

Francilene Barbosa dos Santos Silva

**Algoritmos genéticos para otimização de estruturas reticuladas baseadas em
modelos adaptativos e lagrangeano aumentado**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Orientador: Prof. D.Sc. Carlos Cristiano Hasenclever Borges

Coorientador: D.Sc. Afonso Celso de Castro Lemonge

Coorientador: D.Sc. Helio José Corrêa Barbosa

Juiz de Fora

2011

Silva, Francilene Barbosa dos Santos.

Algoritmos genéticos para otimização de estruturas reticuladas baseadas em modelos adaptativos e lagrangeano aumentado / Francilene Barbosa dos Santos Silva. – 2011.

186 f. : il.

Dissertação (Mestrado em Modelagem Computacional)–Universidade Federal de Juiz de Fora, Juiz de Fora, 2011.

1. Algoritmos genéticos. 2. Otimização. 3. I. Título.

CDU 681.3.055

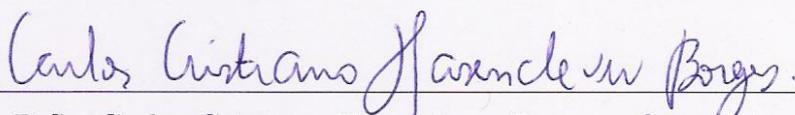
Francilene Barbosa dos Santos Silva

**Algoritmos genéticos para otimização de estruturas reticuladas baseadas em
modelos adaptativos e lagrangeano aumentado**

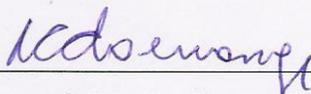
Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Aprovada em 31 de Agosto de 2011.

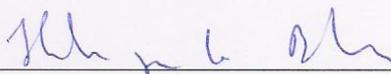
BANCA EXAMINADORA



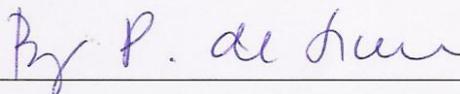
Prof. D.Sc. Carlos Cristiano Hasenclever Borges - Orientador
Universidade Federal de Juiz de Fora



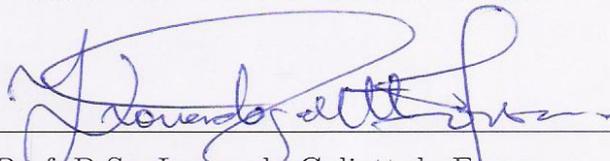
Prof. D.Sc. Afonso Celso de Castro Lemonge - Coorientador
Universidade Federal de Juiz de Fora



Prof. D.Sc. Helio José Corrêa Barbosa - Coorientador
Universidade Federal de Juiz de Fora
Laboratório Nacional de Computação Científica



Prof. D.Sc. Beatriz de Souza Leite Pires de Lima
Universidade Federal do Rio de Janeiro



Prof. D.Sc. Leonardo Goliatt da Fonseca
Universidade Federal de Juiz de Fora

*Dedico este trabalho aos meus
familiares e, em especial, ao meu
esposo*

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus pela oportunidade.

Aos meus pais, pelo carinho e atenção que sempre tiveram comigo, ao meu irmão, Fabrício, por acreditar nos meus ideais. Ao meu esposo, Jeanerson, pelo grande apoio e compreensão nas minhas ausências. À minha sogra, Maria Isabel, pelos grandes conselhos e motivações no decorrer do curso.

Ao meu orientador, Prof. Carlos Cristiano, por ter se prontificado a me orientar, desde o início do curso, também pela sabedoria e paciência que resultou no desencadeamento desse trabalho.

Ao meu coorientador, Prof. Lemonge, pelo apoio e ensinamento proporcionado, além de seu grande empenho que fizeram com que esse trabalho fosse possível.

Ao amigo, Heder Bernardino, pelo essencial auxílio com a ferramenta “perfil de desempenho” e disposição para ajudar sempre.

Ao corpo docente do Mestrado em Modelagem Computacional, especialmente aos professores: Fernanda, Luiz Paulo, Lobosco, Ciro, Rodrigo e Henrique. E também, a Natália, pela dedicação e empenho durante todo o curso.

A todos os professores do Departamento de Mecânica Aplicada Computacional: Flávia, Michèle, Patrícia, Leonardo e Flávio pelo excelente convívio, permitindo, assim, um agradável ambiente de trabalho.

À todos os colegas do Mestrado em Modelagem Computacional e em especial, aos amigos que estiveram mais próximos durante o curso: Michelli, Bárbara, Ana Paula, Victor, Nívea, Carlos Henrique e Eduardo.

Ao programa de pós graduação da Universidade Federal de Juiz de Fora, pelo auxílio financeiro.

À todos que contribuíram, direta ou indiretamente, com este trabalho, bem como com minha formação.

*“O único homem que está isento
de erros, é aquele que não
arrisca acertar.”*

Albert Einstein

RESUMO

Estratégias de penalização são muito utilizadas no trato de problemas com restrições. Problemas inerentes a escolha de valores adequados para os termos de penalização dificultam a obtenção de resultados confiáveis e robustos na sua aplicação em problemas da otimização estrutural. Técnicas baseadas em modelos de penalização adaptativa tem apresentado relativo sucesso quando aplicadas em conjunto com algoritmos evolucionistas. Apresenta-se aqui uma nova alternativa utilizando uma estratégia de lagrangeano aumentado para o trato das restrições do problema de otimização estrutural.

Encontra-se na literatura modelos para penalização adaptativa bem como o uso do lagrangeano aumentado em conjunto com algoritmos genéticos geracionais. O objetivo desse trabalho é adaptar um modelo de penalização para um algoritmo genético não geracional, bem como criar um algoritmo baseado em lagrangeano aumentado também para o algoritmo não-geracional. Esses algoritmos foram aplicados em estruturas reticuladas, muito utilizadas na construção civil como coberturas de ginásios, hangares, galpões, etc. O desempenho desses tipos de estruturas e funções matemáticas foi analisado com as técnicas de tratamento de restrição apresentadas nesse trabalho. Isso foi feito durante a busca de soluções ótimas na tentativa de minimizar os custos e satisfazer as restrições adequadas para diversas estruturas e funções matemáticas.

Palavras-chave: otimização com restrição. penalização. lagrangeano aumentado. algoritmos genéticos.

ABSTRACT

Penalty strategies are widely used in dealing with problems with constraints. Problems inherent in the choice of appropriate values for the terms of penalties difficult to obtain reliable and strong results in its application in problems of structural optimization. Techniques based on models of adaptive penalty has shown some success when applied in conjunction with evolutionary algorithms. Here is presented a new alternative using augmented Lagrangian strategy for dealing with the problem of constrained structural optimizations.

It is found in the literature models for adaptive penalties as well as the use of the augmented Lagrangian together with generational genetic algorithms. The aim of this work is to adapt a model of penalization for non-generational genetic algorithm, as well as create an algorithm based on augmented Lagrangian as also for a non-generational algorithm. These algorithms were applied to structures, widely used in construction as coverage of gymnasiums, hangars, etc.. The performance of these types of structures and functions was analyzed using mathematical techniques for handling constraints presented in this work. This was done during the search for optimal solutions in an attempt to minimize costs and satisfy the constraints appropriate for various structures and mathematical functions.

Keywords: constrained optimization. penalization. augmented lagrangian. genetic algorithm.

SUMÁRIO

1	INTRODUÇÃO.....	16
2	OTIMIZAÇÃO ESTRUTURAL.....	18
2.1	O problema de otimização com restrição	19
2.2	Otimização de estruturas reticuladas	23
3	COMPUTAÇÃO EVOLUCIONISTA APLICADA AO PROBLEMA DE OTIMIZAÇÃO COM RESTRIÇÃO.....	27
3.1	Algoritmos evolucionistas	27
3.2	Algoritmos genéticos	28
3.2.1	<i>Operadores genéticos</i>	33
3.2.1.1	<i>Recombinação</i>	33
3.2.1.2	<i>Mutação</i>	36
3.3	Abordagens inspiradas na natureza	38
3.4	Tratamento das restrições na otimização evolucionista	43
3.4.1	<i>Técnicas de penalização</i>	44
3.4.2	<i>Modelos para otimização com restrição</i>	49
3.4.3	<i>Considerações</i>	51
4	UMA ESTRATÉGIA DE PENALIZAÇÃO ADAPTATIVA PARA AL- GORITMOS GENÉTICOS NÃO GERACIONAIS	53
4.1	Modelos adaptativos para otimização com restrição usando AE's ...	53
4.2	Um modelo adaptativo para AG não-geracional	58
5	UMA ESTRATÉGIA BASEADA EM LAGRANGEANO AUMENTADO E ALGORITMOS GENÉTICOS NÃO-GERACIONAIS	63
5.1	O modelo dos multiplicadores de lagrange para otimização	64
5.2	Técnicas de aplicação do lagrangeano aumentado	65
5.3	AE's baseados em lagrangeano aumentado	69
5.4	Considerações	71
5.5	Um Algoritmo evolucionista baseado em lagrangeano aumentado ..	73

6 EXPERIMENTOS NUMÉRICOS	79
6.1 Descrição das estratégias de comparação utilizadas	80
6.2 Experimentos na suíte de funções	84
6.3 Experimentos em problemas da engenharia	113
6.3.1 Projeto de tração/compressão da mola	114
6.3.2 Projeto redutor de velocidade	115
6.3.3 Projeto da viga soldada	118
6.3.4 O Projeto do vaso de pressão	119
6.3.5 O projeto da viga engastada e livre	120
6.3.6 Discussão dos problemas de engenharia mecânica	122
6.4 Problemas de otimização estrutural	125
6.4.1 Treliça de 10 barras	125
6.4.2 Treliça de 25 barras	128
6.4.3 Treliça de 52 barras	130
6.4.4 Treliça de 72 barras	134
6.4.5 Treliça de 120 barras	137
6.4.6 Treliça de 200 barras	140
7 CONCLUSÕES	145
REFERÊNCIAS	147
8 APÊNDICE	157

LISTA DE ILUSTRAÇÕES

2.1	Treliça de 3 barras	25
3.1	Pseudo-código para algoritmos evolucionistas	28
3.2	Pseudo-código para algoritmo genético geracional	31
3.3	Pseudo-código para algoritmo genético não-geracional	32
3.4	Pseudo-código para o algoritmo do recozimento simulado	40
3.5	Pseudo-código para algoritmo sistema de colônia de formigas	41
3.6	Pseudo-código para algoritmo enxame de partículas	42
3.7	Pseudo-código para algoritmo colônias artificiais de abelhas	43
4.1	Pseudo-código para algoritmo do ranqueamento estocástico	57
4.2	Esquema de penalização adaptativa	60
4.3	Pseudo-código para o algoritmo não-geracional utilizando penalização adaptativa	62
5.1	Pseudo-código para algoritmo de lagrangeano aumentado	64
5.2	Pseudo-código para algoritmo evolucionista baseado em lagrangeano aumentado	78
6.1	Desempenho de 9 problemas obtidos através dos algoritmos A,B,C	82
6.2	Taxa de Desempenho dos algoritmos A, B e C	83
6.3	Comparação dos resultados obtidos no CEC2006	84
6.4	Gráfico em Linhas - Desempenho da medida melhor para as 30 combinações	90
6.5	Gráfico em Barras - Desempenho da medida melhor para as 30 combinações	91
6.6	Gráfico em Linhas - Desempenho da medida média para as 30 combinações	92
6.7	Gráfico em Barras - Desempenho da medida média para as 30 combinações	93
6.8	Gráfico em Linhas - Desempenho da medida mediana para as 30 combinações	94
6.9	Gráfico em Barras - Desempenho da medida mediana para as 30 combinações	95
6.10	Gráfico em Linhas - Desempenho da medida pior para as 30 combinações	96
6.11	Gráfico em Barras - Desempenho da medida pior para as 30 combinações	97
6.12	Gráfico em Linhas - Desempenho da medida nma para as 30 combinações	98
6.13	Gráfico em Barras - Desempenho da medida nma para as 30 combinações	99
6.14	Gráfico desempenho da variação do ninser em H: melhor	101

6.15	Gráfico desempenho da variação do ninser em H: média	101
6.16	Gráfico desempenho da variação do ninser em H: mediana	102
6.17	Gráfico desempenho da variação do ninser em H: pior	102
6.18	Gráfico desempenho da variação do ninser em H: nma	102
6.19	Comparação das duas técnicas: Desempenho do melhor	111
6.20	Comparação das duas técnicas: Desempenho da média	111
6.21	Comparação das duas técnicas: Desempenho da mediana	112
6.22	Comparação das duas técnicas: Desempenho do pior	112
6.23	Comparação das duas técnicas: Desempenho do nma	112
6.24	A tração/compressão da Mola	115
6.25	O redutor de velocidade	116
6.26	Viga soldada	118
6.27	Vaso de pressão	120
6.28	Viga engastada e livre	121
6.29	Comparação dos problemas de engenharia - Desempenho do melhor	123
6.30	Comparação dos problemas de engenharia - Desempenho da média	123
6.31	Comparação dos problemas de engenharia - Desempenho da mediana	124
6.32	Comparação dos problemas de engenharia - Desempenho do pior	124
6.33	Treliça de 10 barras	126
6.34	Treliça de 25 barras.	129
6.35	Treliça de 52 barras.	132
6.36	Treliça de 72 barras	135
6.37	Treliça de 120 barras	138
6.38	Módulo 120 barras.	139
6.39	Treliça de 200 barras.	141

LISTA DE TABELAS

6.1	Funções componentes do G-Suíte	85
6.2	Continuação da tabela com as funções do G-Suíte	86
6.3	Informações sobre o G-Suíte de Funções	87
6.4	Combinação de operadores genéticos	89
6.5	Quadro resumo desempenho do APM [1] do G-Suíte em cada combinação . . .	100
6.6	Resultados do melhor operador com 5 000 avaliações	103
6.7	Resultados do melhor operador com 50 000 avaliações	104
6.8	Resultados do melhor operador com 500 000 avaliações	105
6.9	Versão 1: Experimentos no G-Suíte com a função lagrangeana	107
6.10	Versão 2: Experimentos no G-Suíte com a função lagrangeana modificada . . .	108
6.11	Versão 2: Experimentos no G-Suíte com a função lagrangeana original	109
6.12	Versão 2: Experimentos no G-Suíte com os melhores resultados das tabelas 6.10 e 6.11	110
6.13	Resultados do Projeto compressão/tração da mola.	114
6.14	Variáveis de Projeto encontradas na tração/compressão da mola	115
6.15	Resultados do projeto redutor de velocidade	117
6.16	Variáveis de projeto para as melhores soluções do redutor de velocidade	117
6.17	Valores encontrados para o custo do Projeto da Viga Soldada	119
6.18	Variáveis de projeto das melhores soluções no projeto da viga soldada	119
6.19	Valores do peso no projeto do vaso de pressão	120
6.20	Variáveis de projeto encontradas para o vaso de pressão	121
6.21	Volume encontrado para o projeto Viga Engastada e Livre	122
6.22	Variáveis de projeto para as melhores soluções da viga engastada e livre	122
6.23	Desempenho do APM ^{ngra} em cada problema de engenharia mecânica	123
6.24	Desempenho do APM ^{ngrl} em cada problema de engenharia mecânica	123
6.25	Valores do peso para a treliça de 10 barras – caso discreto	126
6.26	Variáveis de projeto encontradas na treliça de 10 barras – caso discreto	127
6.27	Valores encontrados no peso final da treliça de 10 barras – caso contínuo . . .	127
6.28	Variáveis de projeto da treliça de 10 barras – caso contínuo	127

6.29	Dados de carregamento para a treliça de 25 barras(kips).	128
6.30	Agrupamento dos membros para as treliças de 25 barras.	128
6.31	Comparação dos resultados para as treliças de 25 barras – caso discreto	131
6.32	Valores encontrados para o peso final do projeto da treliça de 25 barras	132
6.33	Dados de carregamento para a treliça de 52 barras(kN).	133
6.34	Membros de agrupamento para a treliça de 52 barras.	133
6.35	Área para a seção transversal para a treliça de 52 barras.	133
6.36	Comparação dos resultados da treliça 52 barras – peso final(Kg).	134
6.37	Valores encontrados para o peso final do projeto da treliça de 52 barras	134
6.38	Dados de carregamento para a treliça de 72 barras(kips).	135
6.39	Membros de grupo treliça de 72 barras.	136
6.40	Valores encontrados para o peso final do projeto da treliça de 72 barras	136
6.41	Comparação dos resultados para a treliça de 72 barras. Peso final(W)-lb.	137
6.42	Áreas da seção transversal treliça de 120 barras.	139
6.43	Sumário estatístico treliça de 120 barras	139
6.44	Membros dos grupos para a treliça de 200 barras	142
6.45	Áreas da seção transversal treliça de 200 barras.	143
6.46	Sumário estatístico das melhores soluções treliça 200 barras	143
6.47	Sumário dos resultados do APM ^{ngra} nos problemas de otimização estrutural	144
6.48	Sumário dos resultados do APM ^{ngrl} nos problemas de otimização estrutural	144
8.1	Experimento 1: Recombinação 1 ponto(0,8) e Mutação Randômica(0,2)	157
8.2	Experimento 2: Recombinação 2 pontos(0,8) e Mutação Randômica(0,2)	158
8.3	Experimento 3: Recombinação Discreto(0,8) e Mutação Randômica(0,2)	159
8.4	Experimento 4: Recombinação Flat(0,8) e Mutação Randômica(0,2)	160
8.5	Experimento 5: Recombinação BLX(0,8) e Mutação Randômica(0,2)	161
8.6	Experimento 6: Recombinação SBX(0,8) e Mutação Randômica(0,2)	162
8.7	Experimento 7: Recombinação Geométrico(0,8) e Mutação Randômica(0,2)	163
8.8	Experimento 8: Recombinação Wriigth(0,8) e Mutação Randômica(0,2)	164
8.9	Experimento 9: Recombinação LSX(0,8) e Mutação Randômica(0,2)	165
8.10	Experimento 10: Recombinação pais múltiplos(0,8) e Mutação Randômica(0,2)	166
8.11	Experimento 11: Recombinação 1 Ponto(0,8) e Mutação Muhenblein(0,2)	167
8.12	Experimento 12: Recombinação 2 Pontos e Mutação Muhenblein	168

8.13	Experimento 13: Recombinação Discreto(0,8) e Mutação de Muhlenbein(0,2)	. 169
8.14	Experimento 14: Recombinação Flat(0,8) e Mutação de Muhlenbein(0,2)	. . . 170
8.15	Experimento 15: Recombinação BLX(0,8) e Mutação de Muhlenbein(0,2)	. . . 171
8.16	Experimento 16: Recombinação SBX((0,8) e Mutação de Muhlenbein(0,2)	. . 172
8.17	Experimento 17: Recombinação Geométrico(0,8) e Mutação de Muhlenbein(0,2)	173
8.18	Experimento 18: Recombinação Wriyth(0,8) e Mutação de Muhlenbein(0,2)	. . 174
8.19	Experimento 19: Recombinação LSX(0,8) e Mutação de Muhlenbein(0,2)	. . . 175
8.20	Experimento 20: Recombinação de pais múltiplos(0,8) e Mutação de muhlenbein(0,2) 176
8.21	Experimento 21: Recombinação 1 Ponto(0,8) e Mutação de não-uniforme(0,2)	177
8.22	Experimento 22: Recombinação 2 Pontos(0,8) e Mutação de não-uniforme(0,2)	178
8.23	Experimento 23: Recombinação BLX(0,8) e Mutação de não-uniforme(0,2)	. . 179
8.24	Experimento 24: Recombinação Flat(0,8)- Mutação de não-uniforme(0,2)	. . . 180
8.25	Experimento 25: Recombinação BLX(0,8) e Mutação de não-uniforme(0,2)	. . 181
8.26	Experimento 26: Recombinação SBX(0,8) e Mutação não-uniforme(0,2) 182
8.27	Experimento 27: Recombinação Geométrico(0,8) e Mutação de não-uniforme(0,2)	183
8.28	Experimento 28: Recombinação Wriyth(0,8) e Mutação de não-uniforme(0,2)	. 184
8.29	Experimento 29: Recombinação LSX(0,8) e Mutação de não-uniforme(0,2)	. . 185
8.30	Experimento 30: Recombinação pais múltiplos(0,8) e Mutação de não-uniforme(0,2)	186

1 INTRODUÇÃO

A otimização consiste na busca de melhores resultados para situações que se apresentam em diversas áreas. A otimização tem como finalidade buscar o máximo ou mínimo de uma função objetivo a ser modelada matematicamente de forma que essa função represente o problema a ser resolvido. Além disso, podem haver restrições que limitem o espaço de busca, ou seja, essas restrições podem ser definidas em equações e inequações, além de limitações nas variáveis que compõem o problema. Trata-se, neste trabalho, de estratégias para a busca do melhor desempenho em diversas situações da engenharia, de forma que tenha-se uma estrutura mais eficiente e, com o custo de material reduzido.

A engenharia estrutural, mais especificamente, tem buscado desenvolver ferramentas para obter projetos com melhor desempenho, quantidade reduzida de material e baixo custo. As técnicas de otimização são utilizadas com esse objetivo de forma que o projeto estrutural disponha seus elementos estruturais para suportar o carregamento a ser submetido sem risco para sua segurança. Assim, surgem as restrições de um problema de otimização de estruturas que estão relacionados aos deslocamentos dos nós, às tensões nas barras, às frequências de vibração, etc.

Uma forma de resolver problemas de otimização é a utilização de métodos numéricos, que podem ser determinísticos ou probabilísticos. Os métodos determinísticos são, geralmente, baseados no cálculo de derivadas, já os métodos probabilísticos são baseados em heurísticas que geram algoritmos muitas vezes construídos com uma população de soluções. Assim, não requerem uma solução inicial de boa qualidade. Entre esses, pode-se citar: os algoritmos genéticos [2], os sistemas imunológicos artificiais [3], enxame de partículas [4], entre outros. Os algoritmos genéticos, abordados nesse trabalho, têm como objetivo resolver problemas de otimização com restrição e para isso, serão acopladas técnicas de tratamento de restrição. A adaptação desses algoritmos quando existem restrições em seu espaço de busca é a forma mais comum de aplicá-los nesta classe de problemas. Dentre essas adaptações, ressaltam-se o uso de funções de penalização, que são acopladas a função objetivo a fim de que o algoritmo obtenha melhor desempenho.

O presente trabalho tem como objetivo utilizar um modelo de penalização adaptativa em um AG não geracional e criar um método de lagrangeano aumentado para o tratamento

da restrição nesse mesmo AG.

O capítulo 2, dá uma visão geral do problema de otimização através de definições e objetivos. Apresenta-se a formulação matemática de um problema de otimização com restrição e os métodos de tratamento de tais restrições. Posteriormente, o problema de otimização estrutural é tratado de forma abrangente, trazendo sua formulação geral e também alguns tipos de restrição comumente utilizadas. Exemplifica-se, ainda, o problema estrutural com uma treliça de três barras.

No capítulo 3, aborda-se definições sobre algoritmos evolucionistas e algoritmos genéticos detalhando-se as diferenças de um AG geracional para um não-geracional. Em seguida, discrimina-se alguns tipos de operadores utilizados na codificação real, codificação esta que será acoplada nos algoritmos desenvolvidos. Ainda nesse capítulo, explicita-se de forma ampla alguns algoritmos estocásticos para o tratamento da restrição em um problema de otimização e, além disso, algumas técnicas de penalização que são acopladas a esses algoritmos.

No capítulo 4, são tratados os métodos de penalização adaptativa presentes na literatura e, mais especificamente, um tipo dessa penalização [1] a qual adapta-se em um algoritmo genético não-geracional com codificação real.

No capítulo 5, apresenta-se as definições e características principais de uma técnica chamada de lagrangeano aumentado. Ainda nesse capítulo, descreve-se o algoritmo de penalização do lagrangeano aumentado acoplada a um algoritmo genético não-geracional, para a otimização de funções com restrições.

Em seguida, no capítulo 6, explicita-se, de forma geral, o perfil de desempenho (*performance profiles*) [5], que é uma ferramenta gráfica para visualização e comparação de vários resultados obtidos nos experimentos computacionais. Entre eles, destaca-se os problemas de otimização de estruturas reticuladas, ou seja, estruturas formadas por barras que são ligadas por nós. Nesse estudo objetiva-se minimizações de pesos dessas estruturas, onde as variáveis de projeto são as áreas das seções transversais das barras. Por fim, são apresentadas as conclusões e propostas para trabalhos futuros.

2 OTIMIZAÇÃO ESTRUTURAL

O problema de otimização estrutural vem sendo objeto de estudo desde os trabalhos pioneiros apresentados pelo cientista Maxwell, no final do século XVIII [6] quando buscava otimizar o uso de materiais na construção de pontes. Seu objetivo era construir uma ponte que utilizasse pouco material e que suportasse as necessidades de uso. Depois de vários estudos Maxwell sugeriu que a forma conceitual de uma estrutura ótima, que utilizasse menos material possível, seria constituída de elementos de treliça. No início do século XX, Michell [6] decidiu aplicar essa teoria para o projeto de vários tipos de estruturas, visando utilizar o menor volume de material. Porém, naquela época, esses estudos foram considerados muito teóricos e sem aplicação prática. Somente mais tarde com o surgimento dos computadores e do método de elementos finitos(MEF) é que problemas práticos de otimização estrutural começaram a ser estudados, principalmente na indústria aeronáutica. Na década de 70, vários algoritmos de otimização foram implementados e na década seguinte, com a implementação de softwares baseados em otimização topológica, os resultados de Michell passaram a ser aplicados na engenharia civil.

Mais especificamente, a otimização estrutural é um processo matemático que tem por finalidade obter um desempenho ótimo (mínima massa, máxima rigidez, entre outros), satisfazendo algumas restrições tanto sobre as variáveis de projeto quanto sobre o comportamento da estrutura. Um dos principais objetivos desta classe de otimização é a diminuição do custo, minimizando a quantidade de material empregado e garantindo, entretanto, que a estrutura suporte todas as restrições mecânicas, estáticas ou dinâmicas.

As três classes de otimização que podem ser utilizadas num projeto são: otimização dimensional, otimização geométrica e otimização topológica. Na **otimização dimensional**, por exemplo, um sólido unidimensional pode considerar como variável de projeto a área da seção transversal. Já em um sólido bidimensional, uma possível variável de projeto seria a sua espessura. Geralmente, no processo de otimização utiliza-se técnicas de discretização do domínio para possibilitar a obtenção de uma solução numérica, através método dos elementos finitos, por exemplo. A **otimização geométrica** tem como objetivo definir a melhor fronteira de um sólido com relação a uma função custo e suas restrições mecânicas de projeto. A **otimização topológica** tem como finalidade defi-

nir da melhor forma possível a distribuição de material em um domínio pré-determinado considerando-se uma função custo e as restrições mecânicas [6].

Geralmente, os problemas de otimização são resolvidos através de algoritmos de otimização determinísticos ou estocásticos [7]. O determinístico apresenta um comportamento onde, a partir de um dado de entrada, encontra-se uma mesma saída. Os modelos mais comuns exigem, geralmente, a primeira derivada da função objetivo em relação às variáveis de projeto. Nos algoritmos estocásticos, dada uma entrada, a saída depende de valores pseudo-rândômicos utilizados na construção do algoritmo. Usualmente, esta classe de algoritmos avalia diretamente a função objetivo, não utilizando derivadas, sendo, portanto, conhecidos como métodos de ordem zero. Desta forma, não há a necessidade da função ser diferenciável e nem contínua.

2.1 O problema de otimização com restrição

Problemas de otimização estão presentes em várias áreas do conhecimento como nas engenharias, na economia, em sistemas biológicos, etc. No caso das engenharias, estão relacionados a diversas atividades associadas a viabilização de produtos finais como: projetos de avião, de componentes mecânicos, de equipamentos elétricos, entre outros. Geralmente, o objetivo principal é a minimização ou maximização de quantidades, através de uma função objetivo sujeita a restrições necessárias e definidas, podendo ser restrições de igualdade ou de desigualdade. Entretanto, na maioria dos problemas de otimização de interesse prático, tais restrições podem ser funções complexas e, a verificação de soluções factíveis tendem a demandar um alto custo computacional.

Geralmente, apresenta-se um problema de otimização da seguinte forma:

$$\text{Minimizar } f(\mathbf{x})$$

Sujeito:

$$g_i(\mathbf{x}) \leq 0, \quad i = 1 \dots m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1 \dots l$$

sendo \mathbf{x} o vetor das variáveis de projeto, dado por: $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$ limitadas pelas

restrições de fronteira $x_i^{min} \leq x_i \leq x_i^{max}$, com $g_i(\mathbf{x})$ definindo as m restrições de desigualdade e $h_j(\mathbf{x})$ as l restrições de igualdade. O vetor \mathbf{x} é determinado por um paralelepípedo em \mathfrak{R}^n definido pelos limites inferior e superior das variáveis que, em conjunto com as restrições de desigualdade e igualdade definem o espaço de busca S . No modelo padrão da otimização, a função objetivo e suas restrições dependem das variáveis de projeto, sendo que essas restrições podem ser lineares ou não lineares. Além disso, o número de restrições de igualdade deve ser, no máximo o número de variáveis de projeto. No caso de restrições de desigualdade esse número é ilimitado.

As técnicas para tratamento das restrições podem ser divididas em: diretas ou indiretas.

Técnicas diretas: utilizam restrições explícitas, ou seja, a função restrição não sofre nenhuma alteração. Alguns métodos que utilizam essa técnica aplicam diretamente as restrições na fronteira, enquanto outros tentam redefini-las. A busca é realizada no espaço contínuo da região factível.

Técnicas indiretas: nesse caso, o problema de restrição é transformado em um problema de otimização sem restrição através do uso de uma função de penalização. Essas funções são conectadas a uma função objetivo e geram uma nova função. Essa técnica considera uma solução inicial para o problema e em seguida avalia se essa solução é factível ou não. Depois disso, são gerados vários pontos em \mathfrak{R}^n .

Com o tratamento de restrições na otimização sendo feito por meio de funções de penalização, a função $f(\mathbf{x})$ a ser otimizada, com restrições g_i e h_j , é transformada em uma nova função, da seguinte forma:

$$F(\mathbf{x}, r) = f(\mathbf{x}) + rP(\mathbf{x}) \quad (2.1)$$

onde a função $F(\mathbf{x}, r)$ é a nova função objetivo, $f(\mathbf{x})$ é a função original, r é o parâmetro de penalização e $P(\mathbf{x})$ é a função de penalização.

As funções de penalização são utilizadas para tratar a violação das restrições, ou seja, evitar que a função atinja um valor fora da região factível. O método sequencial de

minimização(SUMT) aborda três formas de métodos de penalização: método de função de penalidade exterior, método de função de penalidade interior e método de função estendida de penalidade interior [8].

Método de função penalidade exterior: a penalização ocorre na parte exterior da região factível, ou seja, a função objetivo realiza o processo de penalização na região infactível. Assim, somente haverá a penalização quando houver alguma restrição violada. Esse método pode ser utilizado para resolver problemas com restrição de igualdade e desigualdade. Para esse método, a função de penalização geralmente, é escrita na forma:

$$P(\mathbf{x}) = \sum_{i=1}^m (\max[0, g_i(\mathbf{x})])^2 + \sum_{j=1}^l (h_j(\mathbf{x}))^2 \quad (2.2)$$

onde $P(\mathbf{x}) = 0$ quando não há nenhuma restrição violada, ou seja, todas as restrições são satisfeitas. Assim, a função de otimização transformada é igual a função objetivo original. Caso existam restrições violadas, a função objetivo fica transformada na forma da equação (2.1), onde $P(\mathbf{x})$ é a função apresentada na equação (2.2) que corresponde à soma do quadrado violações das restrições $g_i(\mathbf{x})$ e $h_j(\mathbf{x})$. O processo de penalização, geralmente, é iniciado com um pequeno valor da penalização r , para evitar um mal-condicionamento do problema. Com isso, a função transformada $F(\mathbf{x})$ será mais facilmente otimizada porém, com uma maior possibilidade de ocorrer violações nas restrições. Assim, durante a otimização, o valor de r deve ser incrementado até a obtenção de valores ótimos para a função na região factível.

Método de função penalidade interior: também conhecido como método de barreira. Nesse método, a função é penalizada no espaço factível quando um ponto se aproxima da fronteira limite desse espaço. Esse processo garante uma sequência de soluções factíveis melhores que a solução inicial. Um exemplo de função de penalização interior é escrita da seguinte forma:

$$P(\mathbf{x}) = \sum_{i=1}^m -\frac{1}{g_i(\mathbf{x})} \quad (2.3)$$

Existem, também, outras funções como, por exemplo, a função de penalização expressa

como:

$$P(\mathbf{x}) = \sum_{i=1}^m -\log[-g_i(\mathbf{x})] \quad (2.4)$$

Esse tipo de penalização é muito utilizado em problemas com restrições de desigualdade. Geralmente, nas restrições de igualdade o método de penalidade exterior é mais conveniente. Dessa forma, baseada nas equações (2.2), (2.3) de penalidade, a função objetivo é escrita como: $F(\mathbf{x}, r', r) = f(\mathbf{x}) + r' \sum_i^m \frac{-1}{g_i(\mathbf{x})} + r \sum_{j=1}^l (h_j(\mathbf{x}))^2$, onde $f(\mathbf{x})$ representa a função original. A segunda parcela da equação representa a função penalidade descrita na equação (2.3) e a terceira parcela representa a função de penalidade exterior para restrições de igualdade. A penalidade r começa com um valor alto e decresce durante o processo de iteração. A grande vantagem desse método é que ficam bem definidos os domínios factível e infactível. A desvantagem de utilização desse método é que a solução ótima factível é encontrada somente quando os valores de r' tendem para o infinito.

método da função estendida de penalidade interior: essa função resume as características positivas da penalidade exterior e da penalidade interior no tratamento de restrições de desigualdade, sendo o processo iniciado geralmente na região infactível. O método é dividido em: função estendida linear de penalidade exterior e função estendida quadrática de penalidade interior, escritas na forma:

$$P(\mathbf{x}) = \sum_{i=1}^m \tilde{g}_i(\mathbf{x}) \begin{cases} \tilde{g}_i(\mathbf{x}) = -\frac{1}{g_i(\mathbf{x})}, & \text{se } g_i(\mathbf{x}) \leq \varepsilon \\ \tilde{g}_i(\mathbf{x}) = -\frac{2\varepsilon - g_i(\mathbf{x})}{\varepsilon^2} & \text{se } g_i(\mathbf{x}) > \varepsilon \end{cases}$$

onde ε é uma constante negativa definida por: $\varepsilon = -C(r')^a$, sendo $\frac{1}{3} \leq a \leq \frac{1}{2}$, de forma que ε garante uma sequência de pontos viáveis.

Método dos multiplicadores de Lagrange: alguns autores consideram a função lagrangiana como um método adicional de penalização, porém apresenta um embasamento teórico superior, com formas de garantir uma convergência mais rápida e com maior estabilidade em relação aos métodos de penalização. O método dos multiplicadores foi proposto em 1951, baseado em fundamentos teóricos de Khun e Tucker [9]. A função

lagrangeana é escrita como:

$$L(\mathbf{x}, \lambda, \mu) = \sum_{i=1}^m \lambda_i (\max[0, g_i(\mathbf{x})]) + \sum_{j=1}^l \mu_j (h_j(\mathbf{x}))$$

onde λ_i e μ_j são variáveis reais denominadas de multiplicadores de Lagrange para as restrições de desigualdade e igualdade, respectivamente.

Mais tarde, em 1968, essa função lagrangeana foi modificada por Hestenes e Powell [10], que apresentaram o método de lagrangeano aumentado, proposto inicialmente para problemas de otimização com restrição de igualdade e adaptado, em 1974 por Rockafellar [11], para o tratamento de restrições de desigualdade [9]. Essa técnica será discutida no capítulo 5.

2.2 Otimização de estruturas reticuladas

Estudos de estruturas reticuladas são bastante frequentes no que se refere à otimização estrutural [7, 12, 13]. Experimentos realizados nesses trabalhos utilizam várias estruturas reticuladas como: treliças, pórticos e domos. Existem diversos parâmetros para otimizar uma estrutura, os quais são denominados variáveis de projeto. Essas variáveis são representadas pelas dimensões ou áreas das seções transversais, posições dos nós da estrutura, número de barras da estrutura, número máximo de tipos de barras, dentre outras. As variáveis podem ser discretas, contínuas ou mistas. As contínuas assumem valores dentro de um intervalo em \Re , já as discretas podem assumir valores isolados que representam, por exemplo, propriedades físicas dos materiais. As restrições em problemas de otimização de estruturas reticuladas se referem, geralmente, a deslocamentos máximos dos nós, tensões máximas admissíveis, frequências de vibração, cargas críticas de flambagem, etc. Análises estruturais usualmente envolvem discretizações com simulações via métodos de elementos finitos.

A função objetivo $f(\mathbf{x})$, numa estrutura reticulada, visando a otimização do peso é escrita como [14]:

$$f(\mathbf{x}) = \sum_{i=1}^n \rho_i A_i L_i \tag{2.5}$$

onde $\mathbf{x} \in \Re^n$ é o vetor de variáveis de projeto com n sendo o número de barras, ρ_i é a

massa específica do material, A_i é a área da seção transversal e L_i é o comprimento da i -ésima barra da estrutura.

Essa função tem a finalidade de encontrar as áreas $\mathbf{x} = [A_1, A_2, \dots, A_n]^T$ as quais minimizam o peso da estrutura. Pode-se incluir, também, um conjunto de coordenadas dos nós da estrutura como variáveis de projeto, cujos valores influenciarão o peso da estrutura. Esses problemas geralmente estão sujeitos a restrições de desigualdade. Dos vários tipos de restrições possíveis, pode-se citar o da tensão normal máxima como um dos mais utilizados, sendo equacionado na seguinte forma [14]:

$$\frac{|\sigma_i|}{\sigma^{\max}} - 1 \leq 0$$

onde $i = 1, 2, 3, \dots, n$ representam as barras monitoradas, sendo n o número de barras, σ_i a tensão normal no i -ésimo membro e σ^{\max} é a tensão máxima. Tal restrição visa evitar que se obtenha estruturas com peças que não estejam de acordo com as tensões máximas admitidas para o material utilizado.

Outro tipo de restrição muito utilizada é a restrição de deslocamento [14]:

$$\frac{|d_j|}{d^{\max}} - 1 \leq 0$$

, com $j = 1, 2, \dots, m$, onde m é o número de graus de liberdade monitorados da estrutura, d_j é o deslocamento do j -ésimo grau de liberdade global, d^{\max} é o valor máximo de deslocamento permitido. Geralmente, a restrição de deslocamento visa obedecer diretrizes normativas bem como evitar a obtenção de estruturas que possam gerar movimentação excessiva e desconforto para usuários.

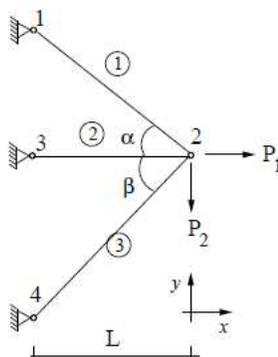
O carregamento a que está sujeita uma estrutura é transmitido para as barras da estrutura que sofrerão deformações gerando as tensões internas. A tensão normal em cada barra é calculada pela razão entre o esforço normal e a área da seção transversal da barra. Para a determinação dos deslocamentos e esforços deve-se resolver um sistema de equações que representa a estrutura discretizada [15]. Esse sistema representa o equilíbrio da estrutura em função do deslocamento que é dada por:

$$[K]\{u\} = \{F\} \quad (2.6)$$

sendo $[K]$ a matriz rigidez da estrutura, $\{F\}$ é o vetor de cargas da estrutura e $\{u\}$ o vetor de deslocamentos a serem determinados.

Exemplificando, apresenta-se uma treliça de 3 barras, apresentadas na referência [15], possuindo 4 nós, com somente 1 nó com deslocamentos livres, conforme a figura 2.1. Para

Figura 2.1: Treliça de 3 barras



esta estrutura reticulada no plano, onde somente o nó 2 possui graus de liberdade, tem-se o sistema:

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{Bmatrix} u_{2x} \\ u_{2y} \end{Bmatrix} = \begin{Bmatrix} P_1 \\ -P_2 \end{Bmatrix}$$

onde k_{ij} são coeficientes de rigidez que dependem das características elásticas e geométricas das barras que compõem a estrutura, P_1 e P_2 os carregamentos aplicados sobre o nó 2 e u_{2x} e u_{2y} os deslocamentos do nó 2 a serem determinados.

Sendo o objetivo minimizar o volume $V(A_i)$ dessa treliça, de maneira que diminua o custo do material, de forma segura, tem-se a função a ser minimizada:

$$V(A_i) = \sum_{i=1}^3 A_i L_i$$

onde L_i são os comprimentos das barras e A_i são as áreas das barras das seções transversais. Com as cargas P_1 e P_2 atuantes, o nó 2 da treliça sofrerá deslocamentos

nas direções dos eixos x e y . Assim, as barras da estrutura sofrerão deformações e estarão submetidas às tensões internas. Considerando σ_i como sendo as tensões nas barras e u_{2x} e u_{2y} os deslocamentos, o problema da figura 2.1 estará sujeito a:

$$\sigma_i \leq \sigma^{\max}, \quad i = 1, 2, 3$$

$$u_{2x}, u_{2y} \leq u_2^{\max}$$

assumindo, ainda, as restrições $A_i^{\min} \leq A_i \leq A_i^{\max}$ das variáveis de projeto. onde σ^{\max} é a tensão máxima nas barras e u_2^{\max} corresponde ao deslocamento máximo no nó 2.

Para uma determinada configuração de áreas adotadas, a solução do sistema 2.6 determina os deslocamentos para esta configuração e, conseqüentemente, os esforços e tensões normais em cada ponto da treliça.

De uma forma mais completa, em um problema de otimização estrutural as variáveis de projeto podem estar relacionadas com as áreas das seções transversais das barras, com as posições dos nós de uma estrutura ou ainda se referir ao número ou tipos das barras. Além disso, esses tipos de barra podem ter diferentes características na sua configuração estrutural, como por exemplo, a exigência de simetria na estrutura.

Os algoritmos evolucionistas tem sido bastante empregados como ferramentas de otimização estrutural. No próximo capítulo, descreve-se esta classe de algoritmos bem como suas principais propriedades e características quando aplicados em problemas com restrições.

3 COMPUTAÇÃO EVOLUCIONISTA APLICADA AO PROBLEMA DE OTIMIZAÇÃO COM RESTRIÇÃO

3.1 Algoritmos evolucionistas

Esta classe de algoritmos baseia-se na teoria da evolução das espécies de Charles Darwin [16] onde, indivíduos com maior capacidade de adaptação ao meio ambiente, têm maiores chances de sobrevivência. Segundo a teoria, as características genéticas são passadas de geração para geração, com isso, os indivíduos gerados possuem material genético dos indivíduos anteriores, geralmente escolhidos entre os mais aptos [16].

A computação evolucionista, inspira-se na teoria da evolução visando desenvolver algoritmos, denominados evolucionistas, cujos principais representantes são [13]: algoritmos genéticos(AG), programação evolucionista(PE) e estratégias evolucionistas(EE). Todos os modelos são baseados em população, que representam possíveis soluções candidatas ao problema proposto. Operadores genéticos são aplicados visando replicar a seleção natural e o processo de reprodução das espécies. Destaca-se, entre os mais utilizados os operadores de seleção, recombinação e mutação. Entre os diversos tipos de algoritmos evolucionistas, os mais utilizados são os algoritmos genéticos, principalmente quando se trata de otimização estrutural. Apesar dos algoritmos desenvolvidos neste trabalho para o trato de restrições poderem ser adaptados para qualquer tipo de algoritmo evolucionista apresenta-se a implementação utilizando um algoritmo genético de codificação real.

Além disto, enfoca-se no desenvolvimento de uma estratégia de reposição conhecida como não-geracional. O interesse neste modelo vem das características específicas que apresenta, as quais serão descritas à frente, e por serem pouco exploradas para problemas com restrição.

Os algoritmos evolucionistas funcionam, basicamente, da seguinte forma: Primeiro,

ocorre a inicialização da população, geralmente de forma randômica, devendo cada indivíduo ser avaliado e associado a uma medida de aptidão. Entra-se, então, em um ciclo evolutivo, até que um critério de convergência seja alcançado onde, indivíduos são selecionados para a reprodução, gerando filhos por recombinação/mutação para uma nova população e, por fim, os indivíduos(filhos) serão avaliados novamente, reiniciando o ciclo. Destaca-se que cada algoritmo evolucionista executa estes passos de uma forma específica. Mostra-se, a seguir, o pseudo-código de um algoritmo evolucionista [7]:

Algoritmo evolucionista

Início

$t=0$

Inicialize população(t)

Avalie população(t)

Enquanto não termina o processo faça:

$t = t + 1$

Selecione a população(t) a partir da população($t-1$)

Altere a população(t)

Fim Enquanto

Fim

Figura 3.1: Pseudo-código para algoritmos evolucionistas

As três diferenças mais significativas entre os algoritmos evolucionistas(AEs) e os métodos de busca clássicos [17] são: (i) os AEs buscam por meio de uma população de soluções em cada iteração, ao invés de um único candidato, (ii) os AEs não necessitam de muita informação do problema, como, por exemplo, gradientes, (iii) os AEs utilizam regras de transição probabilística, e não determinísticas. Estas características tornam os AEs flexíveis, permitindo que sejam aplicados e/ou adaptados para diversos tipos de problemas. A seguir, descreve-se o AE conhecido como algoritmo genético.

3.2 Algoritmos genéticos

Em 1950 e 1960, biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. John Holland propôs um algoritmo genético (AG) para simular sistemas adaptativos em 1975, publicando o livro *Adaptação em sistemas naturais e artificiais* (*Adaptation in Natural and Artificial Systems*) [18]. Após os trabalhos de Holland(1975),

seu ex-aluno David Goldberg(1989) [19, 20], aprofundou os estudos dos AGs seguindo o princípio da seleção natural de sobrevivência do mais apto de Charles Darwin(1859). Princípio esse que, baseia-se na idéia de que indivíduos melhores adaptados ao ambiente tem mais chance de sobreviver e gerar descendentes.

Os algoritmos genéticos como métodos de otimização e busca, inspiram-se em mecanismos de evolução com objetivo de varrer eficientemente espaços de busca encontrando resultados mais próximos da melhor solução para o problema de maximização ou minimização. As soluções obtidas, ótimas ou sub-ótimas, podem ser interpretadas como resultados do processo evolutivo simulado pelo algoritmo. Os AGs, como integrante dos AEs, partem de uma população de indivíduos que se desenvolvem através da aplicação de operadores genéticos que visam aprimorar a aptidão média da população no decorrer das gerações. Cada um desses indivíduos, representa uma possível solução para um problema que, no caso dos AGs, tem as variáveis de projeto codificadas em espaço específico para a aplicação dos operadores. Por construção, os AGs são definidos para maximização. Porém, a adaptação para minimização é trivial. Assim como todos AEs, os AGs são adequados para otimização sem restrição, sendo necessário adaptações para o uso em problemas com restrições.

Diferente dos métodos clássicos de otimização, como o método de Newton ou dos gradientes conjugados, os algoritmos genéticos não necessitam de calcular derivadas para determinar a direção de busca visando localizar a próxima posição da possível solução. Os AGs fazem uma busca sobre um conjunto de indivíduos pertencente à população, não ficando restrito a um único candidato, que pode ter sido iniciado em posição desfavorável no espaço de busca, dificultando a localização da região ótima. Aplicam-se a uma grande diversidade de problemas, apresentando desempenho adequado para uma em boa parcela destes problemas, podendo-se destacar problemas de otimização global e otimização inteira. Os algoritmos genéticos são naturalmente desenvolvidos para problemas de otimização sem restrição. Como visto, na seção anterior, a otimização estrutural, geralmente, envolve muitos tipos de restrições, tais como: tensões e deformações máximas. Porém, a adaptação para o trato de restrições pode ser feita utilizando os mesmos modelos da programação não-linear. Logicamente, estudos contínuos tem sido feitos visando obter desempenho similar ou superior dos AGs adaptados para problemas com restrição em relação aos algoritmos da programação matemática.

O funcionamento de um algoritmo genético é similar ao dos AEs já descritos. Porém, nota-se o uso de uma nomenclatura muito mais próxima da utilizada em genética das populações para definir seus componentes e operadores. Assim, gerada uma população inicial de **cromossomos**, que representam possíveis soluções de um problema,, a população é avaliada e cada cromossomo recebe uma avaliação ou seja, a sua **aptidão**. Baseado na lei de Darwin são **selecionados** os cromossomos para a próxima **geração**, com maior probabilidade para os mais aptos. Os selecionados transferem suas características para seus **descendentes** através da **reprodução**, simulada por meio dos operadores **genéticos** de **recombinação** e **mutação**. Todo esse processo é repetido até que seja encontrada uma solução satisfatória. Goldberg [19] explica alguns fenômenos e termos da biologia utilizado nos AGs:

Cromossomo e Genoma: representam a estrutura de dados que codifica a solução para um problema, ou seja, eles representam um ponto no espaço de busca;

Genoma: conjuntos de genes para formar um indivíduo;

Gene: parâmetro codificado no cromossomo, ou seja, um elemento do vetor que representa o cromossomo. A representação de cada parâmetro de acordo com o alfabeto utilizado(binário, inteiro ou real);

Indivíduo: um conjunto de valores $x = x_1, x_2, x_3, \dots, x_n$ para as variáveis de projeto que definem um ponto no qual se pode avaliar a função objetivo;

Genótipo: representa a informação contida no cromossomo ou genoma;

Fenótipo: estrutura construída a partir das informações do genótipo. É o cromossomo decodificado;

Alelo: representa os valores que o gene pode assumir;

População: conjunto de soluções;

Seleção: processo pelo qual indivíduos de uma população são selecionados de acordo com o grau de adaptação ao meio. Privilegia os mais aptos a permanecerem e aumentarem

sua participação na população. Os métodos de seleção mais utilizados são: roleta, torneio, elitismo;

Recombinação: operador que simula a reprodução, onde o material genético de indivíduos pais é combinado a fim de produzirem filhos com características herdadas dos pais.

Epistasia: interação entre genes do cromossomo, ou seja, quando o valor de um gene influencia no valor de outro gene.

O algoritmo genético realiza uma decodificação das variáveis que correspondem as variáveis de projeto, o fenótipo. No espaço dos genótipos os operadores genéticos são aplicados e, no espaço dos fenótipos obtém-se a aptidão da população que viabilizará o processo de seleção estocástica. Em relação a estratégia de reposição da população nos AGs, destacam-se o algoritmo genético geracional e o algoritmo genético não-geracional (*steady-state*). Nos algoritmos genéticos geracionais toda população é substituída a cada geração, ocasionando grande variabilidade do material genético entre as gerações podendo, inclusive, causar perdas de difícil recuperação. Em 3.2 é apresentado o pseudo-código de um algoritmo genético geracional [13]:

Algoritmo genético geracional

Início

Inicialize a população P aleatoriamente

Avalie os indivíduos na população P

Repita

Repita

Selecione 2 indivíduos em P

Aplique operadores de recombinação com probabilidade p_r

Aplique operadores de mutação com probabilidade p_m

Insira novos indivíduos em P'

Até população P' completa

Avalie os indivíduos na população P'

$P \leftarrow P'$

Até o critério de parada ser satisfeito

Fim

Figura 3.2: Pseudo-código para algoritmo genético geracional

No caso do AG não-geracional, somente uma pequena parcela da população (geralmente 2 indivíduos) é substituída a cada iteração. O comportamento é semelhante ao de populações biológicas, ou seja, cria um indivíduo de cada vez. A cada criação é realizada uma nova avaliação, onde um indivíduo da população pode ser substituído pelo novo indivíduo, de acordo com algum critério pré-definido [21]. Este critério é crucial para o desempenho do AG não-geracional, podendo gerar convergência prematura com maior frequência que o modelo geracional. Geralmente, como é o caso deste trabalho, adota-se a substituição do pior indivíduo da população. Em 3.3 apresenta-se o pseudo-código de um algoritmo do tipo não-geracional [13]:

Algoritmo genético não-geracional

Início

Inicialize a população P aleatoriamente

Avalie indivíduos na população P

Repita

Selecione operador genético

Selecione indivíduo(s) para reprodução

Aplique operador genético

Avalie indivíduo(s) gerado(s)

Selecione indivíduo f para sobreviver

Se f é melhor que o pior elemento de P **Então**

Insira f em P de acordo com seu “ranking”

Até critério de parada satisfeito

Fim

Figura 3.3: Pseudo-código para algoritmo genético não-geracional

Apesar de apresentar diversas vantagens em relação aos métodos clássicos de otimização em problemas de otimização global, os AGs tem um custo computacional elevado, principalmente quando a avaliação dos indivíduos pela função objetivo é computacionalmente cara. Apesar disso, eles são muito utilizados como verifica-se: na otimização multimodal, otimização combinatorial, otimização inteira, otimização multiobjetivo podendo-se citar aplicações em otimização de planejamento, no problema do caixeiro viajante, no problema de otimização de rota de veículos, na otimização de layout de circuitos, em otimização de distribuição, em otimização em negócios, em síntese de circuitos eletrônicos, entre muitas outras.

3.2.1 Operadores genéticos

Algoritmos genéticos necessitam de parâmetros de controle da sua estrutura de funcionamento. A eficiência do funcionamento do algoritmo genético depende diretamente destes parâmetros, destacando-se: o tamanho da população e os operadores genéticos, com suas respectivas probabilidades de aplicação.

O tamanho da população refere-se a quantidade de indivíduos que compõem a população sendo, geralmente, constante durante a evolução. Quando o número de indivíduos é pequeno existem muitas chances de perda da diversidade necessária para a busca da melhor solução. Por outro lado, se a população for muito grande o algoritmo pode tornar-se ineficiente em se tratando do tempo necessário para avaliar a população, exigindo assim, mais recursos computacionais [22].

O princípio básico dos operadores genéticos é a transformação da população através de sucessivas gerações afim de obter indivíduos cada vez mais aptos. Esses operadores são necessários para combinar e variar o material genético representado nas gerações. No algoritmo genético tradicional, a codificação das variáveis do problema é feita utilizando uma cadeia binária, sendo portanto, comum a chamada codificação binária. Porém, existem alguns algoritmos genéticos que utilizam a codificação de indivíduos diretamente no espaço das variáveis de projeto, geralmente reais [22]. Cada codificação apresenta comportamento e desempenho que dependem do problema em que estão sendo utilizados.

Uma abordagem com codificação real será utilizada nos algoritmos desenvolvidos para otimização com restrição. A codificação real, ou de ponto flutuante, é indicada para problemas que utilizam variáveis do domínio contínuo. Nessa representação o cromossomo é um vetor em \mathbb{R}^n com n sendo o número de variáveis do problema. A precisão das variáveis nesse tipo de codificação está associada à precisão da máquina em que o algoritmo está sendo executado. Nessa codificação o indivíduo já representa uma possível solução do problema, com cada gene representando uma variável de projeto. Existem diversos tipos de operadores de recombinação e mutação específicos para a codificação real [13]. Alguns deles serão descritos a seguir.

3.2.1.1 Recombinação

É responsável pela troca de material genético entre os pais durante o processo de reprodução, tendo a herança das características pelos filhos como consequência. Este ope-

rador é utilizado de acordo com uma probabilidade indicativa de sua aplicação entre pais previamente selecionados. O processo de seleção tem a finalidade de determinar os locais mais promissores da busca e o operador de recombinação combina esses locais de forma a tentar gerar soluções de maior qualidade para o problema. Geralmente, este operador é aplicado com uma taxa de probabilidade bem alta com valores variando entre taxas de 60% a 95% [21].

Inicialmente são considerados dois cromossomos C_1 e C_2 onde:

$$C_1 = c_1^1, c_2^1, \dots, c_n^1$$

$$C_2 = c_1^2, c_2^2, \dots, c_n^2$$

Descreve-se, então, alguns modelos de operadores de recombinação (*crossover*) para codificação real [13]:

Recombinação uniforme: é gerado um filho $F = f_1, f_2, \dots, f_n$ onde os valores f_i são escolhidos randomicamente no intervalo $[c_i^1, c_i^2]$. Assim: $c_i^1 \leq f_i \leq c_i^2$;

Recombinação simples: escolhe-se um ponto i , randomicamente, no intervalo entre $1, 2, \dots, n - 1$ e em seguida, realiza-se a troca do material genético entre os cromossomos C_1 e C_2 .

$$F_1 = (c_1^1, c_2^1, \dots, c_i^1, c_1^2, c_{i+1}^2, \dots, c_n^2)$$

$$F_2 = (c_1^2, c_2^2, \dots, c_i^2, c_{i+1}^1, \dots, c_n^1)$$

Recombinação aritmética: são gerados dois filhos F_1 e F_2 , onde: $F_1 = (f_1^1, f_2^1, \dots, f_n^1)$ e $F_2 = (f_1^2, f_2^2, \dots, f_n^2)$. sendo: $f_i^1 = \lambda c_i^1 + (1 - \lambda)c_i^2$ e $f_i^2 = \lambda c_i^2 + (1 - \lambda)c_i^1$,

onde λ é uma constante, geralmente com valor no intervalo $[0, 1]$ ou uma variável com seu valor sendo calculado em função da geração em que se encontra.

Recombinação $BLX-\alpha$ (Blended crossover): é gerado um filho $F_1 = (f_1^1, f_2^1, \dots, f_n^1)$, onde f_i é um número escolhido aleatoriamente no intervalo $[c_{min} - I.\alpha, c_{max} + I.\alpha]$, de

forma que: $c_{max} = \max(c_i, c_i)$, $c_{min} = \min(c_i, c_i)$, $I = c_{max} - c_{min}$, sendo α uma constante escolhida no intervalo $0 \leq \alpha \leq 0.5$; Quando a taxa α for igual a zero, tem-se a recombinação uniforme.

Recombinação discreto: é gerado um filho F_i , onde f_i é um valor aleatoriamente escolhido dentro do conjunto $\{c_i^1; c_i^2\}$;

Recombinação de linha estendida(extended line crossover): um elemento do cromossomo do filho é gerado da seguinte forma: $f_i = c_i^1 + \alpha(c_i^2 - c_i^1)$ onde α é um valor acrescido aleatoriamente no intervalo $[-0.25, 1.25]$;

Recombinação intermediária estendida: igual ao operador de recombinação de linha estendida com uma única diferença: α é uma variável, escolhida aleatoriamente, para cada f_i de forma que: $f_i = c_i^1 + \alpha_i(c_i^2 - c_i^1)$.

Recombinação heurística de Wright: para gerar um filho escolhe-se o pai com maior aptidão onde cada variável do cromossomo filho é: $f_i = \alpha \cdot (c_i^1 - c_i^2) + c_i^1$ sendo α um número aleatório pertencente ao intervalo $[0, 1]$;

Recombinação linear BGA(Breeder genetic algorithm): gera-se um filho com a seguinte expressão:

$$f_i = c_i^1 \pm rang_i \cdot \gamma \cdot \Lambda$$

em que

$$\Lambda = \frac{c_i^2 - c_i^1}{\|C_1 - C_2\|}$$

quando utiliza-se o sinal negativo(escolhido randomicamente) na expressão de f_i é escolhida uma taxa de probabilidade de 0.9 e para o termo $rang_i$ geralmente escolhe-se: $0.5 \cdot (b_i - a_i)$, , sendo a_i e b_i a faixa definida para a variável em questão. Onde a determinação de γ é feita por:

$$\gamma = \sum_{k=0}^{15} a_k 2^{-k}$$

com $\alpha_i \in \{0,1\}$ que é obtido aleatoriamente com probabilidade $p(\alpha_i = 1) = \frac{1}{16}$;

Recombinação LS(line search crossover): gera-se um filho que é a média entre os pais:

$$F_1 = 0.5(c_1^1 + c_1^2, \dots, c_n^1 + c_n^2)$$

sendo C_1 é o pai com melhor aptidão, e se a aptidão de F_1 for melhor que a aptidão de C_1 , então o novo filho será definido como F_1 . Caso contrário, um novo filho F_2 é gerado, onde:

$$F_2 = (c_1^1 + 0.5(c_1^1 - c_1^2), \dots, c_n^1 + 0.5(c_n^1 - c_n^2))$$

Por fim, escolhe-se o melhor filho entre F_1 e F_2 .

Recombinação geométrica[23]: utiliza dois pais para gerar um filho. Considerando-se os pais C_1 e C_2 o filho F_3 será dado por:

$$F^3 = (\sqrt{c_1^1 \cdot c_1^2}, \dots, \sqrt{x_n^1 \cdot x_n^2})$$

Porém, a recombinação geométrica pode ser generalizada da seguinte forma:

$$F_3 = \langle (c_1^1)^\alpha \cdot (c_1^2)^{(1-\alpha)}, \dots, (c_n^1)^\alpha \cdot (c_n^2)^{(1-\alpha)} \rangle$$

para $0 \leq \alpha \leq 1$

E também é possível incluir diversos pais onde:

$$F^{k+1} = ((c_1^1)^{\alpha_1} \cdot (c_1^2)^{\alpha_2} \dots (c_1^k)^{\alpha_k}, \dots, (c_n^1)^{\alpha_1} \cdot (c_n^2)^{\alpha_2} \dots (c_n^k)^{\alpha_k})$$

sendo $\alpha_1 + \dots + \alpha_k = 1$.

3.2.1.2 Mutaçãõ

A mutaçãõ é necessária para a introduçãõ do material genético na populaçãõ, sendo utilizada também de acordo com uma taxa de probabilidade, indicativa de sua da aplicaçãõ nos indivíduos selecionados da populaçãõ [22]. Se a taxa de probabilidade de mutaçãõ for muito baixa pode haver falta/perda de diversidade na populaçãõ. Por outro lado,

se for alta haverá perturbações aleatórias, com os filhos podendo perder as semelhanças com os pais, o que atrapalha o direcionamento da população no espaço de busca com consequências diretamente na convergência. A aplicação deste operador aos indivíduos é feita através de uma taxa de mutação pequena no caso dos AGs pois é um operador genético secundário (*background*). Devem cumprir a finalidade de gerar e recuperar a perda de material genético. Assim, como nos operadores de recombinação, existem diversos operadores de mutação para codificação real [13], sendo alguns apresentados abaixo:

Mutação aleatória: aplica-se a mutação no cromossomo c_i e obtendo-se um novo c_i^* . Nesse processo o c_i^* será um número escolhido aleatoriamente pertencente ao intervalo $[a_i, b_i]$;

Mutação não-uniforme: quando aplicado em uma geração t e considerando t_{max} o número máximo de gerações e definida por:

$$c_i^* = \begin{cases} c_i + \Delta(t, b_i - c_i), & \text{se } \tau = 0 \\ c_i + \Delta(t, c_i - a_i), & \text{se } \tau = 1 \end{cases}$$

Sendo τ escolhido aleatoriamente entre 0 e 1. E a função $\Delta(t, y)$ é dada pela seguinte fórmula:

$$\Delta(t, y) = y(1 - r^{(1 - \frac{t}{t_{max}})^\alpha})$$

onde r é um número aleatório no intervalo de $[0, 1]$ e α é um parâmetro escolhido pelo usuário e determina o grau de dependência do número de iterações.

Mutação small-creep e big-creep: esses operadores têm suas diferenças associadas ao valor máximo dos deslocamentos a serem aplicados na variável que sofre a mutação. Gera novos elementos de tal forma que se distanciam em níveis diferentes da variável original (menos para small-creep e mais para big-creep). Isso é feito introduzindo “deslocamentos” nos cromossomos controlados por quantidades escolhidas aleatoriamente.

Mutação de Muhlenbein: o elemento c_i^* que sofreu mutação é dado por:

$$c_i^* = c_i \pm rang_i \cdot \gamma, \quad \text{onde: } rang_i = 0.1 * (b_i - a_i)$$

sendo que os operadores “+” ou “-” escolhidos na expressão c_i^* com a probabilidade de 0.5 e $\gamma = \sum_{k=0}^{15} a_k \cdot 2^{-k}$, em que $a_i \in [0, 1]$ e é obtido aleatoriamente com probabilidade $p(\alpha_i = 1) = \frac{1}{16}$. É muito comum encontrar variações nesse operador na forma de apresentação de γ ;

Mutação de uma unidade: o filho é gerado através da adição ou subtração aleatória de uma unidade da variável c_i escolhida aleatoriamente.

3.3 Abordagens inspiradas na natureza

Apresenta-se agora, alguns algoritmos que são bastante utilizados como algoritmos estocásticos adaptados para otimização com restrição e que são referenciados nos capítulos seguintes. Esses algoritmos de comportamento estocástico utilizam a avaliação da função objetivo e não necessitam do cálculo da derivada durante o processo de otimização.

Os métodos bio-inspirados são aplicados em problemas de otimização e baseiam-se em população para a busca da solução factível durante a pesquisa estocástica. Existem diversos trabalhos na literatura para o tratamento de restrições utilizando esses métodos [24], que também podem ser chamados de meta-heurística. Existem diversos algoritmos bio-inspirados tais como: algoritmos evolucionistas, sistemas imunológicos artificiais, exame de partículas, entre outros. Essas técnicas tendem a trabalhar somente com as funções avaliações, onde a informação aplica a função sem derivadas na população estocástica. Quando surgem problemas com restrição, muitas vezes, a meta-heurística é acoplada com uma técnica de tratamento de restrição, como por exemplo, técnicas de penalidade. Essa adaptação é realizada somente se as restrições não sejam satisfeitas automaticamente pelas soluções geradas.

Ressalta-se, que a forma de adaptação para a restrição destes e outros algoritmos, costuma ser a mesma utilizada para os AEs. Diferem, principalmente, pelas propriedades que cada algoritmo bio-inspirado apresenta na sua forma original, propriedades estas

que tendem a ser conservadas no algoritmo adaptado para restrição. A seguir, serão apresentados alguns tipos de algoritmos inspirados na natureza.

Recozimento simulado

O modelo baseado no recozimento simulado, fenômeno da termodinâmica, consiste em uma meta-heurística para otimização global. Não utiliza população, modificando somente uma única candidata a solução definida inicialmente. Esta meta-heurística simula um processo térmico utilizado em metalurgia para obtenção de estados de baixa energia num sólido, de modo a reduzir os defeitos de estruturas metálicas. A primeira etapa, se inicia com um aumento da temperatura do sólido para um valor máximo, no qual ele se funde; Na segunda etapa, realiza-se o resfriamento lentamente até que o material se solidifique, sendo acompanhado e controlado esse arrefecimento [25]. Conforme visto, o algoritmo de recozimento simulado substitui a solução atual por uma solução próxima, escolhida de acordo com uma função objetivo e com uma variável de temperatura. Quanto maior for essa variável maior será a variação na próxima solução escolhida. Durante a execução do algoritmo a temperatura vai diminuindo, simulando uma busca local, de forma que o algoritmo atinja a solução ótima. Esse algoritmo é vantajoso porque permite saltos, possibilitando a busca em regiões distantes da solução ótima atual. Desta forma permite que se possa escapar de ótimos locais durante o processo de busca. Uma desvantagem é que não existe um critério de parada bem definido para obter a melhor solução possível. Em seguida, em 3.4, será apresentado o pseudo-código desse algoritmo [13]:

Sistemas imunológicos artificiais(SIA)

Os sistemas imunológicos artificiais são inspirados no sistema imunológico biológico humano, cuja principal função é defender o organismo de diversos agentes externos. No sistema imunológico dos seres humanos, os melhores anticorpos são clonados, hipermutados e selecionados. Além disso, as vezes são gerados anticorpos randômicos(dos seres humanos) para obter maior diversidade na população. Outro fato interessante é quando um organismo é atacado novamente por um antígeno, rapidamente é desenvolvida uma imunização. Essa habilidade de adaptação é conhecida como seleção clonal [26] que é um algoritmo similar aos métodos de busca estocástica, onde os indivíduos da população são os anticorpos. Baseia a evolução no princípio em que cada indivíduo é clonado, hiper-

Algoritmo recozimento simulado**Início**

$t \leftarrow 0$
 Iniciar temperatura T
 Selecionar uma solução x_k aleatoriamente
 Avaliar $f(x_k)$

Repita**Repita**

Selecionar uma nova solução x_n na vizinhança de x_k
 através de uma pequena modificação em x_k

Se $f(x_k) < f(x_n)$ **faça**

$x_k \leftarrow x_n$

Senão

Calcular $p = e^{(f(x_n)-f(x_k))/T}$

Escolher r entre $[0, 1]$ aleatoriamente

Se $r < p$ **faça**

$x_k \leftarrow x_n$

Fim se**Fim se**

Até critério de terminação satisfeito

Atualizar T de acordo com uma função $g(T, t)$.

$t \leftarrow t + 1$

Até critério de parada satisfeito

Fim

Figura 3.4: Pseudo-código para o algoritmo do recozimento simulado

mutado com os indivíduos com maior afinidade são selecionados. O conceito de afinidade está relacionada ao valor da função aptidão [27].

Os sistemas imunológicos artificiais(SIA) são ferramentas computacionais baseadas nesses conceitos biológicos com o objetivo de resolver problemas com soluções reais de otimização em diversas áreas. Tem características que permitem busca local e busca global, com capacidade de encontrar soluções de boa precisão e com pouco ajuste de parâmetros [17].

Algoritmos inspirados em colônias de formigas

É um algoritmo inspirado no comportamento de formigas que utiliza a comunicação via feromônio para guardar um caminho entre a colônia e o local onde se encontra a comida. Isso acontece quando as formigas saem de sua colônia e caminham aleatoriamente até encontrar a comida. Quando a comida é encontrada, as formigas retornam à colônia

deixando um rastro de feromônio para que as outras formigas da colônia retornem e sigam essa trilha para buscar a comida.

Quando aplicado em problemas de otimização, o conceito de evaporação do feromônio estão associados a possibilidade de se obter a rota ótima do problema(ótimo do problema), evitando a convergência para uma solução local ótima [28].

No algoritmo de sistema de colônia de formigas cada solução candidata é representada por uma formiga, gerada aleatoriamente. Esse tipo de algoritmo é aplicado com maior eficiência em problemas de otimização discreta. Cada formiga escolhe uma rota e deposita uma certa quantidade de feromônio em cada rota percorrida. As melhores rotas seriam identificadas pela quantidade de feromônio, que é atualizada a cada iteração do algoritmo [29]. Em 3.5 é apresentado o pseudo-código de acordo com [30].

Algoritmo sistema de colônia de formigas

Início

Para cada colônia faça

Para cada formiga faça

Gerar uma rota

Avaliar a rota

Fim para

Evaporar o feromônio em todas as rotas

Depositar o feromônio na melhor rota

Fim para

Fim

Figura 3.5: Pseudo-código para algoritmo sistema de colônia de formigas

Algoritmos inspirados em enxames de partículas

É uma técnica de otimização estocástica baseada em populações que implementa o comportamento social entre indivíduos de um grupo, onde as partículas são os indivíduos e o enxame é o grupo. A técnica foi inspirada num enxame de partículas baseado no movimento de um bando de aves durante a busca por alimentos em uma região. Durante essa observação foi constatado que o comportamento do grupo é influenciado pela experiência individual de cada componente do enxame [31].

Nesse algoritmo, também chamado de PSO, o enxame é inicializado randomicamente, com uma população de partículas(soluções candidatas) onde, cada partícula é inicializada com uma posição e uma velocidade randômica. O PSO se baseia na informação da tra-

jetória dos indivíduos e nos pontos do espaço de busca, além disso, usa-se também uma memória para preservar os melhores locais visitados. O movimento das partículas, em cada iteração, depende da informação da melhor posição do enxame e da partícula. Essa técnica de enxame de partículas tem sido utilizada em problemas contínuos de funções não-lineares, com poucas aplicações em problemas discretos. A seguir em 3.6 será apresentado o pseudo-código segundo [31]:

Algoritmo enxame de partículas

Início

Iniciar cada indivíduo aleatoriamente

Até atingir o critério de parada

Para cada indivíduo faça

Avaliar a função objetivo do indivíduo

Atualizar a melhor posição do indivíduo

Fim para

Atualizar a melhor posição da população

Atualizar a posição do indivíduo no tempo $k+1$ baseado na posição atual e velocidade

Fim para

Fim

Figura 3.6: Pseudo-código para algoritmo enxame de partículas

Algoritmos inspirados em colônias artificiais de abelhas

Busca a inspiração para construção do algoritmo no comportamento de abelhas produtoras de mel. Observou-se que as abelhas saem em busca de alimentos e, em seguida, retornam a colméia. As colméias recrutam grupos de abelhas para explorar novas áreas em busca de pólen e néctar. Cada abelha visita um local trazendo o caminho, a direção, a distância e uma amostra de mel do local visitado. Ao retornarem compartilham a informação adquirida e, baseadas nestas informações, novas abelhas são designadas a fim de explorar os melhores locais.

As regiões mais promissoras são mais exploradas e as piores acabam sendo descartadas. Este ciclo se repete, com a visitação de novas regiões a cada iteração. É um algoritmo de busca populacional que combina busca aleatória global (graças à atuação das “abelhas”) com busca local (com a maior exploração nas regiões promissoras do espaço de busca). Sua principal aplicação tem sido em problemas de otimização e busca combinatória [32].

Esse algoritmo de busca é inspirado na procura de comida pelas abelhas, requer um

certo número de parâmetros. Esses parâmetros são: número de abelhas batedoras, o número de locais selecionados dentro dos visitados, número de melhores locais dos selecionados, número de abelhas recrutadas, para o melhor local, número de abelhas recrutadas para outros locais selecionados, o tamanho inicial dos campos que inclui: local, sua vizinhança e finalmente, critério de parada. O algoritmo inicia com o número de abelhas batedoras sendo colocadas aleatoriamente no espaço de busca. Depois, as aptidões dos locais visitados são avaliadas pelas abelhas. O local visitado com as informações obtidas seria uma solução do espaço de busca do problema de otimização. Na figura 3.7 é apresentado o pseudo-código desse algoritmo:

Algoritmo colônias artificiais de abelhas

Início

Inicializa a população com soluções randômicas

Avalie a aptidão da população

Enquanto não atinge o critério de parada

Forme uma nova população

Selecione áreas vizinhas para a busca

Recrute abelhas para seleção áreas e avalie aptidão

Selecione a melhor abelha para cada caminho

Selecione novas abelhas randomicamente e avalie sua aptidão

Fim Enquanto

Fim

Figura 3.7: Pseudo-código para algoritmo colônias artificiais de abelhas

3.4 Tratamento das restrições na otimização evolucionista

Os métodos de tratamento das restrições para otimização usando AEs apresentam algumas características [33]. Geralmente são métodos genéricos, que exploram uma estrutura matemática (seja linear ou não-linear) da restrição. Entre os métodos genéricos podemos citar as funções de penalização, o método do multiplicador de Lagrange, e métodos de busca, que são adaptados facilmente sem mudanças significativas no algoritmo. Existem também, métodos específicos que são aplicáveis a algum tipo especial de restrição e podem ser acoplados a diversos tipos de algoritmos entre eles: algoritmos genéticos, enxame de

partículas, colônia de formigas, etc. Os algoritmos genéticos, que é foco desse trabalho, tem sido usados juntamente com as funções penalização para o tratamento de restrição tanto de igualdade quanto desigualdade. A seguir será feita uma abordagem de alguns desses métodos para tratamento de restrição.

3.4.1 Técnicas de penalização

As funções de penalização, propostas em 1940 [24], são técnicas muito utilizadas para tratamento de restrições nos AEs. É uma técnica indireta, pois transforma um problema de otimização com restrição em um problema sem restrição. Conforme explicitado na seção 2.1, essas funções de penalização podem ser exteriores ou interiores. A abordagem mais comum utilizada nos AEs são as técnicas de penalizações exteriores, principalmente porque este modelo não requer uma solução factível logo de início. Uma função de penalização pode ser escrita como [17]:

$$F(\mathbf{x}) = f(\mathbf{x}) + \left(r \sum_{i=1}^m \cdot G_i + r' \sum_{j=1}^l \cdot H_j \right) \quad (3.1)$$

onde $F(\mathbf{x})$ é a nova função objetivo a ser otimizada, G_i e H_j são as funções geradas com as restrições do problema r e r' são os fatores de penalização (constantes positivas). Geralmente as funções restrições são apresentadas da seguinte forma:

$$G_i = \max[0, g_i(\mathbf{x})]$$

$$H_j = |h_j(\mathbf{x})|^2$$

Encontra-se diversas abordagens sobre os valores dos fatores de penalização [24]. Os AEs são muito flexíveis, não exigindo que as funções, nem as restrições sejam contínuas e diferenciáveis. Mas, geralmente, as técnicas de manipulação tendem a tratar apenas com restrição de desigualdade. Quando surgem problemas com restrição de igualdade, estas são transformadas em restrições de desigualdade da seguinte forma:

$$|h_j(\mathbf{x})| - \epsilon \leq 0 \quad (3.2)$$

sendo $\epsilon > 0$ uma tolerância permitida. Existem diversos tipos de funções de penalização utilizadas nos AEs. A seguir, serão abordados alguns tipos presentes na literatura [24].

Penalização de morte

Esse tipo de penalização descarta as soluções inactíveis, gerando outras soluções aleatoriamente ou definindo a aptidão como nula. Essa penalização é recomendada somente para espaços de busca convexos e com regiões factíveis com tamanho relativamente grande em relação ao espaço de busca. Esse método não é vantajoso em problemas complexos onde a população inicial apresenta somente soluções inactíveis. Nesse caso, o processo ficará parado porque todos os indivíduos terão aptidão nula. A penalização de morte trabalha com problemas que tem restrição de desigualdade, pois tem dificuldades de gerar soluções para satisfazer restrições de igualdade.

Penalizações estáticas

Existem diferentes abordagens para esta penalização. Nesse tipo os fatores de penalização não mudam durante o processo evolucionista. A função aptidão pode ser calculada, por exemplo, utilizando a fórmula abaixo, apresentada em [34]:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se a solução é factível} \\ k - \sum_{i=1}^m \frac{k}{m+l} & , \text{ caso contrário} \end{cases} \quad (3.3)$$

onde $F(\mathbf{x})$ é a nova função do problema sem restrição, $f(\mathbf{x})$ é uma função objetivo sem penalização, s é o número de restrições satisfeitas, $m+l$ é o número total de restrições de igualdade e desigualdade e k é uma constante. Apesar de eficiente, essa fórmula requer o uso de um algoritmo para gerar soluções factíveis na população inicial.

Na referência [35] é proposto o uso de níveis de penalização onde os fatores são obtidos dependendo do tamanho da violação para cada restrição do problema. A fórmula proposta para a função aptidão $F(\mathbf{x})$:

$$F(\mathbf{x}) = \sum_{i=1}^{m+l} (R_{k,i}) \cdot (\max[0, g_i(\mathbf{x})]^2) \quad (3.4)$$

onde $R_{k,i}$ são os coeficientes de penalização utilizados, $m+l$ é o número total de restrições, e $k = 1, 2, 3, \dots, L$, onde L é o número de níveis de violação. O objetivo dessa proposta é que

defina-se fatores de penalização específicos para cada restrição em cada nível de violação. Uma desvantagem desse método [35] é que ele requer um grande número de parâmetros a serem definidos gerando uma complexidade na utilização do método.

Outra proposta apresentada em [36] utiliza a seguinte fórmula para a função aptidão $F(\mathbf{x})$:

$$F(\mathbf{x}) = f(\mathbf{x}) \pm \sqrt{\sum_{i=0}^m H[g_i(\mathbf{x})] \cdot \max[0, g_i(\mathbf{x})]^2} \quad (3.5)$$

onde $H : \Re \rightarrow [0, 1]$ é uma função:

$$H(j) = \begin{cases} 1, & \text{se } j < 0 \\ 0, & \text{se } j \geq 0 \end{cases}$$

sendo H a função de Heaviside. Essa proposta apresenta diversas aplicações em problemas reais[37].

Penalizações dinâmicas

Penalizações dinâmicas são estratégias que utilizam a variação dos parâmetros no decorrer do processo de evolução onde, nas primeiras gerações o fator penalização geralmente é baixo. Durante o procedimento os parâmetros de penalização devem crescer, de acordo com alguma função/estratégia, até que seja atingido o número máximo de gerações. Nesse tipo de penalização, se a escolha do parâmetro não for adequada o algoritmo evolucionista pode convergir para outra solução factível não ótima (pois o fator penalização é muito alto) ou para soluções infactíveis se o fator de penalização é muito baixo. Foi proposta a seguinte fórmula para a função aptidão $F(\mathbf{x})$ [38]:

$$F(\mathbf{x}) = f(\mathbf{x}) + (0.5 \cdot t)^2 \sum_{j=1}^l \max[0, g(\mathbf{x})]^2 \quad (3.6)$$

O parâmetro de penalização $(0.5 \cdot t)^2$ cresce com o número t de gerações atingindo seu valor máximo no final do processo de evolutivo.

Outra proposta adotada utiliza um fator de penalização de acordo com o número

geracional chamado de técnica de variação da função aptidão [39]:

$$F(\mathbf{x}) = f(\mathbf{x}) + V(t) \cdot \left(A \sum_{i=1}^m (\delta_i \cdot \omega_i \cdot \Phi(d_i(S))) + B \right) \cdot \delta_s \quad (3.7)$$

onde: A é um fator de “gravidade”, m é o número total de restrições de desigualdade, ω_i é o fator peso para a restrição i , $d_i(S)$ é a medida do grau de violação da restrição quando i é introduzido na solução S , $\Phi_i(\cdot)$ é a função dessa medida, B é um fator de penalização, δ_s é um fator binário, $V(t)$ é uma função crescente de t no intervalo $[0, 1]$ onde $V(t)$ representa a parte dinâmica sendo definido por:

$$V(t) = \left(\frac{t}{T} \right)^2$$

onde t é um número geracional e T é o número total de gerações. O valor de $V(t)$ cresce de acordo com o número da geração.

A principal desvantagem desse método é que ele requer diversos parâmetros de acordo com cada problema e cuja definição não é trivial. Essa técnica foi introduzida num GA tradicional [40] com a finalidade de realizar uma busca global e um micro GA para realizar a busca local. Outros projetos de engenharia testaram essa técnica que obteve resultados competitivos porém, a definição de parâmetros para os AGs continuou sendo um problema.

Penalizações de recozimento (Annealing)

É um tipo de penalização dinâmica que utiliza a idéia do recozimento simulado. Michalewicz e Attia [41] modificaram o fator de penalização nas gerações de forma que, o fator de penalização é aumentado ao longo do tempo fazendo com que os indivíduos in-factíveis sejam penalizados nas gerações finais. A função penalização $F(\mathbf{x})$ foi definida como:

$$F(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2\tau} \sum_{i \in A} \phi_i^2(\mathbf{x}) \quad (3.8)$$

onde τ representa o tempo do arrefecimento, e

$$\phi_i(\mathbf{x}) = \begin{cases} \max[0, g_i(\mathbf{x})], & \text{se } 1 \leq i \leq m \\ |h_i(\mathbf{x})| & \text{se } m + 1 \leq i \leq (m + l) \end{cases}$$

onde $m + l$ é o número total de restrições. O interessante é a população não ter diversidade mas consistir de cópias de indivíduos para satisfazer todas as restrições. Durante a programação percebeu-se a sensibilidade do parâmetro de arrefecimento, que é a principal desvantagem desse método.

Penalizações coevolucionárias

Nos algoritmos genéticos com penalizações coevolucionárias tem-se pelo menos duas populações a evoluir ao longo das gerações, uma em função da outra. Dessa forma, admitem-se dois tipos principais de coevolução: a cooperativa que proporciona ajuda entre as populações para melhorar sua aptidão e a competitiva quando melhora a adaptação dos indivíduos da primeira população piora a aptidão da segunda população e vice-versa. Devido a isso, ao longo das gerações, tem-se uma disputa entre as duas populações [42].

O funcionamento desse tipo de AG envolve subpopulações onde penalizações são aplicadas ou seja, inicia-se a primeira e a segunda população e em seguida, avalia-se a primeira em função da segunda população e depois avalia-se a segunda população em função da primeira. Esse procedimento continua até que seja satisfeito um critério de parada [42].

Nesse tipo de algoritmo a penalização é feita com a definição dos parâmetros de cada AG. É levado em conta o quanto as restrições são violadas e a função aptidão $F(\mathbf{x})$ é definida como [43]:

$$F(\mathbf{x}) = f(\mathbf{x}) + \left(\sum_{i=1}^m \max[0, g_i(x)] \cdot k_1 + v \cdot k_2 \right) \quad (3.9)$$

onde $f(\mathbf{x})$ é o valor da função objetivo; k_1 e k_2 são os dois fatores de penalização inteiros, $\sum_{i=1}^m \max[0, g_i(\mathbf{x})]$ corresponde a soma de todas as restrições violadas, v é um fator inteiro que se inicia com zero sendo incrementado para cada restrição violada no problema.

A abordagem utiliza duas populações diferentes com a segunda população evoluindo os dois pesos (k_1 e k_2) para calcular o valor aptidão dos indivíduos da primeira população. Assim, a primeira população é utilizada para encontrar as soluções (como num AE) e a outra para evoluir fatores de penalização k_1 e k_2 . Cada indivíduo é decodificado e os fatores de penalização produzidos são utilizados na primeira população durante um certo número de gerações. A desvantagem desse método é o alto custo computacional e a necessidade de definição de diversos parâmetros nos problemas definidos.

Algoritmo genético segregado

O AG segregado propõe uma população de soluções onde cada solução seria avaliada usando dois fatores de penalização diferentes. Nesse algoritmo é realizado dois ranqueamentos: um baseado em penalização moderada e outro em altas penalizações da aptidão. Depois, é feita uma seleção das melhores soluções encontradas nesse ranqueamento para a reprodução de forma que, os melhores resultados sejam mantidos na próxima geração. Essa abordagem foi utilizada em [44] e obteve excelentes resultados, porém, a desvantagem desse método foi a definição dos dois fatores de penalização no problema.

Penalização Fuzzy

Essa abordagem é feita utilizando um espaço com as regras fuzzy para atualizar o fator de penalização. Sendo a função aptidão $F(\mathbf{x})$ dada por [45]:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se a solução é factível} \\ f(\mathbf{x}) + r_f \cdot G(\mathbf{x}) & , \text{ caso contrário} \end{cases}$$

onde $G(\mathbf{x}) = \sum_{j=1}^l \max[0, g_j(\mathbf{x})] + \sum_{i=1}^m |h_i(\mathbf{x})|$ e r_f é o fator de atualizado baseado em regras fuzzy. Vários experimentos foram realizados utilizando esse tipo de penalização [45], apresentando resultados competitivos.

3.4.2 Modelos para otimização com restrição

Apresenta-se, agora, recentes estudos e avanços de técnicas de otimização para o tratamento de restrições com AEs e os algoritmos bio-inspirados descritos acima [46] onde uma breve abordagem será feita sobre os principais modelos. Alguns trabalhos são baseados em otimização de enxame de partículas. Entre eles destaca-se o trabalho que trata da otimização com restrição utilizando variáveis contínuas chamado de algoritmo COPSO, onde os autores baseiam-se num algoritmo PSO. Nesse novo algoritmo COPSO realiza-se uma hibridização de um PSO [47] com a utilização de alguns operadores para manter a diversidade da população. Esse trabalho compara o desempenho com outros resultados presentes na literatura. O COPSO apresenta um resultado superior nas soluções factíveis devido a tolerância dinâmica para tratar as restrições de igualdade. Outro componente desse algoritmo é a perturbação de dois operadores para manter a diversidade da po-

pulação e melhorar a busca de forma que, são realizadas modificações somente na melhor partícula da população. O algoritmo híbrido COPSO mostra eficiência quando aplicado no G-Suíte de funções [48] com um número fixo de funções de avaliação.

Ainda na referência [46] cita-se trabalhos para técnicas de manipulação de restrição para otimização evolucionista multiobjetivo. Destaca-se a transformação da restrição em função de penalização tratando as restrições via otimização multiobjetivo. Um dos trabalhos, utiliza uma técnica para tratamento de múltiplos objetivos com base em uma função de penalização adaptativa, juntamente com uma medida de distância. Essas duas funções variam de acordo com o valor da função objetivo e a soma de violações de restrição de um indivíduo. Assim, a estratégia de busca é modificada para atender as violações e o desempenho das restrições em cada indivíduo. Essa modificação é feita para facilitar a busca de soluções ótimas, não só no espaço factível, mas, também no infactível. O número de soluções factíveis na população determina o comportamento do processo, que leva a busca ou encontro de mais soluções factíveis ou, ainda, localização da solução ótima. Esse método não necessita de nenhum ajuste de parâmetros e foi testado com outros trabalhos de otimização multiobjetiva mostrando resultados competitivos.

Existem, também, aplicações dos SIAs para problemas de otimização com restrição, destacando-se: Hajela e Lee [49] que adaptam soluções infactíveis nos indivíduos factíveis usando um processo de correspondência por bits. O desempenho dessa abordagem depende de uma seleção de anticorpos(indivíduos infactíveis), que são expostos aos antígenos durante a simulação. Outra abordagem é feita em [50] com a utilização de um modelo de seleção via sistema imunológico artificial para problemas de otimização com restrição propondo, ainda, uma versão paralela para este algoritmo [51]. Os resultados obtidos foram melhores do que a versão original do algoritmo. Porém, traz como desvantagem a necessidade de definição de alguns parâmetros extras.

Outro trabalho interessante citado em [46] é a combinação da utilização de um sistema imunológico artificial(SIA) com um algoritmo genético padrão(AG). O trabalho “Um AG-SIA híbrido para problemas de otimização com restrição na engenharia” trata da hibridização de um AG com um SIA como uma alternativa para problemas de otimização com restrições na engenharia. A técnica de sistemas imunológicos artificiais é colocado dentro de um AG para movimentação da população para a região factível. Além disso, é incluída uma variação do algoritmo como meio de compensação para melhorar a diver-

sidade da população. Foram utilizadas duas técnicas de penalização para o tratamento das restrições um método de penalização adaptativa e o ranqueamento estocástico [52]. O resultado da hibridização é testado em vários problemas com espaço contínuo, inteiro e misto inteiro-contínuo. Utilizou-se nos experimentos o G-suite, problemas da engenharia mecânica e da engenharia estrutural.

Outra técnica para o tratamento de restrição foi abordada no trabalho: “Otimização com restrição baseada em algoritmos genéticos com aproximação quadrática” [46]. A metodologia proposta trabalha junto com o algoritmo genético para otimização com restrição de desigualdade. Esse método pode ser visto como um operador de busca local que utiliza aproximações quadráticas tanto para a função objetivo como para as restrições. O operador local é baseado na aproximação de dois tipos de funções restrição: as restrições locais convexas são aproximadas por funções quadráticas e as restrições locais não-convexas são aproximadas por restrições lineares. Esse operador local tem como objetivo melhorar a satisfação das restrições.

Uma técnica de manipulação de restrição para problema aerodinâmico e otimização de projeto multidisciplinares é apresentada em [46]. Essa técnica é própria para problemas em que o número de avaliações deve ser baixo. É apresentada uma técnica para tratamento de restrição geométrica, utilizada para otimizações desses tipos de projetos. A combinação do domínio em espaços de restrição e um mecanismo ajuda na abordagem para chegar a região factível. Experimentos indicam que a técnica apresenta resultados competitivos.

3.4.3 Considerações

Como visto até agora, diversos são os algoritmos bio-inspirados e as formas de adaptação de tais algoritmos para o trato de problemas com restrições. Deve-se ressaltar, que a maioria das estratégias apresentadas para viabilizar a manipulação de restrições, podem ser acopladas a qualquer um dos algoritmos bio-inspirados baseados em população. Considera-se que todos os modelos tem pontos fortes e fracos, conforme apresentado acima. Todos, porém, necessitam de parâmetros a serem definidos pelo usuário, parâmetros tais que podem influenciar bastante a qualidade dos resultados.

Um modelo sem parâmetros a serem definidos é possível quando utiliza-se uma estratégia adaptativa. Algumas estratégias adaptativas existentes serão apresentas no próximo capítulo.

Devido a eficiência dos algoritmos genéticos na maximização e minimização de funções reais, o mesmo será o representante dos AEs, bem como dos outros algoritmos bio-inspirados, para a implementação dos modelos desenvolvidos para o trato das restrições. Além disso, tem-se experiência anterior em se utilizar AGs o que facilita a interpretação dos resultados obtidos.

Os desenvolvimentos em relação a forma de trato das restrições via um AG terão duas vertentes:

A primeira, se concentrará no desenvolvimento de uma estratégia de penalização adaptativa. Os modelos adaptativos, que serão descritos no próximo capítulo, tem sido considerados mais eficientes para um melhor ajuste das penalidades pois, utilizam a própria população para gerar o valor adequado das penalidades. A adaptação de uma estratégia para AGs não-geracionais, que apresentou bons resultados para um AG geracional, será a meta a ser alcançada.

A segunda, se baseará em uma técnica da otimização não-linear, conhecida por lagrangeano aumentado, visando sua adaptação para AEs não-geracionais. Espera-se que obtenha bons resultados com a combinação, pelas propriedades específicas tanto do algoritmo não-geracional, quanto da técnica da programação não-linear considerada.

4 UMA ESTRATÉGIA DE PENALIZAÇÃO ADAPTATIVA PARA ALGORITMOS GENÉTICOS NÃO GERACIONAIS

4.1 Modelos adaptativos para otimização com restrição usando AE's

Os métodos de penalização tem sido amplamente utilizados na literatura devido a sua simplicidade e fácil implementação. A função penalização geralmente é aplicável em algum tipo de restrição (linear ou não linear) porém, muitas vezes é difícil encontrar um parâmetro apropriado de penalização para determinar o ótimo restrito.

A introdução de termos de penalização transforma um problema de otimização com restrição em um problema sem restrição de forma que, a penalização é geralmente, determinada por uma sequência de coeficientes de penalização em cada geração. O uso de funções de penalização envolvem a definição de um número de parâmetros de penalização para determinar as soluções factíveis. Devido a essa dependência de parâmetros de penalização busca-se diversas abordagens de funções penalização tais como: penalização estática, penalização dinâmica e penalização adaptativa.

As penalizações estáticas e dinâmicas requerem que os usuários definam os parâmetros heurísticamente enquanto, na penalização adaptativa os valores dos coeficientes de penalização são ajustados dinamicamente ou automaticamente por um algoritmo evolucionista. Nessa penalização pode acontecer por exemplo, que a informação da população seja usada para ajustar os valores dos coeficientes.

Existem diversos estudos em que utilizou-se o parâmetro de penalização adaptativo. Um deles é o trabalho apresentado na referência [53] que desenvolve um método em que os parâmetros de penalização mudam de acordo com as informações do processo de busca

e a função aptidão $F(\mathbf{x})$ é dada por [53]:

$$F(\mathbf{x}) = f(\mathbf{x}) + \gamma(t) \sum_{j=1}^l [v_j(\mathbf{x})]^\beta \quad (4.1)$$

onde o parâmetro de penalização $\gamma(t)$ é atualizado a cada geração i da seguinte forma:

$$\gamma(t+1) = \begin{cases} \frac{1}{\beta_1} \gamma(t), & \text{se } b^i \in F \text{ para todo } t-k+1 \leq i \leq t \\ \beta_2 \gamma(t), & \text{se } b^i \notin F \text{ para todo } t-k+1 \leq i \leq t \\ \gamma(t), & \text{em caso contrário} \end{cases}$$

sendo b^i o melhor indivíduo da geração i , F é a região factível, $\beta_1 \neq \beta_2, \beta_1 > \beta_2$ e $\beta_1, \beta_2 > 1$. Onde $\gamma(t+1)$ representa o parâmetro da geração seguinte e esse parâmetro diminui se todos os indivíduos da melhor geração são factíveis ou aumenta se todos são infactíveis. Caso ocorra a presença de indivíduos factíveis e infactíveis na melhor geração o parâmetro $\gamma(t+1)$ mantém-se inalterado. A principal desvantagem desse método é a definição dos parâmetros β_1, β_2 . Esse tipo de penalização pode ser interessante pois, pode ajustar de forma inteligente o fator de penalização evitando que a população seja totalmente factível ou infactível.

Outra abordagem para o tratamento de restrição baseado em função auto-adaptativa foi proposto em [54]. Essa abordagem evita o uso da definição de um novo parâmetro no algoritmo, implementou-se em 3 três etapas :

- 1) normalização da soma das restrições violadas para cada solução;
- 2) são identificadas as melhores e piores soluções na população;
- 3) a função penalização é aplicada em duas partes: na primeira parte é aplicado somente se uma ou mais soluções factíveis tem uma função objetivo com valor melhor que a solução mais próxima encontrada. O objetivo é aumentar a aptidão das soluções infactíveis. Na segunda parte, aumenta-se a aptidão das soluções infactíveis para favorecer as soluções que são quase factíveis e também para obter um bom valor para a função objetivo.

Essa abordagem define o fator penalização em termos das melhores e piores soluções,

além disso, os autores usaram um algoritmo genético com representação binária e o processo de seleção roleta.

O método proposto em [55] manipula regiões inactiváveis próximas a região factível através de parametrização e a função aptidão $F(\mathbf{x})$ é proposta como:

$$F(\mathbf{x}) = f(\mathbf{x}) + (F_{feas}(t) - F_{todos}(t)) \sum_{i=1}^m \frac{f_i^k(\mathbf{x})}{q(t)} \quad (4.2)$$

onde F_{todos} é o valor da aptidão do melhor indivíduo obtido até a geração t ; F_{feas} é o valor da aptidão do melhor dos indivíduos inactiváveis da geração t ; k é uma constante que ajusta a penalização e segue-se nesse trabalho $k = 2$; $q(t)$ é uma função que define uma região em que as soluções inactiváveis são pouco penalizadas. Em seguida, foi proposta por Coit uma variação de [55] usando uma forma multiplicativa da função aptidão da seguinte forma[56]:

$$F(\mathbf{x}) = f(\mathbf{x}) \cdot P(\mathbf{x})$$

sendo $P(\mathbf{x})$ definido:

$$P(\mathbf{x}) = 1 - \frac{1}{m} \sum_{i=1}^m \left(\frac{\Delta v_i(\mathbf{x})}{v_i} \right)^k \quad (4.3)$$

onde: $\Delta v_i(\mathbf{x}) = \max[0, g_i(\mathbf{x}) - v_i]$ e $v_i(\mathbf{x})$ refere-se a violação da restrição i .

A abordagem de [56] foi refinada mais tarde, introduzindo um novo parâmetro para soluções inactiváveis [57] onde:

$$P(\mathbf{x}) = 1 - \frac{1}{m} \sum_{i=1}^m \left(\frac{\Delta v_i(\mathbf{x})}{\Delta v_i^{max}} \right)^k \quad (4.4)$$

sendo: $\Delta v_i(\mathbf{x}) = \max[0, g_i(\mathbf{x}) - v_i]$ e $\Delta v_i^{max} = \max[\epsilon, \Delta v_i(\mathbf{x}), \mathbf{x} \in P(t)]$

De forma que $\Delta v_i(\mathbf{x})$ representa o valor da violação da restrição i no n -ésimo cromossomo; Δv_i^{max} é a violação máxima da restrição i e ϵ é um pequeno número positivo.

Uma algoritmo adaptativo segregacionista para o tratamento de restrição evolucionária (ASCHEA) proposto em [58] utiliza estratégia de evolução baseada em: função pena-

lização adaptativa, recombinação através das restrições e o operador de seleção segregacionista. O parâmetro adaptativo é adotado da seguinte forma:

$$F(x) = \begin{cases} f(\mathbf{x}) & \text{se a solução é factível} \\ f(\mathbf{x}) - P(\mathbf{x}) & \text{caso contrário} \end{cases} \quad (4.5)$$

onde:

$$P(x) = \alpha \sum_{j=1}^q \max[0, g_j(\mathbf{x})] + \alpha \sum_{j=q+1}^m |h_j(\mathbf{x})| \quad (4.6)$$

sendo, $\max[0, g_j(\mathbf{x})]$ a parte positiva de $g_j(\mathbf{x})$ e α é o fator de penalização adotado para todas as restrições. O fator de penalização é adaptado baseado numa taxa desejada para soluções factíveis τ_{alvo} e na taxa da geração corrente t :

$$\alpha = \begin{cases} \text{se} & (\tau_t > \tau_{alvo}) \quad \alpha(t+1) = \alpha(t)/fact \\ \text{caso contrário} & \alpha(t+1) = \alpha(t) * fact \end{cases} \quad (4.7)$$

onde $fact > 1$ e τ_{alvo} que são parâmetros definidos pelo usuário. τ_t denota a proporção de indivíduos inactíveis na população na geração t . O coeficiente inicial de penalização $\alpha(0)$ é computado usando a primeira população de forma que equilibre a função objetivo e a violação das restrições:

Se o somatório da violação das restrições for nulo, $\alpha(0) = 1$, caso contrário:

$$\alpha(0) = \left| \frac{\sum_{i=1}^n f_i(\mathbf{x})}{\sum_{i=1}^n V_i(\mathbf{x})} \right| * 1000$$

onde $V_i(\mathbf{x})$ é a soma da violação da restrição do indivíduo i . Numa outra versão em 2002 [59] do ASCHEA utiliza um fator de penalização para cada restrição de forma a permitir mais precisão nos fatores de penalização. Dessa forma esse tipo penalização é estendido para o uso de múltiplos coeficientes nas penalizações. Nessa versão foi introduzida uma nova técnica com taxa de adaptação para tratamento de funções multimodais. Esse trabalho mostrou grande precisão e eficiência nos testes realizados.

Uma técnica muito competitiva apresentada em [60] é o ranqueamento estocástico (*Stochastic Ranking*). Consiste em uma estratégia que tenta equilibrar a influência da função

objetivo e a função penalização estocasticamente atribuindo a aptidão para um indivíduo. É introduzida uma probabilidade p_f onde são utilizadas somente funções objetivo para comparação no ranqueamento de regiões inactíveis do espaço de busca. Esse método de ranqueamento foi testado usando uma estratégia de evolução nas treze primeiras funções do G-Suíte e obteve bons resultados. O algoritmo do ranqueamento estocástico é baseado no valor do parâmetro p_f . É realizada uma comparação entre dois indivíduos adjacentes através da taxa de probabilidade de acordo com a função objetivo. Onde a função objetivo será 1 se ambos os indivíduos são factíveis e será p_f se forem inactíveis. No algoritmo do ranqueamento estocástico $U = (0, 1)$ é um número randômico gerador uniforme, N é o número de varreduras por toda a população e p é o número de indivíduos na população. Inicialmente o ranqueamento é gerado randomicamente. Em [60] é apresentado esse algoritmo conforme figura 4.1:

Algoritmo ranqueamento estocástico

Início

$I_j = j \forall j \in \{1, \dots, p\}$

Para $i = 1$ até N

Para $j = 1$ até $p - 1$

$u = U(0, 1)$

Se $(F(I_j)) = F(I_{i+j}) = 0$ ou $(u < p_f)$

Troca (I_j, I_{i+j})

Caso contrário

Se $F(I_j) > F(I_{i+j})$

Troca (I_j, I_{i+j})

Fim para

Se a troca não é realizada Pare

Fim para

Fim

Figura 4.1: Pseudo-código para algoritmo do ranqueamento estocástico

Outra técnica de penalização adaptativa chamada GA-RRWS [33], baseada em ranqueamento, de forma que os parâmetros adaptativos atribui valores entre as soluções factíveis e não factíveis e assim fornecendo uma direção para as regiões factíveis. É utilizado um ranquemamento baseado em roleta bem como operador de seleção(RRWS) que permite continuamente que o GA encontre melhores soluções factíveis gradualmente, principalmente a busca próxima da verdadeira solução ótima. Esse método do GA-RRWS propõe um

coeficiente de penalização:

$$P_i(\mathbf{x}) = \frac{f_{i+1} - f_i}{F(\mathbf{x})_{i+1} - F_i} \quad (4.8)$$

para $F_i \neq F_{i+1}$. Sendo $P_i(\mathbf{x})$ o coeficiente de penalização; f_i a função a ser otimizada; $F_i(\mathbf{x})$ é a função f_i na forma transformada. É definido também um coeficiente de penalização que é determinado durante o ranqueamento de indivíduos ($P_g(\mathbf{x})$). Três casos podem acontecer quando $P_g(\mathbf{x}) \geq 0$:

1) Se $f_i \leq f_{i+1}$ e $F_i(\mathbf{x}) \geq F_i(\mathbf{x}+\mathbf{1})$. Diz-se que: $0 \leq P_g(\mathbf{x}) \leq P_i(\mathbf{x})$. Ou seja, $P_g(\mathbf{x})$ tem valor maior que $P_i(\mathbf{x})$

2) Se $f_i \geq f_{i+1}$ e $F_i(\mathbf{x}) < F_i(\mathbf{x}+\mathbf{1})$, de forma que: $0 \leq P_i(\mathbf{x}) \leq P_g(\mathbf{x})$

3) Se $f_i < f_{i+1}$ e $F_i(\mathbf{x}) < F_i(\mathbf{x}+\mathbf{1})$, sendo $P_i(\mathbf{x}) < 0$

O ranqueamento é motivado por análises dos métodos de penalização do ponto de vista de dominância. Esse algoritmo GA adaptativo introduz uma variação de operadores e não requer nenhum conhecimento anterior do problema para o uso do coeficiente de penalização na função.

Em [61] é proposta uma função penalização adaptativa que foi acoplada em um algoritmo genético geracional para problemas de otimização estrutural. Essa técnica proposta em [62] usa a informação da população em torno dos parâmetros da penalização. Onde a média da função objetivo da população corrente e o nível de violação de cada restrição durante o processo evolucionário é usado para definir os parâmetros de penalização. Esse tipo de penalização adaptativa juntamente aplicada a um algoritmo genético não-geracional foi alvo de estudo neste trabalho [1] e será explicitada com mais detalhes na próxima seção.

4.2 Um modelo adaptativo para AG não-geracional

Um esquema de penalização adaptativa denominada APM (*Adaptive Penalty Method*) usando um AG geracional foi proposto em [62] por Barbosa e Lemonge. A penalização é feita através de um procedimento evolucionário ao qual um parâmetro de penalização é definido automaticamente para cada restrição do problema em cada função. O usuário não

precisa se preocupar, *a priori*, com os diferentes valores de penalização. Esse procedimento mostrou-se eficaz e robusto quando foi aplicado em problemas da matemática e engenharia estrutural, apresentados na literatura [61]. A estratégia de penalização não precisa de nenhuma intervenção do usuário e o esquema adaptativo é utilizado com as informações da população tais como a média da função objetivo e o nível de violação de cada restrição durante a evolução. A função aptidão foi escrita como [62]:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{se } \mathbf{x} \text{ é factível} \\ u(\mathbf{x}) + \sum_{j=1}^m K_j v_j(\mathbf{x}), & \text{caso contrário} \end{cases} \quad (4.9)$$

onde,

$$u(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle, \\ \langle f(\mathbf{x}) \rangle, & \text{caso contrário} \end{cases} \quad (4.10)$$

Sendo $\langle f(\mathbf{x}) \rangle$ a média da função objetivo na população atual. O parâmetro de penalização k é definido em cada geração como:

$$K_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (4.11)$$

Tal que: $\langle v_l(\mathbf{x}) \rangle$ é a média da violação da l -ésima da restrição calculada sobre a população atual.

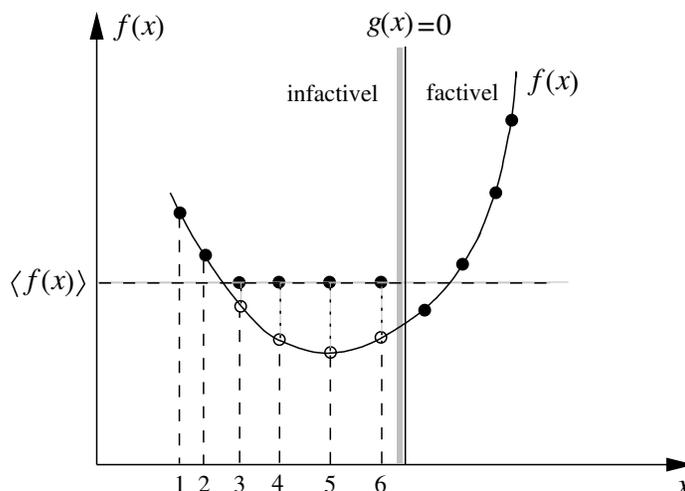
A figura 4.2 apresentada em [62] descreve uma simulação para o valor de $u(\mathbf{x})$ para uma determinada função $f(\mathbf{x})$. Se um indivíduo é infactível em um problema de minimização, por exemplo, como mostra a figura 4.2 a sua função objetivo será alterada para a média da função objetivo ($\langle f(\mathbf{x}) \rangle$), se a mesma for menor que este valor. Se a sua função objetivo for maior, a mesma não terá o seu valor alterado. De acordo com o explicitado anteriormente, a figura 4.2 mostra os pontos infactíveis (3, 4, 5 e 6) que terão os valores das suas funções objetivo alterados para o valor da função objetivo média.

Considerando pop o tamanho da população, o parâmetro de penalização k_j é escrito como:

$$K_j = \frac{|\sum_{i=1}^{pop} f(\mathbf{x}^i)|}{\sum_{l=1}^m [\sum_{i=1}^{pop} v_l(\mathbf{x}^i)]^2} \sum_{i=1}^{pop} v_j(\mathbf{x}^i) \quad (4.12)$$

onde os valores dos coeficientes de penalização são distribuídos de modo que as restrições mais difíceis de serem satisfeitas terão um coeficiente de penalização maior. As demons-

Figura 4.2: Esquema de penalização adaptativa



trações do esquema proposto estão disponíveis em [62].

A partir de experimentos realizados para verificar a eficiência e robustez do esquema de adaptação proposto os autores alertam que a precisão nos resultados finais da busca dependem dos componentes de algoritmo como: codificação, operadores e esquema de seleção.

Devido a eficiência do APM proposto no trabalho [62], os autores adaptaram esse parâmetro de penalização para a utilização em algoritmos genéticos não-geracionais [1], com codificação real, afim de buscar mais precisão nos resultados, principalmente dos problemas com variáveis contínuas. Assim, foi apresentado um novo esquema de penalização para algoritmos genéticos não-geracionais(steady-state). Para cada restrição, o parâmetro de penalização é adaptativamente computado de acordo com a informação extraída da população assim como a existência de indivíduos factíveis e o nível de violação de cada restrição. Deve-se ressaltar que quando encontra-se um individuo melhor que o melhor factível da população a contagem das inserções é zerada.

Num algoritmo genético geracional, simplesmente pode-se atualizar cada coeficiente, a cada geração. Como o funcionamento desse algoritmo não é preparado para o uso em AG não-geracional, foram introduzidas algumas modificações. A principal modificação

ocorreu na determinação do valor $h(x)$ da equação (4.10):

$$h(\mathbf{x}) = \begin{cases} f(\mathbf{x}_{pior}), & \text{se não existe elemento inactivo na população} \\ f(\mathbf{x}_{melhor\,factive}), & \text{caso contrário} \end{cases} \quad (4.13)$$

E o coeficiente de penalização é dado por:

$$k_j = u(\mathbf{x}) \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (4.14)$$

Outra modificação diz respeito ao parâmetro definido como **ninser** que controla as atualizações dos parâmetros de penalização. O valor de **ninser** indica quando os parâmetros de penalização deverão ser atualizados em relação ao número de inserções de indivíduos na população (**inser**). É normalizado pelo tamanho da população, sendo que seu valor indica quantos indivíduos devem ser inseridos em relação ao tamanho da população para que a penalização seja atualizada. Assim, seu valor é comumente um inteiro maior que a unidade. Quando o limite estabelecido pelo **ninser** é atingido, os parâmetros de penalização são atualizados.

O pseudo-código para o esquema adaptativo de penalização proposto para um algoritmos não-geracional foi apresentado na referência [1]. Esse algoritmo genético não-geracional apresentado na figura 4.3 não foi plenamente avaliado em [1], o que será feito neste trabalho. Inicialmente foram testadas algumas combinações de operadores de recombinação e mutação para codificação real visando avaliar o desempenho no algoritmo apresentado. Um estudo visando avaliar o valor ótimo para o parâmetro **ninser** também será feito. Além disso, serão acrescentadas mais algumas funções que foram propostas em [48], serão apresentados alguns experimentos com problemas de engenharia mecânica [63] e de otimização estrutural [61]. Finalmente, será realizada a comparação com alguns resultados propostos na literatura.

APM do Steady-State

Início

Inicialize a população

Calcule a função objetivo e os valores da restrição de violação

Se existe um elemento não factível então:

$h \leftarrow$ pior valor da função objetivo

Caso contrário

$h \leftarrow$ o valor da função objetivo é o melhor indivíduo factível

Fim se

Calcule os coeficientes de penalização

Calcule os valores da função aptidão

$inserir = 0$

Repita

Selecionar o operador

Selecionar os pais

Gerar os filhos

Avaliar os filhos

Manter o melhor filho

Se o filho é o novo melhor elemento factível então

atualize os coeficientes de penalização e os valores aptos

$inserir = 0$

Fim se

Se o filho é melhor que o pior na população então:

O pior é removido

O filho é inserido

$inserir = inserir + 1$

Fim se

Se ($inserir \text{ tamanho da população} \geq n_{inserir}$) então

Atualize os coeficientes de penalização e os valores aptidão

$inserir = 0$

Fim se

Até que o número máximo de avaliações ser atingido

Fim

Figura 4.3: Pseudo-código para o algoritmo não-geracional utilizando penalização adaptativa

5 UMA ESTRATÉGIA BASEADA EM LAGRANGEANO AUMENTADO E ALGORITMOS GENÉTICOS NÃO-GERACIONAIS

Nesse capítulo será apresentada uma proposta de um algoritmo genético não-geracional baseada em uma estratégia particular de penalização: o método do lagrangeano aumentado. O método do lagrangeano aumentado transforma um problema de otimização com restrição em uma sequência de problemas sem restrições. Conforme explicitado na seção 2.1, dado um problema de otimizar $f(\mathbf{x})$ sujeito a restrições de igualdades $h(\mathbf{x})$ e desigualdades $g(\mathbf{x})$, a função lagrangeana é definida como:

$$L(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \max[0, g_i(\mathbf{x})] + \sum_{j=1}^l \mu_j |h_j(\mathbf{x})| \quad (5.1)$$

sendo $\lambda \in \Re^m$ e $\mu \in \Re^l$ os multiplicadores de Lagrange associados as restrições. Esses multiplicadores se apresentam como variáveis a serem determinadas no processo de otimização que, de certa forma, penalizam as restrições. Para a construção do chamado lagrangeano aumentado adiciona-se a esta parcela, referente ao multiplicadores, um termo extra relativo a penalização. Logo, um lagrangeano aumentado pode ser escrito como:

$$F(\mathbf{x}, \lambda, \mu, r) = L(\mathbf{x}, \lambda, \mu) + P(\mathbf{x}, r) \quad (5.2)$$

onde P é uma função de penalidade e $r \in \Re^{m+l}$ são parâmetros de penalidade. A função $F(\mathbf{x}, \lambda, \mu, r)$ é denominada de função de lagrangeano aumentado.

De acordo com [64] a abordagem dessa técnica é melhor do que o uso de funções penalização porque: (i) o lagrangeano aumentado é baseado nas propriedades teóricas de convergência de Kuhn-Tucker, (ii) essa abordagem não distorce a função objetivo original, mas muda a função tornando o problema de minimização com restrição em

um problema sem restrição, (iii) o lagrangeano fornece, através dos multiplicadores de Lagrange, valores mais precisos da violação de cada restrição. Ainda, segundo [64] existe uma desvantagem desse método devido a necessidade de uma série de passos (iterações) no processo de otimização, tornando o algoritmo caro computacionalmente em relação a um modelo de penalização que considere os valores das penalizações constantes.

Na referência é apresentado o pseudo-código para esta estratégia [65]:

Algoritmo de lagrangeano aumentado

Início

Inicialize $k = 0$, λ^0 , μ^0 e os parâmetros de penalidade r^0

Enquanto critério de convergência não for satisfeito **faça**

Minimize $F(\mathbf{x}, \lambda^k, \mu^k, r^k)$ em relação a \mathbf{x} , obtendo \mathbf{x}^k

Atualize λ^{k+1} , μ^{k+1} e r^{k+1} , se necessário

Atualize $k = k + 1$

Fim Enquanto

Fim

Figura 5.1: Pseudo-código para algoritmo de lagrangeano aumentado

Detalhes das estratégias de atualização dos multiplicadores e penalizações, bem como a definição da função F para as restrições de desigualdade e igualdade serão vistos à frente. A referência [64] avalia alguns aspectos interessantes dessa técnica. O primeiro aspecto é referente as propriedades teóricas de convergência que indicam uma distorção mínima da função objetivo original, permitindo um melhor desempenho na busca do ótimo. Além disso, uma discussão também é feita em relação a atualização dos parâmetros durante o processo de otimização. Através de uma atualização feita de forma adaptativa a função penalizada muda dinamicamente o ótimo do ponto mínimo com restrição, para um ponto mínimo sem restrição durante o processo iterativo. Nesse caso, o parâmetro de penalização não é mantido constante durante a otimização. Nas próximas seções será feita uma abordagem mais detalhada sobre esse método.

5.1 O modelo dos multiplicadores de lagrange para otimização

O uso de funções de penalização foi formalizada em 1943 com os trabalhos de Fritz John(1948) e Kuhn-Tucker(1951) trazendo resultados necessários e suficientes para so-

lucionar um problema não-linear. A partir daí, surge a função lagrangeana, sujeita às condições de otimalidade. Determinam as condições suficientes e necessárias que devem ser obedecidas pelas restrições e multiplicadores na solução do problema de otimização. Logo, para otimizar uma função $f(\mathbf{x})$ sujeita as restrições $g_i(\mathbf{x}) \leq 0$ e $h_j(\mathbf{x}) = 0$, são as seguintes condições a serem respeitadas [9]:

Condições de Kuhn-Tucker: Assumindo que f, g_i e h_j são diferenciáveis as condições de Kuhn-Tucker consiste em encontrar a solução de um sistema de equações não-lineares. Ou seja, encontrar vetores \mathbf{x}, λ e μ que satisfazem:

$$\nabla f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{x}) + \sum_{j=1}^l \mu_j \nabla h_j(\mathbf{x}) = 0 \quad (5.3)$$

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \quad (5.4)$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, l \quad (5.5)$$

$$\lambda_i g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \quad (5.6)$$

$$\lambda_i \geq 0, \quad i = 1, 2, \dots, m \quad (5.7)$$

Teorema das Condições necessárias Kuhn-Tucker : Considere o problema 5.3 sendo f, g e h funções diferenciáveis e \mathbf{x}^* uma solução factível do problema. Considere $I = \{i \mid g_i(\mathbf{x}^*) = 0\}$ de forma que mais tarde, $\nabla g_i(\mathbf{x}^*)$ para $i \in I$ e $\nabla h_j(\mathbf{x}^*)$ são linearmente independentes. Se \mathbf{x}^* é uma solução ótima do problema, então existe um par (λ^*, μ^*) , tais que $(\mathbf{x}^*, \lambda^*, \mu^*)$ resolvem o problema explicitado nas equações (5.3) – (5.5).

5.2 Técnicas de aplicação do lagrangeano aumentado

Os métodos de lagrangeano aumentado utilizados na otimização, para minimização e maximização de problemas restritos, começaram a ser estudados em 1968 por Hestenes [66] e Powell [10] em problemas com restrição de igualdade [67]. Powell, em 1969, combinou o método de função penalização com o método primal-dual [67] no qual o quadrado das funções restrições são adicionados ao lagrangeano. Ou seja, uma série de funções penalização são minimizadas de forma que o algoritmo deverá convergir com a transferência da parcela relativa a função de penalização para a parcela dos multiplicadores. A função

penalização para resolução de problemas de otimização com restrição de igualdade de Powell, é escrita da seguinte forma:

$$P(\mathbf{x}, \theta, r) = \frac{1}{2} \sum_{j=1}^l r (h_j(\mathbf{x}) + \theta_j)^2 \quad (5.8)$$

onde $r > 0$ é a constante de penalização (podendo ser um vetor de penalizações) e θ_j parâmetros reais associados a restrição de igualdade. Assim, um problema de minimização para otimização com restrição é transformado no problema sem restrição:

$$F(\mathbf{x}, \theta, r) = f(\mathbf{x}) + P(\mathbf{x}, \theta, r) \quad (5.9)$$

Dessa forma, a função a ser otimizada fica:

$$F(\mathbf{x}, \theta, r) = f(\mathbf{x}) + \sum_{j=1}^l \mu_j h_j(\mathbf{x}) + \frac{1}{2} \sum_{j=1}^l r [(h_j(\mathbf{x}))^2 + \theta_j^2] \quad (5.10)$$

A equação 5.10 é chamada de função lagrangeano aumentado. Sendo θ_j o multiplicador de Lagrange da restrição $h_j(\mathbf{x}) = 0$ com $\mu_j = r \theta_j$. Powell(1969) apresenta uma estratégia para a atualização dos multiplicadores de Lagrange de acordo com as informações obtidas na iteração anterior, sendo que a penalização não necessita ter valores crescentes para garantir a convergência. Esta característica é o diferencial no método de lagrangeano aumentado pois, os métodos de transformação (penalidade interior, estendida, exterior) são dependentes de parâmetros de penalização com valores altos. Já o método de lagrangeano aumentado converge para o ponto ótimo sem a necessidade do parâmetro tender ao infinito.

Hestenes(1969) apresentou um método chamado “método dos multiplicadores”. Nesse método, a função penalização é definida como [66]:

$$P(\mathbf{x}, \mu, r) = \sum_{j=1}^l \mu_j h_j(\mathbf{x}) + \frac{r}{2} \sum_{j=1}^l (h_j(\mathbf{x}))^2 \quad (5.11)$$

sendo função lagrangeano aumentada, para o tratamento de restrições de igualdade, de-

finida de acordo com Hestenes [68]:

$$F(\mathbf{x}, \mu, r) = f(\mathbf{x}) + \sum_{j=1}^l \mu_j h_j(\mathbf{x}) + \frac{r}{2} \sum_{j=1}^l (h_j(\mathbf{x}))^2 \quad (5.12)$$

Em meados dos anos 70 Rockafellar [11] introduziu o lagrangeano aumentado para o tratamento de restrições de desigualdade na seguinte forma:

$$F(\mathbf{x}, \lambda, r) = f(\mathbf{x}) + \begin{cases} \sum_{i=1}^m \lambda_i g_i(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^m r (g_i(\mathbf{x}))^2, & \text{se } g_i(\mathbf{x}) \geq \frac{-\lambda_i}{r_i} \\ \sum_{i=1}^m -\frac{\lambda_i^2}{2r_i}, & \text{caso contrário} \end{cases}$$

onde λ_i são os multiplicadores de Lagrange para a restrição de desigualdade e r é o parâmetro (ou vetor) de penalidade.

Nas funções de lagrangeano aumentado, tanto para a restrição de igualdade como para a desigualdade, os dois primeiros termos são chamados de função lagrangeana, sendo o terceiro o termo aumentado de penalização complementado, assim, o lagrangeano aumentado. O termo de penalização, na formulação tradicional é quadrático porém, outras formas podem ser utilizadas [68].

O tratamento das restrições de igualdade e desigualdade de forma conjunta, é feito com a combinação dos lagrangeanos aumentados para os casos de restrições de igualdade e desigualdade, descritos acima, sendo conhecido como função de Powell-Hestenes-Rockafellar (PHR) [68].

De acordo com [69] o lagrangeano aumentado apresenta multiplicadores estimados para cada restrição com a função lagrangeana podendo ser usada como pseudo-função objetiva sem restrição desde que a solução do problema de otimização sem restrição tenha definido os corretos e exatos multiplicadores como um ponto estacionário para a função lagrangeana aumentada. Mas, o ponto estacionário não precisa ser necessariamente um mínimo para a função de Lagrange [69]. Além disso, o uso do termo aumentado na função lagrangeana preserva as propriedades estacionárias da solução e assegura a conservação do mínimo [69]. Segundo [70], um lagrangeano aumentado é uma função penalização em que um valor finito dos parâmetros de penalização é suficiente para produzir a convergência para a solução do problema com restrição.

Na função lagrangeano aumentado, geralmente, cada restrição infactível é penalizada

individualmente, usando um fator de penalização $r \in \Re^{m+l}$ específico. Os multiplicadores λ , μ e os parâmetros r são todos fixados em cada iteração para a solução do subproblema irrestrito obtendo \mathbf{x}^k , sendo k a iteração atual. Depois, os multiplicadores de lagrange são atualizados baseado na solução aproximada \mathbf{x}^k . A atualização também pode ser aplicada nos parâmetros de penalização. Uma técnica do lagrangeano aumentado resolve uma seqüência de subproblemas muito simples, onde a função objetivo penaliza todas as restrições infactíveis do subproblema tratado. Os métodos do lagrangeano tendem a ser localmente convergentes se seus subproblemas são resolvidos para valores suficientemente pequenos para o parâmetro de penalização [71].

Apresenta-se agora os esquemas de atualização utilizados para os multiplicadores de lagrange, tanto para o modelo de Powell quanto o de Hestenes [9]:

$$\text{Powell: } \theta_i^{k+1} = \theta_i^k + g_i(\mathbf{x})^k$$

$$\text{Hestenes: } \lambda_i^{k+1} = \lambda_i^k + r_i g_i(\mathbf{x})^k$$

Outras formas de de atualização utilizando o gradiente das restrições também são utilizadas como, por exemplo, no método de Rosen:

$$\text{Rosen: } \lambda_i^{k+1} = -[\nabla(g_i(\mathbf{x})^k)^T \nabla g_i(\mathbf{x})^k]^{-1} \nabla(g_i(\mathbf{x})^k)^T \nabla f(\mathbf{x})^k$$

com o operador ∇ sendo o gradiente da função. Nota-se que, a vantagem dos procedimento de atualização de Hestenes e Powell é não necessitar do cálculo de derivadas.

É mais vantajoso utilizar um vetor de parâmetros de penalização do que um escalar pois, os parâmetros de penalidade r_i quando atualizados podem garantir a convergência global. Dessa forma, ajusta-se para cada restrição um parâmetro r_i com a finalidade de evitar mal condicionamento numérico. Existem alguns critérios para a atualização das penalizações. Um critério comum é que r_i^{k+1} seja igual a βr_i^k , onde $\beta \in \Re$ seja superior a um β_0 para evitar uma convergência muito lenta.

A seguir, apresenta-se a utilização do método de lagrangeano aumentado em conjunto com algoritmos evolucionistas e bio-inspirados de forma geral.

5.3 AE's baseados em lagrangeano aumentado

Quando algum método é acoplado com outra técnica, seja ela heurística ou da programação matemática, essa junção é chamada de algoritmos híbridos. Alguns algoritmos híbridos utilizando o método do lagrangeano aumentado estão presentes na literatura [67, 72, 73, 17, 74, 70, 71, 69].

Adeli e Cheng [67] propuseram um método de função penalização hibridizado para otimização estrutural. Os autores utilizam a hibridização do lagrangeano aumentado com um algoritmo genético geracional para a otimização de estruturas espaciais. A hibridização é feita em dois laços: um interno para a atualização dos multiplicadores de Lagrange e outro externo, onde o algoritmo genético minimiza a função objetivo penalizada associada com os multiplicadores de Lagrange. Os autores escolheram uma sequência de valores crescentes para o coeficiente da penalização parando-se o processo sempre que a precisão é atingida. Testes foram realizados em quatro tipos de estruturas espaciais: treliças de 12, 25 e 72 barras e num domo geodésico. Para cada experimento foi utilizado população de 50 indivíduos e três operadores de recombinação diferentes, com taxa de probabilidade de 0.8. Para o operador de mutação a probabilidade foi de 0.05. Os resultados foram comparados com outros presentes na literatura (Powell 1969, Fletcher 1975 e Arora 1984) e apresentarem resultados superiores.

Em [72] é utilizado um algoritmo híbrido do algoritmo genético com o lagrangeano aumentado. O objetivo era minimizar uma peça muito utilizado para prender telhados em paredes de forma que se obtivesse pesos e dimensões mínimas, sujeito a algumas restrições dadas.

Wah e Chen [73] utilizam um método híbrido do recozimento simulado com os algoritmos genéticos para otimização. Para as restrições foi adotado o modelo de lagrangeano aumentado. Os experimentos foram realizados em 11 funções do G-Suíte de funções [48].

Mezura [17] cita o trabalho “Otimização evolucionista baseada em penalização interior lagrangeana” de Myung e Kim. A adordagem dos autores garantem a geração de soluções factíveis durante o processo de busca. A primeira fase do algoritmo consiste na otimização da função aptidão penalizada $f(\mathbf{x}) = f(\mathbf{x}) + \frac{r}{2}(\sum_{i=1}^m (\max[0, g_i(\mathbf{x})])^2 + \sum_{j=1}^l (h_j(\mathbf{x}))^2)$ onde r é uma constante de penalização. A primeira fase do trabalho as restrições violadas aumentam de acordo com o que o usuário desejar. Na segunda etapa, os multiplicadores de lagrange ajustam a função penalização de acordo com a função recebida durante o

processo evolucionário. A desvantagem dessa abordagem é a dificuldade na definição dos parâmetros requeridos pelo algoritmo.

Na referência [74] “Usando o lagrangeano aumentado em enxame de partículas para otimização de problemas com restrição em engenharia” utilizam o método de enxame de partículas(PSO) para resolução de problemas não-lineares, não diferenciáveis e não convexos. Os autores relatam que a otimização via enxame de partículas permite uma implementação mais eficiente no tratamento de restrições de igualdade. Além disso, os autores combinam a técnica do enxame com a função de penalidade do método do multiplicador de Lagrange para o tratamento das restrições. Essa combinação é chamada de ALPSO. O algoritmo detecta automaticamente restrições ativas e fornece estimativas exatas do multiplicador de Lagrange se a formulação do problema é contínua. Além dos problemas de engenharia os testes foram realizados com o grupo de funções(G-Suíte)[48].

O trabalho de [70] apresenta uma metodologia do lagrangeano aumentado com o algoritmo bio-inspirado no comportamento de cardumes de peixes, que é um algoritmo estocástico baseado em população para resolver problemas de otimização global. Simula o comportamento de um cardume artificial de peixes(AFS) e tem apresentado resultados competitivos em problemas de engenharia. Nesse artigo [70], os autores utilizam as funções do lagrangeano aumentado integradas com a heurística do enxame de peixes. Analisa a convergência do algoritmo estocástico adaptado para o trato de restrições e, também, o efeito prático de alguns parâmetros no desempenho do algoritmo. Os testes foram realizados com um grupo de funções(G-Suíte) [48] e comparados com outros algoritmos também baseados em enxame de partículas. Nessa comparação os autores concluem que a junção proposta mostra eficiência na convergência de soluções com tolerância especificada e não tem alto custo computacional quanto se trata do número de funções avaliações.

O trabalho intitulado “Hibridização de um algoritmo genético com o método busca padrão usando lagrangeano aumentado”[71] combina duas técnicas de otimização: uma de busca global e outra de busca local, que trata as restrições. Desenvolve um algoritmo genético híbrido com base em uma pesquisa local usando a busca padrão com a técnica de penalização do lagrangeano aumentado para o tratamento de restrições. A busca global é realizada pelo algoritmo genético. Foram testados, diferentes esquemas de hibridização em relação ao tratamento de populações bem como o refinamento do espaço de busca local visando obter melhores soluções. Realizou-se uma análise comparativa sobre o tamanho

da população, bem como sobre a manipulação da população em relação a diferentes modelos híbridos. Por fim, os autores utilizaram o perfil de desempenho [5] para avaliar o comportamento do algoritmo híbrido proposto e atestar sua eficiência.

No artigo de [69] “Otimização estrutural com restrição via uma estratégia paralela de enxame de partículas com lagrangeano aumentado” apresenta-se uma extensão da abordagem da otimização via enxame de partículas, para a solução de problemas de otimização com restrição aplicado em projetos de engenharia [74], usando ambiente de computação paralela. A abordagem destaca a vantagem do PSO, eficiente para encontrar ótimos globais em problemas e projetos que possuem espaços complexos, em conjunto com o método de lagrangeano aumentado para o trato das restrições. Implementa uma versão paralela visando diminuir o custo computacional.

5.4 Considerações

De acordo com a seção anterior, descreve-se o uso da técnica de lagrangeano aumentado em conjunto com algoritmos evolucionistas geracionais. Todos os modelos descritos, apresentam uma implementação que segue os modelos tradicionais de Hestenes, Powel e Rockfellar de uma forma bastante rigorosa.

Ressalta-se, agora, algumas características desta aplicação de lagrangeano aumentado com técnicas evolucionistas que devem ser consideradas. Inicialmente, observa-se que tem-se somente um multiplicador e/ou variável de penalização para cada restrição do problema de otimização. Desta forma, todos os indivíduos da população devem utilizar o mesmo multiplicador/penalização para esta restrição específica. Ou seja, independente da posição em que se encontra no espaço de busca, um mesmo multiplicador será utilizado para construção do lagrangeano aumentado de todos os indivíduos da população. Isto se dá porque a técnica de lagrangeano aumentado não foi construída para trabalhar diretamente com população de soluções. Em programação matemática, tem-se um processo iterativo onde uma única candidata a solução vai sendo direcionada para o ótimo factível do problema por meio de uma estratégia de atualização da candidata a solução e do multiplicador, respectivamente.

Outra questão a considerar é em relação a atualização das penalidades e multiplicadores. Tem-se as fórmulas para tal, definidas na seção anterior, dos modelos de Hestenes,

Powell e Rosen. Porém, deve-se escolher qual valor das restrições violadas pelos indivíduos da população serão utilizadas na atualização dos multiplicadores. Geralmente, quando utiliza-se algoritmos evolucionistas em conjunto com lagrangeano aumentado, adota-se um valor que seja mais representativo da condição da restrição em relação a população vigente. Em [67], por exemplo, adota-se a média de cada restrição para toda a população. Encontram-se, também, implementações onde adota-se o valor da restrição do melhor indivíduo da população. De qualquer forma, esta é uma escolha que influencia muito a qualidade da busca pois, define diretamente como cada restrição do problema será considerada de acordo com a população do algoritmo evolucionista no momento da atualização dos multiplicadores e penalidades. Apesar da importância desta definição, pouco se analisa sobre este detalhe nos trabalhos correlatos.

Um aspecto também relevante é o momento da atualização dos multiplicadores. Em se tratando de algoritmos evolucionistas geracionais, tem-se, na prática, dois laços no procedimento de otimização utilizando lagrangeano aumentado. O laço relativo ao processo evolutivo, onde avalia-se toda a população para a geração da próxima população, com os multiplicadores sendo mantidos fixos e o laço referente as iterações em que os os multiplicadores devem ser atualizados. Tem-se, então, a necessidade de se definir para o laço evolutivo, mais interno, o momento em que se deve atualizar os multiplicadores. Geralmente, tal definição é feita pelo usuário através da determinação de uma variável de atualização, por exemplo, a cada número de ciclos evolutivos.

Porém, o aspecto mais relevante em relação a utilização de lagrangeano aumentado como uma estratégia da programação matemática e sua adaptação para uso com algoritmos evolucionistas está na forma diferenciada de utilização da função lagrangeano em cada modelo. No caso da programação matemática, a função lagrangeana é base para a aplicação de um otimizador para problemas não-lineares sem restrição, que irá calcular uma solução aproximada da função lagrangeana aumentada, definindo o próximo valor da solução procurada dentro do processo iterativo. Esta sequência visa diminuir a influência da parcela relativa a penalização com a transferência da violação da restrição para o multiplicador de Lagrange, o que ocorre no processo de atualização do mesmo. Busca-se a convergência, sob condições adequadas [75], tanto da solução, para o ótimo, como dos multiplicadores neste processo iterativo.

No caso do lagrangeano aumentado com algoritmos evolucionistas, o funcionamento

não é tão acoplado entre a solução e os multiplicadores. A função lagrangeano define a função objetivo que indica quão bom é o indivíduo em relação aos demais indivíduos da população, viabilizando o cálculo da aptidão. Ou seja, os multiplicadores não tem como objetivo direto diminuir a influência do termo de penalização que, geralmente, introduz mal condicionamento na resolução do sistema. Isto porque a escolha dos novos indivíduos, na computação evolucionista, se dá pela aplicação dos operadores genéticos e não pela otimização do problema irrestrito gerado. Ou seja, o funcionamento entre solução e multiplicador é mais desacoplado, sendo, inclusive, mais impreciso por usar o mesmo multiplicador da restrição para todos os indivíduos da população e pela sua forma de atualização, conforme descrito acima.

Feitas estas observações, delinea-se a estratégia para a construção de um algoritmo de lagrangeano aumentado específico para funcionamento em conjunto com algoritmos evolucionistas, que será descrito na próxima seção.

5.5 Um Algoritmo evolucionista baseado em lagrangeano aumentado

Apresenta-se agora, o desenvolvimento de um algoritmo evolucionista para problemas de otimização com restrição construído tendo como referência a estratégia de lagrangeano aumentado.

Inicialmente, tem-se como referência trabalhos que utilizam-se dois níveis (laços) para algoritmos evolucionistas em conjunto com lagrangeano aumentado [67]. Isto se dá, principalmente, pelo modelo geracional dos algoritmos utilizados. Visando evitar esses dois laços, optou-se por adotar um modelo de algoritmo genético não-geracional [1]. Além de se evitar o laço evolutivo, tem-se a vantagem adicional de se ter um processo menos abrupto na variação das infactibilidades da população, visto que, somente uma pequena parcela da população é substituída a cada iteração. Neste caso específico [1], um ou dois indivíduos, dependendo do operador evolutivo utilizado. Esta estratégia torna-se ainda mais relevante quando se pensa na atualização dos multiplicadores. A medida representativa do estado atual da infactibilidade para determinada restrição também sofre menos impacto com este procedimento de atualização mais suave da população.

No que tange a função lagrangeana, a maneira diferenciada de comportamento do

lagrangeano aumentado evolutivo para o modelo utilizando programação matemática indica a possibilidade de se utilizar de uma forma mais efetiva a função lagrangeana no processo evolutivo. Isto porque a função lagrangeana não é a responsável direta para atualizar a solução para a região admissível ótima mas, para apresentar uma medida mais fiel da qualidade de determinado indivíduo. Assim, procura-se um modelo para a função lagrangeana mas, principalmente, para a atualização dos multiplicadores e penalizações que procure refinar esta medida no decorrer do processo evolutivo.

Propõe-se, para este modelo, a consideração da função lagrangeano como composta de dois termos que penalizam o indivíduo ineficaz, a saber:

- um termo linear (relativo ao multiplicador);
- um termo quadrático (relativo a constante de penalização).

O controle dos valores dos multiplicadores e das penalizações para cada restrição, deverá ser construído de acordo com as expectativas que se tenha para o comportamento de cada um destes termos, no decorrer do processo evolutivo. Pretende-se que, no início do processo evolutivo, a penalização seja preponderante em relação ao multiplicador de Lagrange, eliminando os indivíduos mais ineficazes rapidamente, visando alocar a população, como um todo, mais próxima de regiões eficazes. Desta forma, a penalização deve ser mais rigorosa nas primeiras iterações do algoritmo não-geracional. Este comportamento implica que o termo quadrático deve ser mais relevante na fase inicial do procedimento evolutivo. No decorrer do processo, a penalização deve ir perdendo relevância em detrimento do multiplicador de lagrange, ou seja:

$$r_i^k \rightarrow 0$$

Assim, durante as interações, vai-se trocando a penalização quadrática pela parcela linear, associada ao multiplicador. A grande vantagem é que a penalização linear é uma medida direta da distância da região eficaz, facilitando um ajuste fino entre a magnitude da função objetivo em relação a parcela de penalização, Ou seja, pode-se monitorar melhor a factibilidade em relação a otimalidade [75]. Deve-se ressaltar que este comportamento é parecido com o que ocorre com a programação matemática, onde, com o acoplamento da solução com os multiplicadores esta transferência é feita naturalmente podendo, inclusive, manter o parâmetro de penalização constante. No caso do uso do lagrangeano aumentado

com algoritmos evolucionistas, a forma de atualização das penalizações e multiplicadores é que forçará este comportamento.

A montagem da função lagrangeano pode ser a tradicional, utilizada nos modelos da programação matemática:

$$F(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \max[0, g_i(\mathbf{x})] + \sum_{i=1}^m r_i (\max[0, g_i(\mathbf{x})])^2 \quad (5.13)$$

Porém, pelas expectativas de comportamento, será utilizada uma nova função que garanta que o termo quadrático seja sempre maior que o termo linear no que tange a infactibilidade. Assim, adota-se a função lagrangeano modificada:

$$F(\mathbf{x}) = f(\mathbf{x}) + \sum_{\forall g_i(\mathbf{x}) > 0} \lambda_i (1 + g_i(\mathbf{x})) + \sum_{\forall g_i(\mathbf{x}) > 0} r_i [1 + g_i(\mathbf{x})]^2 \quad (5.14)$$

ou seja, considera-se somente as restrições ativas. Serão realizados experimentos com ambas funções, visando avaliar o desempenho nos problemas de otimização.

Em relação aos valores de inicialização, para os multiplicadores assume-se, como é comum, que sejam nulos. Para as penalidades, uma escolha que se mostrou razoável foi assumir uma relação direta de seu valor com o nível de infactibilidade da população (nig_i) para cada restrição $g_i(\mathbf{x})$ do problema.

O cálculo do nível de infactibilidade da população para determinada restrição leva em conta somente os indivíduos da população que estejam infactíveis para a restrição considerada. Determina-se, assim, o número de indivíduos infactíveis (iig_i) para a restrição $g_i(\mathbf{x})$, ou seja, conta-se o número de indivíduos da população onde $g_i(\mathbf{x}) > 0$. Soma-se, então, todas as violações destes indivíduos e obtém-se a média:

$$nig_i = \frac{\sum_{n=1}^{npop} \max[0, g_i(\mathbf{x})]_n}{iig_i} \quad (5.15)$$

sendo i a i -ésima restrição considerada e $npop$ o tamanho da população. Este cálculo deve ser feito para cada uma das restrições do problema. Caso $iig_i = 0$ não é necessário o cálculo de nig_i para a i -ésima restrição pois, nenhum indivíduo da população é infactível para esta restrição não sendo, portanto, necessário penalizar esta restrição na população

vigente. Adota-se então, como valor das penalizações iniciais:

$$r_i^0 = nig_i \quad (5.16)$$

Define-se, a seguir as estratégias de atualização dos multiplicadores e penalizações. Em relação ao tamanho do ciclo, para que seja feita a atualização, foi adotada uma estratégia similar ao modelo não-geracional adaptativo [1]. Logicamente, esta forma pode ser mais especializada para caso do lagrangeano aumentado mas, nesta primeira versão, preferiu-se simplificar esta etapa por meio desta adoção. Experimentos mostraram um comportamento razoável com esta escolha. Na prática, para o modelo não-geracional, poderia abolir o ciclo/laço do AG geracional [67] atualizando os multiplicadores e penalizações a cada iteração do processo evolutivo.

A atualização dos multiplicadores e penalizações devem levar em conta o comportamento esperado, principalmente para a penalização no decorrer das iterações. Além disto, será feita buscando manter um equilíbrio entre os valores relativos da função objetivo $f(\mathbf{x})$ e das parcelas de penalização e multiplicadores. Isto será feito, usando como referência na atualização o melhor indivíduo presente na população (f_{melhor}) e da média das funções objetivo da população (f_{media}). Ambos os valores são calculados em relação a função $f(\mathbf{x})$, isto é, sem considerar as restrições. Assim:

$$\lambda_i^{k+1} = f(f_{melhor}^k, f_{media}^k) \quad (5.17)$$

Considera-se, ainda, uma forma de normalização para as restrições que são violadas por um ou mais indivíduos da população na hora da atualização. O valor nig_i representa a média do valor de violação da restrição i para a população atual. Existem indivíduos que apresentam maior violação que nig_i e indivíduos que apresentam valores mais baixos para a violação desta restrição. A idéia é que se tenha um controle para o valor a ser considerado na restrição tendo com referência indivíduos que estejam próximos da média da restrição considerada. Assim:

$$\lambda_i^{k+1} = f(f_{melhor}^k, f_{media}^k, nig_i^k) \quad (5.18)$$

Tem-se, então, as seguintes estratégias para atualização dos multiplicadores:

$$\lambda_i^{k+1} = \frac{(|f_{media}^k - f_{melhor}^k|/f_{media}^k + 1) \cdot (|f_{melhor}^k| + 1)}{nig_i^k + 1} \quad (5.19)$$

Uma diferença fundamental da atualização em relação ao lagrangeano aumentado tradicional é a independência da atualização do multiplicador em relação aos valor anterior do multiplicador e penalidade. Depende de uma informação global da população, representada por f_{media} , uma informação local da mesma população, definida por f_{melhor} e uma informação geral do desempenho da população na restrição, definida em nig_i . No caso da atualização das penalizações, seguem-se as mesmas idéias, adotando para a atualização:

$$r_i^{k+1} = \frac{|(f_{media}^k - f_{melhor}^k)/f_{media}^k| + 1}{nig_i^k + 1} \quad (5.20)$$

os valores unitários somados a nig_i não visam evitar divisão por zero pois é sempre positivo. O objetivo é evitar que os termos cresçam quando se tem valores de nig_i muito pequenos(próximos de zero). A expectativa é que no decorrer do processo evolutivo usando um algoritmo evolutivo não-geracional o termo $f_{media} - f_{melhor}$ tenda para zero, gerando o comportamento esperado para o termo de penalidade. Descreve-se, a seguir, o algoritmo evolucionista não-geracional baseado em lagrangeano aumentado em 5.2:

Finalizando, deve-se ressaltar que o algoritmo baseado em lagrangeano aumentado apresentado, tem uma estrutura bastante diferente dos modelos tradicionais de lagrangeano aumentado [10, 76]. Esta flexibilização foi possível pelas características inerentes aos algoritmos evolucionistas usados no processo de otimização. Perde-se propriedades de convergência inerentes da análise do lagrangeano aumentado no contexto da programação matemática [75] mas, tem-se uma estratégia mais adequada a expectativa de comportamento de um algoritmo evolucionista. Como no caso do modelo adaptativo, adotou-se um algoritmo genético não-geracional [1] para a implementação do modelo baseado em lagrangeano aumentado. Alguns testes numéricos foram realizados com um modelo de lagrangeano aumentado utilizando as estratégias tradicionais de atualização dos multiplicadores e penalizações, implementado também em um algoritmo genético não-geracional. Desta forma, tem-se uma referência na análise do desempenho desta nova estratégia apresentada. Os experimentos numéricos serão apresentados no próximo capítulo.

Algoritmo evolucionista baseado em lagrangeano aumentado

Início

Gere a população inicial
 Avalie a população inicial
 Calcule as violações das restrições pelos indivíduos da população inicial
 Calcule nig_i para cada restrição i
 Determine $r^0 = nig_i$, $\lambda^0 = 0$, para cada restrição i
 Determine ATUALIZA = falso
 Calcule a função lagrangeano para a população

Repita

Aplique operador genético
 Avalie indivíduo(s) gerado(s)
 Calcule a função lagrangeano
 Selecione indivíduo f para sobreviver
 Aloque o novo indivíduo na população
 Atualize a variável ATUALIZA

Se ATUALIZA = verdadeiro Então

Calcule nig_i
 Atualize λ baseado em (5.19)
 Atualize r baseado em (5.20)
 ATUALIZA = falso

Fim se

Até critério de parada satisfeito

Fim

Figura 5.2: Pseudo-código para algoritmo evolucionista baseado em lagrangeano aumentado

6 EXPERIMENTOS NUMÉRICOS

Para realizar os experimentos foi utilizado um AG não-geracional do tipo steady-state proposto por [1] com codificação real e a linguagem de programação Fortran. A finalidade desses experimentos era obter o ponto de “máximo ou mínimo” de funções matemáticas, problemas da engenharia mecânica e estruturas da engenharia civil. As funções matemáticas são 24 funções do G-Suíte propostas por [48]. Os problemas da engenharia mecânica são: projeto de tensão compressão da mola, redutor de velocidade, viga soldada, vaso de pressão, viga engastada livre. Nas estruturas da engenharia civil objetivou-se minimizar o peso de 6 tipos de estruturas reticuladas sendo elas: treliças de 10 barras, 25 barras, 72 barras, 120 barras e por último, uma treliça de 200 barras. O espaço de busca para as variáveis de projeto são discretos ou contínuos.

No G-Suíte de funções [48] foram utilizados os operadores de mutação: randômica, Muhlenblein e não uniforme($b = 5$) e os operadores de recombinação(crossover): discreto, SBX, 1 ponto, 2 pontos, flat, BLX- $\alpha = 0.3$, LS, pais múltiplos, Wright e geométrico. No restante dos experimentos, ou seja, nos problemas de engenharia mecânica e treliças reticuladas, os operadores utilizados foram mutação randômica e recombinação Wright, ambos com taxa de probabilidade de 0.2 e 0.8, respectivamente. Nos experimentos utilizou-se dois métodos de penalização: o método de penalização adaptativo discutido na seção 4.2 e o método do lagrangeano aumentado proposto na seção 5.4 dessa dissertação.

A organização desse capítulo será feita da seguinte forma: Na primeira seção explicita-se um pouco sobre a ferramenta “perfil de desempenho” que é uma estratégia de comparação de resultados na forma gráfica. Essa ferramenta será utilizada para comparar o desempenho de diversas combinações dos operadores citados acima e a comparação com alguns trabalhos da literatura. Nas próximas seções, serão explicitados os experimentos no G-Suíte [48], em problemas da engenharia mecânica e por último nos projetos de otimização estrutural.

6.1 Descrição das estratégias de comparação utilizadas

A comparação dos resultados da otimização de funções G-Suite [48] é realizada utilizando uma ferramenta gráfica chamada de perfil de desempenho (*performance profiles*). Essa ferramenta foi utilizada recentemente em alguns trabalhos como: em [77] para analisar os resultados do CEC 2006, em [78] para comparar dois algoritmos com codificação binária e por último em [79] onde apresenta-se uma nova proposta para o “perfil de desempenho” baseado em resultados probabilísticos, sendo este uma alternativa para situações em que os dados são obtidos de um processo estocástico.

O perfil de desempenho facilita a interpretação e visualização de experimentos com grande quantidade de dados. Diversos novos métodos surgem a cada dia para resolver problemas e estes exigem uma estratégia clara para avaliar a qualidade de algoritmos candidatos. Apesar de a princípio parecer simples, a avaliação experimental de algoritmos na prática, apresenta algumas dificuldades como: a definição de conjuntos problemas é heterogênea com um pequeno domínio para a realização dos experimentos; a determinação de medidas de desempenho para avaliar o algoritmo [80]; a decisão de como representar e interpretar os resultados obtidos nos experimentos. Geralmente, quando vamos trabalhar com o G-Suite o número de algoritmos ou problemas é muito grande para a representação de todos os resultados.

O perfil de desempenho foi proposto pela primeira vez em 2002 [5] para facilitar a visualização dos resultados obtidos por algoritmos de otimização. Essa ferramenta facilita trabalhar com a variabilidade encontrada na maioria das configurações experimentais. Devido a eficiência dessa ferramenta com algoritmos determinísticos, surgiu a idéia de uma nova proposta para trabalhar com algoritmos estocásticos [79]. Isso porque, geralmente os algoritmos estocásticos trazem muita incerteza devido aos diferentes desempenhos nas suas diversas execuções mesmo quando se mantém os mesmos parâmetros na mesma rodada. Sem essa ferramenta seria muito difícil a elaboração e análise de relatórios de experimentos envolvendo um grande número de problemas teste e algoritmos.

Uma extensão desta ferramenta analítica original [5] foi apresentada em [79] e recebeu o nome de “perfil de desempenho probabilístico”. O objetivo era adaptar uma ferramenta a princípio projetada para um ambiente determinístico e assim facilitar a análise do de-

sempenho de algoritmos estocásticos. Um exemplo utilizando a variância dos resultados é apresentada, assumindo que esses dados possam ser aproximados por uma distribuição normal. Segundo os autores de [79] essa ferramenta pode ser especializada para situações particulares. Além disso, mostram-se as vantagens do novo método com dados de um experimento real que envolvam vários problemas e algoritmos, onde desvendam tendências que são difíceis de detectar nos dados brutos e que não seriam apresentadas pelo uso do "perfil de desempenho" padrão.

Para explicitar o perfil de desempenho considere um conjunto P de problemas testes p_j , com $j = 1, 2, \dots, n_p$, e A um conjunto de algoritmos a_i onde $i = 1, 2, \dots, n_a$. Em cada problema p_j é definido um custo $c_{ij} > 0$ que mede o desempenho do i -ésimo algoritmo no j -ésimo problema, ou seja, essa taxa representa a melhor variação do desempenho em cada problema onde essa variável vai ser definida como um número positivo. Sendo o custo c_{ij} uma medida de desempenho e assumindo-se seu valor como positivo que pode ser definida como:

$$r_{ij} \equiv \frac{c_{ij}}{b_j} \quad (6.1)$$

onde o r_{ij} é uma medida que representa o desempenho relativo do algoritmo a_i sobre o problema p_j , b_j é o melhor desempenho obtido pelos algoritmos em A no j -ésimo, ou seja, $b_j = (\min_k c_{kj})$. Note que o r_{ij} representa as medidas em uma escala comum.

Então, dado um conjunto de problemas P e um conjunto de algoritmos A , o "perfil de desempenho" de um algoritmo a_i em P é uma função dada por:

$$\rho_i : \mathfrak{R}_+ \longrightarrow [0, 1]$$

$$\rho_i(\tau) \equiv \frac{1}{n_p} \sum_{j=1}^{n_p} \delta(r_{ij}, \tau) \quad (6.2)$$

onde $\rho_i(\tau)$ determina a probabilidade de um problema P ser resolvido dentro de um nível de tolerância τ , e:

$$\delta(r_{ij}, \tau) = \begin{cases} 1 & \text{se } r_{ij} \leq \tau, \\ 0 & \text{se caso contrário} \end{cases}$$

Assim em qualquer ponto τ tem-se uma classificação do algoritmo e quanto maior o valor de $\rho_i(\tau)$ melhor o desempenho do algoritmo.

Existe também, a medida: $\zeta(a_i) = \sup \tau : \rho_i(\tau) < 1$ que mede a confiança dos algoritmos. Logo o algoritmo será mais confiável quando tiver um pequeno valor para $\zeta(a_i)$. Mas, a escolha do melhor algoritmo vai depender do que se espera dele, ou seja, se o método é bem executado num subconjunto de problemas. Ou ainda, se o algoritmo é mais confiável ou se o algoritmo tem um desempenho num certo nível de tolerância τ .

É importante observar algumas propriedades da função (6.2) que calcula o melhor desempenho do algoritmo nessa ferramenta [77]:

- Na equação (6.2), $\rho_i(\tau)$ é uma função crescente, constante e por partes.
- Quando $\tau = 1$, o $\rho_i(\tau)$ é a fração de problemas em P em que o algoritmo apresenta melhor desempenho quando comparado com os demais algoritmos.
- Quando $\tau = \infty$, o $\rho_i(\tau)$ representa a fração de problemas que o algoritmo é capaz de resolver.
- $\rho_i(1)$ é a porcentagem de problemas em que o algoritmo i tem melhor desempenho. Dados dois algoritmos A e B se $\rho_A(1) > \rho_B(1)$ então, o algoritmo A resolve um maior número de problemas do que o algoritmo B .
- A área sob a curva $\rho_i(\text{AUC})$ é um conjunto indicador de desempenho, quanto maior a AUC maior a eficiência do algoritmo.

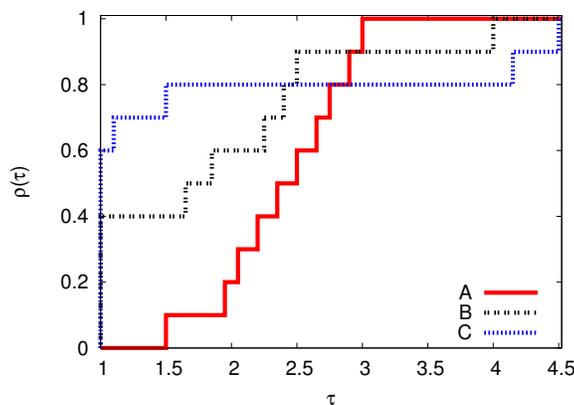
A referência [77] mostra outro exemplo de utilização de três algoritmos A, B e C para resolver 9 problemas e analisar o desempenho dos algoritmos através dessa ferramenta.

Figura 6.1: Desempenho de 9 problemas obtidos através dos algoritmos A,B,C

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
A	1.5	2.0	2.0	2.2	2.4	2.5	2.6	2.8	2.9	3.0
B	1.0	1.0	1.0	1.0	1.6	1.8	2.2	2.4	2.5	4.0
C	4.5	4.2	1.5	1.1	1.0	1.0	1.0	1.0	1.0	1.0

Utilizando a ferramenta perfil de desempenho para essa tabela 6.1, apresentou-se em [77] o gráfico 6.2:

Figura 6.2: Taxa de Desempenho dos algoritmos A, B e C

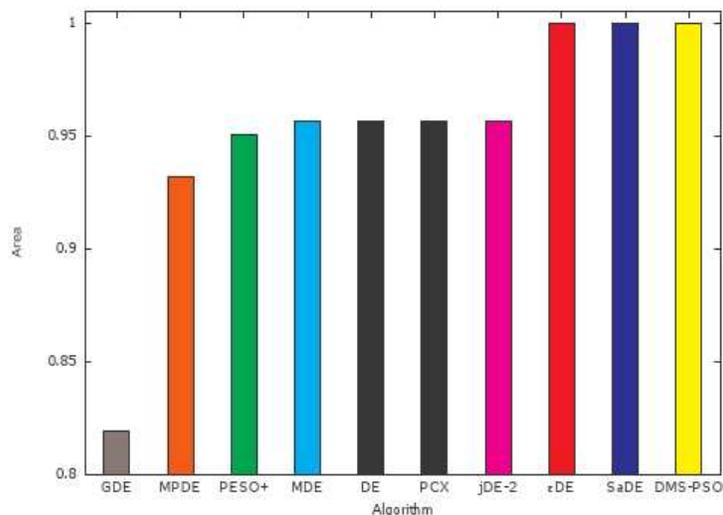


- A partir da figura 6.2 observa-se que: Como $p_C(1) > p_B(1) > p_A(1)$ o algoritmo C é o algoritmo que apresenta melhores resultados no conjunto P
- $\rho_A(1) = 0$ apresenta o pior desempenho obtido pelos algoritmos.
- Considerando $AUC_i = \int \rho_i(t)dt$ pela figura 6.2 $AUC_A = 21.15$, $AUC_B = 26.35$ e $AUC_C = 27.75$. Como $AUC_C > AUC_B > AUC_A$ configurando C como a mais eficiente.

Em [77] os autores explicam mais alguns detalhes importantes sobre a ferramenta perfil de desempenho. Um deles é o fato de que o desempenho do algoritmo pode mudar totalmente dependendo do critério adotado durante a avaliação. Durante a utilização da ferramenta é obtida uma medida que é mostrada através de uma curva AUC. Onde essa curva é plotada com linhas paralelas ao eixo horizontal. De forma que o melhor desempenho corresponde aos valores mais altos apresentados ao longo do eixo vertical. Os gráficos do perfil de desempenho também podem ser feitos por meio de áreas dessas curvas obtidas como por exemplo, as mostradas no gráfico apresentado em 6.2. A figura 6.3 mostra um exemplo de utilização do “perfil de desempenho” através do gráfico em barras. Nesse gráfico observa-se que a última barra, corresponde a maior área obtida, ao se comparar com outros algoritmos. Logo esta barra representa o algoritmo que obteve melhor desempenho. Nesse gráfico mostrado em 6.3, da referência [77] os autores comparam os resultados do G-Suíte [48] com a utilização de alguns algoritmos que foram apresentados no CEC 2006.

Depois de realizar diversos experimentos no G-Suíte de funções [48] foi utilizado o

Figura 6.3: Comparação dos resultados obtidos no CEC2006



perfil de desempenho para comparar os resultados obtidos com os operadores e assim, determinar qual operador apresentou melhor desempenho. Mais detalhes serão explicitados na seção seguinte.

6.2 Experimentos na suíte de funções

A fim de investigar o desempenho do procedimento de penalização, são consideradas as 24 conhecidas funções (G1-G24) propostas por Suganthan [48], em 2006, num congresso de computação evolucionista. O G-Suite é constituído de diferentes tipos de funções que envolvem restrições de igualdade linear, desigualdade linear e, também, desigualdade não linear. Uma extensa discussão envolvendo cada um desses problemas e outras técnicas da literatura de computação evolucionista pode ser encontrada em [48]. O G-Suite é formado por 24 problemas teste como descrito nas tabelas 6.1 e 6.2.

Ainda na referência [48] são apresentados alguns detalhes desses problemas conforme a tabela 6.3, onde p representa a taxa estimada em porcentagem entre a região factível e o espaço de busca. Outros detalhes também são fornecidos na tabela como: número de variáveis, número de restrições (total de igualdade+desigualdade) e o tipo de funções apresentadas no problema.

Tabela 6.1: Funções componentes do G-Suíte

Prob	função	ótimo
G01	$5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$	-15.00000
G02	$-\frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}}$	-0.80361
G03	$-(\sqrt{n})^n \prod_{i=1}^n x_i$	-1.00050
G04	$5.3578547 \cdot x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$	$-3.06655e^4$
G05	$3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$	5126.49671
G06	$(x_1 - 10)^3 + (x_2 - 20)^3$	-6961.81387
G07	$x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$	24.30620
G08	$\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$	-0.09582
G09	$(x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$	680.63005
G10	$x_1 + x_2 + x_3$	7049.24802
G11	$x_1^2 + (x_2 - 1)^2$	0.749900
G12	$-(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100$	-1.00000

Tabela 6.2: Continuação da tabela com as funções do G-Suíte

Prob	função	ótimo
G13	$e^{x_1x_2x_3x_4x_5}$	0.05394
G14	$\sum_{i=1}^{10} x_i (c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j})$	-47.764888
G15	$1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$	961.71502
G16	$0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} + 0.0321y_{12}$ $0.004324y_5 + 0.0001 \frac{c_{15}}{c_{16}} + 37.48 \frac{y_2}{c_{12}} - 0.0000005843y_{17}$	-1.90515
G17	$f(x_1) + f(x_2)$	8853.53967
G18	$-0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$	-0.86602
G19	$\sum_{j=1}^5 \sum_{i=1}^5 c_{ij} x_{(10+i)x(10+j)} + 2 \sum_{j=1}^5 d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i$	32.65559
G20	$\sum_{i=1}^{24} a_i x_i$	0.20497
G21	x_1	193.72451
G22	x_1	236.43097
G23	$-9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$	-400.05510
G24	$-x_1 - x_2$	-5.50801

Tabela 6.3: Informações sobre o G-Suíte de Funções

Problema	variáveis	tipo de função	p%	restrições
G01	13	quadrática	0.0111	9
G02	20	não linear	99.9971	2
G03	10	polinomial	0.0000	1
G04	5	quadrática	52.1230	6
G05	4	cúbica	0.0000	5
G06	2	cúbica	0.0066	2
G07	10	quadrática	0.0003	8
G08	2	não linear	0.8560	2
G09	7	polinomial	0.5121	4
G10	8	linear	0.0010	6
G11	2	quadrática	0.0000	1
G12	3	quadrática	4.7713	1
G13	5	não linear	0.0000	3
G14	10	não linear	0.0000	3
G15	3	quadrática	0.0000	2
G16	5	não linear	0.0204	38
G17	6	não linear	0.0000	4
G18	9	quadrática	0.0000	13
G19	15	não linear	33.4761	5
G20	24	linear	0.0000	20
G21	7	linear	0.0000	6
G22	22	linear	0.0000	20
G23	9	linear	0.0000	6
G24	2	linear	79.6556	2

Técnicas de tratamento de restrições consideradas aqui, nesse trabalho, são testadas utilizando as 24 funções onde, 3 níveis de avaliação da função aptidão são considerados: 5 000, 50 000 e 500 000. Além dos 5 operadores utilizados em [1] foram utilizados mais 9 operadores que não tinham sido relatados no trabalho proposto pelos autores de [1]. Os operadores que foram acrescentados são: recombinação discreto, recombinação 1 ponto, recombinação 2 pontos, recombinação flutuante, recombinação BLX- $\alpha=0.3$, recombinação LS, recombinação pais múltiplos, recombinação Wright, recombinação geométrica. As tabelas 8.1 a 8.30 que estão no apêndice, são relativas aos resultados obtidos para as funções teste(G1-G24) com 25 rodadas independentes, usando a população contendo 500 indivíduos e o número máximo de funções avaliação(neval) foi definido como 50000. Foi utilizado um parâmetro fixo `ninser = 3` em todos os experimentos. Essa escolha do valor do parâmetro `ninser` foi baseado nos estudos da referência [1] que foram ponto de partida desse trabalho.

Nesses experimentos foram utilizados todos os operadores citados no início desse capítulo conforme mostram as tabelas 8.1 a 8.30 do apêndice. A técnica de penalização utilizada foi o APM da seção 4.2. Primeiramente utilizou-se combinações do operador de mutação randômica com cada um dos operadores: recombinação discreta, SBX, 1 ponto, 2 pontos, flutuante, BLX- $\alpha=0.3$, LS, pais múltiplos, Wright, geométrico. Os resultados dessas combinação foram apresentados nas tabelas 8.01 a 8.10 do apêndice. A taxa de probabilidade do operador de mutação foi de 0.2 e do operador de recombinação 0.8. Em seguida, utilizou-se o operador de mutação de Muhlenblein(0.2) com todos os operadores de recombinação(0.8) explicitados anteriormente. Os resultados desses operadores foram mostrados nas tabelas 8.11 a 8.20 do apêndice. Finalmente, utilizou-se o operador de mutação não uniforme(0.2) com os outros operadores de recombinação(0.8) e os resultados estão presentes nas tabelas 8.21 a 8.30 do apêndice.

As medidas obtidas nos experimento foram as seguintes:

- **fcn:** corresponde ao número da função de acordo com a tabela 6.1 e 6.2.
- **melhor:** corresponde ao melhor valor encontrado pelo algoritmo.
- **mediana:** corresponde a medida de localização do centro da distribuição dos valores prováveis para o ótimo.
- **média:** corresponde ao valor da média aritmética dos valores candidatos ao ótimo.
- **dv. padrão:** corresponde ao desvio padrão, ou seja, ao grau de dispersão dos valores candidatos ao ótimo em torno do valor da média.
- **pioor:** corresponde ao menor valor encontrado pelo algoritmo entre todos os valores candidatos ao ótimo.
- **nma:** corresponde ao número médio de avaliações em que o valor candidato ao ótimo foi encontrado.
- **frun:** número da rodada em que o ótimo foi encontrado.

Em segundo estágio, foi feita uma análise gráfica desses experimentos utilizando o perfil de desempenho e a partir desse gráfico foi retirada a melhor combinação de operadores, ou seja, a combinação que apresentou melhor desempenho. Essas combinações utilizaram um

operador de mutação e um de recombinação de forma que as primeiras foram feitas com o operador de mutação randômica com todos os operadores de recombinação utilizados nesse trabalho. Em seguida, realizou-se combinações com o operador de mutação Muhlenbein com os operadores de recombinação e por último, o operador de mutação não uniforme com os operadores de recombinação. Essas combinações foram definidas na tabela 6.4 onde a taxa de probabilidade de recombinação é de 0.8 e para a mutação 0.2. A idéia

Tabela 6.4: Combinação de operadores genéticos

Combinação	Recombinação	Mutação
A	1 ponto	randômica
B	2 pontos	randômica
C	discreto	randômica
D	flat	randômica
E	BLX($\alpha = 3$)	randômica
F	SBX	randômica
G	geométrico	randômica
H	wrigh	randômica
I	LSX	randômica
J	pais múltiplos	randômica
K	1 ponto	muhlenbein
L	2 pontos	muhlenbein
M	discreto	muhlenbein
N	flat	muhlenbein
O	BLX($\alpha = 3$)	muhlenbein
P	SBX	muhlenbein
Q	geométrico	muhlenbein
R	wrigh	muhlenbein
S	LSX	muhlenbein
T	pais múltiplos	muhlenbein
U	1 ponto	não uniforme(b=5)
V	2 pontos	não uniforme(b=5)
X	discreto	não uniforme(b=5)
W	flat	não uniforme(b=5)
Y	BLX	não uniforme(b=5)
Z	SBX	não uniforme(b=5)
AZ	geométrico	não uniforme(b=5)
BZ	wrigh	não uniforme(b=5)
CZ	LSX	não uniforme(b=5)
DZ	pais múltiplos	não uniforme(b=5)

do estudo é verificar qual das 30 combinações determina o melhor desempenho. Nesse trabalho foram obtidos os valores das medidas: **média, melhor, pior, mediana, desvio padrão e número médio de avaliações**. Os gráficos de 6.4 a 6.13 mostram o perfil de desempenho dessas medidas afim de encontrar as melhores soluções nas 25 rodadas

independentes, respectivamente. Relataram os perfis métricos da **média**, **mediana** e **pior** com os melhores desempenhos da combinação nomeada “H” correspondente aos operadores de recombinação Wright e mutação randômica.

Figura 6.4: Gráfico em Linhas - Desempenho da medida **melhor** para as 30 combinações

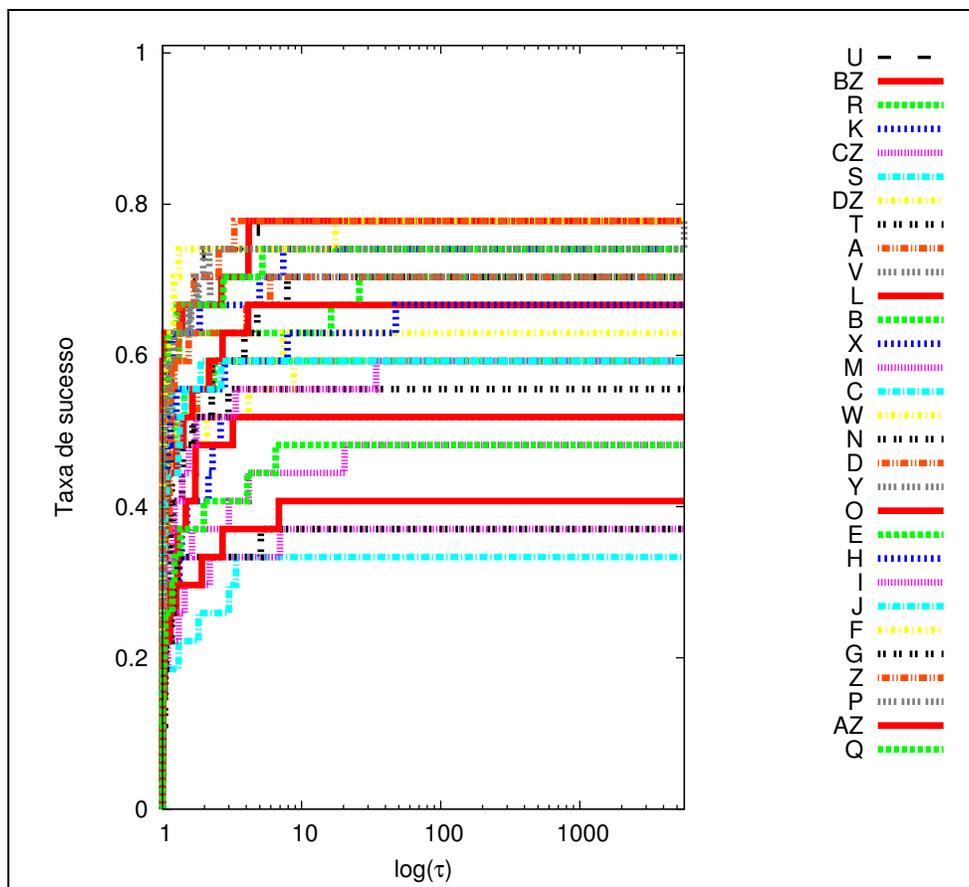


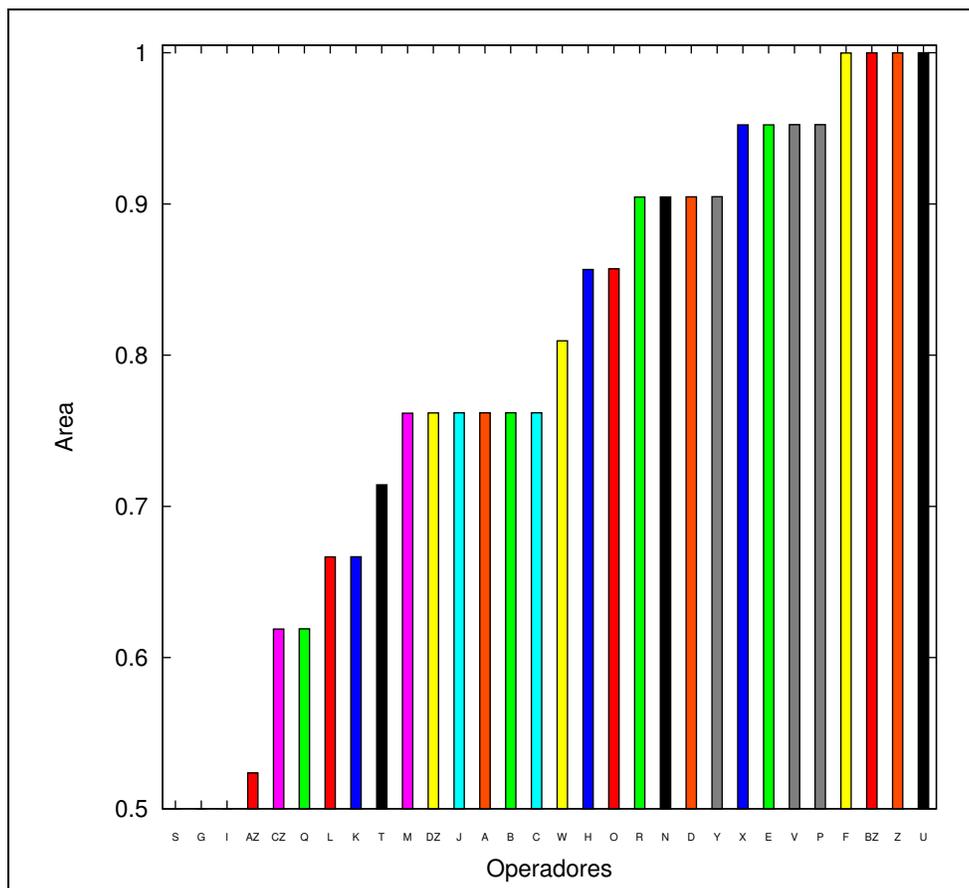
Figura 6.5: Gráfico em Barras - Desempenho da medida **melhor** para as 30 combinações

Figura 6.6: Gráfico em Linhas - Desempenho da medida **média** para as 30 combinações

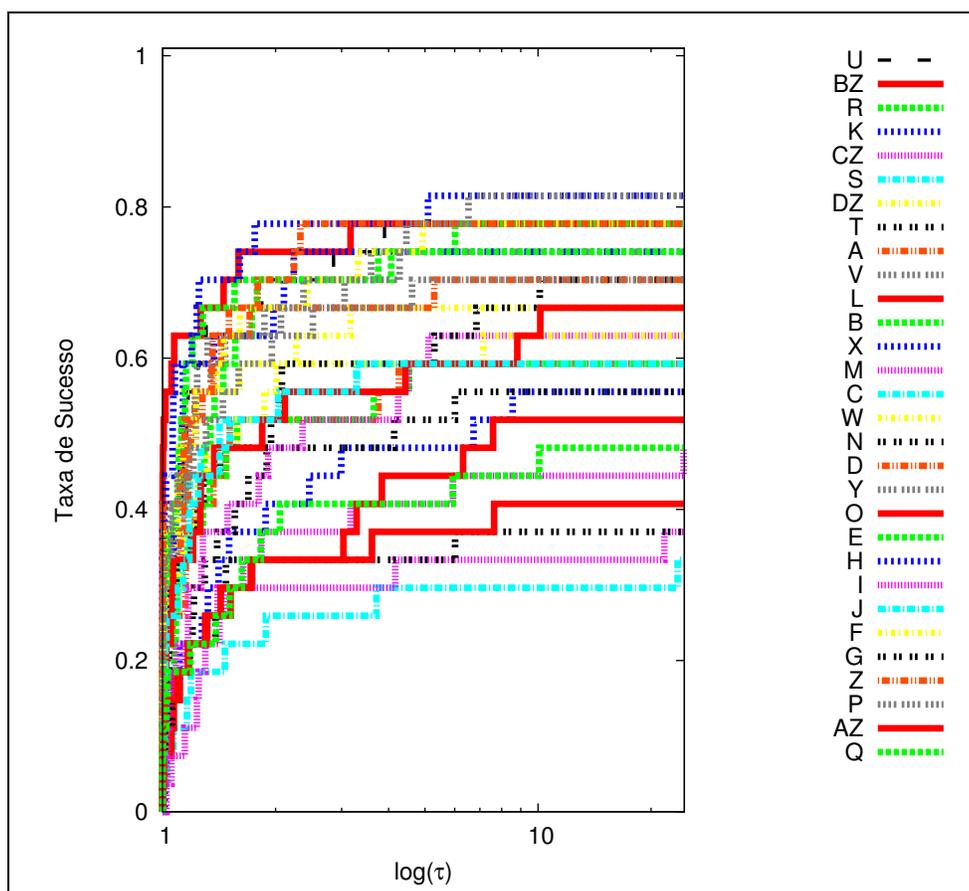


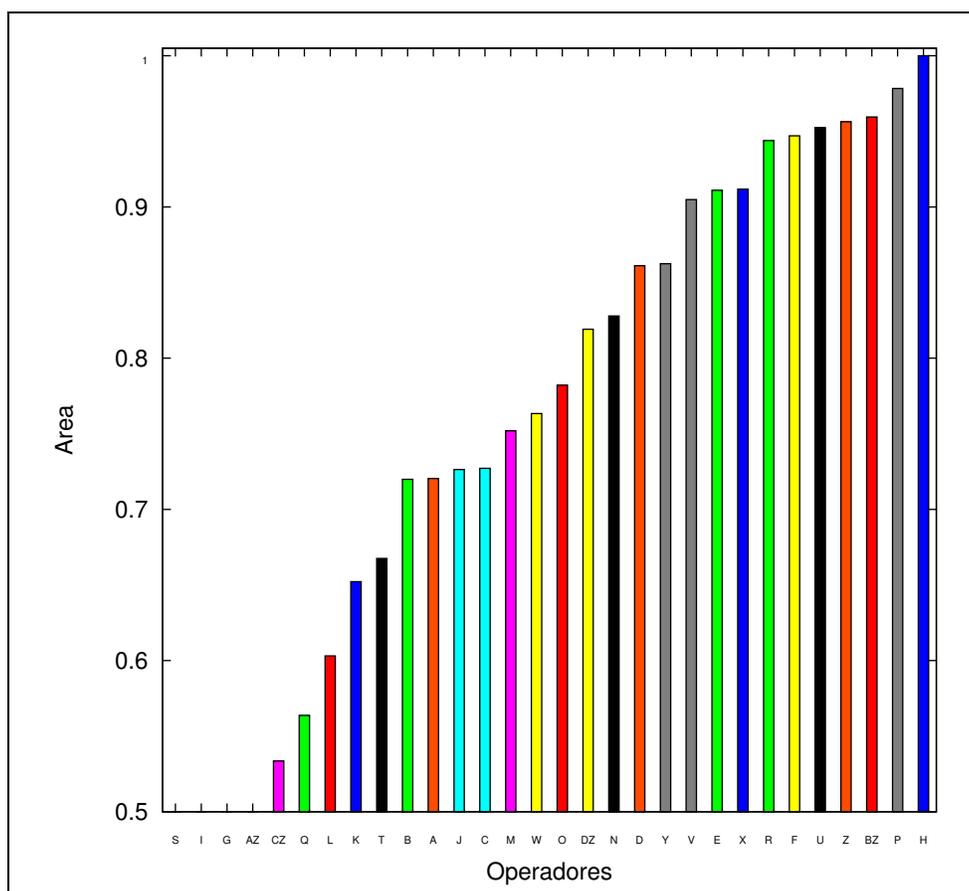
Figura 6.7: Gráfico em Barras - Desempenho da medida **média** para as 30 combinações

Figura 6.8: Gráfico em Linhas - Desempenho da medida **mediana** para as 30 combinações

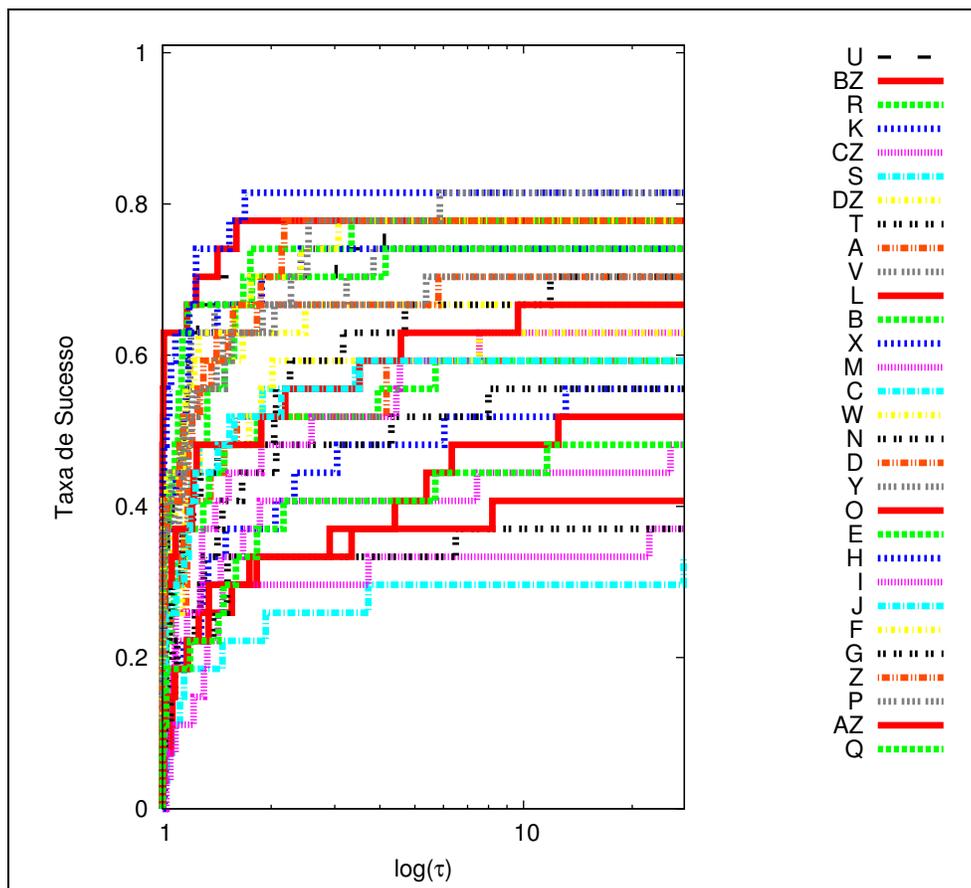


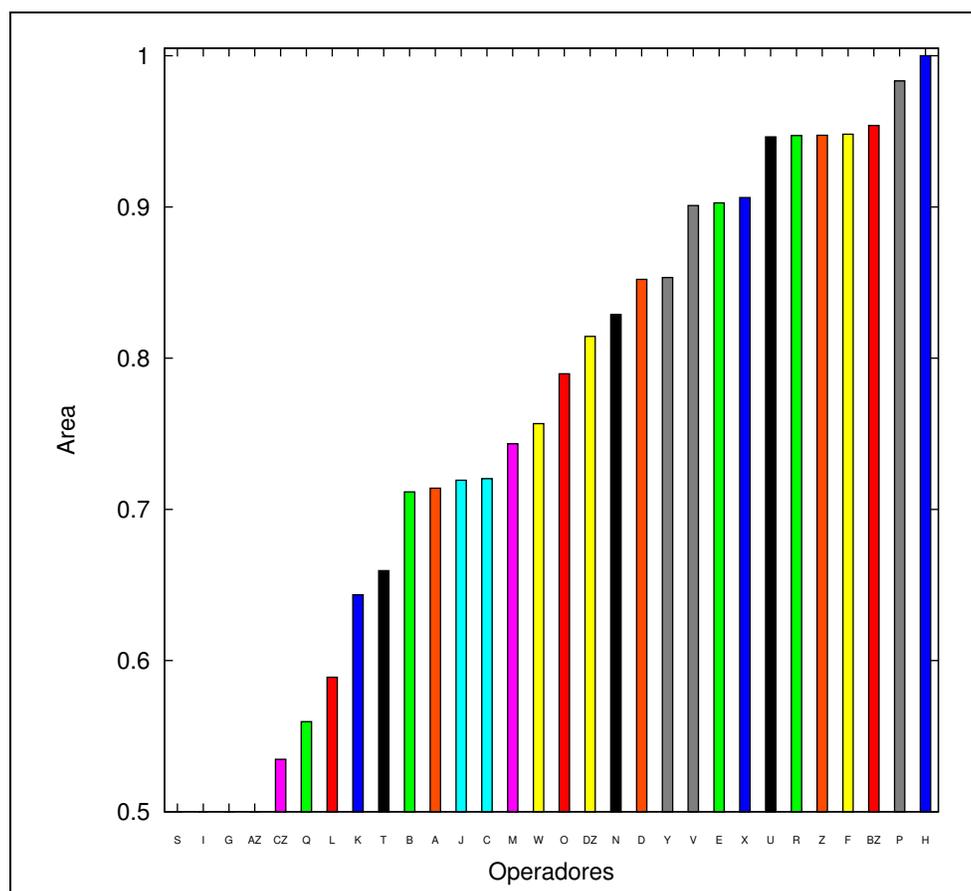
Figura 6.9: Gráfico em Barras - Desempenho da medida **mediana** para as 30 combinações

Figura 6.10: Gráfico em Linhas - Desempenho da medida **pior** para as 30 combinações

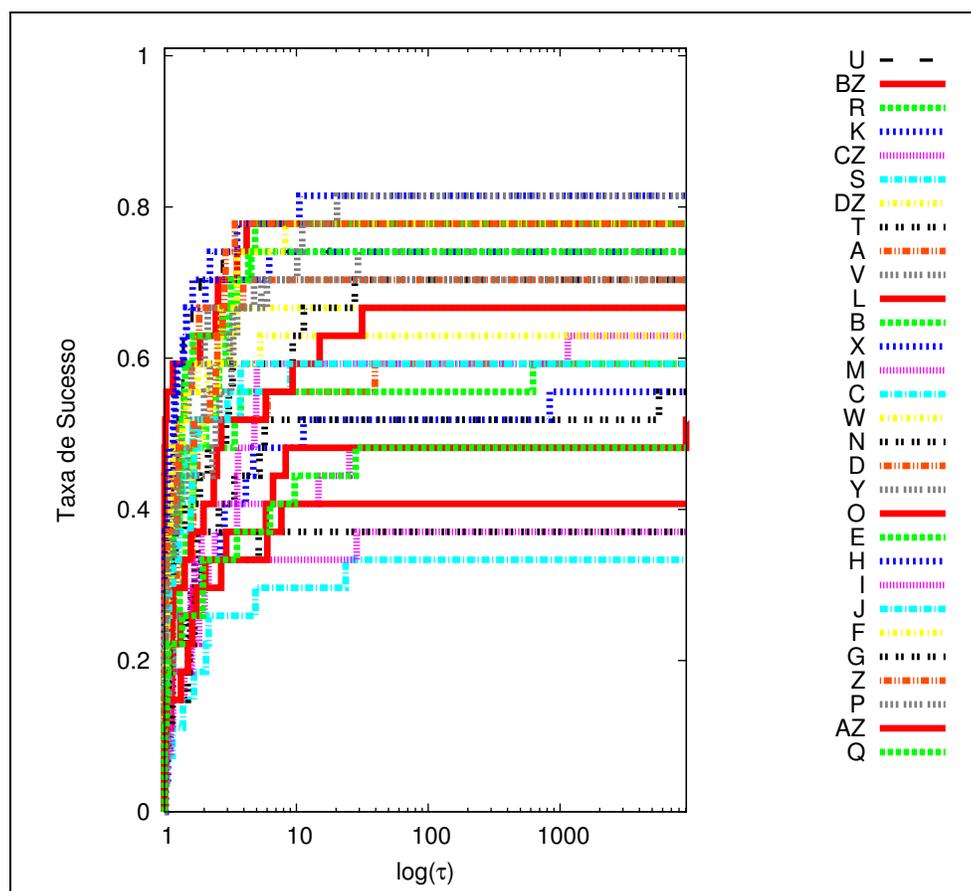


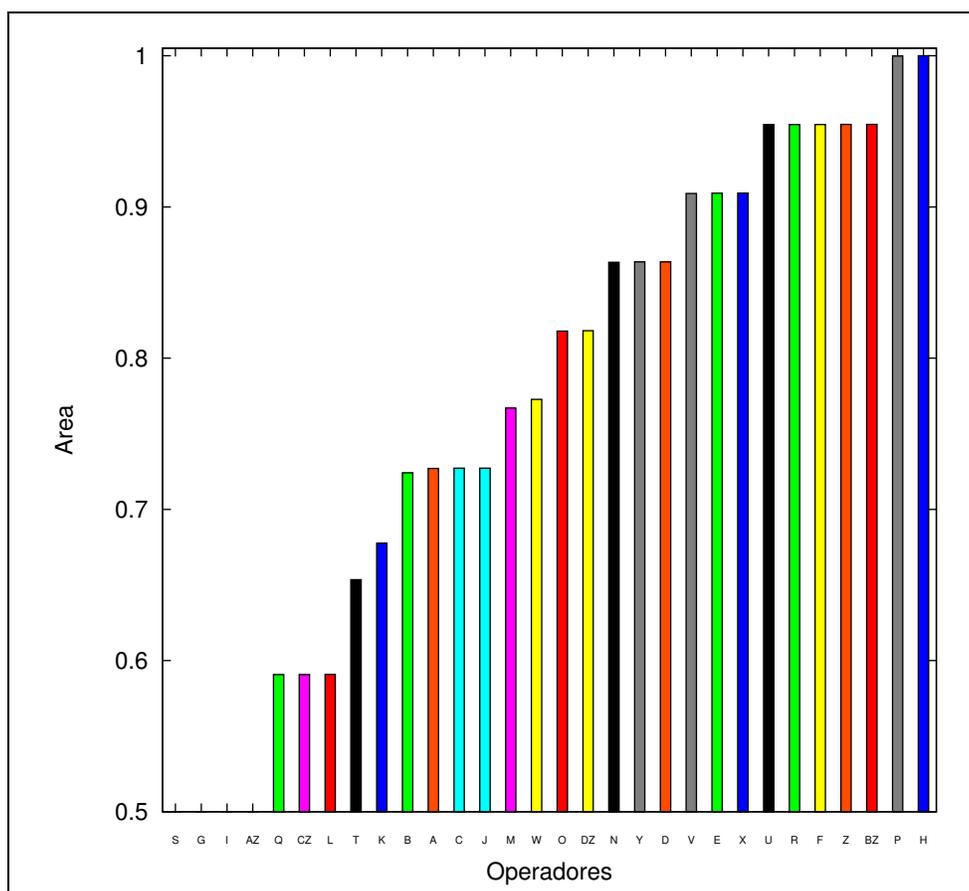
Figura 6.11: Gráfico em Barras - Desempenho da medida **pior** para as 30 combinações

Figura 6.12: Gráfico em Linhas - Desempenho da medida **nma** para as 30 combinações

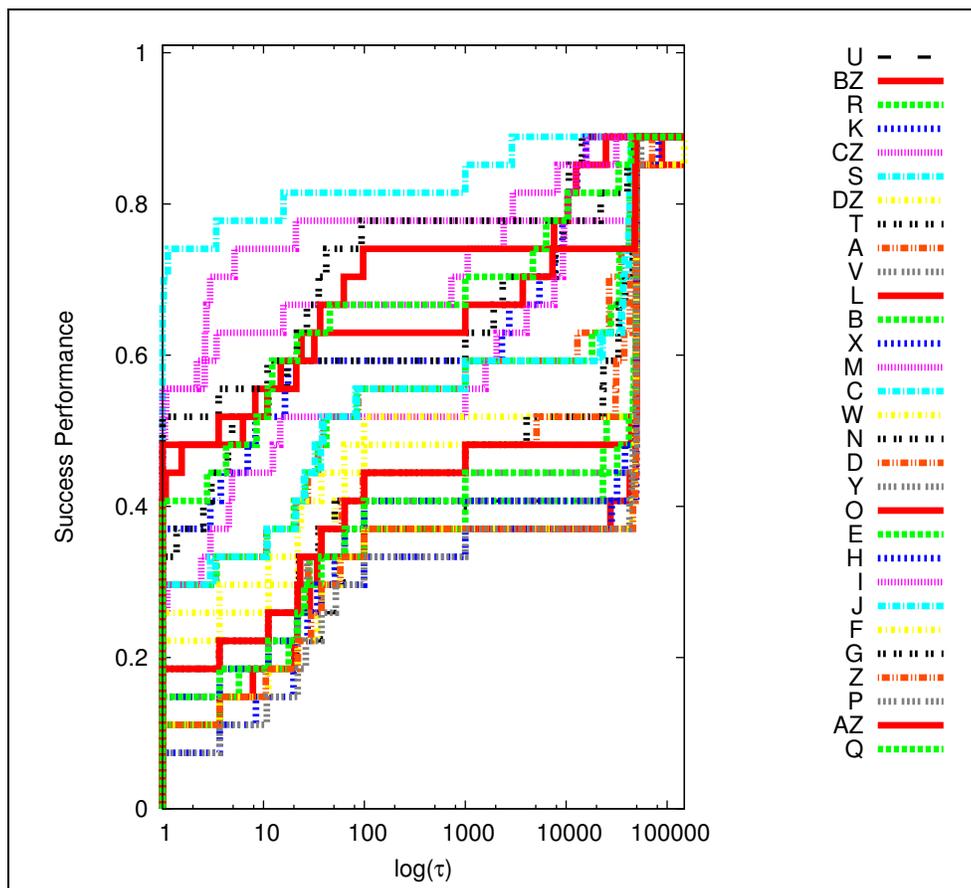
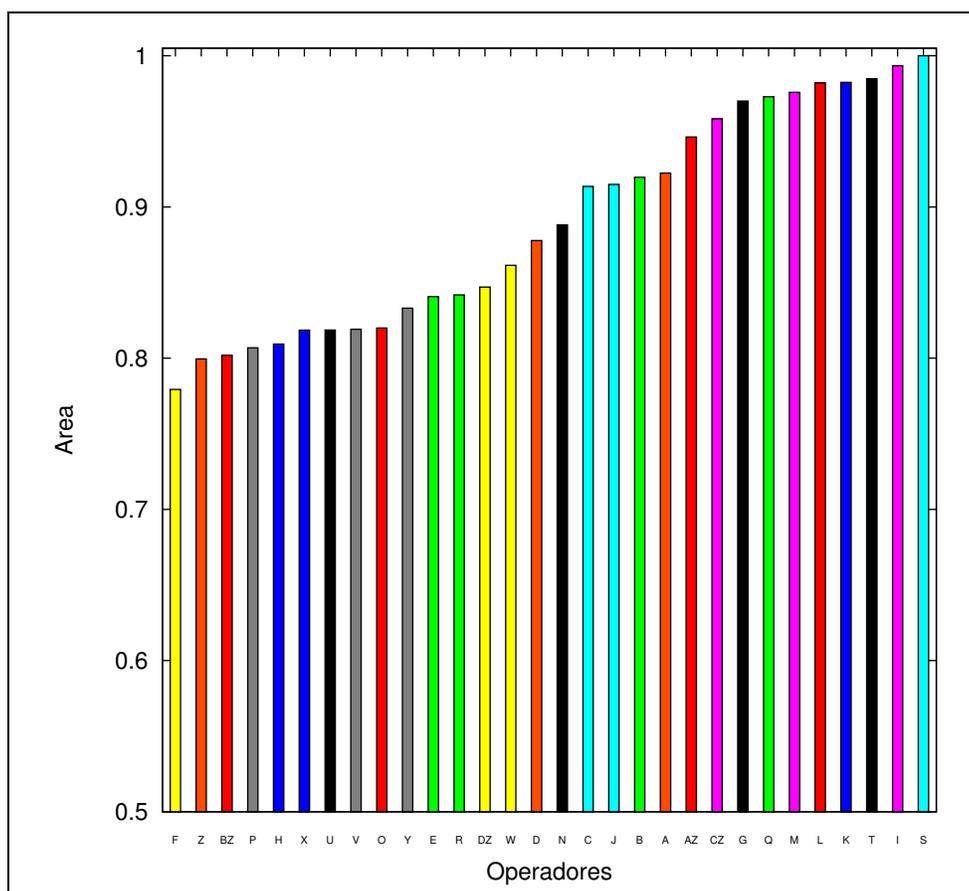


Figura 6.13: Gráfico em Barras - Desempenho da medida **nma** para as 30 combinações

Em seguida, apresenta-se um quadro resumo (6.5) para justificar a escolha da combinação.

Tabela 6.5: Quadro resumo desempenho do APM [1] do G-Suíte em cada combinação

Classificação	melhor	média	mediana	pior
1°	U	H	H	H
2°	Z	P	P	P
3°	BZ	BZ	BZ	BZ
4°	F	Z	F	Z
5°	P	U	Z	F

Observando o quadro, a combinação encontrada, H, é definida e experimentada usando o parâmetro **ninser** variando de 1 a 6 e utilizando a função avaliação. Novamente, o perfil de desempenho mostrou graficamente qual o melhor valor para o parâmetro *ttinser* da combinação H, que teve melhor desempenho.

Na tabela 6.5 observa-se que o resultado da medida **nma** não é válida para a comparação das combinações dos operadores. Isso ocorre porque os melhores valores para o ótimo foram obtidos com uma combinação de operadores diferente da combinação que apresentou melhores resultados no **nma**. Essa medida **nma** só é válida para comparação quando o algoritmo utilizado apresenta melhores desempenhos tanto no **melhor** quanto no **nma**. Quando isso acontecer, o algoritmo estará achando mais rápido o ótimo naquele algoritmo.

As figuras 6.14 a 6.18 mostram o perfil de desempenho, definido pela equação (6.2), para a **melhor, mediana, média, pior, número de funções avaliações** para encontrar as melhores soluções nas 25 rodadas independentes, respectivamente, variando-se o parâmetro **ninser** de 1 a 6.

De acordo com esses gráficos, escolhe-se o melhor valor para o parâmetro **ninser=3**, baseando-se no resultado da **média** e **mediana**. Definido o valor do parâmetro **ninser**, experimentou-se a melhor combinação, H, mutação randômica juntamente com o operador de recombinação Wrigth com taxas de probabilidade de 0.2 e 0.8, respectivamente. Utilizou-se o **ninser=3** e 3 níveis diferentes de avaliação 5000, 50000 e 500000, respectivamente. As tabelas 6.6 a 6.8 apresentam os resultados obtidos.

Combinação do operador de melhor desempenho(H):
 recombinação Wrigth(0.8) e mutação randômica(0.2)

Figura 6.14: Gráfico desempenho da variação do ninser em H: **melhor**

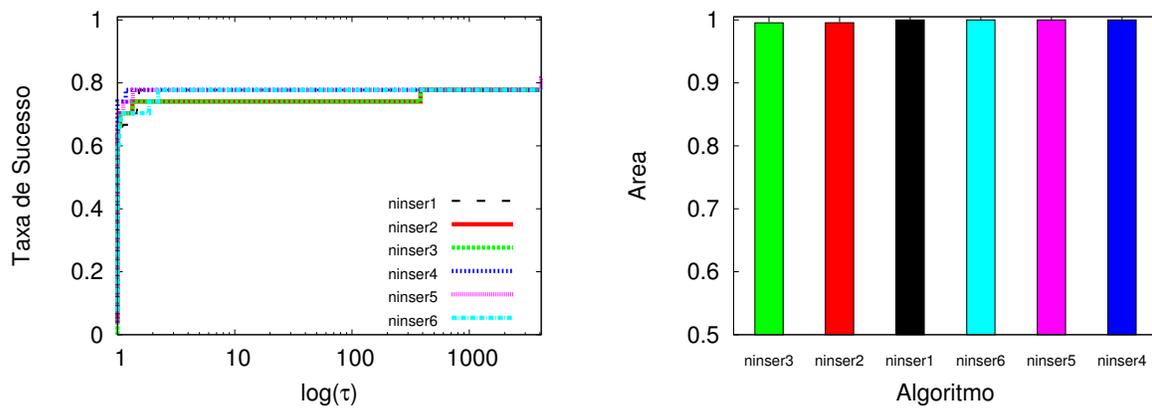


Figura 6.15: Gráfico desempenho da variação do ninser em H: **média**

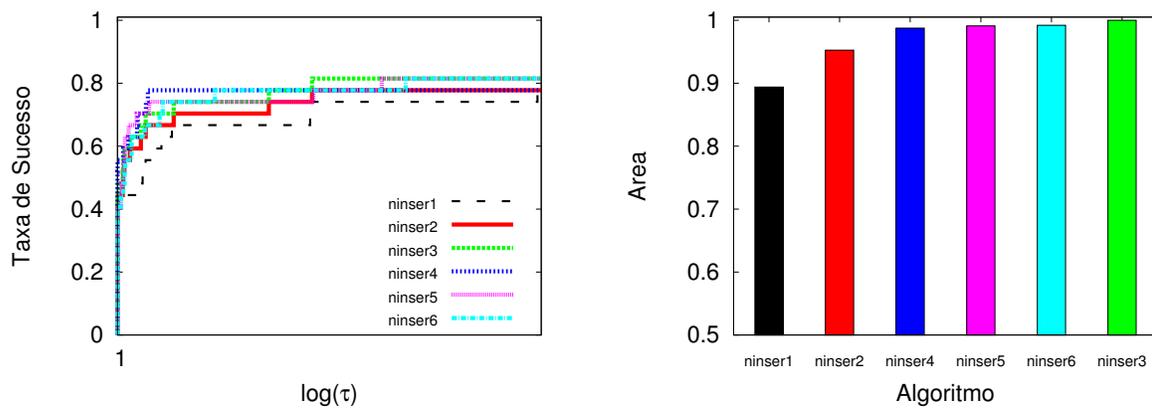


Figura 6.16: Gráfico desempenho da variação do `ninser` em H: **mediana**

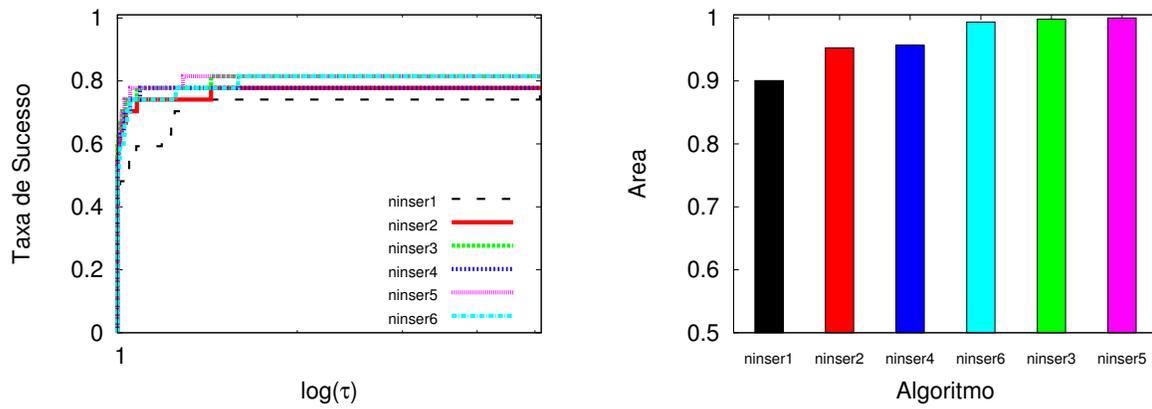


Figura 6.17: Gráfico desempenho da variação do `ninser` em H: **pior**

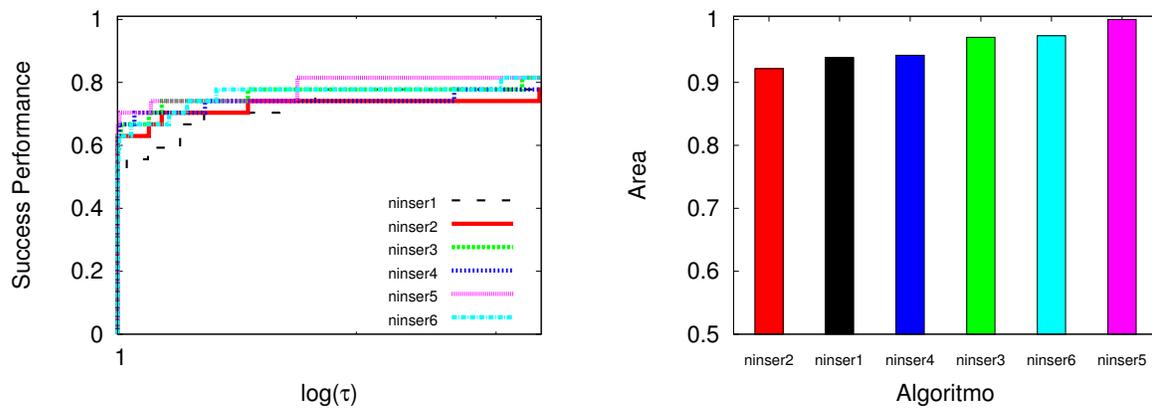


Figura 6.18: Gráfico desempenho da variação do `ninser` em H: **nma**

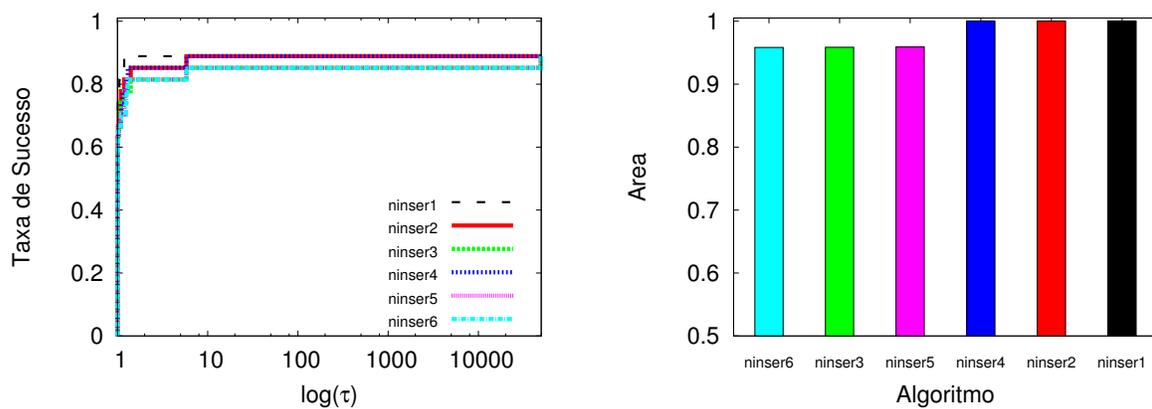


Tabela 6.6: Resultados do melhor operador com 5 000 avaliações

fcn	ótimo	melhor	mediana	média	dv. padrão	pioir	frun	nma
g01	-15.000	-11.924	-8.997	-9.287	1.30E+00	-6.283	25	4626
g02	-0.803	-0.374	-0.262	-0.268	2.99E-02	-0.215	25	4673
g03	-1.000	-0.667	-0.314	-0.293	0.18E+00	-0.005	25	4462
g04	-30665.538	-30664.485	-30659.783	-30657.420	7.76E+00	-30632.020	25	4680
g05	5126.496	5156.331	5202.184	5217.532	4.48E+01	5323.130	25	4776
g06	-6961.813	-6959.661	-6938.059	-6933.640	2.78E+01	-6850.727	20	4641
g07	24.306	60.667	207.254	319.420	2.92E+02	1249.580	25	4842
g08	-0.095	-0.095	-0.095	-0.095	9.58E-05	-0.095	25	4635
g09	680.630	691.310	723.571	745.708	6.39E+01	995.047	25	4726
g10	7049.248	9882.964	12274.450	12617.154	1.86E+03	17275.560	23	4630
g11	0.749	0.749	0.971	0.913	0.11E+00	1.000	25	2602
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.889	-1.822	-1.820	4.37E-02	-1.737	25	4709
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.382	-0.244	-0.244	0.19E+00	-0.107	2	4920
g19	32.655	72.762	225.792	248.054	1.81E+02	716.014	25	4764
g20	0.097	1.184	10.254	11.283	6.75E+00	22.520	15	4813
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-0.045	556.268	495.466	4.38E+02	900.000	8	5594
g24	-5.508	-5.507	-5.507	-5.506	9.43E-04	-5.504	25	4662

Tabela 6.7: Resultados do melhor operador com 50 000 avaliações

fcn	ótimo	melhor	mediana	média	st.padrão	pior	frun	nma
g01	-15.000	-14.999	-12.000	-11.879	2.02E+00	-9.000	25	37456
g02	-0.803	-0.735	-0.661	-0.653	4.63E-02	-0.539	25	49855
g03	-1.000	-0.877	-0.634	-0.601	0.19E+00	-0.083	25	49635
g04	-30665.538	-30665.538	-30665.538	-30665.538	1.11E-11	-30665.538	25	28739
g05	5126.496	5126.498	5126.498	5126.498	1.88E-09	5126.498	25	49586
g06	-6961.813	-6961.810	-6961.810	-6961.810	4.13E-12	-6961.810	25	28213
g07	24.306	24.688	26.731	28.264	3.74E+00	40.429	25	49764
g08	-0.095	-0.095	-0.095	-0.095	3.16E-17	-0.095	25	26554
g09	680.630	680.631	680.683	680.817	0.37E+00	682.388	25	49834
g10	7049.248	7074.699	7260.241	7298.739	2.36E+02	8334.070	25	49730
g11	0.749	0.749	0.751	0.832	0.11E+00	1.000	25	38265
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.311	0.933	0.931	0.34E+00	2.210	25	49791
g14	-47.764	-45.848	-44.361	-43.279	2.63E+00	-37.195	25	49727
g15	961.715	962.476	963.912	965.435	3.04E+00	971.429	25	49291
g16	-1.905	-1.905	-1.905	-1.905	3.15E-06	-1.905	25	49740
g17	8853.539	8865.232	8970.099	9012.058	1.13E+02	9274.089	25	49225
g18	-0.866	-0.865	-0.828	-0.754	0.12E+00	-0.499	25	49738
g19	32.655	36.198	48.607	53.371	1.56E+01	88.186	25	45084
g20	0.097	5.374	9.185	8.443	2.07E+00	10.412	6	82429
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-0.515	149.948	388.433	4.26E+02	900.000	8	31913
g24	-5.508	-5.508	-5.508	-5.508	0.00E+00	-5.508	25	26667

Tabela 6.8: Resultados do melhor operador com 500 000 avaliações

fcn	ótimo	melhor	mediana	média	st.padrão	pior	frun	nma
g01	-15.000	-15.000	-12.000	-11.960	1.98E+00	-9.000	25	45684
g02	-0.803	-0.738	-0.645	-0.646	5.14E-02	-0.541	25	492745
g03	-1.000	-0.907	-0.680	-0.674	0.16E+00	-0.281	25	488010
g04	-30665.538	-30665.538	-30665.538	-30665.538	1.11E-11	-30665.538	25	28662
g05	5126.496	5126.498	5126.498	5126.498	2.99E-12	5126.498	25	113539
g06	-6961.813	-6961.810	-6961.810	-6961.810	4.06E-12	-6961.810	20	28165
g07	24.306	24.580	25.768	26.408	2.14E+00	33.282	25	499844
g08	-0.095	-0.095	-0.095	-0.095	2.83E-17	-0.095	25	47258
g09	680.630	680.630	680.637	680.745	0.36E+00	682.277	25	494099
g10	7049.248	7049.601	7098.679	7156.553	1.01E+02	7402.499	25	499849
g11	0.749	0.749	0.749	0.829	0.11E+00	0.99	25	59832
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.082	0.509	0.534	0.25E+00	0.997	25	497723
g14	-47.764	-45.855	-45.826	-45.462	0.93E+00	-42.238	25	479662
g15	961.715	962.476	962.476	962.479	1.29E-02	962.540	25	236581
g16	-1.905	-1.905	-1.905	-1.905	5.76E-07	-1.905	25	118177
g17	8853.539	8853.533	8931.315	8951.818	9.93E+01	9223.169	25	495376
g18	-0.866	-0.866	-0.865	-0.784	0.11E+00	-0.499	25	499484
g19	32.655	33.601	46.244	51.001	1.72E+01	88.106	25	319980
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-100.043	149.940	234.954	3.62E+02	900.000	10	142446
g24	-5.508	-5.508	-5.508	-5.508	0.00E+00	-5.508	25	26711

Nessa segunda parte dos experimentos, observou-se o desempenho do método da função lagrangeana no algoritmo genético não-geracional. No G-Suíte [48] os parâmetros utilizados foram os mesmos da combinação “H” que obteve melhor desempenho nos experimentos anteriores, mutação randômica(0.2) e recombinação Wright(0.8), **ninser**=3, 50 000 avaliações e 25 rodadas independentes.

Os resultados são mostrados na tabela 6.9 a 6.12. A chamada versão 1, apresentada na tabela 6.9, utiliza o algoritmo de lagrangeano tradicional definido por Hestenes e Rockafellar. Os resultados dessa versão foram interessantes porém, utilizou uma nova versão chamada de versão 2. Nessa versão apresentada na tabela 6.10 e 6.11, utiliza o modelo de lagrangeano desenvolvido nesse trabalho. Os testes foram feitos com a função objetivo baseada em lagrangeano aumentado apresentada em (5.13) e a função objetivo baseada em LA com modificação apresentada em (5.14).

A tabela 6.12 corresponde a versão 2, onde é feita uma seleção dos melhores resultados apresentados com os dois tipos de função lagrangeana propostos (5.13) e (5.14), apresentados nas tabelas 6.10 e 6.11. Sendo na maior parte dos resultados apresentados em 6.12 referentes da função lagrangeana original(tabela 6.11). Os resultados que apresentam “*” na tabela 6.12, correspondem aos resultados obtidos pela tabela 6.10, pois, nesse caso, os resultados foram superiores em relação aos da tabela 6.11.

Em seguida, é feita uma análise do desempenho do algoritmo genético não-geracional [1] utilizando as duas técnicas de tratamento de restrição apresentadas nas seções 4.2 e 5.4. Ou seja, realizou-se uma comparação da técnica de penalização adaptativa com o método do lagrangeano aumentado. Para comparação dessas técnicas utilizou-se os resultados apresentados na tabela 6.7 e 6.12.

Os gráficos apresentados na figura 6.19 a 6.23, obtidos através da ferramenta perfil de desempenho, mostram que o desempenho no resultado das medidas: **melhor**, **mediana** e **nma** nesse algoritmo foi superior quando utilizou-se o método do lagrangeano aumentado em comparação com a técnica de penalização adaptativa. No restante dos resultados, **pior** e **média** a técnica de penalização adaptativa apresentou melhores desempenhos. Nesses resultados a medida **nma** é considerada na comparação das técnicas porque, o método do lagrangeano aumentado apresentou desempenho superior na medida **melhor**. Dessa forma como os melhores valores para o ótimo foram obtidos com o método do lagrangeano aumentado, essa medida **nma** serve como referência, nessa comparação.

Tabela 6.9: Versão 1: Experimentos no G-Suíte com a função lagrangeana

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
G1	-15.0000	-14.999	-12.000	-11.519	1.87E+00	-9.000	25	34490
G2	-0.803	-0.767	-0.669	-0.671	4.28E-02	-0.580	25	49818
G3	-1.000	-	-	-	-	-	-	-
G4	-30665.538	-30423.695	-29796.554	-29850.045	2.50E+02	-29308.827	25	1034
G5	5126.496	5126.550	5151.615	5195.395	9.02E+01	5427.943	25	48155
G6	-6961.813	-6804.384	-6322.044	-5328.469	1.68E+03	-1430.146	15	1649
G7	24.306	24.472	26.044	27.694	3.93E+00	40.076	25	49790
G8	-0.095	-0.095	-0.095	-0.095	3.60E-17	-0.095	25	25608
G9	680.630	680.631	680.729	681.024	0.87E+00	684.955	25	49799
G10	7049.248	-	-	-	-	-	-	-
G11	0.749	0.749	0.750	0.750	7.46E-04	0.752	25	49487
G12	-1.000	-	-	-	-	-	-	-
G13	0.053	0.321	0.629	0.664	0.23E+00	1.004	25	49704
G14	-47.76400	-45.844	-42.138	-41.906	3.25E+00	-36.049	25	49770
G15	961.715	962.476	962.476	962.525	0.11E+00	963.034	25	45577
G16	-1.905	-1.905	-1.905	-1.905	8.90E-06	-1.905	25	49651
G17	8853.539	8853.881	8947.040	8966.977	9.05E+01	9236.030	25	44288
G18	-0.866	-0.862	-0.680	-0.700	0.14E+00	-0.498	25	41710
G19	32.655	34.013	40.499	44.308	9.32E+00	70.500	25	49261
G20	0.097	-	-	-	-	-	-	-
G21	193.724	-	-	-	-	-	-	-
G22	236.430	-	-	-	-	-	0	0
G23	-400.055	900.000	900.000	900.000	0.00E+00	900.000	3	8962
G24	-5.508	-5.508	-5.508	-5.508	1.35E-09	-5.508	25	24544

Tabela 6.10: Versão 2: Experimentos no G-Suíte com a função lagrangeana modificada

fcn	ótimo	melhor	mediana	média	dv. padrão	pioir	frun	nma
g01	-15.000	-14.999	-12.000	-11.639	1.65E+00	-9.000	25	38116
g02	-0.804	-0.697	-0.567	-0.460	0.20E+00	-0.179	25	32255
g03	-1.001	-0.501	-7.920E-15	-0.049	0.14E+00	-3.79E-23	14	43971
g04	-30665.539	-30665.538	-30645.782	-30589.175	1.74E+02	-29973.432	25	19691
g05	5126.497	5126.498	5126.498	5126.498	5.499E-09	5126.498	25	49615
g06	-6961.814	-6018.302	-3687.189	-3852.098	1.33E+03	-2236.188	18	1044
g07	24.306	24.508	27.384	29.440	4.75E+00	39.750	25	49773
g08	-0.096	-0.095	-0.095	-0.095	2.953E-05	-0.095	25	26315
g09	680.630	680.634	680.729	680.953	0.73E+00	684.230	25	49814
g10	7049.248	7058.140	7433.406	7771.449	8.50E+02	9794.341	16	49791
g11	0.750	0.749	0.999	0.918	0.11E+00	1.000	25	21236
g12	-1.0000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.054	0.054	0.320	0.362	0.23E+00	0.925	25	45989
g14	-47.765	-45.830	-42.876	-42.836	2.70E+00	-37.283	25	45165
g15	961.715	962.476	963.664	964.654	2.76E+00	971.637	25	46425
g16	-1.905	-1.905	-1.905	-1.905	8.74E-07	-1.905	25	49720
g17	8853.540	8857.843	8993.884	9002.611	8.80E+01	9271.317	25	41821
g18	-0.866	-0.864	-0.664	-0.687	0.15E+00	-0.393	25	49717
g19	32.656	34.394	40.633	45.451	1.61E+01	111.002	25	49141
g20	0.097	-	-	-	-	-	-	-
g21	193.725	-	-	-	-	-	-	-
g22	236.431	-	-	-	-	-	-	-
g23	-400.055	-218.310	900.000	512.063	4.72E+02	900.000	9	21996
g24	-5.508	-5.508	-5.508	-5.505	6.22E-03	-5.477	25	31193

Tabela 6.11: Versão 2: Experimentos no G-Suíte com a função lagrangeana original

fcn	ótimo	melhor	mediana	média	dv. padrão	pioir	frun	nma
g01	-15.000	-14.999	-14.999	-14.319	1.24E+00	-12.000	25	48300
g02	-0.804	-0.745	-0.621	-0.476	0.25E+00	-0.179	25	27034
g03	-1.001	-0.940	-7.470E-14	-0.150	-2.58E-34	0.331	17	43430
g04	-30665.539	-30665.538	-30665.538	-30593.599	2.48E+02	-29763.871	25	31989
g05	5126.497	5126.498	5126.500	5126.501	4.61E-03	5126.518	25	39461
g06	-6961.814	-6018.302	-3619.332	-3745.041	1.29E+03	-2236.188	18	1060
g07	24.306	24.835	26.739	27.894	3.35E+00	37.878	25	49301
g08	-0.096	-0.095	-0.095	-0.095	8.651E-06	-0.095	25	26836
g09	680.630	680.630	680.652	680.703	0.11E+00	681.104	25	47128
g10	7049.248	-	-	-	-	-	-	-
g11	0.750	0.749	0.749	0.749	3.10E-08	0.749	25	27747
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.054	0.054	0.077	0.219	0.18E+00	0.481	25	44372
g14	-47.765	-45.844	-44.401	-44.151	1.61E+00	-41.291	25	39978
g15	-961.715	962.476	962.997	963.984	2.26E+00	969.544	25	43247
g16	-1.905	-1.905	-1.905	-1.905	9.71E-06	-1.905	25	42954
g17	8853.540	8889.609	8959.531	8988.792	9.13E+01	9264.597	25	39540
g18	-0.866	-0.864	-0.666	-0.693	0.14E+00	-0.489	25	43808
g19	32.656	33.882	40.295	43.167	9.28E+00	70.500	25	48709
g20	-	-	-	-	-	-	-	-
g21	-	-	-	-	-	-	-	-
g22	-	-	-	-	-	-	-	-
g23	-400.055	-0.027	151.715	124.105	8.50E+01	193.020	4	12687
g24	-5.508	-5.508	-5.508	-5.508	8.48E-10	-5.508	25	27497

Tabela 6.12: Versão 2: Experimentos no G-Suíte com os melhores resultados das tabelas 6.10 e 6.11

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.999	-14.999	-14.319	1.24E+00	-12.000	25	48300
g02	-0.804	-0.745	-0.621	-0.476	0.25E+00	-0.179	25	27034
g03	-1.001	-0.940	-7.470E-14	-0.150	0.33E+00	-2.58E-034	17	43430
g04	-30665.539	-30665.538	-30665.538	-30593.599	2.48E+02	-29763.871	25	31989
g05	5126.497	5126.498	5126.500	5126.501	4.61E-03	5126.518	25	39461
g06	-6961.814	-6018.302	-3619.332	-3745.041	1.29E+03	-2236.188	18	1060
g07*	24.306	24.508	27.384	29.440	4.75E+00	39.750	25	49773
g08	-0.096	-0.095	-0.095	-0.095	8.651E-06	-0.095	25	26836
g09	680.630	680.630	680.652	680.703	0.11E+00	681.104	25	47128
g10*	7049.248	7058.140	7433.406	7771.449	8.50E+02	9794.341	16	49791
g11	0.750	0.749	0.749	0.749	3.102E-08	0.749	25	27747
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.054	0.054	0.077	0.219	0.18E+00	0.481	25	44372
g14	-47.765	-45.844	-44.401	-44.151	1.61E+00	-41.291	25	39978
g15	-961.715	962.476	962.997	963.984	2.26E+00	969.544	25	43247
g16	-1.905	-1.905	-1.905	-1.905	9.71E-06	-1.905	25	42954
g17*	8853.540	8857.843	8993.884	9002.611	8.80E+01	9271.317	25	41821
g18	-0.866	-0.864	-0.666	-0.693	0.14E+00	-0.489	25	43808
g19	32.656	33.882	40.295	43.167	9.28E+00	70.500	25	48709
g20	-	-	-	-	-	-	-	-
g21	-	-	-	-	-	-	-	-
g22	-	-	-	-	-	-	-	-
g23*	-400.055	-218.310	900.000	512.063	4.72E+02	900.000	9	21996
g24	-5.508	-5.508	-5.508	-5.508	8.48E-10	-5.508	25	27497

Desempenho do algoritmo genético não-geracional utilizando as técnicas de tratamento da restrição:
penalização adaptativa e lagrangeano aumentado

Figura 6.19: Comparação das duas técnicas: Desempenho do **melhor**

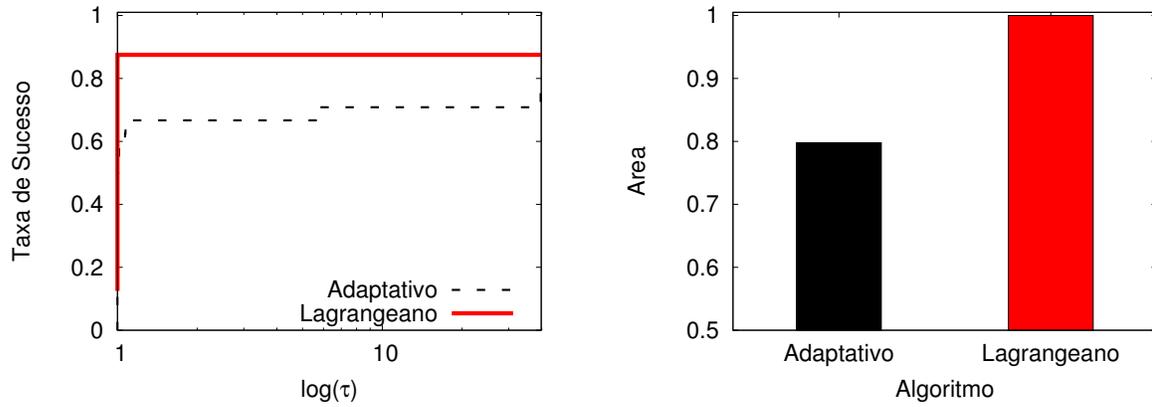


Figura 6.20: Comparação das duas técnicas: Desempenho da **média**

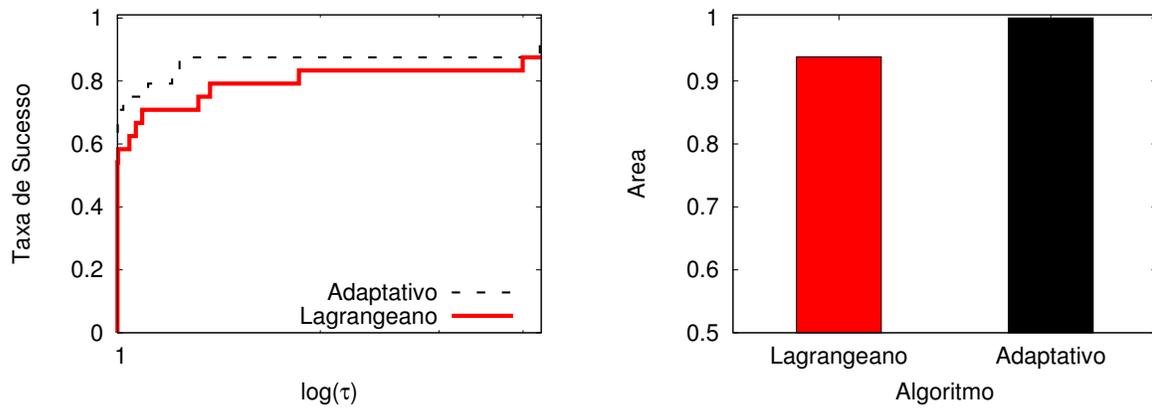


Figura 6.21: Comparação das duas técnicas: Desempenho da **mediana**

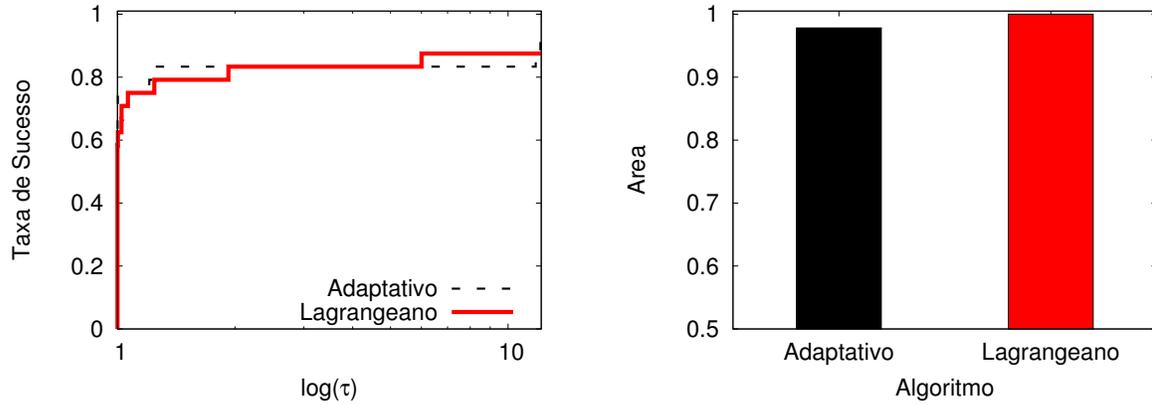


Figura 6.22: Comparação das duas técnicas: Desempenho do **pior**

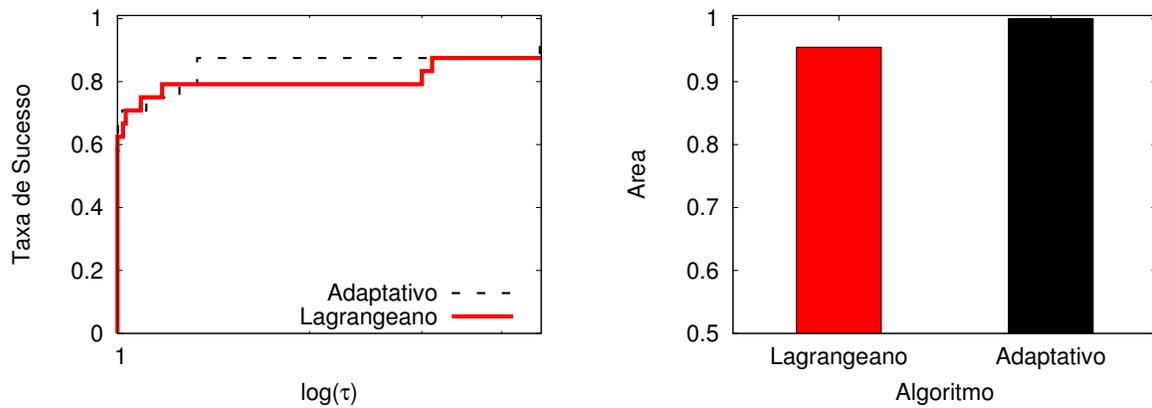
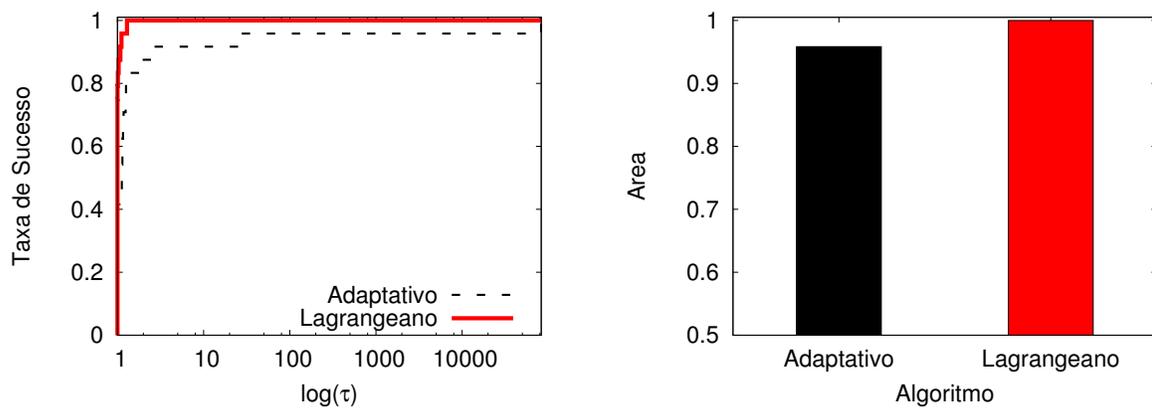


Figura 6.23: Comparação das duas técnicas: Desempenho do **nma**



6.3 Experimentos em problemas da engenharia

Baseados em alguns problemas da engenharia foi analisado o desempenho do algoritmo proposto [1] utilizando as duas técnicas de tratamento de restrição: o modelo adaptativo e o método do lagrangeano aumentado. Obtidos os resultados desses dois experimentos comparou-se com outras técnicas presentes na literatura. As técnicas utilizadas na comparação foram: Algoritmos de hibridização de sistemas imunológicos com GA(AIS-GA e AIS-GA^C) proposto em [81]; um algoritmo genético geracional utilizando o método de penalização adaptativa com “codificação binária”(APM^{gba}) apresentado em [61]; uma hibridização de um GA com sistemas imunológicos artificiais(AIS-GA^H) utilizada em [82]; Um algoritmo utilizando o raqueamento estocástico SR(*Stochastic Ranking technique*) proposto em [60]; Um algoritmo utilizando a estratégia evolucionista(ES-Coello) apresentada por [83] e por último a utilização de um GA(GAOS-Erbatur) proposto em [84].

Os parâmetros utilizados no APM^{gba} [82] e no Método SR [60] foram: população com tamanho de 100 indivíduos, uma taxa de mutação igual a 0.004. Já no AIS-GA^H o tamanho da população foi 50 e a taxa de mutação 0.01. No AIS a taxa de mutação foi 0.03 no seu valor máximo. Em todos os trabalhos, as técnicas utilizaram o código binário de Gray com 25 bits para cada variável contínua, e a taxa de probabilidade do operador de recombinação igual a 0.9.

Nessa dissertação, os testes foram aplicados em todos os problemas, em um algoritmo genético não-geracional utilizando as técnicas propostas nas seções 4.2 e 5.4. Foram utilizadas a notação de APM^{ngra} para o algoritmo genético não-geracional com a técnica de penalização adaptativa e APM^{ngrl} para o algoritmo genético não-geracional utilizando o método do lagrangeano aumentado. Tanto em APM^{ngrl} quanto no APM^{ngra} foi utilizada codificação real. Nos experimentos foram obtidos os valores do **melhor**, **média**, **mediana**, **desvio padrão**, **pior** e o **frun** que corresponde ao número de rodadas de cada solução factível.

Anteriormente, foram apresentados no Cilamce 2010 [85] os resultados do APM^{rc}, correspondentes a testes com o algoritmo genético não-geracional utilizando o método de penalização adaptativa, porém com outras combinações de operadores. O APM^{rc} [85] utilizou codificação real com um esquema de seleção de ranqueamento linear e utilizou os operadores: mutação randômica, mutação não uniforme, mutação de Muhlenbein, recombinação discreta com pais múltiplos e recombinação SBX de Deb. Isso foi feito

sem nenhum ajuste de parâmetros. Todos os operadores foram utilizados com a mesma taxa de probabilidade(0.2) e η foi definido como 2 em SBX. Nesse trabalho utilizou-se nos experimentos esse mesmo APM^{rc} [85] porém, agora, o testes foram realizados com o melhor operador obtido nos experimentos realizados na seção 6.2. Os operadores de mutação randômica e de recombinação Wrigth foram utilizados com taxa de probabilidade 0.2 e 0.8, respectivamente.

6.3.1 Projeto de tração/compressão da mola

O propósito desse projeto é minimizar o volume V da tração/compressão da mola. As variáveis de projeto são o número de espirais ativas das molas ($N = x_1 \in [2, 15]$), o diâmetro do enrolamento($D = x_2 \in [0.25, 1.3]$), e o diâmetro do fio($d = x_3 \in [0.05, 2]$). O volume e as restrições mecânicas são dadas por:

$$\begin{aligned}
 V(x) &= (x_1 + 2)x_2x_3^2 \\
 g_1(x) &= 1 - \frac{x_2^3x_1}{71785x_3^4} \leq 0 \\
 g_2(x) &= \frac{4x_2^2 - x_3x_2}{12566(x_2x_3^3 - x_3^4)} + \frac{1}{5108x_3^2} - 1 \leq 0 \\
 g_3(x) &= 1 - \frac{140.45x_3}{x_2^2x_1} \leq 0 \\
 g_4(x) &= \frac{x_2 + x_3}{1.5} - 1 \leq 0
 \end{aligned}$$

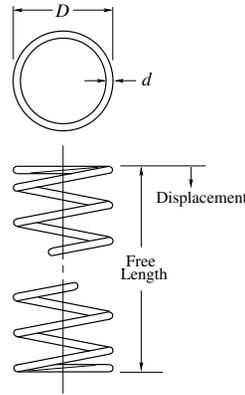
onde

$$2 \leq x_1 \leq 15 \quad 0.25 \leq x_2 \leq 1.3 \quad 0.05 \leq x_3 \leq 2$$

Tabela 6.13: Resultados do Projeto compressão/tração da mola.

	melhor	mediana	média	desvio padrão	pior	fr
AIS-GA [81]	0.012668	—	0.013481	—	0.016155	—
AIS-GA ^C [81]	0.012666	—	0.012974	—	0.013880	—
AIS-GA ^H [82]	0.012666	0.012892	0.013131	6.28E - 4	0.015318	50
APM ^{gba} [61]	0.012684	0.013575	0.014022	1.47E - 3	0.017794	50
SR [60]	0.012679	0.013655	0.013993	1.27E - 3	0.017796	50
APM ^{ngra}	0.012678	0.012678	0.012767	1.45E - 4	0.014533	50
APM ^{ngrl}	0.012678	0.012678	0.012682	1.30E - 5	0.012733	50

Figura 6.24: A tração/compressão da Mola



Os experimentos foram realizados com 200 indivíduos na população e o número de funções avaliações foi 36.000. Os resultados são comparados na tabela 6.13. Nessa tabela, a medida **fr** corresponde ao número da rodada onde o ótimo foi encontrado. Os melhores resultados são encontrados em AIS-GA^C, AIS-GA^H e no APM^{ngra} e APM^{ngrl}. O melhor volume final da comparação foi igual a 0.012666. A tabela 6.14 mostra que os valores são encontrados para as variáveis de projeto correspondendo as melhores soluções quais são todas factíveis.

Tabela 6.14: Variáveis de Projeto encontradas na tração/compressão da mola

	x_1	x_2	x_3	V
AIS-GA [81]	11.85217	0.347475	0.051302	0.012668
AIS-GA ^C [81]	11.32955	0.356032	0.051661	0.012666
AIS-GA ^H [82]	11.66119	0.3505298	0.051430	0.012666
APM ^{gba} [61]	12.07074	0.344304	0.051168	0.0126838
SR [60]	11.37579	0.355485	0.357848	0.051638
APM ^{ngra}	11.23705	0.358485	0.517359	0.012679
APM ^{ngrl}	11.29120	0.356920	0.051697	0.012678

6.3.2 Projeto redutor de velocidade

O objetivo foi minimizar o peso W do redutor de velocidade. As variáveis de projeto são a largura da face ($b = x_1 \in [2.6, 3.6]$), o módulo dos dentes ($m = x_2 \in [0.7, 0.8]$), o número de dentes do pinhão ($n = x_3 \in [17, 28]$), o comprimento do eixo 1 entre os rolamentos ($l_1 = x_4 \in [7.3, 8.3]$), o comprimento do eixo entre os dois rolamentos ($l_2 = x_5 \in [7.8, 8.3]$), o

diâmetro do eixo 1 ($d_1 = x_6 \in [2.9, 3.9]$), e, finalmente, o diâmetro do eixo 2 ($d_2 = x_7$). A variável x_3 é inteira e todas as outras são contínuas. O peso e as restrições mecânicas são dadas por:

$$\begin{aligned}
 W &= 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
 &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \\
 &\quad + 0.7854(x_4x_6^2 + x_5x_7^2) \\
 g_1(x) &= 27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1 \\
 g_2(x) &= 397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1 \\
 g_3(x) &= 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1 \\
 g_4(x) &= 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1 \\
 g_5(x) &= \frac{1}{0.1x_6^3} \left[\left(\frac{745x_4}{x_2x_3} \right)^2 + \{16.9\}10^6 \right]^{0.5} \leq 1100 \\
 g_6(x) &= \frac{1}{0.1x_7^3} \left[\left(\frac{745x_5}{x_2x_3} \right)^2 + (157.5)10^6 \right]^{0.5} \leq 850 \\
 g_7(x) &= x_2x_3 \leq 40 \quad g_8(x) = x_1/x_2 \geq 5 \\
 g_9(x) &= x_1/x_2 \leq 12 \quad g_{10}(x) = (1.5x_6 + 1.9)x_4^{-1} \leq 1 \\
 g_{11}(x) &= (1.1x_7 + 1.9)x_5^{-1} \leq 1
 \end{aligned}$$

A tabela 6.15 apresenta a comparação dos resultados encontrados pelo algoritmo proposto e outros da literatura. O número de funções avaliações foram definidos igual a 36000 com 200 indivíduos na população. Os melhores valores para o peso foram encontrados pelo APM^{ngra}, APM^{ngrl} e AIS-GA^C. A tabela 6.16 apresenta o valor final das variáveis de projeto além disso, todas as soluções são factíveisis.

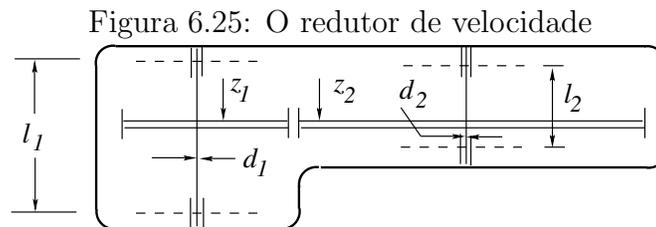


Tabela 6.15: Resultados do projeto redutor de velocidade

	melhor	mediana	média	desvio padrão	pior	fr
ES-Coello [83]	3025.0051	—	3088.7778	—	3078.5918	—
AIS-GA* [81]	2996.3494	2996.356	2996.3643	$4.35E - 3$	2996.6277	50
AIS-GA ^C * [81]	2996.3484	2996.3484	2996.3484	$1.46E - 6$	2996.3486	50
AIS-GA ^H [82]	2996.3483	2996.3495	2996.3501	$7.45E - 3$	2996.3599	50
APM ^{gba} [61]	2996.3482	2996.3482	3033.8807	$1.10E + 2$	3459.0948	19
SR [60]	2996.3483	2996.3488	2996.3491	$1.01E - 3$	2996.3535	50
APM ^{ngra}	2996.3481	2996.3481	3053.7382	$3.23E + 2$	5205.0287	50
APM ^{ngrl}	2996.3481	2996.3482	3003.1981	$4.69E - 1$	3318.2931	47

Tabela 6.16: Variáveis de projeto para as melhores soluções do redutor de velocidade

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	W
ES-Coello [83]	3.506163	0.7008	17	7.460181	7.962143	3.3629	5.308949	3025.0051
AIS-GA* [81]	3.500001	0.7000	17	7.300019	7.800013	3.3502	5.286684	2996.3494
AIS-GA ^C * [81]	3.500000	0.7000	17	7.300001	7.800000	3.3502	5.286684	2996.3484
AIS-GA ^H [82]	3.500001	0.7000	17	7.300008	7.800001	3.3502	5.286683	2996.3483
APM ^{gba} [61]	3.500000	0.7000	17	7.300000	7.800000	3.3502	5.286683	2996.3482
SR [60]	3.500000	0.70000	17	7.300001	7.800001	3.3502	5.286683	2996.3483
APM ^{ngra}	3.500000	0.7000	17	7.300000	7.800000	3.3502	5.286683	2996.3482
APM ^{ngrl}	3.500000	0.7000	17	7.300000	7.800000	3.3502	5.286683	2996.3481

6.3.3 Projeto da viga soldada

Esse projeto visa minimizar o custo $C(h, l, t, b)$ da viga onde $h \in [0.125, 10]$, and $0.1 \leq l, t, b \leq 10$. A função objetivo e restrições são:

$$C(h, l, t, b) = 1.10471h^2l + 0.04811tb(14.0 + l)$$

$$g_1(\tau) = 13,600 - \tau \geq 0 \quad g_2(\sigma) = 30,000 - \sigma \geq 0$$

$$g_3(b, h) = b - h \geq 0 \quad g_4(P_c) = P_c - 6,000 \geq 0$$

$$g_5(\delta) = 0.25 - \delta \geq 0$$

As expressões para τ , σ , P_c , e δ são dadas por:

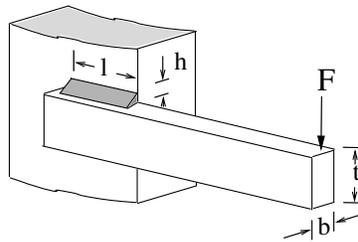
$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'\tau''/\alpha} \quad \tau' = \frac{6000}{\sqrt{2hl}}$$

$$\alpha = \sqrt{0.25(l^2 + (h + t)^2)} \quad \sigma = \frac{504000}{t^2b}$$

$$P_c = 64746.022(1 - 0.0282346t)tb^3 \quad \delta = \frac{2.1952}{t^3b}$$

$$\tau'' = \frac{6000(14 + 0.5l)\alpha}{2(0.707hl(l^2/12 + 0.25(h + t)^2))}$$

Figura 6.26: Viga soldada



A tabela 6.17 mostra uma comparação dos resultados onde os melhores valores encontrados (custo final igual a 2.38113) correspondente ao APM^{ngra} e APM^{ngrl}. Esse último proposto aqui em nosso trabalho. A tabela 6.18 mostra as variáveis de projeto correspondentes as melhores soluções encontradas por cada técnica. Todas as soluções são factíveis e o número de funções avaliação foram definidas como 320.000 com 200 indivíduos na população.

Tabela 6.17: Valores encontrados para o custo do Projeto da Viga Soldada

	melhor	mediana	média	desvio padrão	pior	fr
AIS-GA [81]	2.38125	–	2.59303	–	3.23815	–
AIS-GA ^C [81]	2.38122	–	2.38992	–	2.41391	–
AIS-GA ^H [82]	2.38335	2.92121	2.99298	2.02E – 1	4.05600	50
APM ^{gba} [61]	2.38144	3.27244	3.49560	9.09E – 1	5.94803	50
SR [60]	2.59610	4.21812	4.33259	1.29E + 0	10.1833	50
APM ^{ngra}	2.38113	2.38113	2.63030	1.34E + 0	11.7472	50
APM ^{ngrl}	2.38113	2.38113	2.99895	2.96E + 0	20.6719	50

Tabela 6.18: Variáveis de projeto das melhores soluções no projeto da viga soldada

	AIS-GA	AIS-GA ^C	AIS-GA ^H	APM ^{gbac}	SR	APM ^{ngra}	APM ^{ngrl}
<i>h</i>	0.2443243	0.2443857	0.2434673	0.2442419	0.2758192	0.244395	0.244368
<i>l</i>	6.2201996	6.2183037	6.2507296	6.2231189	5.0052613	6.218086	6.218605
<i>t</i>	8.291464	8.291165	8.291472	8.291471	8.626110	8.291043	8.291473
<i>b</i>	0.2443694	0.2443875	0.2443690	0.2443690	0.2758194	0.244395	0.244368
Custo	2.381246	2.38122	2.38335	2.38144	2.59610	2.38113	2.38113

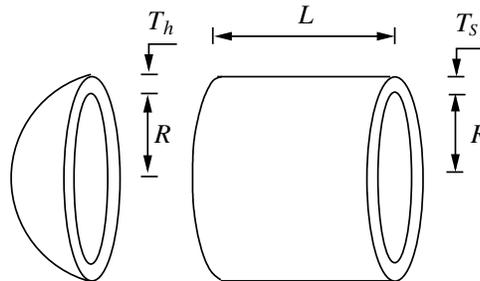
6.3.4 O Projeto do vaso de pressão

O problema corresponde a minimização do peso do vaso de pressão cilíndrico com duas cabeças esféricas. Existem quatro variáveis de projeto (em polegadas) a espessura do vaso de pressão (T_s), a espessura da cabeça (T_h), o raio do interior do vaso (R) e o tamanho do componente cilíndrico (L). Desde que existam duas variáveis discretas (T_s e T_h) e duas variáveis contínuas (R e L), alguns problemas tem restrições não lineares misturando casos de otimização contínuas e discretas. Os limites das variáveis de projeto são: $0.0625 \leq T_s, T_h \leq 5$ (em valores discretos constantes 0.0625) e $10 \leq R, L \leq 200$. O peso, deve ser minimizado, e as restrições são dadas por:

$$\begin{aligned}
W(T_s, T_h, R, L) &= 0,6224T_sT_hR + \\
&+ 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R \\
g_1(T_s, R) &= T_s - 0.0193R \geq 0 \\
g_2(T_h, R) &= T_h - 0.00954R \geq 0 \\
g_3(R, L) &= \pi R^2L + 4/3\pi R^3 - 1,296,000 \geq 0 \\
g_4(L) &= -L + 240 \geq 0
\end{aligned}$$

As duas primeiras restrições estabelecem um limite inferior para as razões T_s/R e T_h/R , respectivamente. A terceira restrição corresponde a um limite inferior para o volume do vaso e por último, a restrição tem um limite superior para o comprimento do componente cilíndrico. A tabela 6.19 mostra a comparação dos resultados obtidos com diferentes

Figura 6.27: Vaso de pressão



algoritmos. Todos algoritmos usam 80.000 funções avaliação, exceto AIS quais utiliza 150.000. O APM^{ngra} e o APM^{ngrl} utilizam 200 indivíduos na população. Observando essa tabela 6.19 a melhor solução foi encontrada por APM^{ngra} e $AISGA^C$. O melhor peso final foi igual a 6059.715 do APM^{ngra} . A tabela 6.20 demonstra que as soluções finais que são factíveis.

Tabela 6.19: Valores do peso no projeto do vaso de pressão

	melhor	mediana	média	desvio padrão	pior	fr
AIS-Coello [83]	6061.123	—	6734.085	—	7368.060	—
AIS-GA [81]	6060.368	—	6743.872	—	7546.750	—
AIS-GA ^C [81]	6060.138	—	6385.942	—	6845.496	—
AIS-GA ^H [82]	6059.855	6426.710	6545.126	1.24E + 2	7388.160	50
APM ^{gba} [61]	6065.822	6434.435	6632.376	5.15E + 2	8248.003	50
SR [60]	6832.584	7073.107	7187.314	2.67E + 2	8012.651	50
APM ^{ngra}	6059.714	6059.714	6152.190	2.24E + 2	7273.510	49
APM ^{ngrl}	7327.868	11101.550	11539.736	2.77E + 3	19193.813	49

6.3.5 O projeto da viga engastada e livre

Esse problema teste corresponde a minimização do volume de uma barra sujeita a uma carga de $P = 50000N$. Existem 10 variáveis de projeto correspondentes a altura (H_i) e largura (B_i) de uma seção transversal retangular para cada cinco passos constantes. As

Tabela 6.20: Variáveis de projeto encontradas para o vaso de pressão

	T_s	T_h	R	L	W
AIS-Coello	0.8125	0.4375	42.0870	176.7791	6061.1229
AIS-GA	0.8125	0.4375	42.0931	176.7031	6060.3677
AIS-GA ^C	0.8125	0.4375	42.0950	176.6797	6060.138
AIS-GA ^H	0.8125	0.4375	42.0973	176.6509	6059.8546
APM ^{gba}	0.8125	0.4375	42.0492	177.2522	6065.8217
SR	1.1250	0.5625	44.5941	176.6368	6832.5836
APM ^{ngra}	0.8125	0.4375	42.0984	176.6368	6059.7150
APM ^{ngrl}	0.5000	1.0000	47.1415	135.8732	7327.868

variáveis B_1 e H_1 são inteiras, B_2 e B_3 assumem valores discretos para serem escolhidos a partir de um conjunto 2.4, 2.6, 2.8, 3.1, H_2 e H_3 são discretas e são escolhidos dentro de um conjunto 45.0, 50.0, 55.0, 60.0 e, finalmente, B_4 , H_4 , B_5 , e H_5 são contínuas. As variáveis são dadas em centímetros e o módulo Young do material é igual a 200 GPa. O volume e as restrições são:

$$V(H_i, B_i) = 100 \sum_{i=1}^5 H_i B_i$$

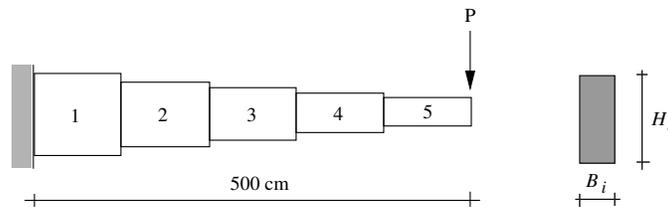
$$g_i(H_i, B_i) = \sigma_i \leq 14000 \text{ N/cm}^2 \quad i = 1, \dots, 5$$

$$g_{i+5}(H_i, B_i) = H_i/B_i \leq 20 \quad i = 1, \dots, 5$$

$$g_{11}(H_i, B_i) = \delta \leq 2.7 \text{ cm}$$

onde δ é a deflexão de ponta do feixe na direção vertical. A tabela 6.21 apresenta os

Figura 6.28: Viga engastada e livre



resultados encontrados usando diferentes técnicas. O número de avaliações de funções foram definidos como 35 000 em todos os casos, exceto na referência que usou 10 000 avaliações de funções em cada um dos três níveis do algoritmo GAOS. Os algoritmos APM^{ngra} e APM^{ngrl} utilizaram 350 indivíduos na população. Sendo que a técnica do

lagrangeano proposta nesse trabalho produziu a melhor solução com um volume igual a 64581.49. A tabela 6.22 mostra os valores das variáveis de projeto correspondentes a melhores soluções (todas factíveis).

Tabela 6.21: Volume encontrado para o projeto Viga Engastada e Livre

	melhor	mediana	média	desvio padrão	pior	fr
GAOS-Erbatur	6481.00	–	–	–	–	–
AIS-GA [81]	65559.60	–	70857.12	–	77272.78	–
AIS-GA ^C [81]	66533.47	–	71821.69	–	76852.86	–
AIS-GA ^H [82]	64834.70	74987.16	76004.24	6.93E + 3	102981.06	50
APM ^{gba} [61]	66030.05	79466.10	83524.21	1.44E + 4	151458.17	50
SR [60]	64599.65	70508.33	71240.03	3.90E + 3	83968.45	47
APM ^{ngra}	64804.65	70547.92	79511.14	2.93E + 4	216403.41	49
APM ^{ngrl}	64581.49	73156.67	76063.32	1.59E + 4	172479.78	48

Tabela 6.22: Variáveis de projeto para as melhores soluções da viga engastada e livre

	B_1	B_2	B_3	B_4	B_5	H_1	H_2	H_3	H_4	H_5	W
GAOS	3	3.1	2.6	2.300	1.800	60	55	50	45.500	35.0000	64815.0
AIS-GA	3	3.1	2.8	2.234	2.003	60	55	50	44.394	32.8787	65559.6
AIS-GA ^C	3	3.1	2.6	2.310	2.225	60	60	50	43.185	31.2502	66533.4
AIS-GA ^H	3	3.1	2.6	2.294	1.825	60	55	50	45.215	35.1191	64834.7
APM ^{gba}	3	3.1	2.6	2.209	2.094	60	60	50	44.042	31.9867	66030.0
SR	3	3.1	2.6	2.283	1.753	60	55	50	45.550	35.0631	64599.6
APM ^{ngra}	3	3.1	2.6	2.297	1.757	60	55	50	45.503	34.9492	64647.8
APM ^{ngrl}	4	3.2	2.6	2.281	1.750	60.9	3.0	2.6	45.612	35.0000	64581.4

Observa-se que o comportamento dos dois algoritmos não geracionais: APM^{ngra} e APM^{ngrl} obtiveram desempenho similares, porém pelos resultados percebe-se que o método do lagrangeano é superior. Comparando-se o algoritmo genético geracional com o não geracional, o desempenho do não geracional foi superior nesse caso.

6.3.6 Discussão dos problemas de engenharia mecânica

Apresenta-se aqui um quadro resumo, destacando em negrito para cada problema, quais foram os melhores resultados em cada peça para as medidas: **média**, **mediana**, **pior**, **melhor**, **desvio-padrão**, quando comparados os algoritmo APM^{ngra} e APM^{ngrl} propostos nesse trabalho. As tabelas 6.23 e 6.24 mostram individualmente os resultados de cada algoritmo. Finalmente construiu-se os gráficos presentes nas figuras 6.29 a 6.32 para a comparação do desempenho dos algoritmos.

Tabela 6.23: Desempenho do APM^{ngra} em cada problema de engenharia mecânica

	melhor	mediana	média	pior
Mola T/C.	0.012678	0.012678	0.012767	0.014533
Red. velocidade	2996.3481	2996.3481	3053.7382	5205.0287
Viga soldada	2.38113	2.38113	2.63030	11.7472
Vaso de pressão	6059.714	6059.714	6152.190	7273.510
Viga eng./livre	64804.65	70547.92	79511.14	216403.41

Tabela 6.24: Desempenho do APM^{ngrl} em cada problema de engenharia mecânica

	melhor	mediana	média	pior
Mola T/C.	0.012678	0.012678	0.012682	0.012733
Red. velocidade	2996.3481	2996.3482	3003.1981	3318.2931
Viga soldada	2.38113	2.38113	2.99895	20.6719
Vaso de pressão	7327.868	11101.550	11539.736	19193.813
Viga eng./livre	64581.49	73156.67	76063.32	172479.78

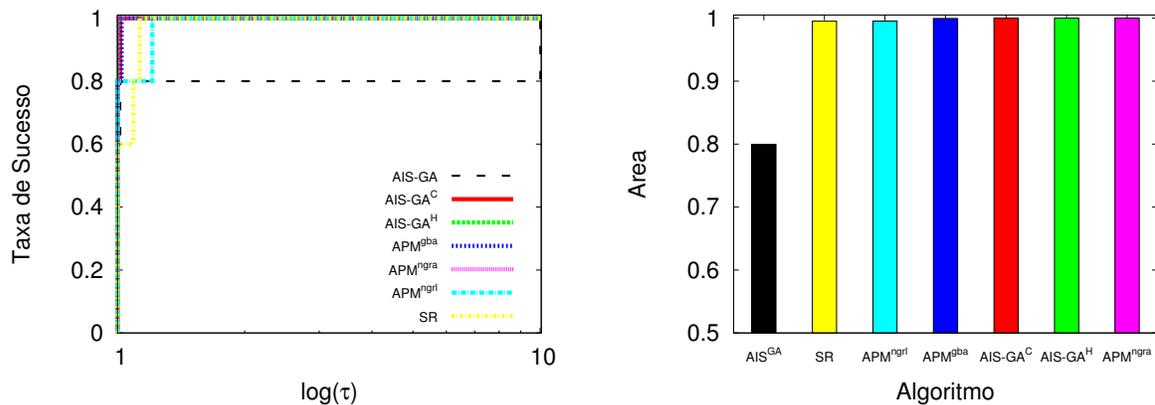
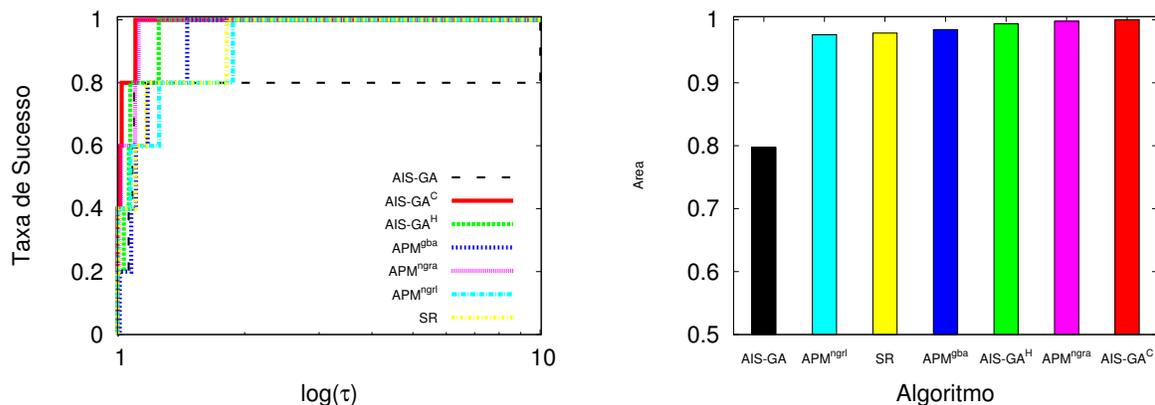
Figura 6.29: Comparação dos problemas de engenharia - Desempenho do **melhor**Figura 6.30: Comparação dos problemas de engenharia - Desempenho da **média**

Figura 6.31: Comparação dos problemas de engenharia - Desempenho da **mediana**

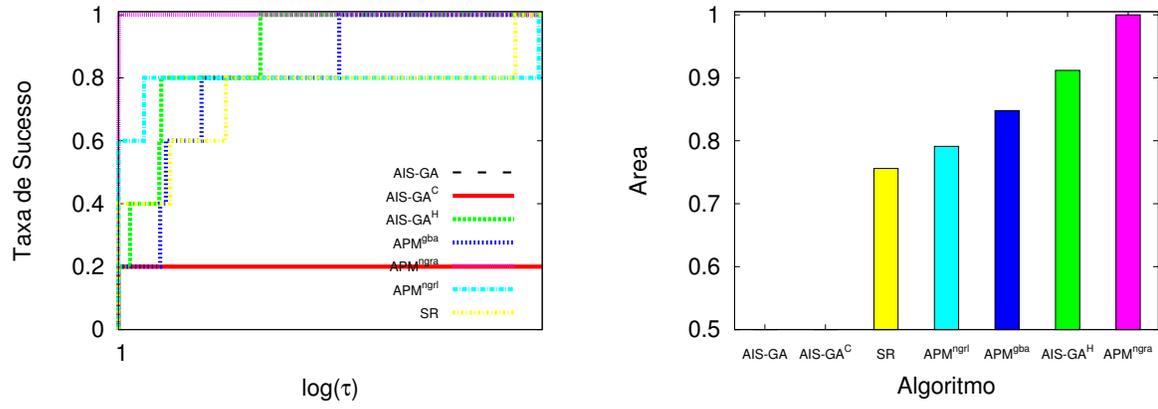
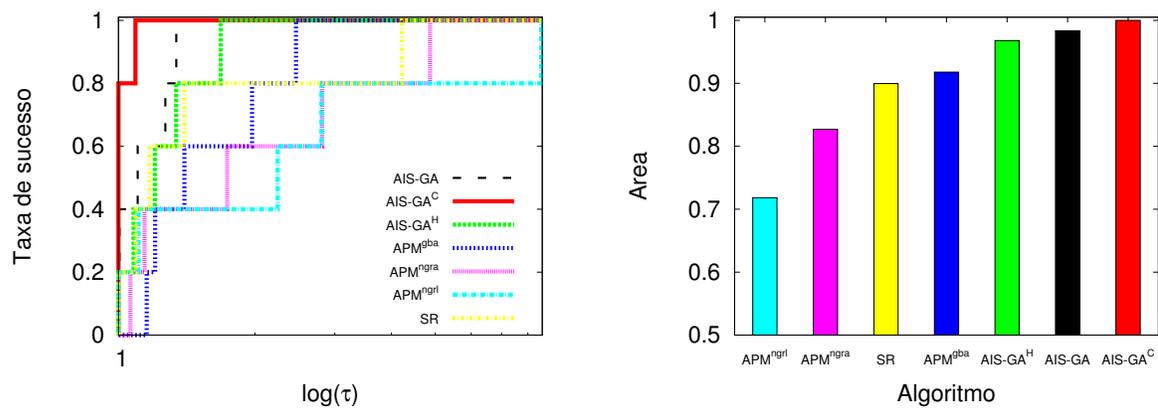


Figura 6.32: Comparação dos problemas de engenharia - Desempenho do **pior**



Percebe-se que no perfil de desempenho o algoritmo APM^{ngra} destacou-se como o melhor desempenho nos resultados do **melhor**, **mediana** e **média**. Na comparação das médias observou-se que essa técnica ficou em segundo lugar no desempenho mostrando, assim, grande eficiência perante os algoritmos analisados. O algoritmo AIS-GA^C apresentou melhor desempenho nas medidas: **média** e **pior**.

O algoritmo utilizando o lagrangeano APM^{ngrl} , no melhor, também apresentou um desempenho similar ao do APM^{ngra} porém ele não ficou tão eficiente, pois, conforme a tabela 6.24 os resultados dos testes realizados com o vaso de pressão apresentaram baixo desempenho nos itens analisados em comparação com os resultados dos outros algoritmos avaliados.

Observou-se também, que a função objetivo da via engastada e livre é não-linear onde o desempenho da maioria dos algoritmos avaliados obtiveram resultados pouco efetivos. Destacando-se, portanto, o método do lagrangeano e o ranqueamento estocástico.

6.4 Problemas de otimização estrutural

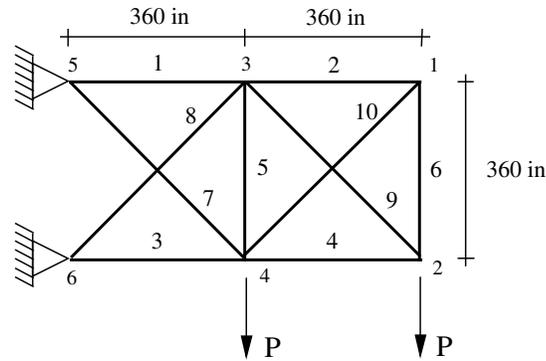
Para verificar a eficiência do APM para o algoritmo genético não-geracional proposto em [1] foram utilizados diversas problemas de otimização estrutural da literatura. Utilizou-se a codificação real, os operadores de mutação randômica e recombinação wright, com taxa de probabilidades 0.2 e 0.8, respectivamente. Minimizou-se o peso de 6 estruturas reticuladas: Treliças de 10, 25, 52, 72, 120 e 200 barras. Adotou-se o método de penalização discutido na seção 4.2 e o método do lagrangeano aumentado proposto no item 5.5.

6.4.1 *Treliça de 10 barras*

Esse problema corresponde a minimização do peso de uma estrutura da treliça de dez barras. As restrições envolvem a tensão de cada membro e deslocamentos nos nós. As variáveis de projeto são as áreas da seção transversal das barras (A_i , $i = 1, 10$). A tensão permitida é limitada em ± 25 ksi e os deslocamentos são limitados para 2 in na direção x e y . A densidade do material é 0.1 lb/in³, o módulo de elasticidade é $E = 10^4$ ksi, e as cargas verticais descendentes $P=100$ kips são aplicadas nos nós 2 e 4.

Dois casos de variáveis são analisados, as discretas e as contínuas. O número avaliações de funções consideradas foi 90 000 para o caso discreto e 280 000 para o caso contínuo. No

Figura 6.33: Treliça de 10 barras



APM^{ngra} e APM^{ngrl} utilizou-se 300 e 400 indivíduos para a população nos casos discreto e contínuo, respectivamente.

Para o caso discreto, os valores da área da seção transversal (in²) são escolhidos entre as possíveis \mathcal{S} 32 opções: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.93, 3.13, 3.38, 3.47, 3.55, 3.63, 3.88, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.97, 11.50, 13.50, 14.20, 15.50, 16.90, 18.80, 19.90, 22.00, 26.50, 30.00, 33.50. A tabela 6.25 apresenta os valores encontrados para o peso final no caso discreto. Os melhores resultados obtidos foram do APM^{ngrl} e AIS-GA^H.

Tabela 6.25: Valores do peso para a treliça de 10 barras – caso discreto

	melhor	mediana	média	desvio padrão	pior	fr
AIS-GA	5539.24	—	5754.97	—	6790.89	—
AIS-GA ^C	5528.09	—	5723.78	—	6239.99	—
AIS-GA ^H	5490.74	5504.54	5513.90	2.56E + 1	5575.28	50
APM ^{bc}	5490.74	5558.74	5585.98	1.48E + 2	6443.23	50
SR	5491.72	5648.46	5664.21	9.64E + 1	6020.77	50
APM ^{ngra}	5507.75	5616.55	5764.74	3.71E + 2	7066.56	48
APM ^{ngrl}	5490.73	5513.31	5515.77	2.75E + 1	5637.15	47

Para o caso contínuo, a área da seção transversal é igual 0.1 in². As tabelas 6.27 e 6.28 mostram os resultados do caso contínuo sendo o melhor resultado do APM^{ngra}. Os resultados apresentados das variáveis de projeto para o caso contínuo e discreto são todos factíveis.

Tabela 6.26: Variáveis de projeto encontradas na treliça de 10 barras – caso discreto

	AIS-GA	AIS-GA ^C	AIS-GA ^H	APM ^{ngba}	SR	APM ^{ngra}	APM ^{ngrl}
1	33.50	33.50	33.50	33.50	33.50	30.00	42.00
2	1.80	1.62	1.62	1.62	1.62	1.62	1.00
3	26.5	22.00	22.90	22.90	22.90	22.90	39.24
4	15.50	14.20	14.20	14.20	15.50	16.90	32.62
5	1.62	1.62	1.62	1.62	1.62	1.62	1.00
6	2.13	1.62	1.62	1.62	1.62	1.62	1.97
7	7.97	5.74	7.97	7.97	7.97	7.97	28.84
8	19.90	26.50	22.90	22.90	22.00	22.90	39.90
9	22.00	22.00	22.00	22.00	22.00	22.90	38.67
10	1.62	1.62	1.62	1.62	1.62	1.62	1.02
W	5539.24	5528.09	5490.74	5490.74	5491.72	5490.74	5490.73

Tabela 6.27: Valores encontrados no peso final da treliça de 10 barras – caso contínuo

	melhor	mediana	média	desvio padrão	pior	fr
AIS-GA	5062.67	–	5075.55	–	5094.89	–
AIS-GA ^C	5064.67	–	5082.52	–	5113.22	–
AIS-GA ^H	5061.16	5064.36	5068.85	7.78	5084.56	50
APM ^{gba}	5062.12	5070.54	5133.22	2.48E + 2	6430.55	50
SR	5061.71	5079.53	5077.67	1.01E + 1	5101.17	50
APM ^{ngra}	5060.85	5076.89	5136.45	2.71E + 2	6460.24	50
APM ^{ngrl}	5060.85	5060.85	5202.49	5.63E + 2	8079.36	50

Tabela 6.28: Variáveis de projeto da treliça de 10 barras – caso contínuo

	AIS-GA	AIS-GA ^C	AIS-GA ^H	APM ^{gba}	SR	APM ^{ngra}	APM ^{ngrl}
1	30.16252	29.78121	30.52684	30.95080	30.01400	30.52008	30.52196
2	0.10004	0.10031	0.10000	0.10000	0.10000	0.10000	0.10000
3	22.81192	22.55140	22.91574	22.92083	26.14460	23.18592	23.19991
4	15.87183	15.50462	15.48294	15.55024	15.29260	15.22868	15.22339
5	0.10000	0.10002	0.10000	0.10000	0.10000	0.10000	0.1000
6	0.51495	0.52377	0.54620	0.60959	0.55610	0.55303	0.55135
7	7.50595	7.52854	7.47594	7.46973	7.43980	7.45773	7.45719
8	21.26408	21.15708	21.01566	20.83562	21.00560	21.03392	21.03632
9	21.38304	22.21351	21.55362	21.35644	21.93900	21.53676	21.52809
10	0.10001	0.10018	0.10000	0.10000	0.10000	0.10000	0.10000
W	5062.67	5064.67	5061.16	5062.12	5061.71	5060.85	5060.85

6.4.2 *Treliça de 25 barras*

Esse problema tem o objetivo de minimizar o peso da treliça de 25 barras mostrada na figura 6.34. Essa treliça é composta por barras de comprimento L_k onde k representa o índice das barras da estrutura. As variáveis de projeto são áreas de seções transversais das barras a_k , e ρ é a massa específica do material. Logo, a finalidade do problema é encontrar o conjunto de áreas $a_k = A_1, A_2, \dots, A_l$, onde A é a área de cada seção transversal, a fim de minimizar o peso da estrutura.

A treliça está sujeita as restrições de tensão e deslocamento. A tensão máxima é ± 40 ksi com deslocamentos máximos nos nós 1 e 2 limitados a $0.35in$, ambos nas direções x e y .

As variáveis de projeto devem ser escolhidas dentro do conjunto com 30 opções diferentes(em in^2): 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.1, 3.2. Essas variáveis são agrupadas conforme a tabela 6.30. A densidade do material de composição das barras é igual a 0.1 lb/in^3 e o módulo de elasticidade é igual a 10^4 ksi. O carregamento da treliça é mostrado na tabela 6.29.

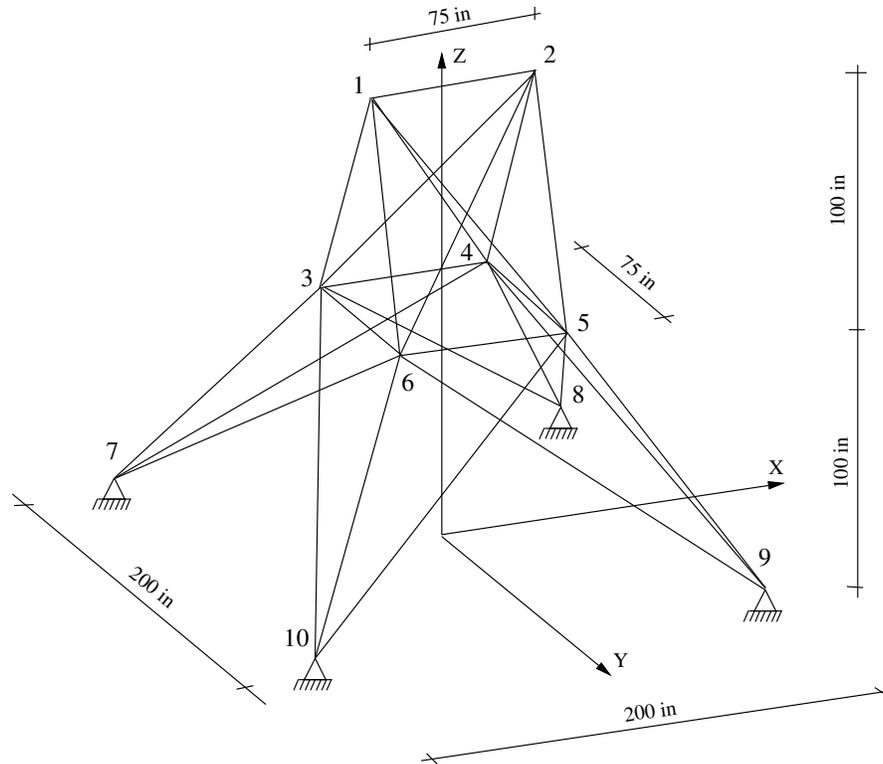
Tabela 6.29: Dados de carregamento para a treliça de 25 barras(kips).

nó	F_x	F_y	F_z
1	1	-10.0	-10.0
2	0	-10.0	-10.0
3	0.5	0	0
6	0.6	0	0

Tabela 6.30: Agrupamento dos membros para as treliças de 25 barras.

Grupo	Conectividades
A_1	1-2
A_2	1-4, 2-3, 1-5, 2-6
A_3	2-5, 2-4, 1-3, 1-6
A_4	3-6, 4-5
A_5	3-4, 5-6
A_6	3-10, 6-7, 4-9, 5-8
A_7	3-8, 4-7, 6-9, 5-10
A_8	3-7, 4-8, 5-9, 6-10

Figura 6.34: Treliça de 25 barras.



Apresentou-se algumas soluções referentes as áreas das seções transversais encontradas na literatura através da tabela 6.31. Nas duas últimas linhas dessa tabela também mostrou-se os valores dos deslocamentos verticais u_{y1} e u_{y2} , nos nós 1 e 2 respectivamente. Os parâmetros utilizados pelas referências apresentadas são os seguintes: As referências [86] e [61] utilizaram um algoritmo genético com codificação binária, a população com 40 indivíduos, 20 rodadas, 800 funções avaliações; O trabalho [84] utilizou uma otimização multinível abordando (GAOS) no caso discreto, com 2 níveis de 10000 funções avaliações em cada um deles; Em [87] utilizou um algoritmo genético não-geracional, codificação binária, método de penalização constante e 40000 funções avaliações. TCell [88] utilizou um APM com a técnica dos sistemas imunológicos com 50 rodadas, 20 000 avaliações de funções por rodada. Os autores do trabalho PSO [89] não citaram o número de funções avaliações somente o número de 20 rodadas. O APM [61] apresentado na penúltima coluna da tabela 6.31 utilizou o AG codificação binária, população com 100 indivíduos e 20000 funções avaliações. O algoritmo de Templeman foi aplicado na referência [90]. O APM^{ngra} e APM^{ngrl} utilizado aqui é um AG não-geracional com codificação real, uma

população com 100 indivíduos e 20 000 avaliações de funções. Observou-se que todas as soluções são factíveis e os projetos são distintos.

Na tabela 6.31 o peso final(W) é representado em lb. Finalmente realizou-se uma comparação com outros trabalhos da literatura dos melhores resultados da função objetiva encontrados para a treliça de 25 barras. A tabela 6.32 traz esses resultados. Na última linha da tabela, colocamos os resultados obtidos pelos APMs estudados nessa dissertação: APM^{ngra} e APM^{ngrl} .

Observando os resultados da tabela 6.32 percebe-se que a referência TCell [88] obteve o melhor desempenho. O restante dos algoritmos analisados obtiveram resultados muito similares. Em se tratando de algoritmo geracional e não-geracional, o desempenho do geracional ainda supera. Comparando-se as duas técnicas de penalização estudadas aqui, para o não-geracional, observa-se que nesse tipo de treliça o método do lagrangeano obtém resultado superior.

6.4.3 Treliça de 52 barras

O objetivo aqui é minimizar o peso da treliça de 52 barras mostrada na figura 6.35. A tabela 6.33 descreve as condições de carregamento para esse tipo de treliça.

Os materiais que compõem essa treliça possuem as propriedades de módulo de Young igual a $2,07 \times 10^5$ MPa e a densidade igual para 7.860 kg/m³. As tensões admissíveis na tração e compressão são definidas a 180 MPa. São definidos doze grupos para os membros dessa treliça conforme a Tabela 6.34. Os valores das áreas transversais devem ser escolhidos a partir das 64 opções apresentadas na Tabela 6.35.

Na tabela 6.36 é feita uma comparação do peso de cada barra e o peso final da treliça de 52 barras. As referências utilizadas foram [87, 13, 61], os APM^{ngra} e APM^{ngrl} .

Em 6.36, a referência [87] corresponde a um GA não-geracional, utilizando os operadores de recombinação de um ponto, na primeira coluna da tabela e dois pontos na segunda coluna, ambos com 60.000 funções avaliações. O trabalho de [13] utilizou um GA geracional, 20 000 avaliações de funções. O APM^{gba} utilizou um tamanho de população igual a 70, 250 gerações e 20 rodadas independentes. Nota-se que as soluções da tabela 6.36 satisfazem ao conjunto de restrições impostas no problema. Por fim, utilizou-se os APM^{ngra} e APM^{ngrl} com população de 100 indivíduos, 17 500 funções avaliações e 20 rodadas independentes. A tabela 6.37 traz um sumário estatístico com uma comparação do algoritmo

Tabela 6.31: Comparação dos resultados para as treliças de 25 barras – caso discreto

Variáveis	Ref.[86]	APM ⁺ [61]	Ref.[90]	Ref.[84]	Ref.[87]	APM ^{gba} [61]	TCell[88]	PSO[89]	APM ^{ngra}	APM ^{ngrl}
A ₁	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
A ₂	1.8	0.7	1.9	1.2	0.5	0.3	0.2	0.4	0.1	0.5
A ₃	2.3	3.4	2.6	3.2	3.4	3.4	3.5	3.4	3.4	3.0
A ₄	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
A ₅	0.1	1.8	0.1	1.1	1.5	2.1	1.9	1.9	2.1	2.0
A ₆	0.8	1.0	0.8	0.9	0.9	1.0	0.8	0.9	1.0	1.0
A ₇	1.8	0.3	2.1	0.4	0.6	0.5	0.1	0.4	0.7	0.4
A ₈	3.0	3.4	2.6	3.4	3.4	3.4	3.7	3.4	3.4	3.0
W	546.01	486.74	562.93	493.80	486.29	484.85	471.33	483.84	488.85	486.49

Tabela 6.32: Valores encontrados para o peso final do projeto da treliça de 25 barras

	melhor	mediana	média	desvio padrão	pior	fr
T-Cell	471.33	--	479.20	$6.17E + 0$	504.50	--
PSO	483.84	--	--	--	489.42	--
APM ^{gba}	484.85	--	485.96	--	490.74	--
APM ^{ngra}	488.85	508.14	508.66	$6.62E + 0$	533.22	19
APM ^{ngrl}	486.49	490.68	493.47	$4.55E + 0$	527.22	19

Figura 6.35: Treliça de 52 barras.

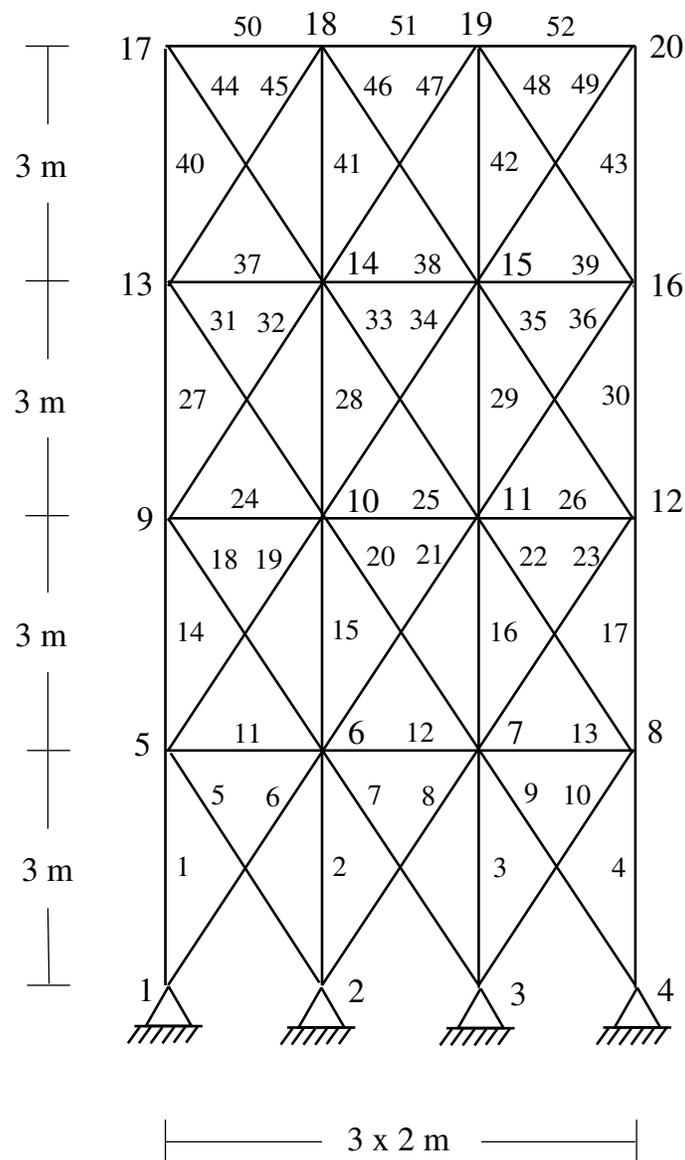


Tabela 6.33: Dados de carregamento para a treliça de 52 barras(kN).

nó	F_x	F_y
17	100.0	200.0
18	100.0	200.0
19	100.0	200.0
20	100.0	200.0

Tabela 6.34: Membros de agrupamento para a treliça de 52 barras.

grupo	membros
A ₁	1, 2, 3, 4
A ₂	5, 6, 7, 8, 9, 10
A ₃	11, 12, 13
A ₄	14, 15, 16, 17
A ₅	18, 19, 20, 21, 22, 23
A ₆	24, 25, 26
A ₇	27, 28, 29, 30
A ₈	31, 32, 33, 34, 35, 36
A ₉	37, 38, 39
A ₁₀	40, 41, 42, 43
A ₁₁	44, 45, 46, 47, 48, 49
A ₁₂	50, 51, 52

Tabela 6.35: Área para a seção transversal para a treliça de 52 barras.

seção	mm ²						
1	71.613	17	1008.385	33	2477.414	49	7419.340
2	90.968	18	1045.159	34	2496.769	50	8709.660
3	126.451	19	1161.288	35	2503.221	51	8967.724
4	161.290	20	1283.868	36	2696.769	52	9161.272
5	198.064	21	1374.191	37	2722.575	53	9999.980
6	252.258	22	1535.481	38	2896.768	54	10322.560
7	285.161	23	1690.319	39	2961.284	55	10903.204
8	363.225	24	1696.771	40	3096.768	56	12129.008
9	388.386	25	1858.061	41	3206.445	57	12838.684
10	494.193	26	1890.319	42	3303.219	58	14193.520
11	506.451	27	1993.544	43	3703.218	59	14774.164
12	641.289	28	2019.351	44	4658.055	60	15806.420
13	645.160	29	2180.641	45	5141.925	61	17096.740
14	792.256	30	2238.705	46	5503.215	62	18064.480
15	816.773	31	2290.318	47	5999.998	63	19354.800
16	940.000	32	2341.191	48	6999.986	64	21612.860

Tabela 6.36: Comparação dos resultados da treliça 52 barras – peso final(Kg).

Variáveis	Ref.[87]	Ref.[87]	Ref.[13]	APM ^{gba}	APM ^{ngra}	APM ^{ngrl}
A ₁	3703.218	4658.055	4658.055	4658.055	4568.055	4658.055
A ₂	2722.575	1161.288	1161.288	1161.288	1161.288	1161.288
A ₃	1858.575	645.160	363.225	494.193	285.161	506.451
A ₄	3206.445	3303.219	3303.219	3303.219	3303.219	3303.219
A ₅	1008.385	1045.159	940.000	940.000	1045.159	940.000
A ₆	1008.385	494.193	641.289	641.289	252.258	494.193
A ₇	2477.414	2477.414	2238.705	2238.705	2896.768	2238.705
A ₈	1008.385	1045.159	1008.385	1008.385	940.000	1008.385
A ₉	388.386	285.161	494.193	363.225	363.225	494.193
A ₁₀	2477.414	1696.771	1283.868	1283.868	1161.288	1283.868
A ₁₁	1008.385	1045.159	1161.288	1161.288	1161.288	1161.288
A ₁₂	1008.383	641.289	494.193	494.193	1374.191	506.451
W	2294.521	1970.142	1903.366	1903.366	1973.422	1903.076

APM^{gba}, APM^{ngra} e o APM^{ngrl} a fim de analisar os melhores valores do peso da treliça de 52 barras.

Tabela 6.37: Valores encontrados para o peso final do projeto da treliça de 52 barras

	melhor	mediana	média	desvio padrão	pior	fr
APM ^{gba}	1903.36	–	2077.46	–	2383.75	–
APM ^{ngra}	1973.42	2273.49	2295.33	$2.22E + 2$	2861.34	19
APM ^{ngrl}	1903.07	1940.93	2035.39	$1.83E + 2$	2537.70	19

6.4.4 Treliça de 72 barras

A treliça de 72 barras descrita na figura 6.36 é utilizada para a minimização do peso. O valor mínimo de cada uma das variáveis de projeto da área da seção transversal variam de 0.1 in^2 à 5 in^2 . São consideradas 72 variáveis de projeto separadas em 16 grupos conforme a tabela 6.39. As restrições envolvem um valor máximo de deslocamento permitido de $|0.25|$ nos nós de 1 a 16 ao longo da direção x e y , e uma tensão máxima permitida em cada barra de $\pm 25 \text{ ksi}$. A densidade do material é 0.1 lb/in^3 e o módulo Young é igual a 10^4 ksi . São definidos dois casos de carregamento considerados para essa estrutura de acordo com a tabela 6.38.

Figura 6.36: Treliça de 72 barras

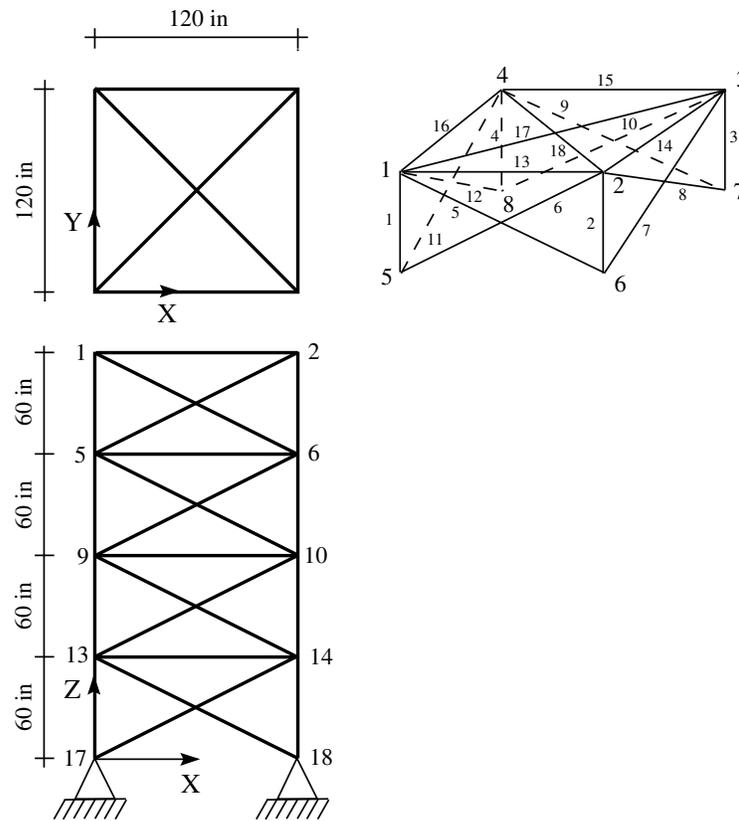


Tabela 6.38: Dados de carregamento para a treliça de 72 barras(kips).

Caso da carga	nó	F_x	F_y	F_z
1	1	5	5	-5
2	1	0	0	-5
	2	0	0	-5
	3	0	0	-5
	4	0	0	-5

Tabela 6.39: Membros de grupo treliça de 72 barras.

Grupo	Membros
A ₁	1, 2, 3, 4
A ₂	5, 6, 7, 8, 9, 10, 11, 12
A ₃	13, 14, 15, 16
A ₄	17, 18
A ₅	19, 20, 21, 22
A ₆	23, 24, 25, 26, 27, 28, 29, 30
A ₇	31, 32, 33, 34
A ₈	35, 36
A ₉	37, 38, 39, 40
A ₁₀	41, 42, 43, 44, 45, 46, 47, 48
A ₁₁	49, 50, 51, 52
A ₁₂	53, 54
A ₁₃	55, 56, 57, 58
A ₁₄	59, 60, 61, 62, 63, 64, 65, 66
A ₁₅	67, 68, 69, 70
A ₁₆	71, 72

Observou-se as seguintes referências: [91, 92] onde utilizou-se técnicas de otimização baseadas em critérios da física; Na referência [93] utilizou-se conceitos aproximados para a otimização estrutural com 20 rodadas independentes, com 70 indivíduos na população, e 500 gerações. As colunas referentes a referência [84], utilizaram a otimização através de uma GA multinível usando GAOS com 20 000 e 30 000 avaliações respectivamente.

Para finalizar usou-se o APM^{ngra} e APM^{ntrl} com 35 000 avaliações, 100 indivíduos na população e 20 rodadas independentes. Apresentou-se na tabela 6.40 um sumário estatístico comparando dos algoritmos APM^{ngra} e APM^{ntrl} para o peso final da treliça de 72 barras. Em seguida, apresenta-se uma tabela 6.41 para comparar os resultados das áreas da seção transversal de cada barra e também o peso final(Kg) encontrado para as treliças de 72 barras. Os valores mostrados na tabela 6.41, que tem as soluções marcadas com um asterisco não são rigorosamente factíveis.

Tabela 6.40: Valores encontrados para o peso final do projeto da treliça de 72 barras

	melhor	mediana	média	desvio padrão	pior	fr
APM^{gba}	387.03	–	402.58	–	432.95	–
APM^{ngra}	415.80	454.48	491.76	$1.07E + 2$	880.93	20
APM^{ntrl}	380.13	399.10	400.33	1.79E + 1	457.65	19

Tabela 6.41: Comparação dos resultados para a treliça de 72 barras. Peso final(W)-lb.

Var.	Ref.[91]	Ref.[92]	Ref.[93]	Ref.[84]	Ref.[84]*	APM ^{gba}	APM ^{ngra}	APM ^{ngrl}
A ₁	0.161	0.1492	0.1585	0.155	0.161	0.155	0.189	0.155
A ₂	0.557	0.7733	0.5936	0.535	0.544	0.545	0.558	0.557
A ₃	0.377	0.4534	0.3414	0.480	0.379	0.274	0.425	0.409
A ₄	0.506	0.3417	0.6076	0.520	0.521	0.518	0.433	0.599
A ₅	0.611	0.5521	0.2643	0.460	0.535	0.603	0.487	0.543
A ₆	0.532	0.6084	0.5480	0.530	0.535	0.666	0.622	0.495
A ₇	0.100	0.1000	0.1000	0.120	0.103	0.101	0.107	0.100
A ₈	0.100	0.1000	0.1509	0.165	0.111	0.130	0.101	0.100
A ₉	1.246	1.0235	1.1067	1.155	1.310	1.199	0.878	1.210
A ₁₀	0.524	0.5421	0.5793	0.585	0.498	0.473	0.445	0.544
A ₁₁	0.100	0.1000	0.1000	0.100	0.110	0.100	0.100	0.100
A ₁₂	0.100	0.1000	0.1000	0.100	0.103	0.109	0.100	0.100
A ₁₃	1.818	1.4636	2.0784	1.755	1.910	1.953	2.699	0.182
A ₁₄	0.524	0.5207	0.5034	0.505	0.525	0.516	0.574	0.506
A ₁₅	0.100	0.1000	0.1000	0.105	0.122	0.100	0.101	0.100
A ₁₆	0.100	0.1000	0.1000	0.155	0.103	0.101	0.186	0.100
W	381.2	395.97	388.63	385.76	383.12	387.03	415.80	380.13

Observando os resultados obtidos nas tabelas 6.40 e 6.41 o algoritmo, APM^{ngrl}, que utilizou o método do lagrangeano aumentado obteve o melhor desempenho, obtendo o peso de 380.13 Kg entre as referências utilizadas.

6.4.5 Treliça de 120 barras

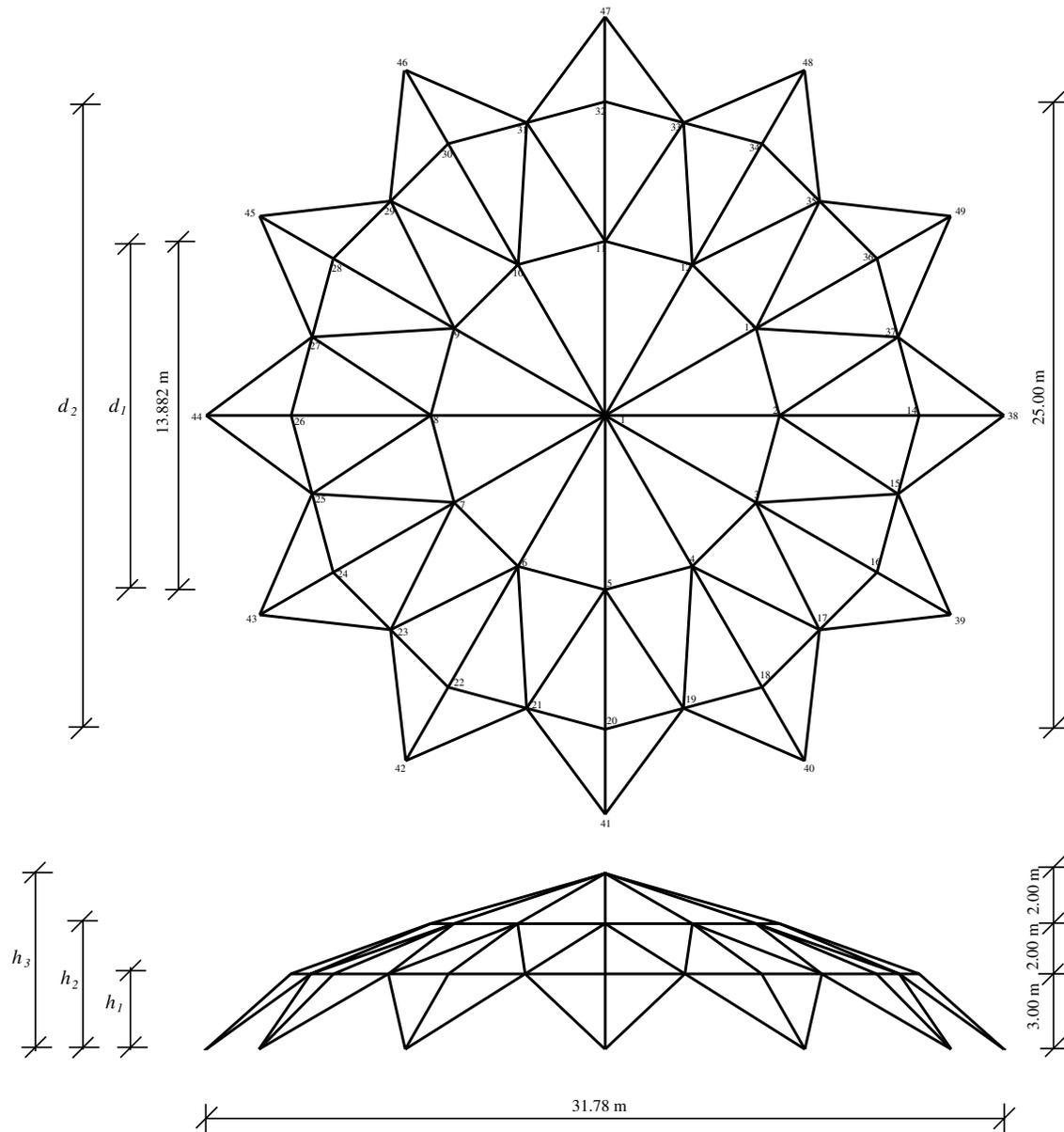
A finalidade desse problema é minimizar o peso da treliça tridimensional em cúpula com 120 barras mostrada na figura 6.37. Essa cúpula está presente em alguns trabalhos da literatura[94, 95, 96].Essa treliça está sujeita a um carregamento vertical de 600 kN em sua coroa(nó 1).

Somente o caso discreto é considerado aqui na minimização do peso da treliça onde variáveis de projeto de dimensionamento são as áreas transversais, que devem ser escolhidas a partir das seções tubulares conforme a tabela 6.42.

A densidade do material é 7.86×10^{-5} kN/cm³ e o módulo de elasticidade é igual a 21000 kN/cm². Os deslocamentos são limitados em 2 cm em qualquer nó nas direções globais x , y , e z . Nota-se que a cúpula é composta de módulos padrão, como descrito na figura 6.38, contendo 10 barras agrupadas em 7 seções transversais distintas. Então, a cúpula mostrada na figura 6.37 apresenta 12 módulos padrão levando a um total de 120

barras.

Figura 6.37: Treliça de 120 barras



Observa-se pelas tabelas 6.42 e 6.43 que o desempenho do método do lagrangeano (APM^{ngrl}) no algoritmo não-geracional foi superior ao desempenho da penalização adaptativa APM^{ngra} .

Figura 6.38: Módulo 120 barras.

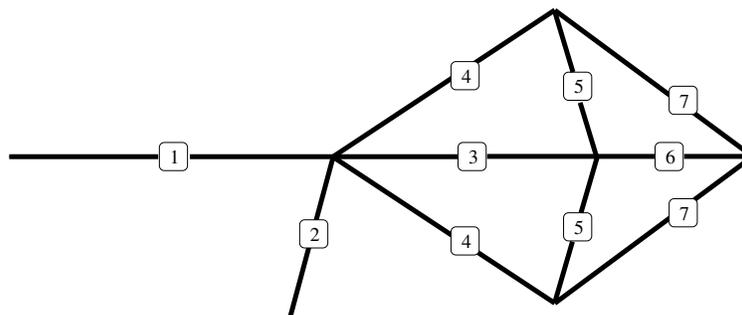


Tabela 6.42: Áreas da seção transversal treliça de 120 barras.

Var.	APM^{ngra}	APM^{ngrl}
A_1	10.848	10.484
A_2	16.824	17.359
A_3	15.628	15.628
A_4	11.101	11.109
A_5	10.956	10.631
A_6	10.407	10.407
A_7	10.036	10.042
W	2016.91	2016.77

Tabela 6.43: Sumário estatístico treliça de 120 barras

	melhor	mediana	média	desvio padrão	pior	fr
APM^{ngra}	2016.91	2031.41	2089.10	1.88E + 2	2867.15	20
APM^{ngrl}	2016.77	2021.53	2072.85	2.23E + 2	3024.10	20

6.4.6 *Treliça de 200 barras*

Essa treliça plana de 200 barras proposta em [97], tem uma estrutura de 77 nós como mostrado na figura 6.39. A finalidade desse problema é determinar a área da seção transversal de cada barra dessa treliça visando minimizar o peso total da treliça. Ela deve ser projetada em função de três condições de carregamento independente e com restrição de tensão em seus membros. As três condições são:

(1) manter a direção x positiva nos pontos do nós: 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, e 71;

(2) 10 kips agindo na direção y negativa nos pontos dos nós: 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22,24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 71, 72, 73, 74, e 75

(3) As condições da carga 1 e 2 agindo em conjunto.

Em [88] propõe-se que os 200 elementos da treliça sejam classificados em 29 grupos conforme a tabela 6.44. As tensões máximas de cada elemento é limitada em 10 ksi para membros com tração e compressão. A tabela 6.46 faz uma comparação dos resultados do algoritmo aqui proposto com outros presentes na literatura[97, 98, 88, 99]. O algoritmo Harmony Search [98] e TCell [88] utilizaram 50 000 avaliações nas funções e 30 rodadas independentes. Os outros trabalhos não citaram o número de funções avaliações utilizados. Os algoritmos apresentados neste trabalho usaram 20 000 funções avaliações e 20 rodadas independentes.

Observando os resultados obtidos na treliça de 200 barras, conforme a tabela 6.46, percebe-se que o desempenho dos algoritmos APM^{ngra} e APM^{ngrl} não apresentaram resultados tão interessantes quando comparou-se com outras técnicas. Nesse tipo de treliça o TCell [88] obteve o melhor desempenho.

Figura 6.39: Treliça de 200 barras.

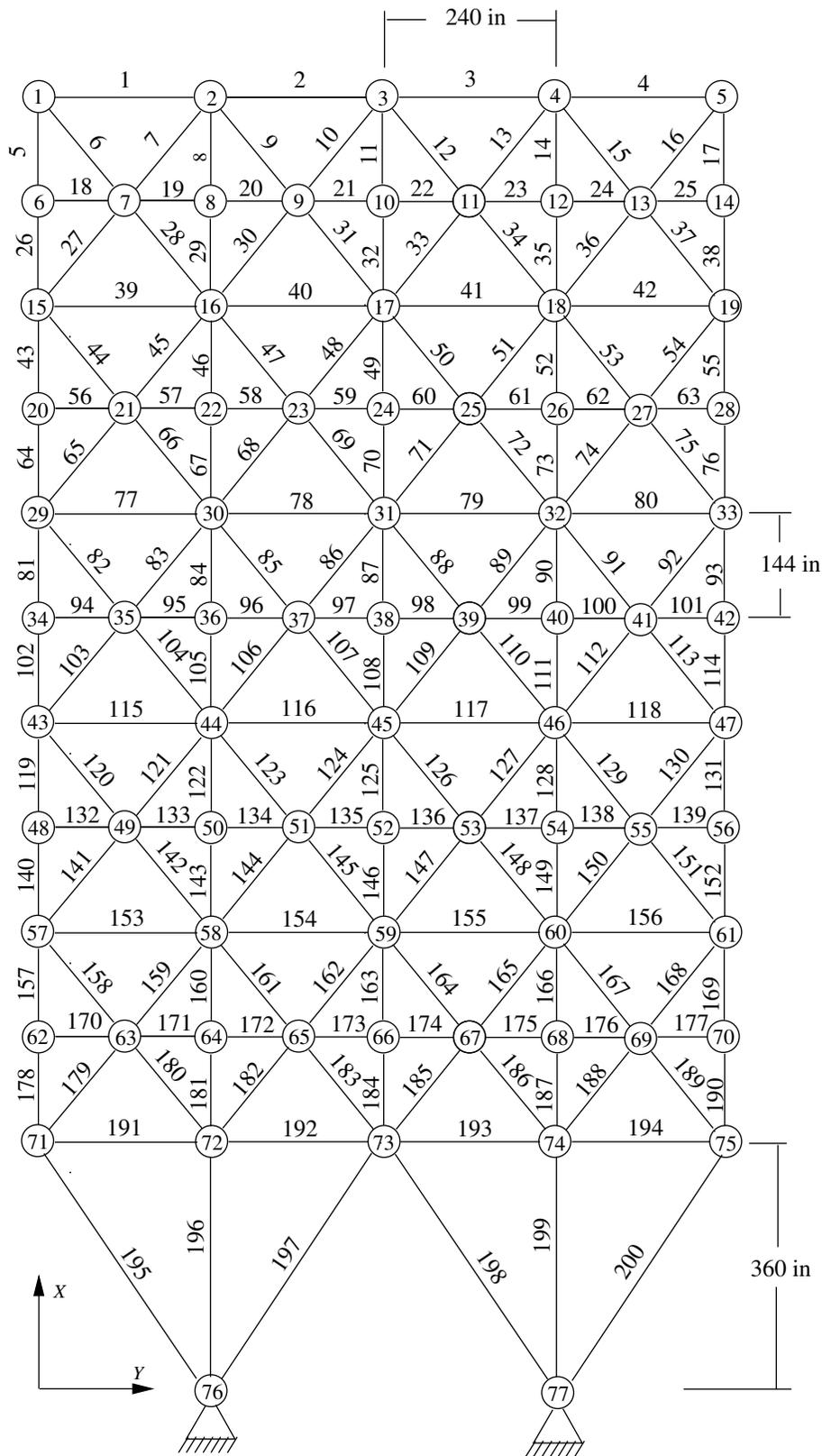


Tabela 6.44: Membros dos grupos para a treliça de 200 barras

N° Grupo	N° Membro
1	1,2,3,4
2	5, 8, 11, 14, 17
3	19, 20, 21, 22, 23, 24
4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177
5	26, 29, 32, 35, 38
6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34, 36, 37
7	39, 40, 41, 42
8	43, 46, 49, 52, 55
9	57, 58, 59, 60, 61, 62
10	64, 67, 70, 73, 76
11	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75
12	77, 78, 79, 80
13	81, 84, 87, 90, 93
14	95, 96, 97, 98, 99, 100
15	95, 96, 97, 98, 99, 100
16	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
17	115, 116, 117, 118
18	119, 122, 125, 128, 131
19	133, 134, 135, 136, 137, 138
20	140, 143, 146, 149, 152
21	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151
22	153, 154, 155, 156
23	157, 160, 163, 166, 169
24	171, 172, 173, 174, 175, 176
25	178, 181, 184, 187, 190
26	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
27	191, 192, 193, 194
28	195, 197, 198, 200
29	196, 199

Tabela 6.45: Áreas da seção transversal treliça de 200 barras.

Var.	T-Cell	Harmony search	APM ^{ngra}	APM ^{ngrl}
A ₁	0.507	0.125	0.100	0.442
A ₂	0.985	1.015	1.106	0.471
A ₃	0.670	0.106	0.154	0.100
A ₄	0.407	0.109	0.943	0.357
A ₅	2.511	1.936	2.536	1.147
A ₆	0.333	0.268	0.441	0.481
A ₇	0.660	0.104	1.987	0.653
A ₈	3.753	2.973	1.718	4.110
A ₉	0.464	0.130	0.102	0.100
A ₁₀	4.372	4.183	2.398	3.417
A ₁₁	0.428	0.396	1.575	1.232
A ₁₂	0.347	0.441	0.655	0.216
A ₁₃	5.019	5.187	4.590	6.588
A ₁₄	0.437	0.191	0.100	0.382
A ₁₅	6.316	6.241	11.045	6.418
A ₁₆	0.485	0.699	0.817	1.096
A ₁₇	0.524	0.115	0.100	2.034
A ₁₈	7.436	7.764	7.973	7.312
A ₁₉	0.401	0.100	0.100	1.449
A ₂₀	7.916	8.827	9.413	8.312
A ₂₁	0.859	0.698	0.581	1.695
A ₂₂	1.697	1.556	2.157	2.180
A ₂₃	10.124	10.980	9.944	11.314
A ₂₄	0.438	0.131	0.410	1.292
A ₂₅	10.962	12.149	10.957	12.314
A ₂₆	1.642	1.637	1.253	2.544
A ₂₇	3.800	10.896	5.003	5.217
A ₂₈	8.419	12.414	9.354	8.977
A ₂₉	13.578	13.257	15.091	16.180
W	24852.58	25447.10	30129.958	29908.904

Tabela 6.46: Sumário estatístico das melhores soluções treliça 200 barras

Algoritmo	melhor	mediana	média	desvio padrão	pior	nr
TCell[88]	24852.58	---	27376.57	2165.0667	33132.30	---
CONMIN[97]	34800.00	---	---	---	---	---
LINRM[97]	33315.00	---	---	---	---	---
SUMT[97]	27564.00	---	---	---	---	---
M-3[97]	26600.00	---	---	---	---	---
M-4[97]	26654.00	---	---	---	---	---
M-5[97]	26262.00	---	---	---	---	---
Harmony Search[98]	25447.10	---	---	---	---	---
TRUST[99]	25500.8	---	---	---	---	---
APM ^{ngra}	30129.95	38343.17	39832.08	1.17E+4	85366.26	20
APM ^{ngrl}	29908.94	34356.30	36570.64	6.24E+3	52897.92	19

Finalmente, confeccionou-se um sumário dos resultados propostos nesse trabalho para problemas de otimização estrutural. Os algoritmos são baseados em AG não-geracional com codificação real dois modelos para o trato das restrições, a saber, uma técnica de penalização adaptativa (APM^{gra}) e o método do lagrangeano (APM^{grl}). Em seguida, as tabelas 6.47 e 6.48 mostram os valores: **melhor**, **média**, **mediana**, **desvio padrão**, **pior** da função objetivo e por último a medida **fr** que corresponde ao número de rodadas.

Observando as duas tabelas percebe-se que o algoritmo genético não-geracional, nos testes com as estruturas reticuladas, apresentou melhor desempenho com a utilização do método do lagrangeano aumentado quando comparada com a técnica de penalização adaptativa.

Tabela 6.47: Sumário dos resultados do APM^{gra} nos problemas de otimização estrutural

Treliça	melhor rodada	média	mediana	desvio padrão	pior rodada	fr
25 barras	488.85	508.14	508.66	$6.62E + 0$	533.22	19
52 barras	1973.42	2273.49	2295.33	$2.22E + 2$	2861.34	19
72 barras	415.80	454.48	491.76	$1.07E + 2$	880.93	20
120 barras	2016.91	2031.41	2089.10	$1.88E + 2$	2867.15	20
200 barras	30129.95	38343.17	39832.08	$1.17E + 4$	85366.26	20

Tabela 6.48: Sumário dos resultados do APM^{grl} nos problemas de otimização estrutural

Treliça	melhor rodada	média	mediana	desvio padrão	pior rodada	fr
25 barras	486.49	490.68	493.47	$4.55E + 0$	527.22	19
52 barras	1903.76	1940.93	2035.39	$1.83E + 2$	2537.70	19
72 barras	380.13	399.10	400.33	$1.79E + 1$	457.65	19
120 barras	2016.77	2021.53	2072.85	$2.23E + 2$	3024.10	20
200 barras	29908.94	34356.30	36570.64	$6.24E + 3$	52897.92	19

7 CONCLUSÕES

O trato de problemas de otimização estrutural com restrição é de vital importância para que se obtenha estruturas confiáveis e econômicas. Modelos da programação matemática e adaptações implementadas em algoritmos evolucionistas se apresentam como ferramentas para tal tarefa.

Os métodos de penalização adaptativa bem como o método do lagrangeano aumentado se destacam quando utilizados para a manipulação das restrições. Dois modelos usando estas técnicas foram desenvolvidos, neste trabalho, em conjunto com um algoritmo genético não-geracional. Destaca-se que, modelos não-geracional não são comumente adaptados para tratar restrições.

Realizou-se um estudo comparativo do algoritmo genético não-geracional com as duas técnicas de penalização. Inicialmente, analisou-se o comportamento de diferentes operadores genéticos com codificação real no AG não-geracional com penalização adaptativa, usando um conjunto de funções(G-Suíte). Em seguida, verificou-se a sensibilidade do parâmetro de atualização da penalidade adaptativa presente no algoritmo. Os resultados foram representados graficamente, para facilitar a determinação dos melhores modelos, utilizando testes denominados “perfil de desempenho”.

Adotando-se a combinação de operadores com melhor desempenho, experimentos numéricos foram realizados com o modelo de lagrangeano aumentado com configuração tradicional e o modelo proposto neste trabalho nas funções(G-Suíte). Mostrando resultado competitivo, adotou-se o modelo desenvolvido como representante dos método de lagrangeano aumentado. Em uma etapa seguinte, aplicou-se, então, os modelos desenvolvidos em problemas da engenharia mecânica, além de estruturas da engenharia civil.

Dentre os resultados apresentados, percebeu-se que no G-Suíte a técnica do lagrangeano aumentado mostrou-se mais eficiente no que se refere as medidas: melhor, mediana e nma. Nos problemas de engenharia mecânica, o modelo de penalização adaptativa no AG não-geracional apresentou resultados muito eficientes em comparação com o método do lagrangeano aumentado e com outros presentes na literatura. Nos testes realizados com as treliças foram analisados o desempenho do AG geracional utilizando o modelo de penalização adaptativa e o desempenho do AG não-geracional utilizando os métodos

de: lagrangeano aumentado e penalização adaptativa. Nessas análises percebeu-se que o AG não geracional juntamente com o método do lagrangeano aumentado obteve melhores resultados quando comparados com o AG geracional.

Deve-se destacar que esta é uma primeira tentativa de utilização de lagrangeano aumentado em AGs não-geracionais. A flexibilidade que se tem, quando combinado com AEs, indica que pode-se obter desempenhos melhores com o aprofundamento das análises e redefinição do modelo, com um maior controle da factibilidade e otimalidade no decorrer da otimização.

Em trabalhos futuros, pretende-se realizar um estudo da combinação de outros operadores testados com o G-Suíte para as estruturas da engenharia civil e mecânica, buscando sempre melhorar o desempenho em relação a outros resultados presentes na literatura. Pretende-se testar o desempenho do método do lagrangeano com os outros operadores utilizados nesse trabalho. Além disso, pode-se verificar o comportamento dessa técnica num AG geracional e comparar com os resultados presentes nesse trabalho. Vale ressaltar que esses experimentos podem ser estendidos a outros tipos de treliças planas ou tridimensionais. Pode-se, ainda, analisar o método do lagrangeano em diferentes tipos de algoritmos baseados em meta-heurísticas.

REFERÊNCIAS

- [1] BARBOSA, H. J. C., LEMONGE, A. C. C., “An Adaptative Penalty Scheme for Steady-State Genetic Algorithms”, *Springer-Verlag Berlin Heidelberg*, pp. 718–729, 2003.
- [2] MICHALEWICZ, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1999.
- [3] CASTRO, L. N. D., TIMMIS, J., “Artificial Imune Systems: A New Computational Intelligence Approach”, *Springer*, v. 1 ed., 2002.
- [4] KENEDY, J., EBERHART, R., “Particle swarm optimization”, *Purdue Scholl of engineering and tecnology*, pp. 1492–1498, 1995.
- [5] DOLAN, E. D., MORÉ, J., “Benchmarking optimization software with performance profiles”, *Mathematical Programming*, v. 91, pp. 201–213, 2002.
- [6] SILVA, E. C. N., “Técnicas de Otimização Aplicadas no Projeto de Peças Mecânicas”, *Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos Escola Politécnica da USP*, 2001.
- [7] SILVA, M. M. D., *Otimização de Estruturas Reticuladas Incluindo Não-Linearidade Geométrica*, Master’s Thesis, Universidade Federal de Juiz de Fora - UFJF, 2011.
- [8] SOUSA, J. A. F. D., *Otimização do Peso de uma grande estrutura espacial em órbita baixa da terra com restrição na frequência fundamental de vibração*, Master’s Thesis, Instituto Nacional de Pesquisas Espaciais, 2002.
- [9] PIERRE, D. A., LOWE, M. J., *Mathematical Programming Via Augmented Lagrangians : An Introduction with Computer Programs*. Addison-Wesley Publishing Company, 1975.
- [10] POWELL, M. J. D., “A method for nonlinear constraints in minimization problems”, *Optimization*, R. Fletcher (ed.), London and New York, Academic Press, pp. 283–298, 1969.

- [11] ROCKAFELLAR, R., “Augmented Lagrange multiplier functions and duality in non-convex programming,” *SIAM Journal on Control and Optimization*, v. 12, pp. 268–285, 1974.
- [12] STARK, R., NICHOLLS, R., “Mathematical Foundations for Design in Civil Engineering Systems. New York,” *McGraw-Hill Book Company*, 1972.
- [13] LEMONGE, A., *Application of Genetic Algorithms in Structural Optimization Problems*, Ph.D. Thesis, Universidade Federal do Rio de Janeiro, 1999.
- [14] LEMONGE, A. C., SILVA, M. M., BARBOSA, H. J., BORGES, C. C., LIMA, E. B., SANTOS, P. P., “A Genetic Algorithm for optimization of Geometrically nonlinear truss structures”, *Simpósio de Mecânica Aplicada e Computacional*, pp. 15, 2010.
- [15] BERNARDINO, H. S., *Hibridização de algoritmos genéticos e sistemas imunológicos artificiais para problemas de otimização com restrições em engenharia*, Master’s Thesis, Universidade Federal de Juiz de Fora, 2008.
- [16] DARWIN, C., *A Origem das Espécies e a Seleção Natural*. 2000.
- [17] MONTES, E. M., *Técnicas Alternativas para o Tratamento de Restrições na Otimização Evolucionária*, Ph.D. Thesis, Centro de Investigação de Estudos Avançados do Instituto Politécnico Nacional - México, 2004.
- [18] HOLLAND, J., *Adaptation in Natural and Artificial Systems*. 1975.
- [19] GOLDBERG, D., *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Co., 1989.
- [20] GOLDBERG, D., L.B., B., HOLLAND, J., “Classifier systems and genetic algorithms”, *Machine Learnings: Paradigms and Methods- MIT/ Press Elsevier*, pp. 235–282, 1989.
- [21] LINDEN, R., *Algoritmos Genéticos*. 2008.
- [22] COSTA, R. D. O., *Algoritmo genético especializado na resolução de problemas com variáveis contínuas e altamente restritos*, Master’s Thesis, Universidade Estadual Paulista, 2009.

- [23] MICHALEWICZ, Z., NAZHIYATH, G., MICHALEWICZ, M., “A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems”, *In Proceedings: Evolutionary Programming*, pp. 305–312 , 1996.
- [24] COELLO, C. A. C., “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art”, *Computer methods in applied mechanics and engineering*, pp. 1245–1287, 2002.
- [25] ROMEIJN, H. E., SMITH, R. L., “Simulated Annealing for Costrained Global Optimization” , , pp. 112–1261983.
- [26] GARRETT, S., “Parameter-free, adaptive clonal selection”, *Congress on Evolutionary Computation, CEC 2004*, v. 1, pp. 1052–1058, 2004.
- [27] WATANABE, K., CAMPELO, F., IGARASHI, H., “Topology optimization based on immune algorithm and multigrid method.” *IEEE Trans. on Magnetics*, v. 43(4), pp. 1637–1640, 2007.
- [28] DORIGO, M.; MANIEZZO, V. C. A., “The Ant System: An Autocatalytic Optimizing Process. Milano, Italy,”, 1991.
- [29] DORIGO, M., STUTZLE, T., “Ant Colony Optimization”, *The Mit Press*, 2004.
- [30] CAPRILLES, P. V. S. Z., FONSECA, L. G. D., BARBOSA, H. J. C., LEMONGE, A. C., “Anty colony algorithms applied to discrete optimization problems”, *Congresso nacional de matemática aplicada e computacional*, pp. 1–8.
- [31] SILVA, A., LIMA, B., JACOB, B., LEMONGE, A., BARBOSA, H., “Solução de Problemas de Otimização com Restrições via um Algoritmo Bio-inspirado utilizando uma Estratégia de Penalização Adaptativa”, *Nono Simpósio de Mecânica Computacional*, pp. 14, 2010.
- [32] KARABOGA, “An idea based on honey bee swarm for numerical optimization”, *Technical report, Erciyes University - Turkey*, 2005.
- [33] JADAAN, O. A., RAJAMANI, L., RAO, C. R., “Adaptative Penalty function for solving constrained evolutionary optimization”, *Journal of Theoretical an Applied Information Technology*, pp. 339–351, 2009.

- [34] KURI-MORALES, A., QUEZADA., C. V., “A universal eclectic genetic algorithm for constrained optimization.” *In Proceedings 6th European Congress on Intelligent Techniques Soft Computing*, pp. 518–522, 1998.
- [35] BACK, T., *Evolutionary Algorithms in Theory and Practice*. New York, 1996.
- [36] HOFFMEISTER, F., J., S., “Problem-independent handling of constraints by use of metric penalty functions”, *Proceedings of the Fifth Annual Conference on Evolutionary Computation .*, pp. 289–294, 1996.
- [37] BACK, T., “Introduction to the special issue: Self-adaptation”, *Evolutionary Computation .*, 2001.
- [38] GOLDBERG, D., “.The Design of Innovation”, *Kluwer Academic Publishers- New York*, v. ISBN 1-4020-7098-5., 2002.
- [39] KAZARLIS, S., PETRIDIS, V., “Varying fitness functions in genetic algorithms: Studying the rate or increase of the dynamic penalty terms”, *In A. E. Eiben, T. Back, M Schoenauer, and H.P. Schwefel, editors Proceedings of the of the 5th Parallel Problem Solving from Nature. Springer-Verlag*, v. 1498, pp. 211–220, 1998.
- [40] KAZARLIS, S. A., PAPADAKIS, S. E., THEOCHARIS, J. B., PETRIDIS, V., “Microgenetic algorithms as generalized hill climbing operators for GA optimization”, *IEEE Transactions on Evolutionary Computation*, pp. 204–217, 2001.
- [41] MICHALEWICZ, Z., ATTIA., N. F., *Evolutionary optimization of constrained problems*. 3rd Annual Conference on Evolutionary Programming, 1994.
- [42] FIGUEREDO, G. P., *Algoritmos Genéticos na Simulação da Evolução das Bibliotecas de Genes do Sistema Imune*, Master’s Thesis, Universidade Federal do Rio de Janeiro - COPPE/UFRJ, 2004.
- [43] COELLO, C. A. C., “Use of a self-adaptive penalty approach for engineering optimization problems”, *Computers in Industry*, pp. 113–127, 2000.
- [44] RICHE, R. G. L., KNOPF-LENOIR, C., HAFTKA, R. T., “A segregated genetic algorithm for constrained structural optimization”, *Sixth International Conference on Genetic Algorithms (ICGA-95)*, pp. 558–565, 1995.

- [45] WU, B., YU, X., “Fuzzy penalty function approach for constrained function optimization with evolutionary algorithms”, *8th International Conference on Neural Information Processing-China*, pp. 299–304, 2001.
- [46] MONTES, F. M., *Constraint-Handling in Evolutionary Optimization*. v. 198. 2009.
- [47] HERNANDEZ, A., MUNOZ, A. AND, V. E., BOTELLO, S., “COPSO: Constrained Optimization via PSO Algorithm.” *Technical Report of the Computer Sciences Department: Centro de Investigacion en Matematicas, Guanajuato, México*, 2007.
- [48] LIANG, J. J., RUNARSSON, T. P., MONTES, E. M., CLERC, M., SUGANTHAN, P. N., COELLO, C. A. C., DEB, K., “Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization”, *Technical Report*, pp. 1–24, 2006.
- [49] HAJELA, P., YOO, J., “Constraint handling in genetic search using expression strategies.” *AIAA Journal*, v. 34(12), pp. 2414–2420, 1996.
- [50] COELLO, C. A. C., CORTES, N. C., “Use of emulations of the immune system to handle constraints in evolutionary algorithms.” *Intelligent Engineerin Systems through Artificial Neural Networks (ANNIE2001)*, v. 11, pp. 141–146, 2001.
- [51] COELLO, C. A. C., CORTES, N. C., “A parallel implementation of an artificial immune system to handle constraints in genetic algorithms: Preliminary results.” *In Proceedings of the Congress on Evolutionary Computation 2002 (CEC2002)*, v. 1, pp. 819–824, 2002.
- [52] RUNARSSON, T. P., “Approximate Evolution Strategy using Stochastic Ranking”, *IEEE Congress on Evolutionary Computation (CEC 2006)*, pp. 1, 2006.
- [53] BEAM, J. C., ALOUANE, A., “A dual genetc algorithm for bounded integer programs”, *The University of Michigan*, pp. 92–53, 1992.
- [54] FARMANI, R., WRIGHT, J. A., “Self-adaptive fitness formulation for constrained optimization”, *IEEE Transactions on Evolutionary Computation*, pp. 7(5):445–455, 2003.

- [55] COIT, D., SMTIH, A., TATE, D., “Adaptative penalty methods for genetic optimization of constrained combinatorial problems”, *Journal on computing*, v. 6(2), pp. 173–182, 1996.
- [56] GEN, M., CHENG, R., “A survey of penalty techniques in genetic algorithms”, *T. Fukuda, T. Furuhashi (Eds.), Proceedings of the 1996 International Conference on Evolutionary Computation, IEEE, Nagoya, Japan.*, pp. 804–809, 1996.
- [57] GEN, M., CHENG, R., “Interval programming using genetic algorithms”, *Proceedings of the Sixth International Symposium on Robotics and Manufacturing*, 1996.
- [58] HAMIDA, S., M., S., “An adaptative algorithm for constrained optimization problems”, *Parallel Problem Solving from Nature PPSN VI, Lecture Notes in Computer Science- Springer:Berlin*, v. 1917, pp. 529–538, 2000.
- [59] BEN HAMIDA, S., SCHOENAUER, M., “ASCHEA: new results using adaptive segregational constraint handling”, *Proceedings of the 2002 Congress on*, v. 1, pp. 884–889, 2002.
- [60] RUNARSSON, T., YAO, X., “Stochastic ranking for constrained evolutionary optimization.” *IEEE Trans. Evotion Computation*, v. 4(3), pp. 284–294, 2000.
- [61] LEMONGE, A. C. C., BARBOSA, H. J. C., “An adaptive penalty scheme for genetic algorithms in structural optimization”, *International Journal for numercial métodos in Engineering*, pp. 703–736, 2004.
- [62] BARBOSA, H. J., LEMONGE, A. C., “An adaptive penalty scheme in genetic algorithms for constrained optimization problems.” *Proc. of the Genetic and Evolutionary Computation Conference New York*, pp. 287–294, 2002.
- [63] DEB, K., “GeneAS: a robust optimal design technique for mechanical component design.” *In Evolutionary Algorithms in Engineering Applications. Dasgupta D, Michalewicz Z(eds.), Springer: Berlin*, pp. 497–514, 1997.
- [64] SRIVASTAVA, S., DEB, K., “A Genetic algorithm based augmented lagrangian method for computationally fast constrained optimization”, *Lecture Notes in Computer Science*, v. 6466/2010, pp. 330–337, 2010.

- [65] SILVA, M. A. D., *Aplicação do Lagrangeano aumentado em otimização estrutural com restrições dinâmicas*, Master's Thesis, Escola Politécnica da Universidade de São Paulo, 1997.
- [66] HESTENES, M. R., "Multiplier and gradient methods", *Journal of Optimization Theory and Applications*, v. 4, pp. 303–320, 1969.
- [67] ADELI, H., CHENG, N. T., "Augmented Lagrangian Genetic Algorithm for Structural Optimization", *Journal of Aerospace Engineering*, v. 6, pp. 315–328, 1993.
- [68] SCHUVERDT, M. L., *Métodos de Lagrangiano aumentado com convergência utilizando a condição de dependência linear constante*, Ph.D. Thesis, Universidade Estadual de Campinas, 2006.
- [69] JANSEN, P., PEREZ, R., "Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach", *Computers and Structures*, pp. 1–15, 2011.
- [70] ROCHA, A. M. A. C., MARTINS, F., FERNANDES, E. M., "An augmented Lagrangian fish swarm based method for global optimization", *Journal of Computation and Applied Mathematics*, pp. 1–19, 2010.
- [71] COSTA, L., ESPIRITO SANTO, I. A., DENYSIUK, R., FERNANDES, E. M. G. P., "Hybridization of a Genetic Algorithm with a Pattern Search Augmented Lagrangian Method", *2nd International Conference on Engineering Optimization Lisbon, Portugal*, 2010.
- [72] WEI LU, P. M., "Augmented Lagrangian Genetic Algorithms for optimal design of Hat-Shaped Cold-Formed Steel Profile", .
- [73] WAH, BENJAMIN W.; CHEN, Y., "Hybrid Constrained Simulated Annealing and Genetic Algorithms for Nonlinear Constrained Optimization", *IEEE*, pp. 925–932, 2001.
- [74] SEDLACZEK, K., EBERHARD, P., "Using augmented Lagrangian particle swarm optimization for constrained problems in engineering", *Springer-Verlag*, pp. 277–286, 2006.

- [75] CASTELANI, E., MARTINEZ, A., SVAITER, J. B. F., “Addressing the greediness phenomenon in nonlinear programming by means of proximal augmented lagrangians”, *Computational Optimization and Applications*, v. 46, pp. 229–245, 2010.
- [76] ROCKAFELLAR, R., “The multiplier method of Hestenes and Powell applied to convex programming”, *Journal of Optimization Theory and Applications*, v. 12, pp. 555–562, 1973.
- [77] BARBOSA, H. J., BERNARDINO, H. S., BARRETO, A. M., “Using Performance Profiles to Analyze the Results of the 2006 CEC Constrained Optimization Competition”, *IEEE World Congress on Computational Intelligence*, pp. 8, 2010.
- [78] BARBOSA, H. J., LEMONGE, A. C., FONSECA, L. G., BERNARDINO, H. S., “Comparing Two Constraint Handling Techniques in a Binary-Coded Genetic Algorithm for Optimization Problems”, *Springer-Verlag Berlin Heidelberg*, pp. 125–134, 2010.
- [79] BARRETO, A. M. S., BERNARDINO, H. S., BARBOSA, H. J. C., “Probabilistic Performance Profiles for the Experimental Evaluation of Stochastic Algorithms”, *GCECO2010 Portland, Oregon, USA.*, pp. 751–758, 2010.
- [80] EIBEN, A., JELASITY, M., “A critical note on experimental research methodology in EC”, *Proceedings of the 2002 Congress on Evolutionary Computation CEC*, pp. 582–587, 2002.
- [81] BERNARDINO, H., BARBOSA, H., LEMONGE, A. C. D. C., “A hybrid genetic algorithm for constrained optimization problems in mechanical engineering.” *In Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 646–653, 2007.
- [82] BERNARDINO, H., BARBOSA, H. J., LEMONGE, A., FONSECA, L. G., “A new hybrid AIS-GA for constrained optimization problems in mechanical engineering.” *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pp. 1455–1462, 2008.

- [83] MONTES, E. M., L., R., “Engineering optimization using a simple evolutionary algorithm.” *In 15th International Conference on Tools with Artificial Intelligence*, pp. 149–156, 2003.
- [84] ERBATUR, F., HASANÇEBİ, O., TÜTÜNCÜ, I., KILÇ, H., “Optimal design of planar and space structures with genetic algorithms”, *Computers & Structures*, v. 75, pp. 209–224, 2000.
- [85] LEMONGE, A. C., BARBOSA, H. J., BORGES, C. C., SILVA, F. B., “Constrained Optimization Problems in Mechanical Engineering Design using a Real-Coded Steady-State Genetic Algorithm”, *CILAMCE - Iberian-Latin-American Congress on Computational Methods in Engineering*, pp. 17, 2010.
- [86] KRISHANMOORTY, C., RAJEEV, S., “Discrete optimization of structures using genetic algorithms”, *Structural Engineering*, v. 118(5), pp. 1233–1250, 1992.
- [87] WU, S. J., CHOW, P. T., “Steady-state genetic algorithms for discrete optimization of trusses”, *Computers & Structures*, pp. 979–991, 1995.
- [88] ARAGON, V. S., ESQUIVEL, S. C., COELLO, C. A. C., “A modified version of a T-Cell Algorithm for constrained optimization problems”, *International Journal for numerical Methods engineering*, v. 84:, pp. 351–378, 2010.
- [89] PEREZ, R., BEHDINAN, K., “Particle swarm optimization in structural design”, *Itech Education an Publishing*, pp. 373–394, 2007.
- [90] ZHU, D., “An improved Templeman’s algorithm for optimum design of trusses with discrete member sizes.” *Engineering Optimization*, v. 9, pp. 303–312, 1986.
- [91] GELLATY, R., BERKE, L., “Optimal structural design”, *Technical Report AFFDL-TR70-165 Air Force Flight Dynamics Laboratory*, 1971.
- [92] VENKAYYA, V., “Design of optimum structures”, *Journal of Computers and Structures*, pp. 265–309, 1971.
- [93] SCHIMIT, L., FARSHI, B., “Some approximation concepts in structural synthesis”, *AIAA Journal*, v. 12, pp. 692–699, 1974.

- [94] SAKA, M., ULKER, M., “Optimum design of geometrically non-linear space trusses”, *Computers and Structures*, v. 41, pp. 1387–1396, 1991.
- [95] EBENAU, C., ROTSSCHAFFER, J., THIERAUF, G., “An advanced evolutionary strategy with an adaptative penalty function for mixed-discrete structural optimisation”, *Advances in Engineering Software*, v. 36, pp. 29–38, 2005.
- [96] CAPRILES, P., FONSECA, L., BARBOSA, H., LEMONGE, A., “Rank-based ant colony algorithms for truss weight minimization with discrete variables”, *Communications in Numerical Methods in Engineering*, v. 26, pp. 553–576, 2007.
- [97] BELEGUNDU, A., *A study of mathematical programming methods for structural optimization*, Ph.D. Thesis, Department of Civil and Environmental USA, 1982.
- [98] LEE, K., GEEM, Z., “A new structural optimization method based on the harmony serach algorithm”, *Computers & Structures*, v. 82, pp. 781–798, 2004.
- [99] SUNAR, M., BELEGUNDU, A., “Trust region methods for structural optimization using exact second order sensitivity”, *International Journal for numerical Methods engineering*, v. 32, pp. 275–293, 1991.

8 APÊNDICE

Sumário estatístico combinação com o operador de mutação randômica

25 rodadas – 50.000 Avaliações – ninser = 3

Tabela 8.1: Experimento 1: Recombinação 1 ponto(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.722	-14.085	-13.967	0.56E+00	-12.434	25	47584
g02	-0.803	-0.740	-0.674	-0.677	3.301E-02	-0.610	25	45575
g03	-1.000	-0.561	-0.233	-0.256	0.14E+00	-0.023	25	26633
g04	-30665.538	-30570.033	-30313.348	-30338.227	1.11E+02	-30093.880	25	36350
g05	5126.496	5142.314	5196.479	5201.425	5.17E+01	5343.584	25	35152
g06	-6961.813	-6694.145	-5946.360	-5874.933	6.56E+02	-3827.560	24	26664
g07	24.306	26.616	37.517	39.263	1.04E+01	64.091	25	45348
g08	-0.095	-0.095	-0.095	-0.095	7.15E-04	-0.092	25	33639
g09	680.630	683.414	690.232	691.360	5.59E+00	704.173	25	43250
g10	7049.248	7795.733	9855.542	9974.259	1.32E+03	13774.778	24	42075
g11	0.749	0.754	0.889	0.896	6.09E-02	0.992	23	12934
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	–	–	–	–	–	0	0
g14	-47.764	–	–	–	–	–	0	0
g15	961.715	–	–	–	–	–	0	0
g16	-1.905	-1.780	-1.619	-1.561	0.19E+00	-1.072	25	43288
g17	8853.539	–	–	–	–	–	0	0
g18	-0.866	-0.737	-0.538	-0.539	9.84E-02	-0.371	22	45247
g19	32.655	76.240	164.269	184.370	8.54E+01	425.848	25	47806
g20	0.097	–	–	–	–	–	0	0
g21	193.724	–	–	–	–	–	0	0
g22	236.430	–	–	–	–	–	0	0
g23	-400.055	–	–	–	–	–	0	0
g24	-5.508	-5.503	-5.452	-5.439	5.25E-02	-5.329	25	30691

Tabela 8.2: Experimento 2: Recombinação 2 pontos(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	píor	frun	nma
g01	-15.000	-14.618	-14.015	-13.819	0.79E+00	-11.288	25	48262
g02	-0.803	-0.760	-0.721	-0.713	2.59E-02	-0.649	25	46383
g03	-1.000	-0.692	-0.170	-0.212	0.18E+00	-0.001	25	18038
g04	-30665.538	-30523.045	-30406.558	-30388.969	9.43E+01	-30188.919	25	34338
g05	5126.496	5139.872	5247.065	5292.063	1.09E+02	5520.051	25	33773
g06	-6961.813	-6770.961	-5371.202	-5434.110	7.90E+02	-4212.069	20	33518
g07	24.306	26.616	37.517	39.263	10.41E+00	64.091	25	45348
g08	-0.095	-0.095	-0.095	-0.087	2.25E-02	-0.025	25	33943
g09	680.630	684.005	691.136	695.139	10.18E+00	725.053	25	42598
g10	7049.248	7757.710	9767.445	9753.829	1.21E+03	13187.745	24	42348
g11	0.749	0.749	0.808	0.845	8.89E-02	0.997	17	27696
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.800	-1.658	-1.635	0.12E+00	-1.370	25	42071
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.785	-0.473	-0.515	9.17E-02	-0.418	24	43785
g19	32.655	78.306	155.473	158.108	5.22E+01	240.072	25	45686
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.503	-5.432	-5.401	8.72E-02	-5.174	25	32701

Tabela 8.3: Experimento 3: Recombinação Discreto(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.640	-14.421	-14.429	0.15E+00	-13.964	25	48694
g02	-0.803	-0.783	-0.767	-0.765	1.24E-02	-0.735	25	45118
g03	-1.000	-0.841	-0.513	-0.471	0.19E+00	-0.103	25	34983
g04	-30665.538	-30587.669	-30396.261	-30422.463	1.09E+02	-30170.116	25	33228
g05	5126.496	5142.535	5219.474	5222.045	5.83E+01	5405.101	25	41922
g06	-6961.813	-6880.693	-5506.036	-5502.479	7.72E+02	-4259.798	21	36799
g07	24.306	25.262	30.033	31.214	4.76E+00	43.113	25	44712
g08	-0.095	-0.095	-0.095	-0.095	1.98E-05	-0.095	25	36028
g09	680.630	682.213	685.512	686.867	3.82E+00	695.934	25	42893
g10	7049.248	7447.632	8341.736	8630.042	9.56E+02	10884.335	25	41830
g11	0.749	0.749	0.905	0.898	8.85E-02	0.999	22	22997
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.840	-1.711	-1.688	0.12E+00	-1.387	25	40796
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.836	-0.546	-0.573	9.50E-02	-0.427	24	42421
g19	32.655	62.548	134.496	142.678	4.70E+01	262.869	25	47807
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.502	-5.468	-5.446	5.15E-02	-5.338	25	30635

Tabela 8.4: Experimento 4: Recombinação Flat(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-13.267	-12.682	-12.689	0.31E+00	-11.967	25	49412
g02	-0.803	-0.758	-0.648	-0.664	5.26E-02	-0.587	25	49063
g03	-1.000	-0.996	-0.966	-0.960	2.79E-02	-0.899	25	30792
g04	-30665.538	-30415.465	-30283.817	-30273.046	8.63E+01	30116.580	25	48589
g05	5126.496	5148.041	5174.579	5177.998	2.63E+01	5266.630	25	43306
g06	-6961.813	-6191.659	-5094.353	-5089.107	4.52E+02	-4318.308	25	42620
g07	24.306	28.270	32.114	32.377	2.21E+00	39.171	25	49185
g08	-0.095	-0.095	-0.095	-0.095	2.90E-17	-0.095	25	26990
g09	680.630	683.037	685.457	685.866	2.21E+00	692.911	25	49146
g10	7049.248	7413.418	8215.957	8684.629	1.10E+03	11679.071	25	48488
g11	0.749	0.751	0.886	0.871	6.52E-02	0.987	25	5080
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.446	0.984	0.875	0.19E+00	0.999	24	37658
g14	-47.764	-39.722	-38.806	-38.619	0.67E+00	-37.172	19	48261
g15	961.715	962.645	965.519	965.445	1.43E+00	967.499	24	31251
g16	-1.905	-1.692	-1.349	-1.331	0.18E+00	-1.039	25	48211
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.561	-0.473	-0.476	5.19E-02	-0.361	12	49064
g19	32.655	197.553	228.821	229.046	1.85E+01	278.584	25	49517
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.505	-5.481	-5.474	1.62E-02	-5.448	25	46601

Tabela 8.5: Experimento 5: Recombinação BLX(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.680	-14.267	-13.721	1.29E+00	-8.998	25	49739
g02	-0.803	-0.733	-0.694	-0.680	3.74E-02	-0.587	25	49703
g03	-1.000	-0.946	-0.876	-0.861	6.86E-02	-0.689	25	45999
g04	-30665.538	-30641.614	-30492.646	-30452.800	1.37E+02	-30167.075	25	49595
g05	5126.496	5130.178	5162.229	5167.431	3.36E+01	5261.356	25	49572
g06	-6961.813	-6961.810	-6929.641	-6775.982	3.96E+02	-5197.205	25	49687
g07	24.306	26.777	29.740	34.015	1.86E+01	122.960	25	49471
g08	-0.095	-0.095	-0.095	-0.095	2.97E-17	-0.095	25	24818
g09	680.630	681.944	687.093	687.322	3.72E+00	696.787	25	49668
g10	7049.248	7211.562	8019.782	8433.907	1.08E+03	11275.691	25	49420
g11	0.749	0.749	0.761	0.800	7.50E-02	0.999	25	32069
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.581	0.891	1.052	0.53E+00	2.764	24	49648
g14	-47.764	-41.915	-40.077	-40.117	0.99E+00	-38.277	8	49698
g15	961.715	962.582	966.361	966.415	2.52E+00	971.818	25	49462
g16	-1.905	-1.892	-1.761	-1.725	0.14E+00	-1.353	25	49612
g17	8853.539	8871.618	8936.288	8916.238	3.87E+01	8940.808	3	41902
g18	-0.866	-0.844	-0.489	-0.523	0.10E+00	-0.422	23	49586
g19	32.655	90.991	163.341	175.879	5.08E+01	305.142	25	49511
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.508	-5.507	-5.507	1.99E-03	-5.499	25	49716

Tabela 8.6: Experimento 6: Recombinação SBX(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.993	-14.444	-13.520	1.50E+00	-9.999	25	49754
g02	-0.803	-0.707	-0.639	-0.638	4.08E-02	-0.551	25	49650
g03	-1.000	-0.826	-0.612	-0.600	0.17E+00	-0.112	25	49549
g04	-30665.538	-30665.538	-30665.483	-30664.827	1.41E+00	-30660.079	25	49684
g05	5126.496	5126.523	5129.454	5138.867	2.11E+01	5211.853	25	49715
g06	-6961.813	-6961.810	-6961.810	-6961.810	7.19E-12	-6961.810	25	44977
g07	24.306	26.278	31.877	32.711	5.47E+00	44.132	25	49585
g08	-0.095	-0.095	-0.095	-0.095	7.88E-17	-0.095	25	33424
g09	680.630	681.020	685.318	686.552	9.39E+00	730.276	25	49779
g10	7049.248	7294.833	8349.289	8288.033	6.74E+02	9321.541	25	49809
g11	0.749	0.749	0.755	0.781	4.40E-02	0.901	25	47985
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.126	0.997	1.219	0.79E+00	3.537	25	49760
g14	-47.764	-45.821	-39.997	-40.289	2.60E+00	-35.516	24	49703
g15	961.715	962.476	963.907	964.683	2.56E+00	972.765	25	49464
g16	-1.905	-1.904	-1.896	-1.883	2.80E-02	-1.800	25	49760
g17	8853.539	8907.059	8948.512	9021.730	1.36E+02	9291.456	9	48380
g18	-0.866	-0.847	-0.492	-0.558	0.14E+00	-0.339	23	49670
g19	32.655	39.966	95.232	105.056	4.41E+01	188.623	25	49619
g20	0.097	11.890	16.691	15.915	2.86E+00	18.389	4	147552
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-261.968	97.456	375.870	4.74E+02	899.973	11	48245
g24	-5.508	-5.508	-5.508	-5.508	4.05E-16	-5.508	25	44604

Tabela 8.7: Experimento 7: Recombinação Geométrico(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	piores	frun	nma
g01	-15.000	-11.683	-10.758	-10.731	0.62E+00	-8.609	25	48888
g02	-0.803	-0.773	-0.744	-0.737	2.53E-02	-0.663	25	48608
g03	-1.000	-0.957	-0.899	-0.871	7.81E-02	-0.620	25	21983
g04	-30665.538	-30438.856	-30208.417	-30218.515	1.30E+02	-30017.987	25	46580
g05	5126.496	-	-	-	-	-	0	0
g06	-6961.813	-6574.379	-4597.830	-4872.055	6.78E+02	-4140.322	25	40508
g07	24.306	-	-	-	-	-	0	0
g08	-0.095	-0.095	-0.095	-0.095	1.94E-11	-0.095	25	26183
g09	680.630	-	-	-	-	-	0	0
g10	7049.248	7391.989	7974.295	8620.361	1.55E+03	12868.053	25	47334
g11	0.749	-	-	-	-	-	0	0
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.636	-1.261	-1.291	0.17E+00	-0.990	25	46884
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-	-	-	-	-	0	0
g19	32.655	169.688	255.602	260.246	3.96E+01	365.231	25	48896
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.507	-5.498	-5.492	1.55E-02	-5.443	25	32655

Tabela 8.8: Experimento 8: Recombinação Wrihth(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.999	-12.000	-11.959	1.98E+00	-9.000	25	37564
g02	-0.803	-0.735	-0.645	-0.642	5.05E-02	-0.539	25	49857
g03	-1.000	-0.891	-0.607	-0.599	0.16E+00	-0.278	25	49754
g04	-30665.538	-30665.538	-30665.538	-30665.538	1.11E-11	-30665.538	25	28662
g05	5126.496	5126.498	5126.498	5126.498	1.84E-09	5126.498	25	49689
g06	-6961.813	-6961.810	-6961.810	-6961.810	4.06E-12	-6961.810	20	28165
g07	24.306	25.190	26.913	28.260	2.81E+00	33.612	25	49781
g08	-0.095	-0.095	-0.095	-0.095	1.39E-16	-0.095	25	29884
g09	680.630	680.631	680.692	680.829	0.37E+00	682.388	25	49839
g10	7049.248	7391.989	7974.295	8620.361	1.55E+03	12868.053	25	47334
g11	0.749	0.749	0.750	0.832	0.11E+00	1.000	25	39028
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.426	0.854	0.824	0.26E+00	1.621	25	49876
g14	-47.764	-45.854	-45.529	-43.848	2.62E+00	-38.531	25	49770
g15	961.715	962.476	963.365	964.689	2.78E+00	970.685	25	49189
g16	-1.905	-1.905	-1.905	-1.905	1.98E-06	-1.905	25	49851
g17	8853.539	8860.524	8978.974	9003.378	1.11E+02	9260.415	25	49343
g18	-0.866	-0.865	-0.840	-0.763	0.12E+00	-0.499	25	49775
g19	32.655	36.198	48.607	53.371	1.56E+01	88.186	25	45084
g20	0.097	4.80	8.886	8.403	1.75E+00	10.193	8	69440
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-99.446	149.941	305.018	4.18E+02	900.000	10	40114
g24	-5.508	-5.508	-5.508	-5.508	0.00E+00	-5.508	25	26711

Tabela 8.9: Experimento 9: Recombinação LSX(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-6.868	-3.995	-3.563	$2.22E+00$	0.246	11	12086
g02	-0.803	-0.642	-0.550	-0.551	$5.46E-02$	-0.450	25	40053
g03	-1.000	–	–	–	–	–	0	0
g04	-30665.538	-30421.744	-29929.008	-29915.828	$1.89E+02$	-29574.377	25	6833
g05	5126.496	6036.910	6677.897	6661.756	$5.75E+02$	7254.322	4	8218
g06	-6961.813	–	–	–	–	–	0	0
g07	24.306	–	–	–	–	–	0	0
g08	-0.095	-0.095	-0.088	-0.083	$1.20E-02$	-0.057	25	20343
g09	680.630	740.965	830.047	840.151	$8.60E+01$	1094.814	25	2958
g10	7049.248	–	–	–	–	–	0	0
g11	0.749	0.862	0.999	0.972	$5.26E-02$	1.000	7	15859
g12	-1.000	-1.000	-1.000	-1.000	$0.00E+00$	-1.000	25	1000
g13	0.053	–	–	–	–	–	0	0
g14	-47.764	–	–	–	–	–	0	0
g15	961.715	–	–	–	–	–	0	0
g16	-1.905	-1.459	-1.299	-1.299	$0.22E+00$	-1.140	2	500
g17	8853.539	–	–	–	–	–	0	0
g18	-0.866	–	–	–	–	–	0	0
g19	32.655	232.445	878.264	939.089	$4.14E+02$	2020.465	25	4982
g20	0.097	–	–	–	–	–	0	0
g21	193.724	–	–	–	–	–	0	0
g22	236.430	–	–	–	–	–	0	0
g23	-400.055	–	–	–	–	–	0	0
g24	-5.508	-5.473	-5.240	-5.215	$0.15E+00$	-4.926	25	2050

Tabela 8.10: Experimento 10: Recombinação pais múltiplos(0,8) e Mutação Randômica(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-13.892	-10.031	-9.814	2.17E+00	-6.000	25	37788
g02	-0.803	-0.726	-0.671	-0.663	3.86E-02	-0.572	25	14533
g03	-1.000	-0.274	-0.058	-0.078	7.77E-02	-0.000	24	5936
g04	-30665.538	-30467.234	-30293.961	-30263.504	1.39E+02	-29965.857	25	6763
g05	5126.496	5252.868	5507.637	5517.660	1.49E+02	5807.759	25	3793
g06	-2737.298	-	-	-	-	-	0	0
g07	24.306	35.930	63.866	73.114	3.09E+01	148.641	25	13053
g08	-0.095	-0.084	-0.021	-0.029	3.10E-02	0.027	25	975
g09	680.630	702.587	724.932	729.419	2.18E+01	784.828	25	7594
g10	7049.248	7879.237	10857.179	10897.111	1.39E+03	13862.588	20	20120
g11	0.862	-	-	-	-	-	20	0
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.775	-1.558	-1.496	0.21E+00	-1.051	25	14223
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.372	-0.241	-0.226	0.10E+00	-0.105	5	11800
g19	32.655	106.697	249.179	273.314	1.19E+02	584.598	25	33743
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.481	-5.218	-5.210	0.15E+00	-4.90	25	1205

Sumário estatístico combinação com o operador de mutação muhlenblein

25 rodadas – 50.000 avaliações – ninser=3

Tabela 8.11: Experimento 11: Recombinação 1 Ponto(0,8) e Mutação Muhlenblein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-13.359	-7.234	-7.888	2.26E+00	-5.570	25	37122
g02	-0.803	-0.701	-0.612	-0.613	4.55E-02	-0.482	25	13526
g03	-1.000	-0.434	-0.074	-0.112	0.11E+00	-1.11E-03	25	7685
g04	-30665.538	-30477.500	-30249.069	-30245.842	1.05E+02	-30034.209	25	9226
g05	5126.496	5190.309	5337.585	5370.626	1.24E+02	5636.192	25	5438
g06	-6961.813	-6669.107	-5205.304	-4939.509	1.44E+03	-1477.754	22	2722
g07	24.306	36.535	58.719	65.404	1.86E+01	109.897	25	12803
g08	-0.095	-0.095	-0.093	-0.090	8.15E-03	-0.059	25	2891
g09	680.630	687.711	708.071	713.184	1.89E+01	757.675	25	9188
g10	7049.248	9689.511	10857.755	10982.872	1.17E+03	12526.466	4	15624
g11	0.749	0.752	0.860	0.864	9.56E-02	0.997	15	2306
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	–	–	–	–	–	0	0
g14	-47.764	–	–	–	–	–	0	0
g15	961.715	–	–	–	–	–	0	0
g16	-1.905	-1.675	-1.463	-1.442	0.14E+00	-0.996	25	8768
g17	8853.539	–	–	–	–	–	0	0
g18	-0.866	-0.405	-0.283	-0.271	9.17E-02	-0.146	6	10325
g19	32.655	87.167	237.537	291.358	1.75E+02	798.573	25	37207
g20	0.097	–	–	–	–	–	0	0
g21	193.724	–	–	–	–	–	0	0
g22	236.430	–	–	–	–	–	0	0
g23	-400.055	–	–	–	–	–	0	0
g24	-5.508	-5.499	-5.378	-5.368	8.86E-02	-5.198	25	2941

Tabela 8.12: Experimento 12: Recombinação 2 Pontos e Mutação Muhlenblein

fcn	ótimo	melhor	mediana	média	dv. padrão	pioor	frun	nma
g01	-15.000	-11.710	-8.149	-8.574	1.91E+00	-5.471	25	36866
g02	-0.803	-0.726	-0.671	-0.663	3.86E-02	-0.572	25	14533
g03	-1.000	-0.577	-0.077	-0.127	0.14E+00	-0.000	22	7305
g04	-30665.538	-30467.234	-30294.432	-30266.389	1.32E+02	-29965.857	25	8276
g05	5126.496	5252.868	5525.669	5523.199	1.49E+02	5807.759	25	3712
g06	-6961.813	-	-	-	-	-	0	0
g07	24.306	35.927	73.557	80.560	4.68E+01	226.126	25	12540
g08	-0.095	-0.084	-0.021	-0.029	3.09E-02	0.027	25	972
g09	680.630	702.587	724.932	729.419	2.18E+01	784.828	25	7594
g10	7049.248	8853.009	11292.602	11077.276	1.54E+03	13733.896	12	24807
g11	0.749	-	-	-	-	-	0	0
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.827	-1.512	-1.466	0.25E+00	-0.647	25	11902
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.503	-0.160	-0.211	0.16E+00	-0.091	5	10345
g19	32.655	106.697	249.179	273.314	1.19E+02	584.598	25	33743
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.481	-5.218	-5.210	0.15E+00	-4.902	25	1205

Tabela 8.13: Experimento 13: Recombinação Discreto(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-13.805	-11.550	-11.704	1.53E+00	-7.497	25	21695
g02	-0.803	-0.758	-0.720	-0.719	2.53E-02	-0.660	25	15078
g03	-1.000	-0.570	-0.128	-0.213	0.19E+00	-0.000	20	9285
g04	-30665.538	-30490.989	-30312.542	-30315.215	1.10E+02	-30013.607	25	6001
g05	5126.496	5203.026	5438.401	5397.920	9.99E+01	5592.279	25	4047
g06	-6961.813	-6200.293	-4563.359	-3860.141	1.65E+03	-1402.769	12	2045
g07	24.306	37.799	47.558	50.797	9.91E+00	71.290	25	10557
g08	-0.095	-0.095	-0.088	-0.081	1.77E-02	-0.019	25	2356
g09	680.630	687.336	702.290	708.957	1.65E+01	751.358	25	7621
g10	7049.248	8351.007	10161.467	10871.297	2.05E+03	15917.440	17	12446
g11	0.749	0.753	0.967	0.911	0.10E+00	0.996	5	1586
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.685	-1.497	-1.488	0.15E+00	-1.206	25	7340
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.623	-0.334	-0.344	0.12E+00	-0.167	16	9518
g19	32.655	111.876	174.951	183.503	5.73E+01	339.493	25	28201
g20	0.097	23.319	24.686	24.506	0.88E+00	25.335	4	31358
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.479	-5.358	-5.355	7.50E-02	-5.214	25	2324

Tabela 8.14: Experimento 14: Recombinação Flat(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-9.107	-7.202	-7.142	0.99E+00	-5.024	25	49769
g02	-0.803	-0.410	-0.384	-0.383	1.55E-02	-0.354	25	49303
g03	-1.000	-0.988	-0.930	-0.927	5.00E-02	-0.800	25	23423
g04	-30665.538	-30381.180	-30059.206	-30096.626	1.33E+02	-29917.996	25	46466
g05	5126.496	5203.026	5438.401	5397.920	9.99E+01	5592.279	25	4047
g06	-6961.813	-6161.240	-3087.997	-3578.636	1.24E+03	-2122.148	25	33438
g07	24.306	55.215	80.075	182.323	2.24E+02	1068.048	25	49508
g08	-0.095	-0.095	-0.095	-0.095	3.03E-17	-0.095	25	21882
g09	680.630	685.960	693.824	708.550	2.55E+01	755.678	25	49751
g10	7049.248	7985.551	9237.996	9375.304	1.04E+03	11771.157	14	45828
g11	0.749	0.751	0.855	0.849	6.96E-02	0.961	25	23130
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.534	0.970	1.086	0.66E+00	3.066	22	38566
g14	-47.764	-39.420	-37.498	-37.082	1.50E+00	-33.402	23	48081
g15	961.715	962.528	966.061	966.010	2.05E+00	969.484	25	33076
g16	-1.905	-1.567	-1.341	-1.340	0.16E+00	-1.029	18	49649
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.223	-0.184	-0.153	8.91E-02	-0.053	3	49736
g19	32.655	263.116	467.101	437.378	1.06E+02	660.954	25	49703
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.498	-5.395	-5.383	8.43E-02	-5.132	25	39682

Tabela 8.15: Experimento 15: Recombinação BLX(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-10.617	-7.919	-8.058	1.35E+00	-5.459	25	49723
g02	-0.803	-0.565	-0.358	-0.368	4.74E-02	-0.320	25	49653
g03	-1.000	-0.948	-0.782	-0.764	0.12E+00	-0.470	25	49630
g04	-30665.538	-30625.565	-30325.830	-30275.986	2.39E+02	-29751.622	25	49714
g05	5126.496	5131.999	5205.570	5243.307	9.74E+01	5517.666	25	49778
g06	-6961.813	-6961.802	-6902.835	-6554.642	8.62E+02	-2785.608	25	49745
g07	24.306	52.855	88.933	269.499	3.24E+02	1213.056	25	49775
g08	-0.095	-0.095	-0.095	-0.095	3.22E-17	-0.095	25	22420
g09	680.630	689.404	721.313	725.584	2.27E+01	776.557	25	49787
g10	7049.248	7638.469	8857.832	9205.593	1.07E+03	11863.872	14	49759
g11	0.749	0.749	0.758	0.819	8.92E-02	0.999	25	44208
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.140	0.934	1.202	1.30E+00	5.857	18	49821
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	962.476	966.930	966.232	2.87E+00	971.724	25	49719
g16	-1.905	-1.827	-1.580	-1.580	0.16E+00	-1.200	25	49827
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.321	-0.189	-0.183	9.37E-02	-0.040	9	49787
g19	32.655	136.950	379.993	381.048	1.29E+02	660.104	25	49661
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.508	-5.506	-5.491	3.34E-02	-5.357	25	49740

Tabela 8.16: Experimento 16: Recombinação SBX((0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.900	-9.993	-10.256	1.98E+00	-6.000	25	48088
g02	-0.803	-0.526	-0.383	-0.399	5.17E-02	-0.328	25	49777
g03	-1.000	-0.919	-0.625	-0.605	0.17E+00	-0.083	25	49661
g04	-30665.538	-30665.538	-30665.413	-30608.316	1.47E+02	-30082.246	25	49707
g05	5126.496	5126.506	5151.366	5176.936	6.80E+01	5341.896	25	49701
g06	-6961.813	-6961.810	-6961.810	-6961.810	4.87E-11	-6961.810	25	42436
g07	24.306	39.224	63.653	113.805	1.63E+02	783.934	25	49802
g08	-0.095	-0.095	-0.095	-0.095	3.16E-17	-0.095	25	25564
g09	680.630	685.575	719.672	724.865	3.72E+01	841.493	25	49859
g10	7049.248	7887.587	9014.665	9530.396	2.64E+03	21744.176	25	49711
g11	0.749	0.749	0.753	0.782	5.28E-02	0.917	25	45805
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.220	0.988	1.058	0.64E+00	3.217	25	49625
g14	-47.764	-45.825	-40.453	-40.359	3.07E+00	-34.221	19	49780
g15	961.715	962.476	964.283	965.284	3.00E+00	971.442	25	49767
g16	-1.905	-1.898	-1.868	-1.799	0.13E+00	-1.433	25	49675
g17	8853.539	8871.018	8888.862	8927.675	8.31E+01	9023.143	3	49980
g18	-0.866	-0.715	-0.381	-0.393	0.15E+00	-0.101	15	49803
g19	32.655	65.440	230.967	282.559	1.79E+02	715.916	25	49704
g20	0.097	0.679	13.783	11.390	7.96E+00	23.655	16	55303
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-0.046	99.997	339.981	4.27E+02	899.998	5	44696
g24	-5.508	-5.508	-5.508	-5.508	00E+00	-5.508	25	40577

Tabela 8.17: Experimento 17: Recombinação Geométrico(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-3.677	-2.609	-2.504	0.75E+00	-0.522	25	49229
g02	-0.803	-0.468	-0.429	-0.427	2.18E-02	-0.388	25	37360
g03	-1.000	-0.953	-0.659	-0.593	0.25E+00	-0.147	14	6348
g04	-30665.538	-30322.061	-29948.424	-29939.328	1.59E+02	-29594.118	25	11320
g05	5126.496	–	–	–	–	–	0	0
g06	-6961.813	-6750.945	-3217.734	-3388.464	1.10E+03	-1964.068	24	4667
g07	24.306	–	–	–	–	–	0	0
g08	-0.095	-0.095	-0.095	-0.095	3.95E-17	-0.095	25	11883
g09	680.630	–	–	–	–	–	0	0
g10	7049.248	8603.013	8658.300	8658.300	7.81E+01	8713.586	2	33088
g11	0.749	–	–	–	–	–	0	0
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	–	–	–	–	–	0	0
g14	-47.764	-34.126	-31.842	-31.842	3.23E+00	-29.558	2	44074
g15	961.715	962.581	966.253	966.698	2.38E+00	970.470	11	10379
g16	-1.905	-1.620	-1.193	-1.263	0.21E+00	-0.976	17	22750
g17	8853.539	–	–	–	–	–	0	0
g18	-0.866	–	–	–	–	–	0	0
g19	32.655	216.364	457.524	435.988	1.20E+02	684.648	25	48976
g20	0.097	–	–	–	–	–	0	0
g21	193.724	–	–	–	–	–	0	0
g22	236.430	–	–	–	–	–	0	0
g23	-400.055	–	–	–	–	–	0	0
g24	-5.508	-5.497	-5.383	-5.363	0.10E+00	-5.172	25	3322

Tabela 8.18: Experimento 18: Recombinação Wrigth(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.996	-10.000	-10.232	2.29E+00	-6.000	25	25412
g02	-0.803	-0.577	-0.447	-0.437	8.11E-02	-0.292	25	49881
g03	-1.000	-0.862	-0.580	-0.558	0.16E+00	-0.204	25	46397
g04	-30665.538	-30665.538	-30665.538	-30665.538	1.11E-11	-30665.538	25	23529
g05	5126.496	-	-	-	-	-	0	0
g06	-6961.813	-6961.810	-6961.810	-6961.810	5.53E-12	-6961.810	21	23101
g07	24.306	27.653	39.972	41.509	7.64E+00	59.018	25	49858
g08	-0.095	-0.095	-0.095	-0.095	3.35E-17	-0.095	25	24938
g09	680.630	680.741	683.812	685.925	6.84E+00	709.884	25	49826
g10	7049.248	7159.143	7574.983	7934.178	7.93E+02	10847.951	25	49813
g11	0.749	0.749	0.999	0.890	0.12E+00	1.000	25	23297
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	0.107	0.986	1.131	0.61E+00	2.913	25	49843
g14	-47.764	-45.840	-42.798	-42.620	3.20E+00	-34.341	25	49788
g15	961.715	962.476	963.537	965.300	3.33E+00	970.844	25	46840
g16	-1.905	-1.905	-1.905	-1.904	1.36E-03	-1.901	25	49809
g17	8853.539	8880.252	8966.993	9010.229	1.17E+02	9284.358	25	49066
g18	-0.866	-0.836	-0.543	-0.591	0.18E+00	-0.124	21	49845
g19	32.655	33.836	52.315	58.065	2.33E+01	145.667	25	48827
g20	0.097	11.052	18.137	18.018	2.73E+00	22.028	14	49289
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-199.989	424.945	458.308	4.39E+02	900.000	12	25190
g24	-5.508	-5.508	-5.508	-5.508	0.00E+00	-5.508	25	22238

Tabela 8.19: Experimento 19: Recombinação LSX(0,8) e Mutação de Muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pioor	frun	nma
g01	-15.000	-5.000	-4.000	-4.000	0.89E+00	-3.000	6	4446
g02	-0.803	-0.510	-0.405	-0.414	4.02E-02	-0.350	25	46083
g03	-1.000	-	-	-	-	-	0	0
g04	-30665.538	-30392.671	-29820.143	-29879.291	225.522	-29504.198	25	1358
g05	5126.496	-	-	-	-	-	0	0
g06	-6961.813	-	-	-	-	-	0	0
g07	24.306	-	-	-	-	-	0	0
g08	-0.095	-0.095	-0.083	-0.080	1.34E-02	-0.046	25	15379
g09	680.630	710.783	760.139	790.442	7.00E+00	939.444	25	2894
g10	7049.248	-	-	-	-	-	0	0
g11	0.749	-	-	-	-	-	0	0
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.459	-1.299	-1.299	0.22E+00	-1.140	2	500
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-	-	-	-	-	0	0
g19	32.655	112.500	1092.871	1018.629	3.65E+02	1667.397	25	2274
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.473	-5.189	-5.154	0.17E+00	-4.776	25	880

Tabela 8.20: Experimento 20: Recombinação de pais múltiplos(0,8) e Mutação de muhlenbein(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.357	-12.083	-12.255	1.19E+00	-9.305	25	19779
g02	-0.803	-0.776	-0.753	-0.754	1.44E-02	-0.710	25	18643
g03	-1.000	-0.585	-0.121	-0.160	0.151	-1.65E-04	20	9421
g04	-30665.538	-30526.108	-30346.441	-30328.426	1.29E+02	-30074.052	25	6478
g05	5126.496	-	-	-	-	-	0	0
g06	-6961.813	-6418.299	-4751.618	-4469.902	1.43E+03	-1323.641	18	2347
g07	24.306	33.823	42.045	45.010	9.44E+00	65.392	25	10430
g08	-0.095	-0.095	-0.0898	-0.082	1.79E-02	-0.028	25	2471
g09	680.630	688.062	699.931	702.054	1.09E+01	732.668	25	7824
g10	7049.248	8317.600	10304.379	10205.975	1.15E+03	12163.704	20	14366
g11	0.749	0.862	0.956	0.936	5.24E-02	0.997	8	1911
g12	-1.000	-1.000	-1.000	-1.000	0.00E+00	-1.000	25	1000
g13	0.053	-	-	-	-	-	0	0
g14	-47764	-	-	-	-	-	0	0
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.728	-1.464	-1.471	0.16E+00	-1.072	24	8563
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.619	-0.418	-0.429	0.11E+00	-0.234	18	10635
g19	32.655	99.131	169.588	179.201	6.58E+01	405.879	25	25405
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.478	-5.361	-5.333	0.10E+00	-5.088	25	2375

Sumário estatístico combinações com o operador de mutação não uniforme

25 rodadas – 50.000 avaliações – 3 atualizações

Tabela 8.21: Experimento 21: Recombinação 1 Ponto(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.449	-13.880	-13.632	0.84E+00	-11.274	25	49749
g02	-0.803	-0.737	-0.667	-0.668	3.75E-02	-0.601	25	49762
g03	-1.000	-0.587	-0.236	-0.247	0.12E+00	-0.051	25	49628
g04	-30665.538	-30591.848	-30358.395	-30377.755	1.21E+02	-30193.921	25	49604
g05	5126.496	5128.664	5170.819	5197.852	7.78E+01	5449.747	25	49568
g06	-6961.813	-6951.567	-6697.115	-6590.904	5.18E+02	-4306.066	25	49544
g07	24.306	25.102	31.687	33.950	9.75E+00	71.858	25	49705
g08	-0.095	-0.095	-0.095	-0.095	3.69E-15	-0.095	25	49619
g09	680.630	682.989	689.580	689.788	6.29E+00	707.862	25	49739
g10	7049.248	7659.068	8512.564	9067.744	1.25E+03	11570.838	23	49538
g11	0.749	0.757	0.889	0.886	6.33E-02	0.992	25	48979
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.543	0.989	0.935	0.11E+00	0.999	25	49803
g14	-47.764	-43.956	-40.394	-39.849	2.00E+00	-35.458	25	49702
g15	961.715	962.508	964.141	965.538	2.92E+00	971.657	22	49013
g16	-1.905	-1.847	-1.681	-1.642	0.14E+00	-1.385	25	49465
g17	8853.539	8896.476	8916.336	8916.336	2.80E+01	8936.195	2	49344
g18	-0.866	-0.862	-0.613	-0.622	0.11E+00	-0.479	25	49814
g19	32.655	66.011	119.092	123.485	31.60E+00	193.854	25	49850
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	57.260	124.966	140.184	8.23E+01	253.543	4	49854
g24	-5.508	-5.508	-5.488	-5.472	4.23E-02	-5.334	25	49176

Tabela 8.22: Experimento 22: Recombinação 2 Pontos(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pioor	frun	nma
g01	-15.000	-14.567	-14.141	-13.912	0.68E+00	-12.044	25	49766
g02	-0.803	-0.767	-0.707	-0.707	3.43E-02	-0.653	25	49779
g03	-1.000	-0.549	-0.253	-0.268	0.141	-3.16E-02	25	49662
g04	-30665.538	-30630.779	-30423.614	-30433.018	1.16E+02	-30204.747	25	49520
g05	5126.496	5132.753	5193.624	5212.819	7.24E+01	5391.896	25	49537
g06	-6961.813	-6745.028	-6103.948	-5674.275	1.47E+03	-1282.328	22	47177
g07	24.306	24.810	30.341	32.615	8.15E+00	56.481	25	49812
g08	-0.095	-0.095	-0.095	-0.087	2.21E-02	-0.029	25	49356
g09	680.630	681.576	687.220	688.429	5.28E+00	702.647	25	49732
g10	7049.248	7233.316	8494.157	9023.383	1.91E+03	16619.564	25	49697
g11	0.749	0.749	0.896	0.880	8.97E-02	0.997	25	49503
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.169	0.993	0.991	0.32E+00	2.213	25	49764
g14	-47.764	-44.181	-40.203	-40.436	1.79E+00	-37.223	25	49727
g15	961.715	962.551	967.558	967.150	3.24E+00	972.245	22	49398
g16	-1.905	-1.837	-1.713	-1.658	0.15E+00	-1.162	25	49762
g17	8853.539	8943.317	9059.857	9059.857	1.64E+02	9176.397	2	49983
g18	-0.866	-0.865	-0.620	-0.617	0.12E+00	-0.410	25	49764
g19	32.655	72.124	127.331	132.145	3.90E+01	219.469	25	49747
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	60.314	191.640	191.640	185.723	3.22E+02	2	49552
g24	-5.508	-5.501	-5.472	-5.410	0.11E+00	-5.088	25	49490

Tabela 8.23: Experimento 23: Recombinação BLX(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	worst	frun	nma
g01	-15.000	-14.865	-14.711	-14.633	0.43E+00	-12.587	25	49841
g02	-0.803	-0.787	-0.774	-0.768	1.72E-02	-0.728	25	49679
g03	-1.000	-0.865	-0.516	-0.489	0.20E+00	-0.149	25	49675
g04	-30665.538	-30655.159	-30499.127	-30486.411	1.04E+02	-30203.835	25	49706
g05	5126.496	5126.807	5141.220	5163.567	4.24E+01	5268.698	25	49611
g06	-6961.813	-6870.217	-6601.811	-6505.878	3.24E+02	-5875.913	25	49688
g07	24.306	24.773	28.656	29.606	3.77E+00	38.532	25	49802
g08	-0.095	-0.095	-0.095	-0.095	4.06E-16	-0.095	25	49711
g09	680.630	681.327	684.780	684.942	3.09E+00	692.552	25	49710
g10	7049.248	7150.560	8294.228	8273.407	8.90E+02	10210.053	25	49742
g11	0.749	0.749	0.889	0.883	8.93E-02	0.998	25	49517
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.821	0.998	0.979	3.97E-02	0.999	25	49752
g14	-47.764	-43.913	-38.587	-39.172	1.80E+00	-36.734	25	49612
g15	961.715	962.566	963.873	965.465	3.22E+00	972.246	24	49673
g16	-1.905	-1.875	-1.748	-1.757	7.67E-02	-1.607	25	49798
g17	8853.539	8905.768	8959.761	9024.790	1.25E+02	9238.178	12	49721
g18	-0.866	-0.774	-0.609	-0.593	7.67E-02	-0.439	25	49827
g19	32.655	61.610	95.214	96.800	2.15E+01	143.935	25	49771
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-52.526	160.480	160.480	3.01E+02	373.488	2	49393
g24	-5.508	-5.507	-5.484	-5.465	4.53E-02	-5.315	25	49705

Tabela 8.24: Experimento 24: Recombinação Flat(0,8)- Mutaç o de n o-uniforme(0,2)

fcn	�timo	melhor	mediana	m�dia	st. dev.	worst	frun	nma
g01	-15.000	-9.237	-7.367	-7.372	0.89E+00	-5.841	25	49598
g02	-0.803	-0.437	-0.417	-0.418	1.21E-02	-0.388	25	49693
g03	-1.000	-0.988	-0.972	-0.966	1.71E-02	-0.929	25	49401
g04	-30665.538	-30521.379	-30284.544	-30284.607	1.05E+02	-30127.536	25	49508
g05	5126.496	5141.392	5174.473	5175.502	2.62E+01	5248.948	25	49592
g06	-6961.813	-6541.150	-6073.952	-6118.488	3.10E+02	-5492.474	25	49089
g07	24.306	29.649	32.111	34.954	1.18E+01	90.590	25	49767
g08	-0.095	-0.095	-0.095	-0.095	2.90E-17	-0.095	25	22883
g09	680.630	682.124	683.463	683.476	0.87E+00	685.460	25	49550
g10	7049.248	7699.793	8363.027	9047.409	1.31E+03	13731.542	25	49652
g11	0.749	0.754	0.868	0.865	6.46E-02	0.971	25	48972
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.603	0.972	0.909	0.13E+00	0.999	25	49340
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	962.642	965.686	965.330	1.76E+00	969.284	25	48819
g16	-1.905	-1.750	-1.568	-1.550	0.13E+00	-1.181	25	49704
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.670	-0.498	-0.513	5.18E-02	-0.448	25	49770
g19	32.655	245.606	296.645	309.614	3.53E+01	375.485	25	49726
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.503	-5.485	-5.478	2.13E-02	-5.406	25	49178

Tabela 8.25: Experimento 25: Recombinação BLX(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-13.157	-10.107	-10.115	1.68E+00	-6.907	25	49814
g02	-0.803	-0.591	-0.414	-0.418	4.14E-02	-0.375	25	49867
g03	-1.000	-0.961	-0.850	-0.831	6.90E-02	-0.690	25	49295
g04	-30665.538	-30665.396	-30527.082	-30496.900	1.45E+02	-30196.211	25	49611
g05	5126.496	5126.510	5136.648	5144.791	2.14E+01	5213.381	25	49486
g06	-6961.813	-6961.810	-6954.620	-6937.632	3.98E+01	-6791.237	25	49646
g07	24.306	24.998	30.546	30.708	3.49E+00	39.886	25	49778
g08	-0.095	-0.095	-0.095	-0.095	2.90E-17	-0.095	25	26421
g09	680.630	681.001	682.257	682.837	2.10E+00	689.888	25	49615
g10	7049.248	7357.044	8512.243	8935.919	1.49E+03	13205.580	25	49619
g11	0.749	0.749	0.767	0.818	8.39E-02	0.993	25	49119
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.111	0.925	0.898	0.29E+00	1.522	25	49653
g14	-47.764	-42.352	-37.853	-37.796	2.52E+00	-32.457	17	49501
g15	961.715	962.476	966.700	966.275	2.97E+00	971.268	25	49631
g16	-1.905	-1.899	-1.850	-1.819	8.57E-02	-1.562	25	49641
g17	8853.539	8914.891	8946.254	8987.148	9.41E+01	9185.176	11	49120
g18	-0.866	-0.863	-0.612	-0.605	0.11E+00	-0.475	25	49807
g19	32.655	73.228	211.644	200.010	6.94E+01	337.292	25	49801
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.508	-5.508	-5.507	4.04E-04	-5.506	25	49562

Tabela 8.26: Experimento 26: Recombinação SBX(0,8) e Mutação não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.989	-11.893	-11.621	$2.14E+00$	-6.000	25	46814
g02	-0.803	-0.550	-0.418	-0.435	$5.95E-02$	-0.341	25	49872
g03	-1.000	-0.856	-0.620	-0.643	$0.14E+00$	-0.369	25	49785
g04	-30665.538	-30665.538	-30665.538	-30662.426	$1.14E+01$	-30611.541	25	49707
g05	5126.496	5126.499	5126.621	5129.470	$9.24E+00$	5172.444	25	49754
g06	-6961.813	-6961.810	-6961.810	-6961.810	$7.02E-11$	-6961.810	25	46003
g07	24.306	25.220	32.840	35.950	$1.06E+01$	64.369	25	49851
g08	-0.095	-0.095	-0.095	-0.095	$3.10E-17$	-0.095	25	28792
g09	680.630	680.672	681.672	682.425	$2.51E+00$	692.655	25	49830
g10	7049.248	7189.765	8078.836	8309.773	$7.86E+02$	10702.203	25	49843
g11	0.749	0.749	0.774	0.779	$3.50E-02$	0.901	25	48709
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.282	0.959	1.002	$0.54E+00$	2.844	25	49798
g14	-47.764	-45.781	-39.903	-39.724	$4.43E+00$	-32.267	24	49748
g15	961.715	962.476	964.176	965.549	$3.14E+00$	972.356	25	49714
g16	-1.905	-1.905	-1.901	-1.899	$8.01E-03$	-1.868	25	49762
g17	8853.539	8870.871	8950.297	8970.424	$9.46E+01$	9219.384	24	49373
g18	-0.866	-0.860	-0.569	-0.598	$0.11E+00$	-0.485	25	49816
g19	32.655	43.467	84.214	96.887	$4.20E+01$	204.837	25	49825
g20	0.097	0.899	11.804	11.173	$7.05E+00$	24.370	9	70996
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-79.783	-0.003	76.230	$2.53E+02$	699.937	8	49684
g24	-5.508	-5.508	-5.508	-5.508	$3.06E-15$	-5.508	25	44999

Tabela 8.27: Experimento 27: Recombinação Geométrico(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pioor	frun	nma
g01	-15.000	-5.561	-4.452	-4.114	1.02E+00	-1.909	25	48657
g02	-0.803	-0.483	-0.450	-0.454	1.47E-02	-0.434	25	48649
g03	-1.000	-0.969	-0.898	-0.868	9.39E-02	-0.616	25	48557
g04	-30665.538	-30466.588	-30250.482	-30241.300	9.75E+01	-29991.721	25	48285
g05	5126.496	-	-	-	-	-	0	0
g06	-6961.813	-6814.433	-5924.346	-5958.835	4.19E+02	-5237.715	25	47776
g07	24.306	-	-	-	-	-	0	0
g08	-0.095	-0.095	-0.095	-0.095	6.35E-16	-0.095	25	31043
g09	680.630	-	-	-	-	-	0	0
g10	7049.248	8123.357	9738.078	10421.692	2.19E+03	17027.486	25	48522
g11	0.749	-	-	-	-	-	0	0
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	962.571	967.103	966.498	1.92E+00	969.255	14	48903
g16	-1.905	-1.730	-1.418	-1.448	0.16E+00	-1.176	25	48593
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-	-	-	-	-	0	0
g19	32.655	228.468	322.896	330.586	4.86E+01	425.112	25	48050
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-	-	-	-	-	0	0
g24	-5.508	-5.508	-5.499	-5.492	1.61E-02	-5.451	25	48757

Tabela 8.28: Experimento 28: Recombinação Wriqth(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.999	-12.000	-11.759	2.53E+00	-6.000	25	34907
g02	-0.803	-0.669	-0.551	-0.537	7.36E-02	-0.407	25	49912
g03	-1.000	-0.963	-0.606	-0.606	0.17E+00	-0.221	25	49838
g04	-30665.538	-30665.538	-30665.538	-30665.538	1.11E-11	-30665.538	25	26727
g05	5126.496	5126.498	5126.498	5126.498	4.29E-09	5126.498	25	49661
g06	-6961.813	-6961.810	-6961.810	-6961.810	1.87E-12	-6961.810	18	27435
g07	24.306	24.467	25.315	26.583	3.67E+00	38.508	25	49802
g08	-0.095	-0.095	-0.095	-0.095	2.68E-04	-0.094	25	27970
g09	680.630	680.630	680.655	680.694	0.13E+00	681.306	25	49851
g10	7049.248	7061.227	7277.322	7431.431	4.22E+02	8332.811	25	49781
g11	0.749	0.749	0.757	0.837	0.10E+00	0.999	25	42756
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.461	0.860	0.925	0.39E+00	2.542	25	49771
g14	-47.764	-45.852	-45.517	-44.191	2.01E+00	-40.172	25	49834
g15	961.715	962.571	967.103	966.498	1.92E+00	969.255	14	48903
g16	-1.905	-1.905	-1.905	-1.905	6.48E-07	-1.905	25	49764
g17	8853.539	8889.287	8961.550	8979.197	7.99E+01	9252.056	25	49664
g18	-0.866	-0.866	-0.864	-0.813	9.18E-02	-0.605	25	49761
g19	32.655	33.310	39.377	43.274	8.86E+00	70.500	25	49207
g20	0.097	1.799	5.433	4.798	2.10E+00	7.114	6	88876
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-62.787	-0.002	241.426	4.06E+02	899.998	19	48697
g24	-5.508	-5.508	-5.508	-5.508	0.00E+00	-5.508	25	26539

Tabela 8.29: Experimento 29: Recombinação LSX(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	melhor	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-5.000	-2.000	-2.500	1.91E+00	-1.000	4	4846
g02	-0.803	-0.569	-0.421	-0.426	4.10E-02	-0.376	25	46397
g03	-1.000	-	-	-	-	-	0	0
g04	-30665.538	-30217.704	-29947.408	-29910.108	1.99E+02	-29417.381	25	1322
g05	5126.496	6036.910	7430.784	7181.414	1.04E+03	8076.549	3	728
g06	-6961.813	-	-	-	-	-	0	0
g07	24.306	-	-	-	-	-	0	0
g08	-0.095	-0.095	-0.087	-0.084	1.23E-02	-0.052	25	15285
g09	680.630	698.288	869.230	896.251	1.97E+02	1642.794	25	2403
g10	7049.248	-	-	-	-	-	0	0
g11	0.749	0.751	0.806	0.806	7.86E-02	0.862	2	1057
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	-	-	-	-	-	0	0
g14	-47.764	-	-	-	-	-	0	0
g15	961.715	962.476	962.940	964.354	2.89E+00	972.168	25	48914
g16	-1.905	-1.459	-1.140	-1.160	0.28E+00	-0.882	3	1309
g17	8853.539	-	-	-	-	-	0	0
g18	-0.866	-0.866	-0.864	-0.813	9.18E-02	-0.605	25	49761
g19	32.655	677.601	1005.917	1056.483	3.21E+02	1779.895	25	2398
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-62.787	-0.002	241.426	4.06E+02	899.998	19	48697
g24	-5.508	-5.473	-5.218	-5.172	0.15E+00	-4.803	25	779

Tabela 8.30: Experimento 30: Recombinação pais múltiplos(0,8) e Mutação de não-uniforme(0,2)

fcn	ótimo	best	mediana	média	dv. padrão	pior	frun	nma
g01	-15.000	-14.934	-14.865	-14.849	0.054	-14.719	25	49871
g02	-0.803	-0.791	-0.783	-0.781	$9.69E-03$	-0.757	25	49536
g03	-1.000	-0.926	-0.582	-0.602	$0.17E+00$	-0.293	25	49432
g04	-30665.538	-30648.087	-30498.670	-30496.573	112.997	-30259.843	25	49450
g05	5126.496	5128.334	5185.873	5201.738	$6.86E+01$	5425.324	25	49678
g06	-6961.813	-6917.482	-6629.047	-6545.627	$2.81E+02$	-5692.489	24	49757
g07	24.306	24.925	28.081	29.283	$4.02E+00$	42.353	25	49691
g08	-0.095	-0.095	-0.095	-0.095	$1.64E-15$	-0.095	25	49600
g09	680.630	681.085	683.237	684.432	$3.11E+00$	692.420	25	49721
g10	7049.248	7254.801	7886.796	8028.652	$6.61E+02$	9863.512	25	49697
g11	0.749	0.750	0.896	0.884	$9.26E-02$	0.999	25	49447
g12	-1.000	-	-	-	-	-	0	0
g13	0.053	0.978	0.999	0.997	$5.01E-03$	0.999	25	49764
g14	-47.764	-43.252	-40.321	-40.186	$2.04E+00$	-36.315	25	49699
g15	961.715	-	-	-	-	-	0	0
g16	-1.905	-1.881	-1.787	-1.765	$8.31E-02$	-1.562	25	49820
g17	8853.539	8885.064	8945.279	8984.566	107.287	9168.778	9	49358
g18	-0.866	-	-	-	-	-	0	0
g19	32.655	69.277	97.842	98.162	18.036	149.461	25	49609
g20	0.097	-	-	-	-	-	0	0
g21	193.724	-	-	-	-	-	0	0
g22	236.430	-	-	-	-	-	0	0
g23	-400.055	-62.787	$-2.97E-003$	241.426	406.129	899.998	19	48697
g24	-5.508	-5.506	-5.491	-5.468	$5.40E-002$	-5.283	25	49789