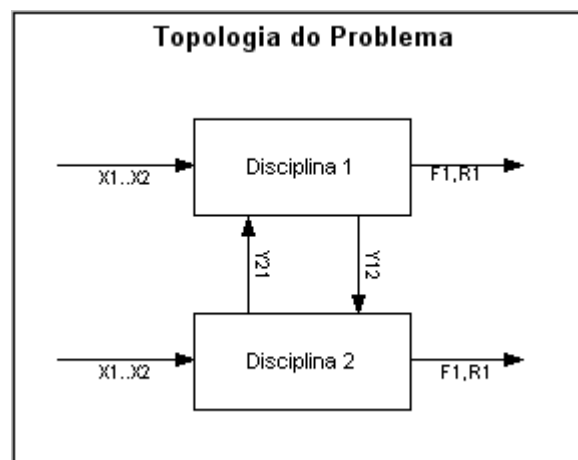


Figura 5.34. Topologia do problema Multidisciplinar Acoplado.

### 5.6.2.1 Solução do problema Multidisciplinar Acoplado

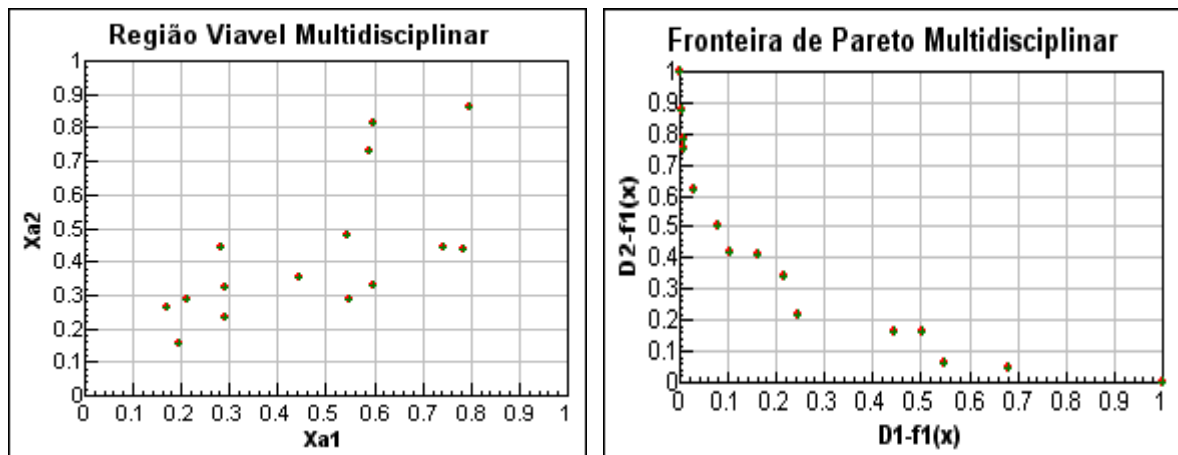


Figura 5.35. Solução do problema Multidisciplinar Acoplado.

### 5.6.3 Múltiplas Funções Objetivo (4Variaveis4Funcoes.txt)

Este problema foi proposto originalmente em **Rabeau (2006)** para avaliar o algoritmo COSMOS por ele formulado.

*Disciplina 1*

$$f_{1,1} = (x_{1,1} - 1)^2 + (x_c - 3)^2$$

$$f_{1,2} = (x_{1,2} - 3)^2 + (x_c - 7)^2$$

*Disciplina 2*

$$f_{2,1} = (x_{2,1} - 3)^2 + (x_c - 5)^2$$

$$f_{2,2} = (x_{2,2} - 5)^2 + (x_c - 9)^2$$

(5.10)

$$x_{1,1} \in [0;20] ; x_{1,2} \in [0;20] ; x_{2,1} \in [0;20] ; x_{2,2} \in [0;20] ; x_c \in [0;10]$$

A topologia do problema Múltiplas Funções Objetivo formulado de forma multidisciplinar é mostrada na figura 5.36.

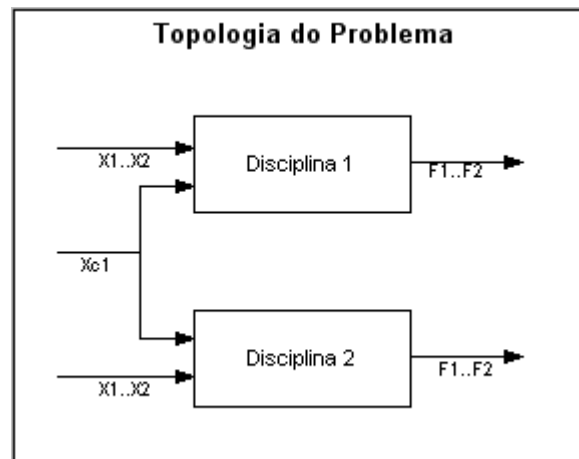


Figura 5.36. Topologia do problema Múltiplas Funções Objetivo

### 5.6.3.1 Solução do problema Múltiplas Funções Objetivo

$$x_{1,1} = 1 ; x_{1,2} = 3 ; x_{2,1} = 3 ; x_{2,2} = 5$$

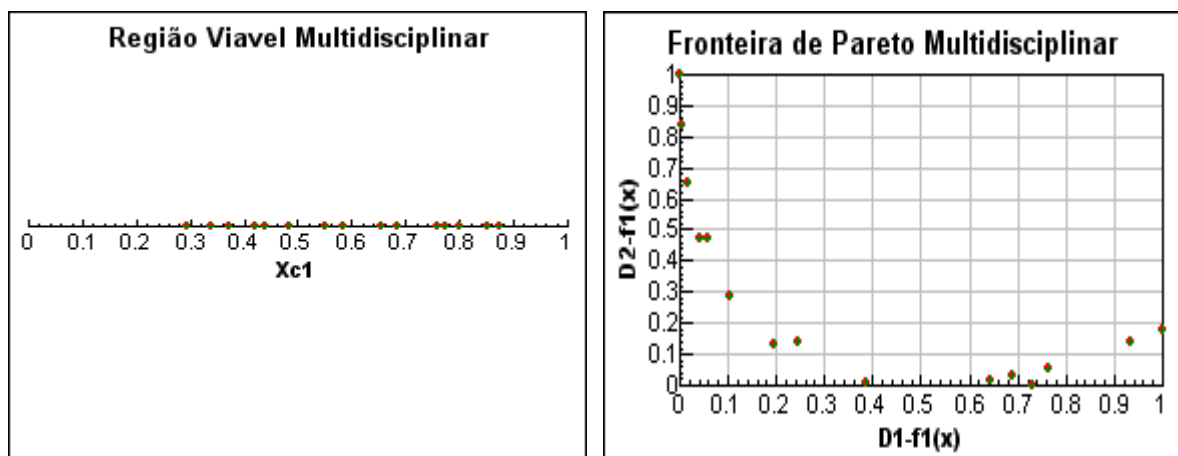


Figura 5.37. Solução do problema Múltiplas Funções Objetivo

Tabela 5.16. Solução do problema Múltiplas Funções Objetivo.

Acoplado		Disciplina 1					Disciplina 2				
N	$x_c$	$x_c$	$x_1$	$x_2$	$f_1$	$f_2$	$x_c$	$x_1$	$x_2$	$f_1$	$f_2$
1	3.05	3.06	1.25	3	0.07	15.52	3.06	3	5	3.76	35.28
2	5.80	5.80	1	3.13	7.84	1.46	5.79	3	5.16	0.62	10.33
3	3.32	3.31	1	3.14	0.10	13.64	3.33	3	5	2.79	32.15
4	7.83	7.82	1	3	23.23	0.67	7.82	3	5.08	7.95	1.40
5	2.94	2.95	1	3	0.00	16.40	2.95	3	5	4.20	36.60
6	4.74	4.75	0.94	3	3.07	5.06	4.75	3	5	0.06	18.06
7	4.33	4.32	1	2.97	1.74	7.18	4.34	3	5	0.44	21.72
8	4.37	4.38	0.94	3	1.91	6.86	4.38	3	5	0.38	21.34
9	3.61	3.62	1.56	3	0.70	11.42	3.62	3	5	1.90	28.94
10	8.73	8.72	1	3	32.72	2.96	8.72	2.77	5	13.89	0.08
11	3.12	3.13	1.25	3	0.08	14.98	3.13	3	5	3.50	34.46
12	5.64	5.63	1	2.81	6.92	1.91	5.63	3	5.25	0.40	11.42
13	3.22	3.23	0.79	3	0.10	14.21	3.23	3	5	3.13	33.29
14	3.45	3.46	1	2.82	0.21	12.56	3.46	3	5	2.37	30.69
15	6.65	6.64	1	2.97	13.25	0.13	6.64	3	5	2.69	5.57

É interessante observar que a solução apresentada por Rabeau define que a Fronteira de Pareto é dada por  $X_c \in [5 ; 7]$  quando na verdade  $X_c \in [3 ; 9]$ , fato facilmente verificável uma vez que a solução  $X_c = 3$  ou  $X_c = 9$  não é dominada por nenhuma solução com  $X_c \in [5 ; 7]$ .

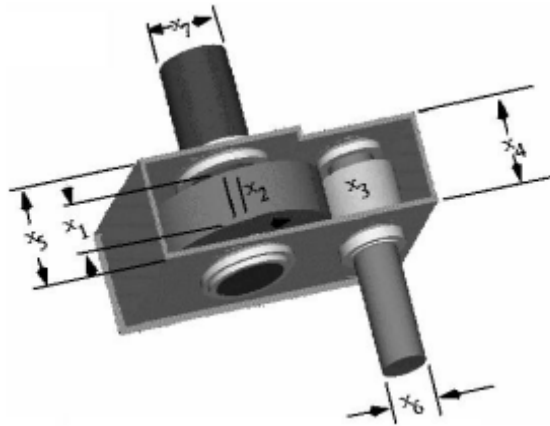
## 5.7 Redutor de Velocidade (Redutor.txt)

Nesta seção será apresentado um exemplo de aplicação do método de otimização multidisciplinar baseado em valores de acoplamento. O problema analisado em **Rabeau (2006)** consiste na otimização de um redutor de velocidade mono estágio e foi proposto originalmente em **Golinski (1970)**. Este mesmo problema foi formulado pela primeira vez de maneira multidisciplinar em **Gunawan (2003)**.

O objetivo é minimizar o volume e a constante elástica de cada árvore respeitando as restrições mecânicas do redutor. Uma representação tridimensional do redutor de velocidade é mostrada da figura 5.38, juntamente com as variáveis de projeto descritas na tabela 5.17.

A proposição de Gunawan foi separar a formulação de cada árvore em uma disciplina mantendo a formulação dos dentes como o vínculo no problema multidisciplinar.





**Figura 5.38.** Representação tridimensional do Redutor

**Tabela 5.17.** Variáveis do problema Redutor como formulado por Golinski.\*

Variável	Mínimo	Máximo	Resolução
$x_1$ : espaçamento entre dentes da engrenagem	2.6	3.6	0.1
$x_2$ : módulo dos dentes da engrenagem	0.7	0.8	0.01
$x_3$ : número de dentes da engrenagem menor	17	28	1
$x_4$ : espaçamento entre rolamentos da árvore menor	7.3	8.3	0.1
$x_5$ : espaçamento entre rolamentos da árvore maior	7.3	8.3	0.1
$x_6$ : diâmetro da árvore menor	2.9	3.9	0.1
$x_7$ : diâmetro da árvore maior	5.0	5.5	0.1


\* A variável  $x_3$  é uma variável natural e portanto deve ser tratada obrigatoriamente com resolução 1.

A correlação entre as variáveis propostas por Golinski e as variáveis da formulação multidisciplinar de Gunawan é mostrada na tabela 5.18.

**Tabela 5.18.** Variáveis do problema Redutor como formulado por Gunawan.

Variável	Disciplina	Denominação
$x_1$ : espaçamento entre dentes da engrenagem	1,2	$x_{c,1}$
$x_2$ : módulo dos dentes da engrenagem	1,2	$x_{c,2}$
$x_3$ : número de dentes da engrenagem menor	1,2	$x_{c,3}$
$x_4$ : espaçamento entre rolamentos da árvore menor	1	$x_{1,1}$
$x_5$ : espaçamento entre rolamentos da árvore maior	2	$x_{2,1}$
$x_6$ : diâmetro da árvore menor	1	$x_{1,2}$
$x_7$ : diâmetro da árvore maior	2	$x_{2,2}$

Na equação 5.11 é mostrado o sub-sistema para a disciplina 1, enquanto na equação 5.12 é mostrado o sub-sistema para a disciplina 2. Por fim, na equação 5.13 são apresentadas as restrições globais do problema.<sup>32</sup>

<sup>32</sup>  Existe uma divergência entre a restrição  $g_{s,2}$  apresentada por Rabeau e por Gunawan. Optou-se pela expressão apresentada por Gunawan, mais coerente ao problema.

$$\left\{ \begin{array}{l} f_{1,1} = 0.7854x_{c1}x_{c2}^2 \left( \frac{10x_{c3}^2}{3} + 14.933x_{c3} - 43.0934 \right) - 1.508x_{c1}x_{1,2}^2 + 0.7854x_{1,1}x_{1,2}^2 \\ f_{1,2} = \frac{\sqrt{\left( \frac{745x_{1,1}^2}{x_{c2}x_{c3}} \right)^2 + 1.69 * 10^7}}{0.1x_{1,2}^2} \\ g_{1,1} : \frac{x_{1,1}^3}{x_{c2}x_{c3}x_{1,2}^4} - \frac{1}{1.93} \leq 0 \\ g_{1,2} : 1.9 - x_{1,1} + 1.5x_{1,2} \leq 0 \\ g_{1,3} : f_{1,2} - 1300 \leq 0 \end{array} \right. \quad (5.11)$$

$$\left\{ \begin{array}{l} f_{2,1} = -1.508x_{c1}x_{2,2}^2 + 7.477x_{2,2}^3 + 0.7854x_{2,1}x_{2,2}^2 \\ f_{2,2} = \frac{\sqrt{\left( \frac{745x_{2,1}}{x_{c2}x_{c3}} \right)^2 + 1.575 * 10^8}}{0.1x_{2,2}^3} \\ g_{2,1} : \frac{x_{2,1}^3}{x_{c2}x_{c3}x_{2,2}^4} - \frac{1}{1.93} \leq 0 \\ g_{2,2} : 1.9 - x_{2,1} + 1.1x_{2,2} \leq 0 \\ g_{2,3} : f_{2,2} - 1100 \leq 0 \end{array} \right. \quad (5.12)$$

$$\left\{ \begin{array}{l} g_{s,1} : \frac{1}{x_{c1}x_{c2}^2x_{c3}} - \frac{1}{27} \leq 0 \\ g_{s,2} : \frac{1}{x_{c1}x_{c2}^2x_{c3}^2} - \frac{1}{397.5} \leq 0 \\ g_{s,3} : x_{c2}x_{c3} - 40 \leq 0 \\ g_{s,4} : \frac{x_{c1}}{x_{c2}} - 12 \leq 0 \\ g_{s,5} : f_{2,2} - 1100 \leq 0 \end{array} \right. \quad (5.13)$$

O acoplamento do problema é dado unicamente pelo vetor de variáveis comuns, conforme mostrado na figura 5.39.

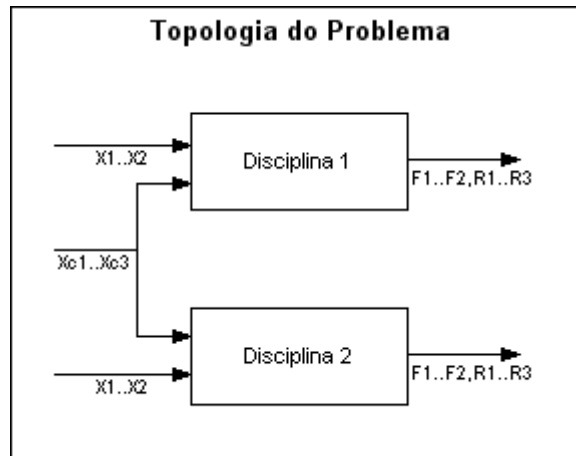


Figura 5.39. Topologia do problema Redutor


### 5.7.1.1 Solução do problema Redutor

Tabela 5.19. Solução do problema Redutor.

Acoplado				Disciplina 1								Disciplina 2							
N	$x_{c1}$	$x_{c2}$	$x_{c3}$	$x_{c1}$	$x_{c2}$	$x_{c3}$	$x_1$	$x_2$	$f_1$	$f_2$	$x_{c1}$	$x_{c2}$	$x_{c3}$	$x_1$	$x_2$	$f_1$	$f_2$		
1	2.82	0.75	23.23	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	2.9	0.76	23.3	7.4	5.0	970.6	1020.8		
2	2.70	0.79	25.36	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	2.8	0.80	25.5	7.6	5.1	1037.3	959.4		
3	2.69	0.80	20.44	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	2.8	0.80	20.5	8.0	5.5	1306.3	774.3		
4	3.47	0.71	17.94	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.6	0.72	18.0	7.6	5.1	1005.9	978.6		
5	3.46	0.77	20.79	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.5	0.78	20.9	7.7	5.2	1072.1	913.1		
6	3.44	0.77	19.71	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.5	0.78	19.8	7.9	5.4	1204.4	819.6		
7	3.29	0.75	18.24	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.4	0.76	18.3	7.8	5.3	1141.2	870.9		
8	2.80	0.78	21.93	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	2.9	0.79	22.0	7.8	5.3	1162.4	861.0		
9	3.24	0.79	25.18	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.3	0.80	25.3	7.6	5.1	1017.7	959.6		
10	3.47	0.72	19.17	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.6	0.73	19.3	7.8	5.3	1132.7	870.2		
11	2.91	0.74	20.10	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.0	0.75	20.2	7.9	5.4	1226.4	820.5		
12	3.51	0.74	22.19	2.7	0.79	20.3	7.6	3.8	2189.5	3399.7	3.6	0.75	22.3	7.8	5.3	1132.7	862.4		
13	2.81	0.79	25.95	2.9	0.80	26.0	7.6	3.8	3810.9	3187.1	2.9	0.80	26.0	8.0	5.5	1301.8	766.8		
14	2.62	0.78	20.22	2.9	0.80	26.0	7.6	3.8	3810.9	3187.1	2.7	0.79	20.3	7.8	5.3	1170.9	864.1		
15	3.51	0.76	26.38	2.9	0.80	26.0	7.6	3.8	3810.9	3187.1	3.6	0.77	26.5	7.9	5.4	1200.0	810.0		

## 5.8 Projeto do hidrofólio de uma embarcação catamarã <sup>33</sup>

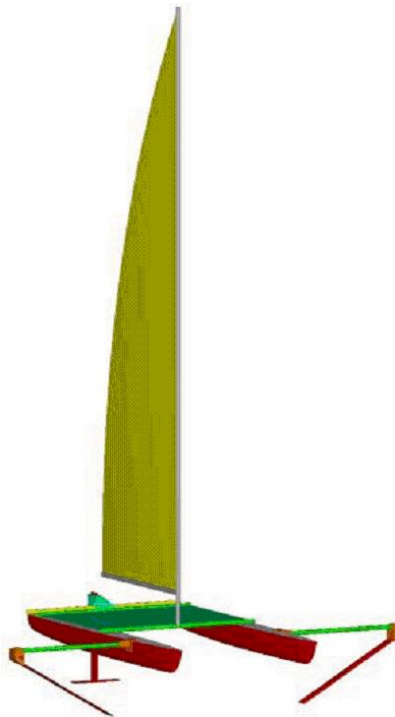
O problema consiste no projeto do hidrofólio de uma embarcação do tipo catamarã. Este projeto é um importante objeto de estudo, pois está inserido no moderno contexto de otimização multidisciplinar, bem como no conceito de projeto e processamento distribuído, apresentados neste trabalho.

<sup>33</sup>  O desenvolvimento desse exemplo foi apresentado no 12º International Congress of the International Maritime Association of the Mediterranean (IMAM 2007), Varna, Bulgária, 2-6 September, 2007 e, a seguir, convidado para ser publicado Journal of Engineering for the Maritime Environment, Vol. 222 Part M, pp 91-104.

Em 2005 a Universidade de São Paulo (Brasil) e a Ecole Centrale de Nantes (França) desenvolveram uma pesquisa sobre o projeto desse tipo de embarcação, e parte dos resultados obtidos pode ser vista em **Dreyer (2005)**.

Existem diversas opções para melhorar a velocidade de uma embarcação: redução de peso, aprimoramento hidrodinâmico, planeio, melhorias na propulsão, etc. A utilização de hidrofólios é uma maneira eficiente de melhorar a velocidade, pois reduz consideravelmente o arrasto. Por essa razão, a cada dia, mais e mais barcos de alta velocidade são construídos utilizando hidrofólios.

Um catamarã é uma embarcação construída sobre dois cascos finos separados transversalmente. Graças a essa configuração, o catamarã pode alcançar boas velocidades, pois os cascos são finos e a relação comprimento por boca (de cada um) é excelente. Outra vantagem é o conforto, pois a grande largura da superestrutura permite a instalação de cabines confortáveis. Uma representação de um catamarã com hidrofólios é mostrada na figura 5.40.



**Figura 5.40.** *Layout* básico de um catamarã com hidrofólios. Dreyer (2005).

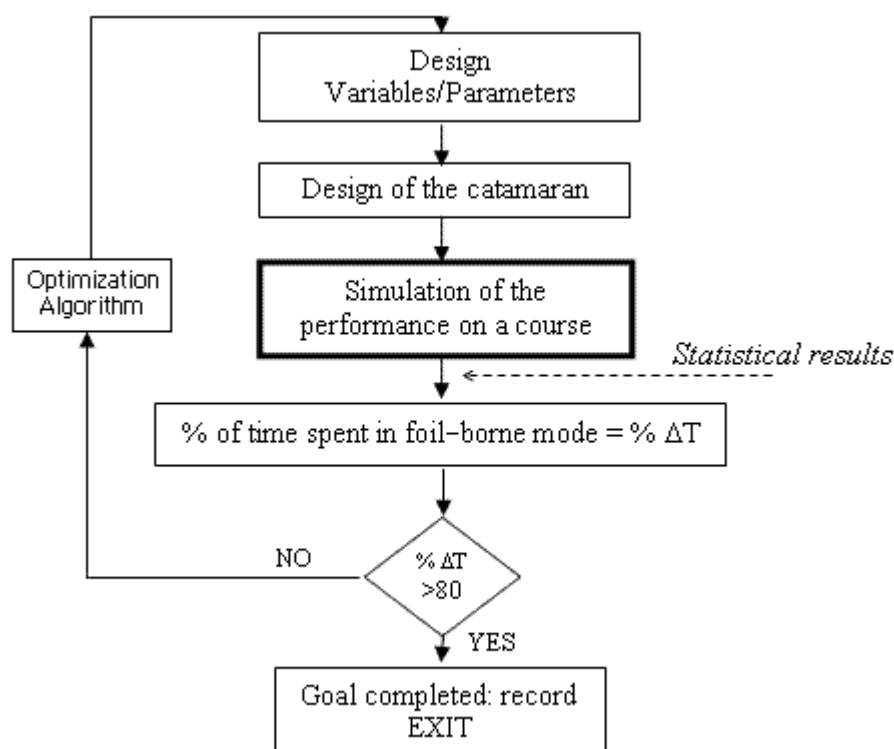
A composição do catamarã com dois cascos assegura grande estabilidade transversal quando este se desloca com os cascos submersos na água. Porém, uma vez que a embarcação alcance uma velocidade específica, o catamarã entra no regime de planeio, onde a sustentação dos hidrofólios suspende os cascos para fora da água, minimizando consideravelmente o arrasto.

Durante o regime de planeio, os cascos da embarcação não tocam a água e, portanto, não interferem na performance do catamarã. Por essa razão, as características dos cascos são importantes apenas para garantirem que a embarcação possa atingir o regime de planeio, fazendo desta uma das restrições mais importante do problema.

### 5.8.1 Descrição do problema multidisciplinar

Assim como o projeto de outros sistemas complexos, o catamarã pode ser dividido em diversos subsistemas, sendo excelente para o desenvolvimento em ambiente distribuído.

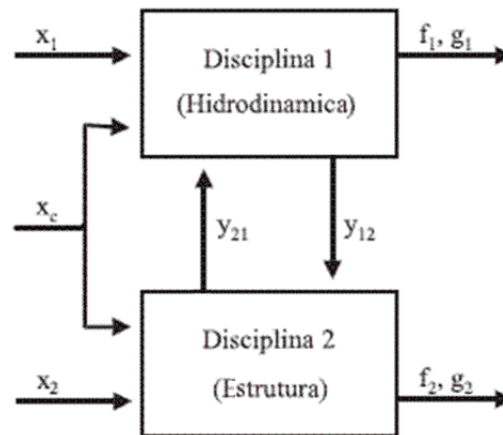
O objetivo é otimizar o projeto para alcançar a maior velocidade média ao longo de uma rota predeterminada. O fluxograma do processo de projeto é mostrada na figura 5.41.



**Figura 5.41.** Fluxograma do processo de projeto. Dreyer (2005).

O problema é que esta não é uma otimização simples, pois existem aspectos hidrodinâmicos e aspectos estruturais, cada qual com seus próprios objetivos e restrições, que podem ser conflitantes.

A abordagem proposta para o problema é separar o projeto em dois subsistemas, que, em geral, são desenvolvidos por equipes especializadas. O acoplamento entre as disciplinas é representado pelas funções de acoplamento  $y_{ij}$  e pelo vetor de variáveis comuns  $X_c$ . Essa divisão do problema é mostrada na figura 5.42.



**Figura 5.42.** As duas disciplinas envolvidas no projeto e o acoplamento entre elas.

O modelo matemático completo do problema não é assunto para este trabalho, porém pode ser visto em **Dreyer (2005)**.

#### **5.8.1.1 O acoplamento entre o problema estrutural e hidrodinâmico**

O projeto do hidrofólio do catamarã normalmente envolve um projeto estrutural e um projeto hidrodinâmico, então as variáveis de projeto devem maximizar os objetivos estruturais e hidrodinâmicos concomitantemente.

Normalmente o projeto deve minimizar a massa estrutural e aumentar a performance da embarcação, além de alcançar o regime de planeio sem comprometer a integridade estrutural dos hidrofólios.

As principais forças que atuam sobre a estrutura provêm dos efeitos hidrodinâmicos de arrasto e sustentação, as quais são fortemente influenciadas pelo ângulo de ataque do hidrofólio em relação ao fluido.

A resistência estrutural está diretamente associada à deformação do hidrofólio, que por sua vez, influencia diretamente o ângulo de ataque deste com o fluido. Logo, a mudança do projeto estrutural resulta em um ângulo de ataque diferente, o qual, por sua vez, influencia o projeto hidrodinâmico. Entretanto, a mudança no projeto hidrodinâmico modifica as forças hidrodinâmicas responsáveis pela deformação estrutural.

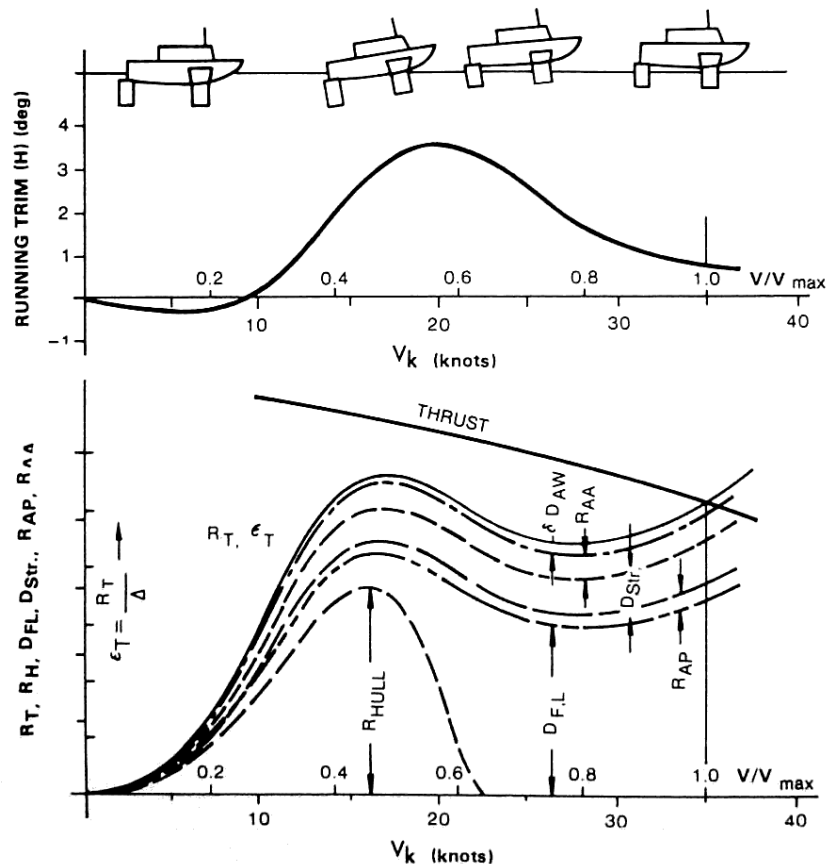
Portanto, existe um claro acoplamento entre o projeto estrutural e o projeto hidrodinâmico. Neste caso a faixa de possibilidades consideradas deve prever uma solução de compromisso que não necessariamente será a melhor solução para um dos critérios específicos.

### 5.8.1.2 O problema hidrodinâmico

Como proposto por **Masuyana (1987)**, o hidrofólio trabalha da mesma maneira que a asa de um avião, com uma exceção: é submerso em água. Assim, os mesmos princípios estabelecidos para aeronaves podem ser aplicados ao projeto de embarcações dotadas de hidrofólios.

Uma embarcação dotada de hidrofólios possui dois modos de operação: o modo normal (com os cascos submersos) e o modo planeio (com os cascos completamente fora da água e os hidrofólios submersos). A capacidade hidrodinâmica para se manter fora da água depende, principalmente, do peso da embarcação e da forma dos hidrofólios (responsáveis pela sustentação).

O comportamento hidrodinâmico esperado para uma embarcação com hidrofólios pode ser visto na figura 5.43. Pode-se observar que para veículos de alta velocidade, quando o casco deixa completamente a água, a resistência é reduzida significativamente. Entretanto, existe um pico de resistência que deve ser superado, a partir do qual a sustentação gerada pelos hidrofólios eleva o casco da água e a resistência ao avanço diminui.



**Figura 5.43.** Hidrodinâmica de uma embarcação com hidrofólios. Adaptado de Molland (1996).

Este modelo de resistência é importante e deve ser considerado como uma restrição do projeto hidrodinâmico, pois caso o regime de planeio não seja alcançado, o processo de otimização deve buscar outras soluções.

Utilizando o modelo proposto por **Holtrop (1982)** e a condição de equilíbrio (somatória de forças e momentos igual a zero), é possível calcular as forças de sustentação e arrasto em função da velocidade.

Algumas hipóteses do modelo hidrodinâmico:

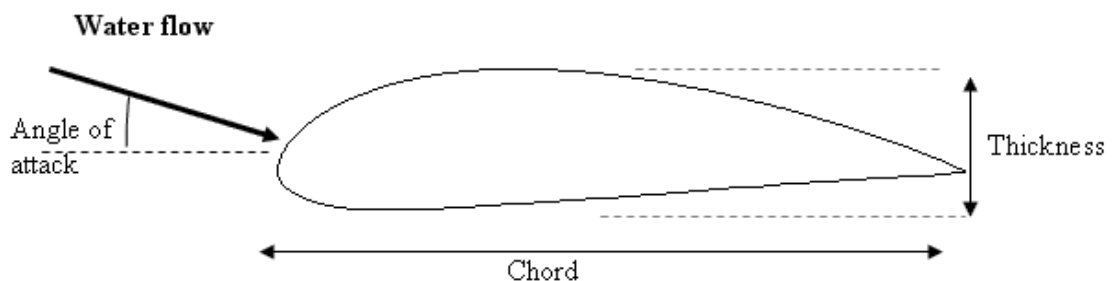
- as forças (peso, sustentação e empuxo) são aplicadas no centro de gravidade da embarcação;
- a embarcação atinge o regime de planeio sem angulo de *trim* ou *roll*;
- a superfície da água é calma e sem ondas;
- os efeitos de cavitação e aeração nos hidrofólios são ignorados.

Portanto, pode-se definir o problema hidrodinâmico como a busca da maior velocidade obtida em regime de planeio, uma vez assegurado que a embarcação consiga atingir tal regime.

### 5.8.1.3 O problema estrutural

Como vistos anteriormente, os hidrofólios são montados em estruturas sobre o casco. O problema estrutural consiste na minimização do peso desta estrutura respeitando o limite de escoamento do material empregado.

Uma vez escolhido o tipo de seção do hidrofólio, existem poucas variáveis de projeto a serem definidas, as quais são mostradas na figura 5.44.

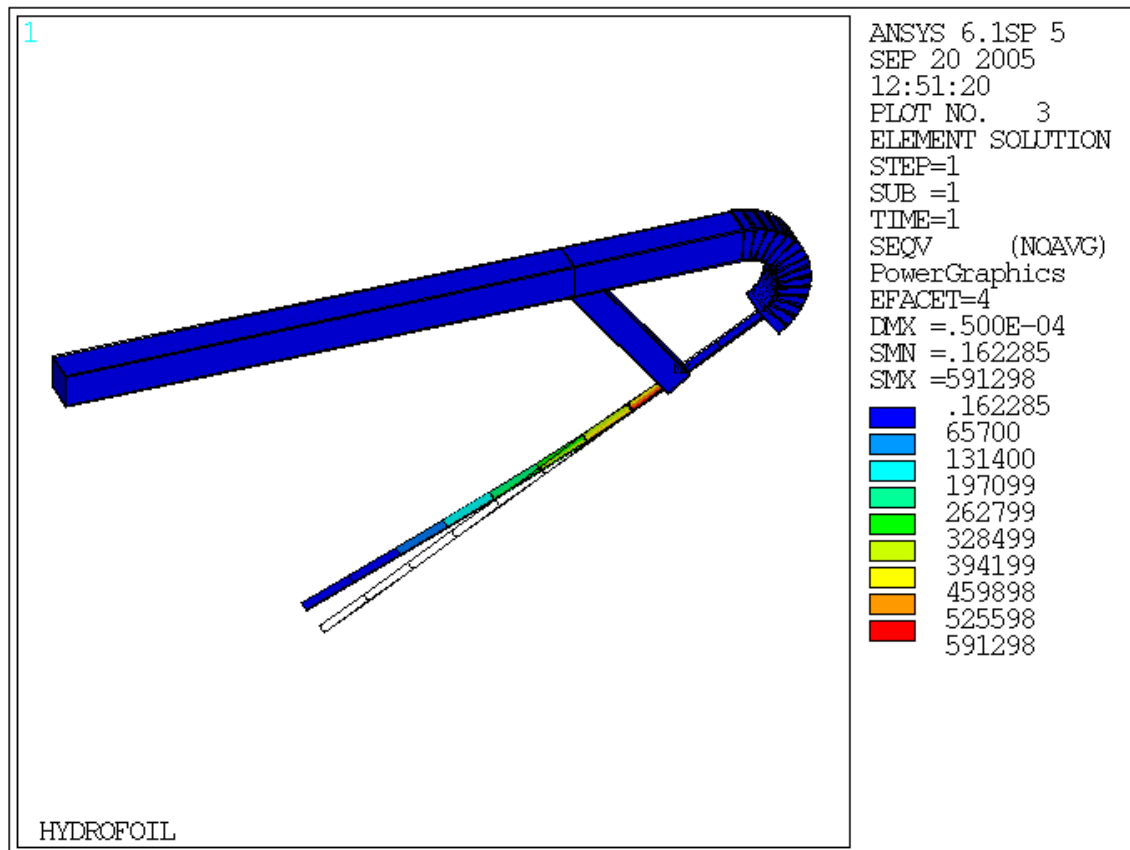


**Figura 5.44.** Variáveis de projeto para a seção do hidrofólio.

No modo planeio é essencial monitorar a deflexão dos hidrofólios, que deve ser sincronizada com o projeto hidrodinâmico, para adaptar o cálculo da força de sustentação, conforme discutido na seção 5.8.1.1.



A deflexão, mostrada na figura 5.45, pode exigir reforços para que a estrutura se torne mais rígida. Evidentemente, esta solução aumenta o peso da estrutura e então a embarcação terá mais dificuldades em atingir o regime de planeio.



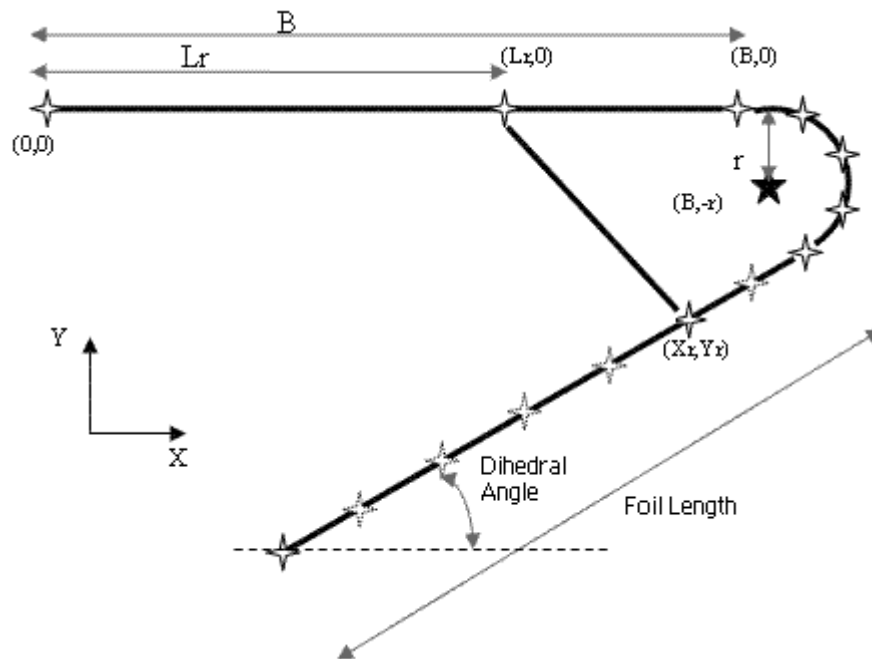
**Figura 5.45.** Deflexão do hidrofólio e distribuição das tensões de von Mises.

O modelo estrutural é dividido em 4 partes: a viga horizontal de suporte, a região de transição, o hidrofólio e, por fim, a barra de reforço transversal. Na figura 5.46 é possível observar todas as regiões e variáveis do projeto estrutural.

A primeira região possui uma das extremidades fixadas ao casco do catamarã, enquanto a outra, é conectada ao hidrofólio. Esta região é assumida como horizontal e todos os deslocamentos e deformações são calculados em relação a esta horizontal.

Em seguida, existe uma região de transição, com um raio de curvatura  $r$ , entre a viga horizontal e o hidrofólio propriamente dito. Por fim, a barra de reforço transversal limita a deflexão do hidrofólio.

As vigas: horizontal, transversal e de transição possuem seções tubulares.



**Figura 5.46.** Variáveis do projeto estrutural do hidrofólio.

A integridade do hidrofólio é essencial para a embarcação, fazendo desta a mais importante restrição do projeto estrutural, cuja modelagem considerou todas as hipóteses usuais da teoria de vigas.

#### 5.8.1.4 Definição do carregamento

Como visto na figura 5.46, o modelo estrutural do hidrofólio é dividido em partes, cada uma definida por dois nós. As forças hidrodinâmicas (de sustentação e arrasto) são calculadas para cada um dos nós

A força de sustentação atua sempre perpendicularmente ao fluido e pode ser decomposta em uma componente vertical, responsável por elevar a embarcação até o regime de planeio, e uma componente horizontal, anulada em virtude da simetria (bombordo - estibordo) da embarcação. A decomposição das forças hidrodinâmicas é mostrada na figura 5.47. A força de arrasto é perpendicular ao plano definido pelas componentes da sustentação e não está representada na figura.

As forças hidrodinâmicas que atuam em cada ponto do hidrofólio podem ser modeladas em função do calado (no ponto considerado), do ângulo de ataque em relação ao fluido e da seção NACA utilizada.

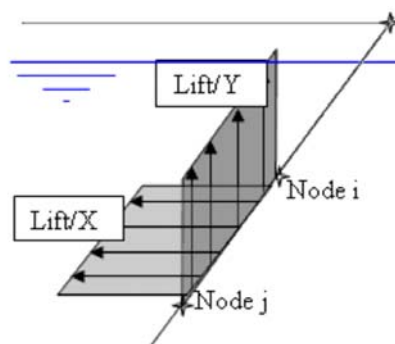


Figura 5.47. Decomposição das forças hidrodinâmicas.

### 5.8.2 As características da embarcação projetada

As dimensões do catamarã projetado são apresentadas na tabela 5.20.

Tabela 5.20. Dimensões do catamarã.\*

Dimensão	Valor
Comprimento entre perpendiculares (m)	6.430
Comprimento na linha da água (m)	6.340
Boca máxima (m)	0.440
Boca na linha da água (m)	0.370
Calado (m)	0.220
Pontal (m)	0.520
Superfície molhada (m <sup>2</sup> )	3.053
Deslocamento (m <sup>3</sup> )**	0.220

L/B	B/H	Cb	Cp	Cm	Cwp	Cv
17.14	1.680	0.429	0.660	0.650	0.741	0.867

\* Relativos a apenas um casco.

\*\* A variação da massa resultante da otimização estrutural é menor do que a variação estatística da massa da tripulação. Por essa razão, neste caso, é possível considerar a massa como uma constante.

### 5.8.3 A abordagem racional do problema

Embora este problema possa ser resolvido utilizando-se as estratégias de otimização multidisciplinar discutidas na seção 2.1.11, o objetivo é concentrar o estudo na aplicação do ambiente de projeto distribuído e na utilização de superfícies dinâmicas de resposta, e para uma melhor comparação com os resultados obtidos por **Dreyer (2005)**, optou-se pela abordagem racional utilizada naquela oportunidade.

O objetivo é determinar o valor das variáveis de projeto que, a um só tempo, maximizem a velocidade da embarcação e minimizem a massa da estrutura, respeitando as restrições estruturais e hidrodinâmicas.

Como discutido na seção 5.8.1.1, existe um claro acoplamento entre as disciplinas do projeto. Entretanto, a massa da estrutura é dada pela soma das massas do casco e do

hidrofólio da embarcação. Logo, a otimização estrutural do hidrofólio modifica apenas uma pequena parcela da massa total da embarcação, e, sabendo que essa modificação é inferior a variação estatística da massa da tripulação, pode-se considerá-la constante, desacoplando parte do problema.

Assim, o problema se concentra apenas na deformação do hidrofólio e na velocidade alcançada pela embarcação, sendo possível construir uma Fronteira de Pareto determinando a velocidade máxima alcançada para cada valor de deformação do hidrofólio.

Para cada ponto da Fronteira de Pareto, os valores das variáveis estruturais são determinados pelo processo de otimização estrutural, fixando-se a deformação do hidrofólio. De maneira análoga, os valores das variáveis hidrodinâmicas são determinados pelo processo de otimização hidrodinâmica, fixando-se o carregamento dinâmico considerado.<sup>34</sup>

#### 5.8.4 Resultados da otimização

O objetivo é maximizar a velocidade do catamarã ao longo de uma rota predeterminada; o que implica em maximizar a velocidade do catamarã para diferentes direções do vento. A melhor maneira de representar essa condição é maximizar a área do gráfico polar representado na figura 5.48. Os valores otimizados das variáveis de projeto são apresentados na tabela 5.21.

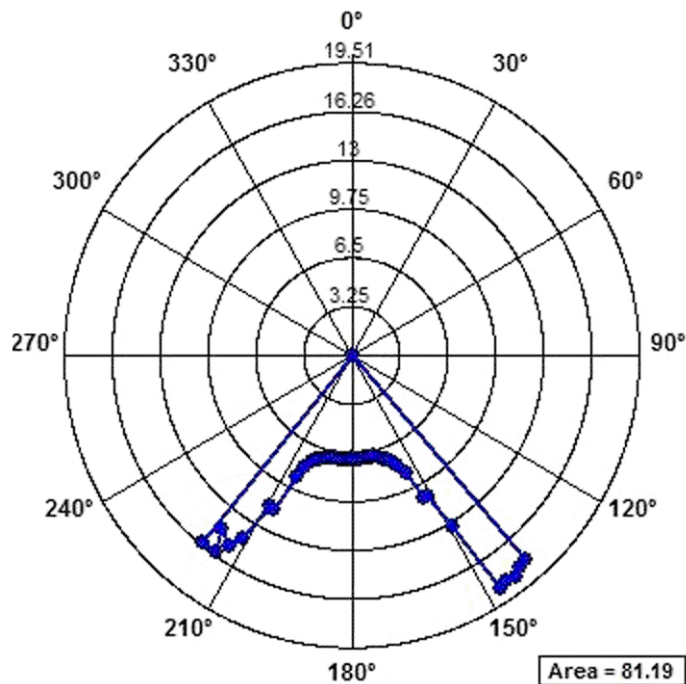
Tabela 5.21. Resultados da otimização do hidrofólio do catamarã.\*

Variável	Mínimo	Máximo	Ótimo
Foil Length 1 (m)	0.5 m	4.0 m	1.23 m
Foil Length 2 (m)	0.5 m	4.0 m	1.23 m
Foil Length 3 (m)	0.2 m	3.0 m	1.79 m
Foil Chord 1 (m)	0.1 m	0.6 m	0.21 m
Foil Chord 2 (m)	0.1 m	0.6 m	0.21 m
Foil Chord 3 (m)	0.1 m	0.6 m	0.18 m
Rudder Chord (m)	0.1 m	0.6 m	0.29 m
Foil Initial Angle 1 (°)	-10°	10°	-6.43°
Foil Initial Angle 2 (°)	-10°	10°	-6.43°
Foil Initial Angle 3 (°)	-10°	10°	-3.75°
Dihedral Angle (°)	-10°	70°	25.54°

\* A otimização foi realizada considerando o ângulo de vento entre 139° e 221°, com variação de 2° e velocidade 20 knots.

\*\* Os ângulos iniciais são diferentes dos ângulos da estrutura deformada (ângulos reais de operação).

<sup>34</sup> É interessante observar que essa abordagem inspirou o desenvolvimento da estratégia de otimização multidisciplinar baseada em valores de acoplamento descrita na seção 4.4. No entanto, neste caso a Fronteira de Pareto foi construída manualmente utilizando-se intervalos discretos predeterminados e otimizações disciplinares independentes.



**Figura 5.48.** Função objetivo maximizada para velocidade do vento de 20 knots.

Os resultados obtidos são consideravelmente próximos aos obtidos em **Dreyer (2005)**, visto que a abordagem e todos os modelos utilizados são os mesmos.

Como visto na seção 2.1.5, normalmente algoritmos de otimização são comparados em relação ao número de iterações que realizam. No entanto, neste caso, é impossível comparar o número de iterações, visto que as iterações dividem-se em avaliações reais e avaliações utilizando superfícies de respostas. Logo, o resultado mais significativo é a redução do tempo total do processo. Em Dreyer, sem o uso de superfícies de resposta ou processamento distribuído, a duração do processo foi de 85 horas, enquanto que no ambiente desenvolvido, a duração foi quase 10 vezes menor.<sup>35</sup>

Quatro computadores foram utilizados no processo, sendo: dois para o treinamento de superfícies de resposta, um para as análises estruturais e um para o cliente de otimização (que ao mesmo tempo atuou como servidor de infra-estrutura).

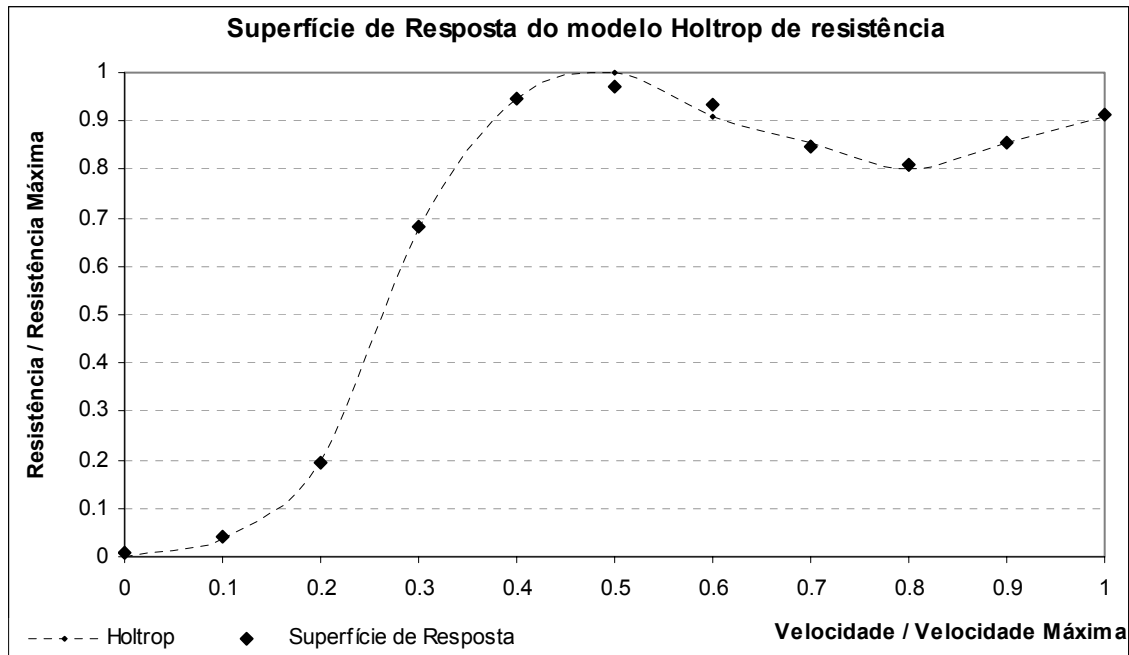
Observa-se ainda que, utilizando o algoritmo MBA, o número de computadores não é relevante, pois apenas 1 avaliação é realizada a cada iteração do algoritmo. Mas, se for utilizado o algoritmo genético, a velocidade de otimização é proporcional ao número de

<sup>35</sup> É importante observar que os computadores utilizados são similares ao computador utilizado em Dreyer (2005), um Pentium III 800 Mhz com 128MB de RAM. Mesmo considerando o aumento da performance computacional (*hardware*), o ganho de desempenho do processo é real.

computadores utilizados para as avaliações estruturais, pois neste caso, todos os indivíduos da população são avaliados a cada iteração do processo.

#### 5.8.4.1 A neuro superfície de resposta para o modelo Holtrop de resistência

Nesta seção é mostrado um exemplo de superfície de resposta calculada durante o processo de otimização. Nesse caso, a superfície de resposta, mostrada na figura 5.49, foi utilizada no problema hidrodinâmico para estimar a resistência ao avanço da embarcação.



**Figura 5.49.** Superfície de resposta calculada para representar o modelo de resistência de Holtrop.

A superfície apresentada na figura 5.49 foi calculada utilizando uma rede neural do tipo MLP com 1 camada escondida composta de 8 neurônios. A superfície foi treinada utilizando um histórico com 11 pares, 1% de erro e 100.000 iterações.

## 6 Conclusões

---

6.1	Conclusões	175
6.2	Perspectivas de pesquisa	178
6.2.1	Contribuições teóricas	178
6.2.2	Implementações computacionais	180

*"Só erra quem produz. Mas, só produz quem não tem medo de errar."*

Octavio Paz

Neste capítulo serão apresentadas as principais conclusões alcançadas nesta pesquisa, bem como perspectivas para novas pesquisas, que por falta de tempo, não puderam ser exploradas.

### 6.1 Conclusões

Muito tempo decorreu e um longo caminho foi percorrido desde que as primeiras idéias desta pesquisa surgiram. Uma caminhada que atravessou muitos campos do conhecimento e cruzou dois continentes.

Os últimos meses foram gastos no esforço em construir um relatório agradável aos leitores, que não se detenha em preciosismos matemáticos desnecessários e que não fuja ao escopo deste trabalho. O relatório se dedica a definir o problema proposto, relatar as atividades desenvolvidas, bem como os principais resultados e contribuições realizadas.

Como resultado final deste trabalho tem-se um conjunto de módulos funcionais capazes de atuar como ambiente para otimização de projetos multidisciplinares sobre neuro superfícies de resposta dinâmicas com processamento distribuído pela Internet.

A primeira contribuição importante foi a utilização da programação distribuída utilizando o paradigma de *Grid* e o protocolo *MPI* para a troca de mensagens pela Internet. Entre os trabalhos discutidos na seção 2.3, o trabalho de **Giassi (2004)** se destaca por propor uma otimização distribuída dividindo as análises de cada disciplina em computadores geograficamente separados. Seu trabalho, no entanto, utilizava o protocolo *POP3/SMTP* para enviar *emails*, responsáveis pela troca de mensagens. Dessa forma o sistema proposto

por Giassi não atua em tempo real e, embora possua otimização distribuída, não o faz de forma paralela.

Os testes de desempenho revelaram que o custo de sincronização e paralelismo, discutidos na seção 2.3.3.3, é muito baixo em relação ao ganho de desempenho ao se utilizar o processamento distribuído, e representa uma possibilidade interessante e de baixo custo para o desenvolvimento de projetos complexos com otimização multidisciplinar. A estabilidade e largura da banda de conexão com a Internet mostraram-se suficientes para a passagem das mensagens e não influenciaram o desempenho do ambiente.

Destaca-se ainda a constatação de que a escolha do algoritmo de otimização influencia diretamente o ganho de desempenho proporcionado pela utilização do processamento distribuído. A utilização de superfícies de resposta paralelas e algoritmos populacionais mostraram os maiores ganhos de performance ao se utilizar processamento distribuído.

A segunda contribuição relevante ocorreu na geração das superfícies de resposta para a aceleração da convergência do problema. Como visto na seção 2.2, diversos trabalhos exploraram a capacidade de representar padrões e prever resultados das redes neurais para a representação de superfícies de resposta.

O trabalho de **Giassi (2004)**, também utiliza redes neurais artificiais para a representação de superfícies de resposta. Embora poucos detalhes sejam fornecidos, sabe-se que o modelo de rede utilizado é o *Multilayer Feed-forward Neural Network*, em linhas gerais o mesmo modelo utilizado nesta pesquisa. A contribuição apresentada, diferente das abordagens vistas em outros trabalhos, é a proposta de uma nova forma de treinamento dinâmico das redes neurais. O treinamento dinâmico propõe uma readaptação da superfície de resposta enfatizando uma de suas regiões, onde se supõe esteja localizado o ponto ótimo do problema. Inclui-se ainda no algoritmo a utilização de um filtro que elimina pontos próximos, acelerando o processo de treinamento e portanto de convergência.

Vale ressaltar também que o ambiente desenvolvido permite desacoplar uma superfície de resposta complexa (representando várias funções objetivo ou de restrição) em múltiplas superfícies de resposta, cada uma representando uma única função, e em seguida distribuir o processo de treinamento em múltiplos computadores, aumentando consideravelmente a velocidade de convergência do problema.

A terceira contribuição relevante também está localizada na utilização das superfícies de resposta. Percebeu-se que qualquer modelo de representação de superfícies de resposta



(neural ou não) pode, em alguns problemas, apresentar superfícies anômalas ou viciadas. A abordagem clássica para esses casos é realizar uma parte das análises utilizando a superfície de resposta e uma parte das análises utilizando o problema real.

Essa abordagem é extremamente eficiente para problemas nos quais cada análise demanda pouco esforço computacional, pois as análises reais não aumentam consideravelmente o tempo de processo. Para problemas complexos de engenharia, no entanto, uma única análise pode representar a maior parte do processo de otimização.

Para solucionar este problema foi proposta uma abordagem utilizando-se regiões tabus para isolar os pontos das superfícies anômalas ou viciadas e assim induzir a otimização a uma segunda solução. Essa segunda solução é analisada para então recalibrar as superfícies; desta forma reduz-se consideravelmente o número de análises reais, e aumenta-se a velocidade de convergência do processo.

A última contribuição ocorre, por fim, na estratégia de otimização multidisciplinar adotada. Como visto anteriormente existem diversas estratégias de otimização multidisciplinares com diferentes níveis de centralização.

Como visto na seção 2.1.14, **Giassi (2004)** utilizou uma variação da estratégia de *Otimização Colaborativa (CO)* que denominou *MORDACE*. Após seu trabalho, **Rabeau (2006)** modificou a estratégia *MORDACE* e desenvolveu a estratégia *COSMOS*. Ambos os trabalhos, porém, afirmam categoricamente que problemas multidisciplinares só podem ser resolvidos se apresentarem acoplamento fraco.

A pesquisa realizada e descrita neste relatório propõe uma abordagem inovadora para o processo. Em vez de se modificar o grau de centralização e de se criarem mecanismos para estimar os valores de acoplamento, o que se faz é buscar quais são os valores de acoplamento que resultam nas melhores soluções para o problema.

Essa estratégia tem, contudo, o inconveniente de aumentar consideravelmente o número de iterações e avaliações do problema, pois a cada par analisado no problema global (o de determinar os valores de acoplamento que otimizam o problema), deve-se realizar uma otimização local para determinar quais os valores ótimos das funções de cada disciplina fixados os valores de acoplamento.

Uma estrutura de superfícies de resposta robusta e eficiente pode, porém, facilmente ser construída, reduzindo esse esforço computacional e viabilizando a implementação da estratégia de otimização multidisciplinar baseada em valores dos acoplamentos.

As possibilidades abertas pelo ambiente desenvolvido permitem considerar que uma nova metodologia de projeto deva ser utilizada pelos engenheiros.

O atual paradigma de projeto (seja utilizando concepção linear, espiral ou simultânea) baseado em ciclos de análise direcionadas por um algoritmo de otimização, como mostrado no fluxograma da figura 5.41, pode não ser adequado as novas ferramentas e recursos computacionais existentes.

Nessa nova metodologia, o problema deve ser pensado sempre globalmente, e o maior esforço deve ser empregado na parametrização e acoplamento dos modelos disciplinares utilizados.

Espera-se que este trabalho abra caminho para muitos outros, pois o potencial está provado, e que em breve possamos optar por novas formas de interface com um novo paradigma de processamento, que possa auxiliar o engenheiro libertando-o para explorar soluções mais complexas e arrojadas.

## **6.2 Perspectivas de pesquisa**

Nesta seção serão discutidos assuntos que, por falta de tempo, não foram amplamente abordados na pesquisa, representando perspectivas ainda a serem exploradas. Dividem-se esses assuntos em dois grupos: as contribuições teóricas e as implementações computacionais.

### **6.2.1 Contribuições teóricas**

#### **6.2.1.1 Aumentar a eficiência do treinamento neural com a inversão do erro**

O uso de redes neurais artificiais na representação de superfícies de resposta para a aceleração do processo de otimização foi amplamente discutido neste relatório.

Observou-se que, em muitos casos, com poucos ciclos de treinamento, a superfície de resposta adquire a “forma” da superfície real do problema; fornecendo ao algoritmo de otimização a tendência real do problema.

Esses casos são especialmente importantes quando as superfícies de respostas são utilizadas na representação das funções objetivo, pois permitem determinar corretamente o vetor de variáveis que representam a solução ótima do problema, sem contudo obter o valor correto da função.

O primeiro passo no treinamento de uma rede neural que represente uma superfície de resposta deve ser, portanto, a busca pela tendência correta da função. Sugere-se alterar o procedimento de treinamento neural para amplificar o erro dos pares que representem uma inversão do erro, e com isso acelerar a convergência do processo de otimização.

A inversão do erro (mudança de sinal na diferença entre o valor real e o valor estimado pela superfície entre dois pontos próximos) representa má aderência entre a superfície neural e o problema real, visto que as inflexões das funções reais do problema estão mal caracterizadas nas superfícies de resposta.

#### **6.2.1.2    *Trabalhar a Fronteira de Pareto utilizando um polinômio interpolador***

Como visto na revisão bibliográfica, **Ticona (2003)** apud **Kalyanmoy (2001)** destaca duas importantes metas na otimização de problemas multiobjetivo:

- um conjunto de soluções o mais próximo possível da Fronteira de Pareto.
- um conjunto de soluções com a maior diversidade possível.

Sugere-se que seja possível condicionar, em alguns momentos, o processo de otimização de tal forma que os resultados obedeçam a essas metas.

O conceito geral a ser trabalhado é o de utilizar um polinômio interpolador, ou outra técnica estatística qualquer para aproximar a Fronteira de Pareto e, a partir dessa aproximação, sugerir pontos que pertençam à Fronteira de Pareto e aumentem a diversidade da fronteira.

Esses pontos seriam introduzidos à população inicial de cada geração. Esse processo poderia ser classificado como um “Elitismo Externo”, pois introduziria pontos fortemente candidatos a pontos ótimos de maneira artificial à evolução genética.

#### **6.2.1.3    *Reajustar os pesos neurais após o ajuste dos limites de normalização***

Um dos maiores custos computacionais envolvidos no ambiente desenvolvido reside sem dúvida na readaptação das superfícies de resposta.

Como descrito anteriormente, o correto funcionamento das redes neurais artificiais na representação de superfícies de resposta depende fortemente da normalização das funções representadas. Porém, a pergunta que se sucede é como normalizar uma função da qual não se conhecem os limites?

Para solucionar esse problema adotou-se uma estratégia de limites dinâmicos, que se modificam ao longo do processo de otimização. Tal abordagem implica, no entanto, que a cada mudança dos limites de normalização uma readaptação dos pesos e limites de ativação seja feita.

A abordagem convencional é reiniciar o treinamento fazendo com que o processo clássico de treinamento se encarregue do ajuste de pesos e limites de ativação. Esse procedimento é robusto e confiável, porém tem a clara desvantagem de representar um grande esforço de treinamento repetido a cada modificação dos limites de normalização. Mesmo utilizando uma modificação de limites com margens de 15%, o número de readaptações devido à modificação dos limites de normalização, em alguns casos, é superior ao número de readaptações para a convergência do problema.

Sugere-se que seja possível formular um procedimento para ajustar os pesos e limites de ativação a partir da variação dos limites de normalização das funções representadas.

### **6.2.2 Implementações computacionais**

#### **6.2.2.1 Otimização do uso da conexão TCP/IP**

Quando o procedimento de otimização distribuída foi implementado, utilizaram-se os padrões estabelecidos pelo protocolo MPI com passagem de mensagem pela Internet.

A estrutura desenvolvida utilizando um servidor de infra-estrutura para centralizar e balancear a distribuição de cargas se mostrou altamente robusta e eficiente. Entretanto, um problema de implementação computacional reduz a eficiência do sistema.

A implementação mais simples foi utilizar, para cada etapa do processo de otimização a ser distribuída, uma conexão independente entre o cliente de otimização e o servidor de infra-estrutura. Esse procedimento se mostrou confiável pois era facilmente controlado, e os dados não se misturavam. Por essa razão foi utilizado ao longo de todo o desenvolvimento da pesquisa.

O custo computacional para se estabelecer uma conexão TCP/IP para cada procedimento a ser paralelizado diminui, no entanto, a eficiência do processo. Sugere-se implementar uma rotina que utilize apenas uma conexão TCP/IP ao longo de todo o processo, distribuindo os fluxos de dados por meio de *sockets* ou mesmo por mensagem codificada.

### **6.2.2.2 Vetor dinâmico para o uso de superfícies de resposta**

Existem basicamente duas formas de se definir um vetor de dados em programação.

A primeira forma, mais simples, é definir um vetor de tamanho fixo. Neste caso o sistema operacional reserva uma quantidade de memória igual ao tamanho do vetor definido no programa. Esta maneira é especialmente útil para vetores com poucos elementos ou elementos que consumam pouca memória.

A segunda maneira é definir um vetor dinâmico, cujo tamanho é modificado ao longo do processo de acordo com a necessidade do problema. Cada linguagem de programação possui uma estrutura própria para definir e redimensionar os vetores dinâmicos. O problema envolvido nessa abordagem é o cuidado necessário para ajustar esses vetores sob pena de erros numéricos ou lógicos.

Por limitações de tempo, o vetor que contém os objetos do tipo superfície de resposta foi definido como um vetor fixo; a quantidade de superfícies máximas em um problema está, portanto, limitada pelo tamanho definido no código de programação.

Essa condição é especialmente prejudicial ao se implementar a estratégia de otimização multidisciplinar que exige a geração de diversos vetores de objetos de otimização, cada qual com seus vetores de superfícies de resposta. Ao se definir um número elevado de superfícies, a quantidade de memória reservada pelo sistema operacional pode chegar a várias dezenas de *megabytes*.

Sugere-se modificar essa implementação, para que o programa utilize vetores dinâmicos para alocar as superfícies de resposta.

### **6.2.2.3 Implementar o paralelismo na estratégia de otimização multidisciplinar**

Na estratégia de otimização multidisciplinar baseada em valores de acoplamento, a cada avaliação do processo global de otimização multidisciplinar é realizado um processo de otimização local para cada disciplina existente.

Portanto é possível paralelizar esse procedimento, para que as otimizações locais sejam feitas de forma simultânea, aumentando consideravelmente a velocidade de convergência do processo.

---

Thiago Pontin Tancredi

## 7 Referencias

---



Artigos, teses e dissertações consultadas e disponíveis no CD-ROM anexo.



Livros consultados nas bibliotecas da Universidade de São Paulo.



Artigos consultados, porém não disponíveis no CD-ROM anexo.

*"Where is the wisdom we have lost in knowledge?  
Where is the knowledge we have lost in information?"*

T.S. Eliot

**Agatonovic-Kustrin, S., Zecevic, M., Zivanovic, Lj. and Tucker, I. G.**, *"Application of neural networks for response surface modeling in HPLC optimization"*. Analytica Chimica Acta, Vol. 364, Issues 1-3, May, pp 265-273, 1998.

**Allison J.**, *"Complex system optimization: a review of analytical target cascading, collaborative optimization and other formulations"*, Master's thesis, University of Michigan, 2004.

**Almasi, G. S. and Gottlieb, A.**, *"A Highly Parallel Computing"*, 2ed., The Benjamins Cumming Publishing Company, 1994.

**Amdahl, G. M.**, *"Validity of the single processor approach to achieving large scale computing capabilities"*, AFIPS spring joint computer conference, 1967.

**Andrews, G. R.**, *"Foundations of Multithread, Parallel and Distributed Programming"*, USA, 1999.

















**Anjum, M. F., Tasadduq, I. and Tasadduq, K.**, *"Response surface methodology: A neural network approach"*, European Journal of Operational Research, Vol. 101, Issue 1, August, pp. 65-73, 1997.

**Augusto, O. B.**, *"Processos de Busca não Linear para o Projeto da Estrutura do Navio"*, BT/PNV/23, EPUSP, Brasil, 1996.

**Azuma, R. T., Baillot, Y., Behringer, R., Feiner, S., Julier, S. and MacIntyre, B.**, *"Recent Advances in Augmented Reality"*, IEEE Computer Graphics and Applications vol.21(6), pp. 34-47, 2001.
















**Balling, R. J. and Sobieszczanski-Sobieski, J.**, *"Optimization of Coupled Systems: A Critical Overview of Approaches"*, American Institute of Aeronautics and Astronautics, AIAA Paper 94-4330-CP, pp. 753-773, 1994.

















**Baş D. and Boyac, İ. H.**, *"Modeling and optimization II: Comparison of estimation capabilities of response surface methodology with artificial neural networks in a bio-chemical reaction"*, Journal of Food Engineering, Vol. 78, Issue 3, February, pp. 846-854, 2007.















- Batill, S. M., Stelmack M. A. and Yu, X. Q.**, *"Multidisciplinary design optimization of an electric-powered unmanned air vehicle"*, Aircraft Design, vol.2, pp. 1-18, 1999. 
- Belcher, D. and Billinghamurst, M.**, *"Using Augmented Reality for Visualizing Complex Graphs in Three Dimensions"*, Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03), 2003. 
- Behdinin, K., Perez, R. E. and Liu, H. T.**, *"Multidisciplinary Design Optimization of Aerospace Systems"*, Canadian Design Engineering Network, 2005. 
- Belegundu, A. D., Halberg, E., Yukish, M. A. and Simpson, T. W.**, *"Attribute-Based Multidisciplinary Optimization of Undersea Vehicles"*, 8th AIAA / USAF / NASA / ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, 2000. 
- Birk, L. and Clauss, G. F.**, *"Automated hull optimization of offshore structures based on rational seakeeping criteria"*, Proc. of 11th Int. Offshore and Polar Engineering Conference ISOPE, Stavanger, Norway 2001. 
- Brasil**, *"Guia de Estruturação e Administração do Ambiente de Cluster e Grid"*, Secretaria de Logística e Tecnologia da Informação, v1.0, Brasília, 2006. 
- Braun R., Gage P., Kroo I., and Sobieszczanski-Sobieski J.**, *"Implementation and Performance Issues in Collaborative Optimization"*, Proceedings 5th AIAA/USAF MDO symposium, AIAA Paper 96-4017, Bellevue, WA, Sept.1996. 
- Campana, E. F., Fasano, G., Peri, D. and Pinto, A.**, *"Nonlinear Programming Approaches in the Multidisciplinary Design Optimization of a Sailing Yacht Keel Fin"*, 9th International Conference on Numerical Ship Hydrodynamics, Michigan, 2007. 
- Castro, R. E.**, *"Otimização de estruturas com multi-objetivos via algoritmos genéticos"*, Dissertação de Doutorado, Universidade Federal do Rio de Janeiro, 2001. 
- Chase, N., Rademacher M., Goodman E., Averill, R. and Sidhu, R.**, *"A Benchmark Study of Optimization Search Algorithms"*, Michigan State University, East Lansing, MI, 2008. 
- Caudill, M. and Butler, C.**, *"Understanding Neural Networks – computer explorations"* volume 1: Basic Net-works, MIT Press, London, England, 1994. 
- Coutinho, K. D.**, *"Método de Otimização Topológica em estruturas tridimensionais"*, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, 2006. 
- Cramer, E., Dennis, J. J., Frank, P., Lewis, R., and Shubin, G.**, *"Problem formulation for multidisciplinary optimization"*, AIAA Symposium on Multidisciplinary Design Optimization, 1993. 
- Cybenko, G. V.**, *"Approximation by Superpositions of a Sigmoidal function"*, Mathematics of Control, Signals and Systems, Vol. 2, no. 4, pp. 303-314, 1989. 
- Dennis, J. E., and Lewis, R. M.**, *"Problem Formulations and Other Optimization Issues in Multidisciplinary Optimization"*, AIAA Paper 94-2196, AIAA Symposium on Fluid Dynamics, Colorado Springs, CO, June 1994. 
- Dreyer, T., Augusto, O. B. and Fouad, B.**, *"Catamaran Project: Report about the structural model of an hydrofoil"*, Universidade de Sao Paulo, Brazil, 2005. 

- Duncan, R.**, "A Survey of Parallel Computers Architectures", IEEE Computer, pp. 5-16, Fevereiro 1990. 📄
- Edwards, I. M. and Jutan, A.**, "Optimization and control using response surface methods", Journal of Computers & Chemical Engineering, Vol. 21, Issue 4, December, pp. 441-453, 1997. 📄
- Fletcher, R.**, "An exact penalty function for nonlinear programming with inequalities". Mathematical Programming, Vol. 5, pp. 129-150, 1973. 🖐️
- Fonseca, C. M. and Fleming, P. J.**, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423, 1993. 📄
- Fortune, S. and Wyllie, J.**, "Parallelism in Random Access Machines", in Proc. 10th ACM Symp. Theory of Computing, San Diego, CA., May 1978. 📄
- Giassi, A., Bennis, F. and Maisonneuve, J.**, "Multi-Objective Optimization of Ship Hull with Distributed Applications", Proceedings of Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DAC-48732, USA, 2003. 📄
- Giassi, A.**, "Optimisation et Conception Collaborative dans le cadre de L'ingenierie Simultanee", Thèse de Doctorat, Ecole Centrale de Nantes, France, 2004. 📄
- Glover, F.**, "Tabu Search – Part I", ORSA Journal on Computing 1, pp. 190-206, 1989. 📄
- Glover, F.**, "Tabu Search – Part II", ORSA Journal on Computing 2, pp. 4-32, 1990. 📄
- Goldberg, D. E.**, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, Inc., Reading, MA, 1989. 📖
- Golinski, J.**, "Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods", Journal of Mechanisms, Vol. 5, pp. 287–309, 1970. 📄
- Groşan, C. and Dumitrescu, D.**, "A Comparison Of Multiobjective Evolutionary Algorithms", Acta Universitatis Apulensis, Acta 4, 2002. 📄
- Gunawan, S., Farhang, A., and Azarm, S.**, "Multi-level Multi-objective Genetic Algorithm Using Entropy to Preserve Diversity", Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, LNCS (Springer, ed.), Vol. 2632, pp. 148–161, 2003. 📄
- Haftka, R. T.**, "Elements of Structural Optimization (Solid Mechanics and its Applications)", Kluwer Academic Publishers, Dordrecht, Netherlands, 1993. 📖
- Haftka, R. T., Sobieszczanski-Sobieski J., and Padula S. L.**, "On Options for Interdisciplinary Analysis and Design Optimization", Structural Optimization, Vol.4/2, pp. 65-74, 1992. 🖐️
- Hajela, P. and Lin, C. Y.**, "Genetic Search Strategies in Multicriterion Optimal Design", Structural Optimization, v. 4, pp. 99-107, 1992. 🖐️
- Hebb, D. O.**, "The organization of behavior: A neuropsychological theory", New York: Wiley, 1949. 📖



- Holtrop, J. and Mennen, G. G. J.**, “*An Approximate Power Prediction Method*”, International Shipbuilding Progress, Vol. 29, pp. 166-170, 1982. 
- Hopfield, J. J.**, “*Neural networks and physical systems with emergent collective computational abilities*”, Proceedings of the National Academy of Sciences USA, Vol. 79, pp. 2554-2558, 1982. 
- Horn, J., and Nafpliotis, N.**, “*Multiobjective Optimization Using the Niche Pareto Genetic Algorithm*”, IlliGAL Report No. 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign, 1993. 
- Huang, Y. M. and Wang, K. J.**, “*Optimization of bicycle frames using genetic algorithms*”, ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, USA, 2007. 
- Júnior, W. F. P.**, “*Hiperquicksort: uma análise prática com Implementação em MPI*”, Monografia de Graduação, Lavras, Minas Gerais, 2002. 
- Kalyanmoy, D.**, “*Multi-objective genetic algorithms: Problem difficulties and construction of test problems*”. Evolutionary Computation, vol.7, pp. 205–230, 1999. 
- Kalyanmoy, D.**, “*Multi-Objective Optimization using Evolutionary Algorithms*”, John Wiley & Sons, New York, USA, 2001. 
- Kodiyalam, S.**, “*Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Phase I*”, Langley Research Center, NASA, 1998. 
- Kursawe F.**, “*A variant of evolution strategies for vector optimization*”, Proceedings of the 1st parallel problem solving form nature (Schwefel H.-P. et Reinhard M"anner S. V., eds.), pp. 193–197, 1991. 
- McCulloch, W. S. and Pitts, W.**, “*A logical calculus of the ideas immanent in nervous activity*”, Bulletin of Mathematical Biology, Vol. 5, n4, pp. 115-133, 1943. 
- Masuyana, Y.**, “*Stability analysis and prediction of performance for a hydrofoil sailing boat I: Equilibrium sailing state analysis*”, International Shipbuilding Progress, Vol. 33, pp. 134-143, 1986. 
- Molland, A. F., Wellicome, J. F. and Couser, P. R.**, “*Resistance experiments on a systematic series of high speed displacement catamarans forms: variation of length-displacement ratio and breadth-draught ratio*”, Transactions, Royal Institution of Naval Architects, Vol. 138A, 1996. 
- Monte-Mor, J. A.**, “*Paralelização de um Método de Aprendizado Indutivo de Máquina Baseado na Teoria de Conjuntos Aproximados*”, Brasil, 2002. 
- Murray, A. F.**, “*Applications of Neural Networks*”, The University of Edinburgh, Kluwer Academic Pub, 1995. 
- Nielsen, H. B., Lophaven, S. N. and Søndergaard, J.**, “*DACE - A MATLAB Kriging Toolbox*”, Technical Report IMM-TR-2002-12, Informatics and mathematical modeling, Technical University of Denmark, 2002. 

- Novotny, A., Feijóo, R., Taroco, E. and Padra, C.,** “*Derivada topológica via análise de sensibilidade à mudança de forma na Otimização Topológica*”, Revista Internacional de Métodos Numéricos para Cálculo y Diseño em Ingeniería, vol. 18, pp. 499-519, 2002. 
- Peri, D., Fasano, G., Dessi, D. and Campana, E. F.,** “*Global Optimization Algorithms in Multi-disciplinary Design Optimization*”, 2th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, pp. 10-12, Canada, 2008. 
- Quinn, M. J.,** “*Parallel Computing Theory and Practice*”, McGraw-Hill, 1994. 
- Rabeau, S., Dépincé, Ph., Bennis, F. and Augusto, O. B.,** “*Amelioration de la vitesse de résolution des problèmes multi-objectifs à l'aide d'algorithmes génétiques*”, 17ème Congrès Français de Mécanique, 2005. 
- Rabeau, S.,** “*Optimisation multi-objectif en conception collaborative*”, These de Doctorat, Ecole Centrale de Nantes, France, 2006. 
- Ribeiro, S. A.,** “*Sistema imune artificial para o problema de escalonamento Job Shop*”, Dissertação de Mestrado, Universidade Federal do Espírito Santo, Brasil, 2006. 
- Rosenblatt, F.,** “*The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*”, Psychological Review, Vol. 65, pp. 386-408, 1958. 
- Rosenbrock, H. H.,** “*An Automated Method of Finding the Greatest or Least Value of a Function*”, Computer Journal, Vol. 3, No. 3, Oct., pp. 175-184, 1960. 
- Rumelhart, D. E., Hinton, G. E. and Williams, R. J.,** “*Learning representations by back-propagating errors*”, Nature, Vol. 323, pp. 533-536, 1986. 
- Schaffer, J. D.,** “*Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*”, D.Sc. Thesis, Department of Electrical Engineering, Vanderbilt University, 1984. 
- Sellar R. S., Batill S. M., and Renaud J. E.,** “*Response Surface Based Concurrent Subspace Optimization for Multidisciplinary System Design*”, American Institute of Aeronautics and Astronautics, AIAA Paper 96-0714, 1996. 
- Shuaeib, F. M., Hamouda, A. M. S., Wong, S. V., Umar, R. S. and Ahmed, M. M. H.,** “*A new motorcycle helmet liner material: The finite element simulation and design of experiment optimization*”, Materials & Design, Vol. 28, Issue 1, pp. 182-195, 2007. 
- Shyy, W., Papila, N., Vaidyanathan R. and Tucker, K.,** “*Global design optimization for aerodynamics and rocket propulsion component*”, Progress in Aerospace Sciences 37, pp. 59-118, 2001. 
- Silva, D.,** “*Rede Neural Artificial Aplicada ao Reconhecimento de Padrões usando o Método Back-Propagation*”, Monografia apresentada para a obtenção do grau de Bacharel, Universitário do Triângulo, Uberlândia, Brasil, 2000. 
- Silva, E. C. N.,** “*Otimização aplicada ao projeto de sistemas mecânicos*”, Departamento de Engenharia Mecatrônica e Sistemas Mecânicos, Universidade de São Paulo, 2007. 
- Smith, J., Hodgins, J., Oppenheim, I. and Witkin, A.,** “*Creating Models of Truss Structures with Optimization*”, International Conference on Computer Graphics and Interactive Techniques, USA, 2002. 

- Sobieszczanski-Sobieski J.**, “*On The Sensitivity Of Complex, Internally Coupled Systems*”, National Aeronautics and Space Administration, Technical Memorandum, NASA/TM-100537, January, 1988. 
- Sobieszczanski-Sobieski J., Jeremy S. A. and Robert R. S. Jr.**, “*Bi-Level Integrated System Synthesis (BLISS)*”, National Aeronautics and Space Administration Technical Manual NASA/TM-1998-208715, August, 1998. 
- Srinivas, N. and Deb, K.**, “*Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*”, Journal of Evolutionary Computation, Vol.2 n.3, pp.221-248, 1994. 
- Stump, F. V.**, “*Otimização Topológica aplicada ao projeto de estruturas tradicionais e estruturas com gradação funcional sujeitas a restrição de tensão*”, Dissertação de Mestrado, Universidade de São Paulo, 2006. 
- Takahara, J., Takayama, K., Isowa, K. and Nagai, T.**, “*Multi-objective simultaneous optimization based on artificial neural network in a ketoprofen hydrogel formula containing O-ethylmenthol as a percutaneous absorption enhancer*”, International Journal of Pharmaceutics, Vol. 158, Issue 2, December, pp. 203-210, 1997. 
- Takahara, J., Takayama, K. and Nagai, T.**, “*Multi-objective simultaneous optimization technique based on an artificial neural network in sustained release formulations*”, Journal of Controlled Release, Vol. 49, Issue 1, November, pp. 11-20, 1997. 
- Ticona, W. G. C.**, “*Aplicação de algoritmos genéticos multi-objetivo para alinhamento de seqüências biológicas*”, Dissertação de Mestrado, Universidade de São Carlos, Brasil, 2003. 
- Wangdahl, G. E.**, “*The External Penalty Function Optimization Technique and Its Application to Ship Design*”, The University of Michigan, Department of Naval Architecture and Marine Engineering, Report N. 129, June, 1972. 
- Ugarte, E. C. T.**, “*Otimização do projeto de embarcações pesqueiras*”, Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia, São Paulo, 2007. 
- Vanderplaats, G. N.**, “*Numerical Optimization Techniques for Engineering Design With Applications*”, McGraw-Hill, Inc., NY, USA, 1984. 
- Zeboudj, S., Eboudj, S., Belhaneche-Bensemra, N. and Belabbes, R.**, “*Use of surface response methodology for the optimization of the concentration of the sweet orange essential oil of Algeria by wiped film evaporator*”, Journal of food engineering, ISSN 0260-8774, Vol. 67, no 4, pp. 507-512, 2005. 
- Zitzler, E. and Thiele, L.**, “*An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*”, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, TIK-Report No. 43, 1998. 
- Zitzler, E., Kalyanmoy, D. and Thiele, L.**, “*Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*”, Evolutionary Computation, Vol. 8, pp. 173–195, 2000. 
- Zhu, J. Z., Chang, C. S., Yan, W. and Xu, G. Y.**, “*Reactive power optimization using an analytic hierarchical process and a nonlinear optimization neural network approach*”, IEEE Pvoc.-Genev. Transm. Distrib., Vol. 145, No. 1, January, 1998. 

# APÊNDICE A

Neste apêndice é mostrado um exemplo de macro utilizada no ambiente desenvolvido: a macro do problema Viga em Balanço com a análise estrutural utilizando o aplicativo *Ansys*.

## A1 - Macro para o ambiente desenvolvido. (Viga em Balanco-FEM-SR.txt)

```
Viga I - Elementos Finitos - Superfície de Resposta
//número de disciplinas
1
//número de variáveis da disciplina 1
3
//número de restrições da disciplina 1
2
//número de objetivos da disciplina 1
1
//valor mínimo para a variável 1
0.200
//valor máximo para a variável 1
1.000
//resolução usada em algoritmos genéticos para a variável 1
0.001
//valor mínimo para a variável 2
0.200
//valor máximo para a variável 2
0.800
//resolução usada em algoritmos genéticos para a variável 2
0.001
//valor mínimo para a variável 3
0.006
//valor máximo para a variável 3
0.040
//resolução usada em algoritmos genéticos para a variável 3
0.001
//número de parâmetros extras
0
//tipo da primeira restrição,      3 = ansys
3
//nome do arquivo de saída que contém o valor da restrição calculada pela macro
TensaoEQVM
//utilizar superfície de resposta?    1 = sim    0 = não
1
//variáveis que influenciam nesta restrição:  x1 = 1 (sim)      x2 = 1 (sim)      x3 = 1 (sim)
111
//valor limite para a restrição 1
<270000
//tipo da segunda restrição,      3 = ansys
3
//nome do arquivo de saída que contém o valor da restrição calculada pela macro
Deslocamentos
//utilizar superfície de resposta?    1 = sim    0 = não
1
//variáveis que influenciam nesta restrição:  x1 = 1 (sim)      x2 = 1 (sim)      x3 = 1 (sim)
111
//valor limite para a restrição 2
<1000
//tipo da função objetivo,      2 = expressão analítica
2
//expressão analítica
X1*X3+2*X2*X3
//utilizar superfície de resposta?    1 = sim    0 = não
0
//descrição das variáveis, restrições, objetivos, parâmetros utilizados
Altura da Alma: // m
Largura do Flange: // m
Espessura Alma/Flange: // m
```

Tensão Máxima Longitudinal: // KPa  
 Deslocamento Maximo: // m  
 Massa da Estrutura: // m<sup>2</sup>  
 //descrição da disciplina  
 Viga I - Elementos Finitos - Superfície de Resposta  
 //número de arquivos de imagem gerados pela análise ansys  
 1  
 //nome do arquivo de imagem gerado pela análise ansys  
 Estrutura  
 //nome da macro parametrizada para o ansys  
 VigaEmBalanco.mac

## A2 - Macro para análise estrutural utilizando o aplicativo *Ansys*. (VigaEmBalanco.mac)

```
/NOPR ! Suppress printing of UNDO process
FINISH ! Make sure we are at BEGIN level
/CLEAR,NOSTART ! Clear model since no SAVE found
*SET, X1, ValorX1
*SET, X2, ValorX2
*SET, X3, ValorX3
*SET, Area, 2*X3*X2+X1*X3
*SET, Izz, X1*X3**3/12+2*X3*X2**3/12
*SET, Iyyp, X1**3*X3/12+2*X3**3*X2/12
*SET, Iyyt, 2*X3*X2*(X1+X3)/2*(X1+X3)/2
*SET, TKz, X1+2*X3
*SET, TKy, X2
/PREP7
K,1,0,0,0,
K,2,100,0,0,
LSTR, 1, 2
ET,1,BEAM4
R,1,Area,Iyyp+Iyyt,Izz,TKy,TKz,,
RMORE, , , , ,
MPTEMP,1,0
MPDATA,EX,1,,270000000
MPDATA,PRXY,1,,0.3
SECTYPE, 1, BEAM, I, , 0
SECOFFSET, CENT
SECDATA,X2,X2,X1+2*X3,X3,X3,X3,0,0,0,0
TYPE, 1
MAT, 1
REAL, 1
ESYS, 0
SECNUM, 1
LESIZE,1,1, , , , , 1
LMESH, 1
DK,1, ,0, ,0,ALL, , , , ,
FK,2,FY,-1
FINISH
/SOL
SOLVE
FINISH
/POST1
/DEV,PSFN,NINC
/SHOW,JPEG,,0.8
JPGPRF,500,100,1
/GRAPHICS,POWER
/view,1,-0.75,1.5,2
plnsol,u,y,1
/show,CLOSE
/RENAME,file.jpg,,pathTemp\Estrutura.jpg
ETABLE,SMAXI,NMISC,1
ESEL,S,ELEM,1,1,1
/output,pathTemp\TensaoEQVM.txt
Pretab,SMAXI,,0,1
/output,pathTemp\Deslocamentos.txt
PRNSOL,u,comp
```

A cada avaliação os valores de  $x_1$ ,  $x_2$  e  $x_3$  são substituídos na macro, que é então processada pelo programa Ansys.

A cada avaliação o caminho “pathTemp” é substituído na macro pela pasta de trabalho do programa de otimização.

# ANEXO A

Neste anexo são apresentados os *softwares* de otimização mais utilizados em engenharia e suas principais áreas de atuação, para conhecimento do leitor.

TSO (WASP) - projeto preliminar da asa e cauda de aviões com restrições aeroelásticas combina análise estrutural (modelos simplificados) e otimização (método de penalização interior);

ACCESS, FASTOP, OPSTAT, OPTCOMP, OPTIMUM, ASOP, STARS, DESAP - integram algoritmos de MEF específicos com otimização;

OPTSYS - integra os programas de MEF genérico ABAQUS e ASKA com algoritmos de otimização;

ASTROS - software de otimização estrutural baseado no NASTRAN;

NISAOPT (SHAPE e STROPT) - baseado no NISA II;

ADS, DOT e DOC (empresa VMA) - derivados do CONMIN. DOT é uma coleção de subrotinas em FORTRAN para otimização. DOC é um programa de controle que facilita o uso do DOT na solução de problemas com múltiplos objetivos;

NEWSUMT - pacote de otimização baseado em métodos de penalização;

OPT - software de otimização baseado no algoritmo de gradientes reduzidos;

IDESIGN - software de otimização baseado no PQS (ou “SQP”);

NASTRAN, I-DEAS, ANSYS - possuem algoritmos de otimização, cálculo de sensibilidade e otimização topológica baseada no método das densidades;

GENESYS - apresenta o MEF já integrado com algoritmos de otimização, incluindo otimização topológica;

OPSTRUCT (empresa ALTAIR - EUA) e OPTISHAPE (empresa QUINT - Japão) - softwares de otimização topológica baseados no método de homogeneização;

modeFrontier – desenvolvido pela Sociedade Sirehna, incorpora otimização multiobjetivo e projeto colaborativo.