

Grasiele Regina Duarte

**Um algoritmo inspirado em colônias de abelhas para otimização numérica
com restrições.**

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. Afonso Celso de Castro Lemonge

Coorientador: Prof. D.Sc. Leonardo Goliatt da Fonseca

Juiz de Fora

2015

Ficha catalográfica elaborada através do Programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Duarte, Grasielle Regina.

Um algoritmo inspirado em colônias de abelhas para otimização numérica com restrições / Grasielle Regina Duarte. -- 2015.

159 f. : il.

Orientador: Afonso Celso de Castro Lemonge

Coorientador: Leonardo Goliatt da Fonseca

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Modelagem Computacional, 2015.

1. Algoritmo Colônia de Abelhas Artificiais. 2. Otimização com restrições. 3. Métodos de penalização. I. Lemonge, Afonso Celso de Castro, orient. II. Fonseca, Leonardo Goliatt da, coorient. III. Título.

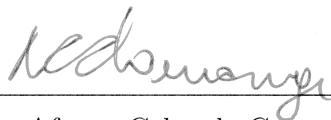
Grasiele Regina Duarte

**Um algoritmo inspirado em colônias de abelhas para otimização numérica
com restrições.**

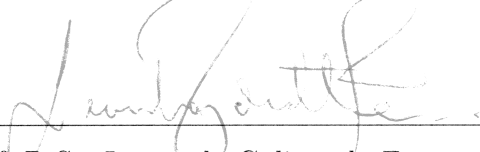
Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Aprovada em 6 de Março de 2015.

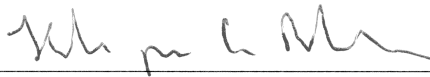
BANCA EXAMINADORA



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



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



Prof. D.Sc. Hélio José Corrêa Barbosa
Universidade Federal de Juiz de Fora



Prof. D.Sc. Eduardo Krempser da Silva
Faculdade de Educação Tecnológica do Estado do Rio de Janeiro

Dedico este trabalho à minha família.

AGRADECIMENTOS

À minha família, pelo amor, carinho, torcida e apoio incondicional.

À UFJF pela estrutura. Aos professores e técnicos do Programa de Pós-graduação em Modelagem Computacional por tornarem isso possível. Aos professores Afonso Lemonge e Leonardo Goliatt em especial, pela excelente orientação durante a realização deste trabalho.

Aos professores Hélio Barbosa e Eduardo Krempser por participarem da avaliação do trabalho e contribuírem com possíveis sugestões e correções.

Aos novos amigos, futuros colegas de profissão, pela ótima convivência, por compartilharem os momentos bons e difíceis desta caminhada.

Aos antigos amigos que de uma forma ou de outra se fizeram presentes nesta etapa da minha formação.

Suba bem alto, inspire o ar rarefeito,
veja o que nunca foi visto;
parta, perca-se,
mas faça a escalada.

Edna St. Vincent Millay

RESUMO

Os problemas de otimização estão presentes em diversas áreas de atuação da sociedade e o uso de algoritmos bio-inspirados para a resolução de problemas complexos deste tipo vem crescendo constantemente. O Algoritmo Colônia de Abelhas Artificiais (ABC – do inglês *Artificial Bee Colony*) é um algoritmo bio-inspirado proposto em 2005 para a resolução de problemas de otimização multimodais e multidimensionais. O fenômeno natural que inspirou o desenvolvimento do ABC foi o comportamento inteligente observado em colônias de abelhas, mais especificamente no forrageamento. O ABC foi proposto inicialmente para ser aplicado na resolução de problemas sem restrições. Este trabalho avalia o desempenho do ABC quando aplicado na resolução de problemas de otimização com restrições. Para o tratamento das restrições, métodos de penalização serão incorporados ao ABC. São analisados diversos métodos de penalização, de diferentes tipos, com o objetivo de identificar com qual deles o algoritmo apresenta melhor desempenho. Além disto, são avaliadas possíveis limitações e cuidados que devem ser tomados ao combinar métodos de penalização ao ABC. O algoritmo proposto é avaliado através da resolução de problemas de otimização encontrados na literatura. Vários experimentos computacionais são realizados e gráficos e tabelas são gerados para demonstração dos resultados obtidos que também são discutidos.

Palavras-chave: Algoritmo Colônia de Abelhas Artificiais. Otimização com restrições. Métodos de penalização.

ABSTRACT

Optimization problems are present in several areas of society and the use of bio-inspired algorithms to solve complex problems of this type has been growing constantly. The Artificial Bee Colony Algorithm (ABC) is a bio-inspired algorithm proposed in 2005 for solving multimodal and multidimensional optimization problems. The natural phenomenon that inspired the development of the ABC was intelligent behavior observed in bee colonies, more specifically in foraging. The ABC was initially proposed to be applied to solve unconstrained problems. This study evaluates the performance of ABC when applied in solving constrained optimization problems. For the treatment of constraints, penalty methods will be incorporated into the ABC. Several penalty methods, of different types, are analyzed with the goal of identifying which of these penalty methods offers better performance. Furthermore, possible limitations and care that should be taken when combining penalty methods to ABC are evaluated. The proposed algorithm is evaluated by solving optimization problems found in the literature. Several computational experiments are performed and graphs and tables are generated for demonstration of the obtained results which are also discussed.

Keywords: Artificial Bee Colony Algorithm. Constrained optimization. Penalties methods.

SUMÁRIO

1	Introdução.....	17
2	Otimização.....	20
2.1	Introdução	20
2.2	Otimização sem restrições	23
2.3	Otimização com restrições	23
2.4	Otimização estrutural	24
2.5	Algoritmos de otimização	24
3	Algoritmo inspirado em colônias de abelhas	27
3.1	Introdução	27
3.2	Inteligência coletiva	27
3.3	Enxame de abelhas – forrageamento	28
3.4	Algoritmo Colônia de Abelhas Artificiais	30
3.5	Variações do ABC	33
3.5.1	<i>Período entre 2005 e 2007.....</i>	34
3.5.2	<i>Período entre 2008 e 2009.....</i>	34
3.5.3	<i>Período entre 2010 e 2011.....</i>	35
3.5.4	<i>Período entre 2012 e 2013.....</i>	37
3.6	Aplicações do ABC	41
4	Tratamento de restrições.....	46
4.1	Introdução	46
4.2	Métodos de penalização	47
4.2.1	<i>Penalização estática.....</i>	47
4.2.2	<i>Penalização dinâmica</i>	48
4.2.3	<i>Penalização adaptativa.....</i>	49
4.2.4	<i>Método de Penalização Adaptativo – APM.....</i>	52
4.3	Alterações propostas	58
4.3.1	<i>Coit Mod.....</i>	59

4.3.2	<i>ADP Log</i>	60
4.3.3	<i>APM Sum</i>	61
5	Experimentos computacionais	62
5.1	Introdução	62
5.2	Implementação do algoritmo	62
5.3	Análise de resultados	66
5.3.1	<i>Perfis de desempenho</i>	66
5.3.2	<i>Escalonamento Multidimensional</i>	68
5.4	Problemas analisados	69
5.5	Experimento 01 – Problemas clássicos da Engenharia Mecânica	70
5.5.1	<i>MTC – Mola sob Tração/Compressão</i>	70
5.5.2	<i>RV – Redutor de Velocidade</i>	74
5.5.3	<i>T3B – Treliça de Três Barras</i>	79
5.5.4	<i>VP – Vaso de Pressão</i>	82
5.5.5	<i>VEL – Viga Engastada e Livre</i>	86
5.5.6	<i>VS – Viga Soldada</i>	92
5.5.7	<i>Discussão sobre os resultados dos problemas clássicos da Engenharia Mecânica</i>	95
5.6	Experimento 02 – Problemas de otimização estrutural	100
5.6.1	<i>T10C – Treliça de 10 Barras – Caso Contínuo</i>	100
5.6.2	<i>T10D – Treliça de 10 Barras – Caso Discreto</i>	106
5.6.3	<i>T25C – Treliça de 25 Barras – Caso Contínuo</i>	113
5.6.4	<i>T72C – Treliça de 72 Barras – Caso Contínuo</i>	119
5.6.5	<i>Discussão sobre os Resultados dos Problemas de Otimização Estrutural</i>	126
5.7	Experimento 03 – <i>G-Suite</i>	131
5.7.1	<i>G-Suite</i>	131
5.7.2	<i>Discussão sobre os resultados do G-Suite</i>	133
6	Conclusões	144
	REFERÊNCIAS	146

LISTA DE ILUSTRAÇÕES

2.1	Regiões de um espaço de busca com restrições.	21
2.2	Classificação dos problemas de otimização.	22
4.1	Definição de $\bar{f}(\mathbf{x})$ no APM.	54
5.1	Exemplo de gráfico da técnica Perfis de Desempenho.	67
5.2	MTC – Mola sob Tração/Compressão.	70
5.3	RV – Redutor de Velocidade.	75
5.4	T3B – Treliça de Três Barras.	79
5.5	VP – Vaso de Pressão.	82
5.6	VEL – Viga Engastada e Livre.	86
5.7	VS – Viga Soldada.	92
5.8	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	98
5.9	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	98
5.10	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	99
5.11	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	99
5.12	T10C – Treliça de 10 Barras – Caso Contínuo.	100
5.13	Dendrograma do resultado da técnica MDS para o problema Treliça de 10 Barras – Caso Discreto (T10D) adotando 5000 avaliações da função objetivo.	108
5.14	T25C – Treliça de 25 Barras – Caso Contínuo.	113
5.15	T72C – Treliça de 72 Barras – Caso Contínuo.	119

5.16	Resultado da técnica MDS para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 5000 avaliações da função objetivo.	124
5.17	Resultado da técnica MDS para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 500000 avaliações da função objetivo.	124
5.18	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	129
5.19	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	129
5.20	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	130
5.21	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	130
5.22	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para o <i>G-Suite</i> adotando a média da função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.	140
5.23	Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para o <i>G-Suite</i> adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.	140

LISTA DE TABELAS

4.1	Novas variantes para o APM.	58
5.1	Métodos de penalização analisados com o ABC G-best.	64
5.2	Resultados obtidos para o problema Mola sob Tração/Compressão (MTC) adotando 50000 avaliações da função objetivo.	72
5.3	Resultados obtidos para o problema Mola sob Tração/Compressão (MTC) adotando 500000 avaliações da função objetivo.	73
5.4	Comparação do resultado obtido para o problema Mola sob Tração/Compressão (MTC) com outros trabalhos da literatura.	74
5.5	Resultados obtidos para o problema Redutor de Velocidade (RV) adotando 50000 avaliações da função objetivo.	76
5.6	Comparação do resultado obtido para o problema Redutor de Velocidade (RV) com outros trabalhos da literatura.	78
5.7	Resultados obtidos para o problema Treliça de Três Barras (T3B) adotando 5000 avaliações da função objetivo.	80
5.8	Resultados obtidos para o problema Treliça de Três Barras (T3B) adotando 500000 avaliações da função objetivo.	81
5.9	Comparação do resultado obtido para o problema Treliça de Três Barras (T3B) com outros trabalhos da literatura.	82
5.10	Resultados obtidos para o problema Vaso de Pressão (VP) adotando 5000 avaliações da função objetivo.	84
5.11	Resultados obtidos para o problema Vaso de Pressão (VP) adotando 500000 avaliações da função objetivo.	85
5.12	Comparação do resultado obtido para o problema Vaso de Pressão (VP) com outros trabalhos da literatura.	86
5.13	Resultados obtidos para o problema Viga Engastada e Livre (VEL) adotando 5000 avaliações da função objetivo.	88
5.14	Resultados obtidos para o problema Viga Engastada e Livre (VEL) adotando 500000 avaliações da função objetivo.	89

5.15	Comparação do resultado obtido para o problema Viga Engastada e Livre (VEL) com outros trabalhos da literatura.	91
5.16	Resultados obtidos para o problema Viga Soldada (VS) adotando 5000 avaliações da função objetivo.	93
5.17	Resultados obtidos para o problema Viga Soldada (VS) adotando 500000 avaliações da função objetivo.	94
5.18	Comparação do resultado obtido para o problema Viga Soldada (VS) com outros trabalhos da literatura.	95
5.19	Melhores resultados obtidos para os problemas da Engenharia Mecânica adotando 5000 avaliações da função objetivo.	96
5.20	Melhores resultados obtidos para os problemas da Engenharia Mecânica adotando 500000 avaliações da função objetivo.	97
5.21	Resultados obtidos para o problema Treliça de 10 Barras – Caso Contínuo (T10C) adotando 5000 avaliações da função objetivo.	102
5.22	Resultados obtidos para o problema Treliça de 10 Barras – Caso Contínuo (T10C) adotando 500000 avaliações da função objetivo.	103
5.23	Comparação do resultado obtido para o problema Treliça de 10 Barras – Caso Contínuo (T10C) com outros trabalhos da literatura.	105
5.24	Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) adotando 5000 avaliações da função objetivo.	107
5.25	Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) adotando 50000 avaliações da função objetivo.	109
5.26	Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) adotando 500000 avaliações da função objetivo.	110
5.27	Comparação do resultado obtido para o problema Treliça de 10 Barras – Caso Discreto (T10D) com outros trabalhos da literatura.	112
5.28	Carregamentos da Treliça de 25 Barras – Caso Contínuo (T25C).	113
5.29	Agrupamento das barras da Treliça de 25 Barras – Caso Contínuo (T25C).	114
5.30	Resultados obtidos para o problema Treliça de 25 Barras – Caso Contínuo (T25C) adotando 5000 avaliações da função objetivo.	115
5.31	Resultados obtidos para o problema Treliça de 25 Barras – Caso Contínuo (T25C) adotando 500000 avaliações da função objetivo.	116

5.32	Comparação do resultado obtido para o problema Treliça de 25 Barras – Caso Contínuo (T25C) com outros trabalhos da literatura.	118
5.33	Carregamentos da Treliça de 72 Barras – Caso Contínuo (T72C).	119
5.34	Agrupamento das barras da Treliça de 72 Barras – Caso Contínuo (T72C). . .	120
5.35	Resultados obtidos para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 5000 avaliações da função objetivo.	121
5.36	Resultados obtidos para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 500000 avaliações da função objetivo.	122
5.37	Comparação do resultado obtido para o problema Treliça de 72 Barras – Caso Contínuo (T72C) com outros trabalhos da literatura.	125
5.38	Melhores resultados obtidos para os problemas de Otimização Estrutural adotando 5000 avaliações da função objetivo.	127
5.39	Melhores resultados obtidos para os problemas de Otimização Estrutural adotando 500000 avaliações da função objetivo.	128
5.40	Detalhes dos 24 problemas do conjunto <i>G-Suite</i>	132
5.41	Melhores resultados obtidos para os problemas do <i>G-Suite</i> (g01 a g12) adotando 5000 avaliações da função objetivo.	134
5.42	Melhores resultados obtidos para os problemas do <i>G-Suite</i> (g13 a g24) adotando 5000 avaliações da função objetivo.	135
5.43	Melhores resultados obtidos para os problemas do <i>G-Suite</i> (g01 a g12) adotando 500000 avaliações da função objetivo.	136
5.44	Melhores resultados obtidos para os problemas do <i>G-Suite</i> (g13 a g24) adotando 500000 avaliações da função objetivo.	137
5.45	Melhores soluções encontradas para os problemas do conjunto <i>G-Suite</i>	138
5.46	Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	141
5.47	Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.	141

5.48 Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando a média da função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.	142
5.49 Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.	142
5.50 Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	143
5.51 Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto <i>G-Suite</i> , adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.	143

1 Introdução

Os problemas de otimização estão presentes nas mais distintas áreas de atuação da sociedade como economia, engenharias, indústrias e diversas ciências. Muitas vezes buscando atingir um determinado objetivo torna-se relevante, por exemplo, o uso reduzido de materiais, baixo custo, aumento do lucro, da produção, entre outros interesses. Além disto, quase sempre haverão condições impostas como limitação dos recursos que estarão disponíveis e requisitos que deverão ser atendidos.

Nesse contexto torna-se importante um ramo da matemática chamado otimização. A otimização pode ser definida como a busca por uma solução que produza os valores máximo ou mínimo de uma função. Neste caso a função representa o objeto de interesse que deverá ser reduzido ou aumentado ao máximo e por este motivo é chamada de função objetivo. As variáveis envolvidas na definição da função objetivo deverão representar os recursos utilizados no projeto, observando que os mesmos serão limitados. Frequentemente, a solução do problema de otimização estará sujeita a restrições, isto significa que alguns requisitos deverão ser atendidos para que esta seja considerada viável.

Os problemas de otimização têm se tornado cada vez mais complexos e de difícil solução. Com isto, torna-se indispensável o uso de recursos computacionais para a sua resolução. Para tanto, é necessária a aplicação de um algoritmo computacional de otimização entre os muitos já propostos na literatura. Alguns algoritmos produzem sempre um mesmo resultado dada uma determinada entrada e por esse motivo são conhecidos como algoritmos determinísticos. Alguns desses algoritmos são difíceis de serem aplicados pois exigem diferenciação das funções, pelo menos de primeira ordem, o que muitas vezes pode ser difícil de ser obtido ou até impossível.

Uma outra classe de algoritmos de otimização, conhecidos como não-determinísticos, são aqueles que a partir de uma determinada entrada podem produzir resultados distintos a cada execução. Esses algoritmos tomam decisões probabilísticas durante a busca pela solução do problema e por este motivo também são conhecidos como estocásticos. Alguns desses algoritmos, são populacionais, ou seja, a partir de um conjunto de soluções iniciais são realizadas operações entre elas até que seja encontrada uma solução ótima para o problema. Entre os algoritmos populacionais, muitos deles foram propostos

tendo como inspiração fenômenos da natureza e por isso são considerados bio-inspirados. Diversos algoritmos bio-inspirados já foram propostos e alguns deles tiveram inspiração em processos evolutivos como é o caso do Algoritmo Genético que foi inspirado na teoria de evolução das espécies proposta por Charles Darwin. Outros algoritmos bio-inspirados tiveram inspiração no comportamento inteligente de algumas espécies sociais como é o caso do Algoritmo Colônia de Abelhas Artificiais descrito no próximo parágrafo.

O Algoritmo Colônia de Abelhas Artificiais (ABC – do inglês *Artificial Bee Colony*) foi inspirado no comportamento social observado em colônias de abelhas reais, em particular durante o forrageamento. Ao descobrirem novas fontes de alimento ou mesmo já explorando uma, ao retornarem para o ninho cada abelha divide com as outras, informações sobre a fonte conhecida. Ao receberem informações sobre as fontes, as abelhas presentes no ninho escolherão aquelas que explorarão. Existe a tendência de que as fontes de melhor qualidade sejam exploradas por um número maior de abelhas. Ao visitarem as fontes conhecidas, caso na vizinhança seja encontrada alguma de qualidade superior, as abelhas poderão memorizar a localização dessa nova fonte e dividir a informação ao retornarem para o ninho. No forrageamento de colônias de abelhas, três tipos de abelhas estão presentes: as abelhas campeiras que exploram as fontes conhecidas e dividem informações ao retornarem para o ninho, as abelhas observadoras que aguardam no ninho e escolhem uma fonte para explorarem de acordo com as informações recebidas e as abelhas exploradoras que buscam novas fontes para substituírem aquelas esgotadas.

Na busca pela solução do problema, o algoritmo ABC simula a atividade de forrageamento de uma colônia de abelhas. Para tanto, de forma artificial estão presentes as fontes de alimento e os três tipos de abelhas envolvidos. As soluções candidatas representam as posições das fontes de alimento. As abelhas observadoras escolhem as soluções conhecidas pelas abelhas campeiras de acordo com a sua qualidade para resolver o problema. As abelhas campeiras e observadoras realizam ainda uma busca local em torno da fonte conhecida e sendo encontrada uma solução melhor, a fonte memorizada será substituída. As fontes conhecidas por abelhas campeiras que não melhoram durante o processo de busca são substituídas por outras geradas aleatoriamente, simulando assim a atividade das abelhas exploradoras.

O ABC é um algoritmo de fácil implementação e para sua execução requer a definição de poucos parâmetros. O algoritmo tem apresentado bons resultados na literatura e por

este motivo será aplicado neste trabalho. Em 2005 o algoritmo foi proposto para resolver problemas de otimização sem restrições. Sabendo que grande parte dos problemas de otimização estão sujeitos a restrições, propõe-se neste trabalho avaliar o desempenho do ABC na resolução deste tipo de problema. O tratamento a ser dado às restrições será por meio do método de penalização.

O método de penalização tem como objetivo transformar um problema de otimização com restrições em um problema sem restrições. Em seguida, pode-se aplicar uma técnica para resolver problemas sem restrições para que o problema original seja resolvido de fato. A aplicação do método de penalização pode ocorrer de maneiras distintas, principalmente quando se trabalha com algoritmos populacionais como é o caso do ABC. Neste trabalho diversos métodos de penalização serão integrados ao algoritmo ABC com o objetivo de identificar aquele que melhor contribuirá com a convergência do algoritmo.

O principal objetivo deste trabalho é avaliar o desempenho do algoritmo ABC na resolução de problemas de otimização com restrições quando combinado a métodos de penalização e identificar o tipo e o método com o qual o algoritmo terá melhor convergência. Para atingir tal objetivo, diversos problemas de otimização encontrados na literatura serão submetidos ao algoritmo proposto. Entre os problemas escolhidos estão alguns da Engenharia Mecânica, outros de Otimização Estrutural e um conjunto composto por 24 problemas de otimização. De posse dos resultados obtidos, estes serão apresentados por meio de tabelas e gráficos e discutidos.

Esta dissertação está organizada em seis capítulos, sendo esta introdução o primeiro deles. O capítulo 2 aborda conceitos teóricos sobre otimização, seguido do capítulo 3 que descreve com maiores detalhes o algoritmo ABC, além de apresentar um breve histórico sobre as contribuições já feitas na literatura para o algoritmo. O capítulo 4 traz a descrição do método de penalização, como se aplica, os parâmetros envolvidos, além de descrever aqueles analisados junto do ABC neste trabalho. Já o capítulo 5 descreve a metodologia adotada para a realização de experimentos computacionais e em seguida apresenta os resultados obtidos. Por fim o capítulo 6 expõe as conclusões tiradas sobre o trabalho.

2 Otimização

2.1 Introdução

A otimização é um ramo da matemática que tem como objetivo encontrar a solução (ponto ou pontos) no domínio de uma função (ou várias funções) que produza os valores máximo ou mínimo que esta assume. Desta forma, otimizar consiste em buscar pelo valor máximo ou mínimo de uma função. O ponto que maximiza ou minimiza uma função é chamado de solução ótima. Normalmente a solução ótima estará sujeita a determinadas condições que deverão ser atendidas para que seja considerada realmente viável para resolver o problema [1, 2, 3].

Para a resolução de problemas de otimização existe uma diversidade de métodos. Esta variedade se deve à distinção entre os problemas, tornando impossível que um único método possa ser aplicado de forma generalizada [3].

De forma geral, um problema de otimização é dado por

$$\left\{ \begin{array}{l} \text{Encontre } \mathbf{x} \text{ que minimize } f(\mathbf{x}) \\ \text{Sujeito a:} \\ g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q \\ \mathbf{x}_I \leq \mathbf{x} \leq \mathbf{x}_S \end{array} \right. , \quad (2.1)$$

onde \mathbf{x} é um vetor n -dimensional, sendo n o número de variáveis de projeto do problema, $f(\mathbf{x})$ é chamada de função objetivo, $g_i(\mathbf{x})$ e $h_j(\mathbf{x})$ são respectivamente as restrições de desigualdade e igualdade impostas ao problema. Os limites inferior e superior para as variáveis de projeto são dados respectivamente pelos vetores \mathbf{x}_I e \mathbf{x}_S , também n -dimensionais.

Um problema de maximização pode ser convertido em um problema de minimização e passar a ser visto como na forma dada pela Equação (2.1). Com isto, maximizar $f(\mathbf{x})$ é o mesmo que minimizar $-f(\mathbf{x})$ [2, 3].

Comumente ocorre a conversão de restrições de igualdade em restrições de desigualdade

na forma

$$|h_j(\mathbf{x})| - \epsilon \leq 0, \quad j = 1, 2, \dots, q, \quad (2.2)$$

que é numericamente aceitável para uma tolerância ϵ de valor suficientemente pequeno, de forma a não afetar a qualidade da solução do problema. Em geral adota-se $\epsilon = 10^{-4}$.

Considerando os problemas de otimização em que além da função objetivo estão presentes restrições de desigualdade (do tipo $g_i(\mathbf{x}) \leq 0$, $i = 1, 2, \dots, m$), a Figura 2.1 exemplifica as funções de tais restrições no espaço de busca. Observa-se que existem duas regiões distintas:

- A região factível, onde encontram-se os pontos que atendem a todas as restrições ($g_i(\mathbf{x}) \leq 0$).
- A região infactível, onde encontram-se os pontos que não atendem as restrições ($g_i(\mathbf{x}) > 0$).

Basicamente o papel das restrições é definir as fronteiras do espaço de busca. Apesar de não aparecerem na Figura 2.1, as possíveis restrições de igualdade (do tipo $h_j(\mathbf{x}) = 0$, $j = 1, 2, \dots, q$) podem ser transformadas em restrições de desigualdade de acordo com a Equação (2.2). Restrições de desigualdade serão consideradas ativas quando $g_i(\mathbf{x}) = 0$ ($i \in \{1, 2, \dots, m\}$).

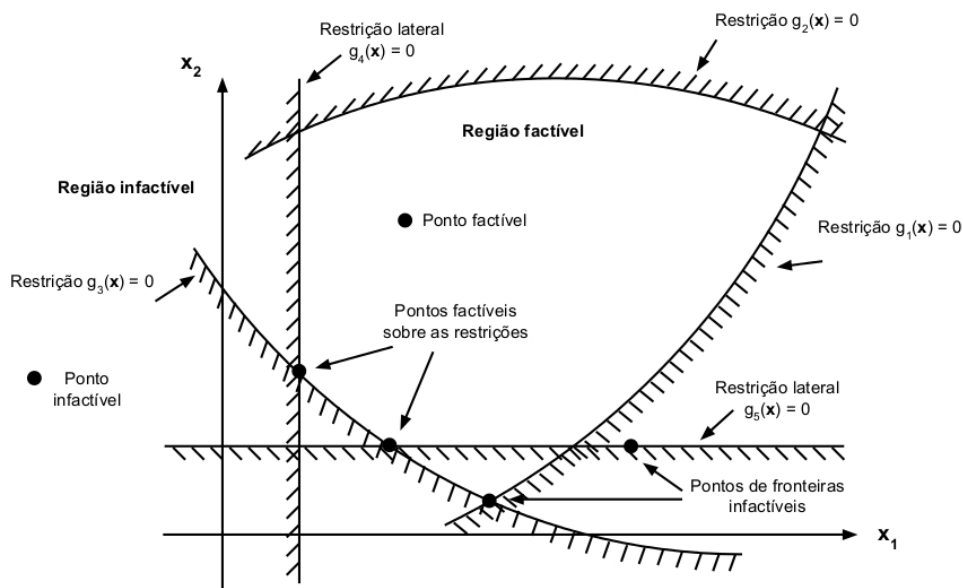


Figura 2.1: Regiões de um espaço de busca com restrições, reproduzida de [4], adaptada de [3].

Os problemas de otimização podem ser classificados de acordo com diversas características envolvidas em suas formulações. A Figura 2.2 apresenta algumas dessas possíveis classificações, bem como as características que as justificam. Por exemplo, a classificação de um problema de otimização de acordo com a ausência ou presença de restrições, poderá ser sem ou com restrições. Um problema de otimização poderá ser classificado como linear ou não-linear de acordo com as definições da função objetivo e/ou restrições.

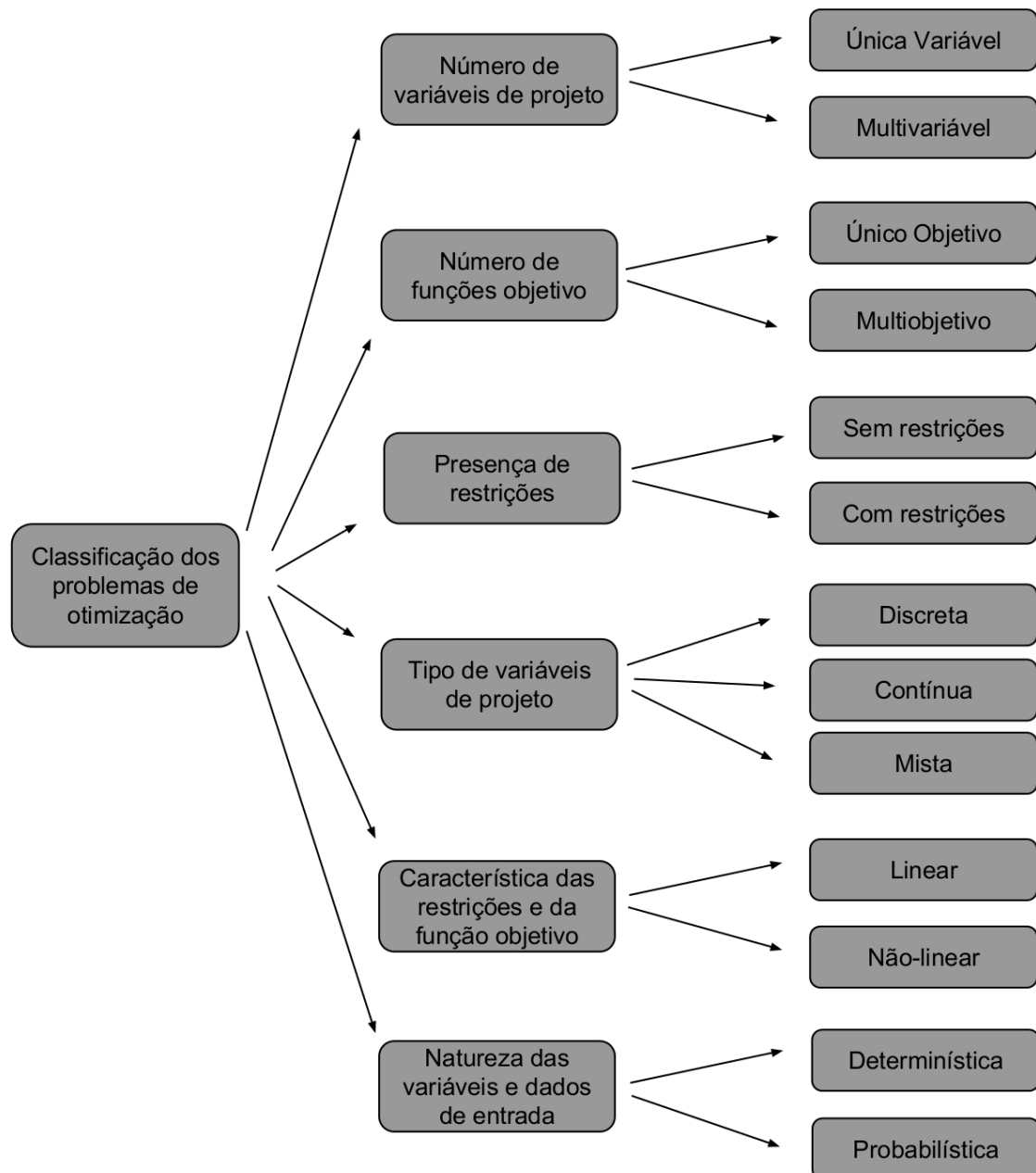


Figura 2.2: Classificação dos problemas de otimização, reproduzida de [4], baseada de [5].

2.2 Otimização sem restrições

Suponha que o problema de otimização seja dado por

$$\text{Minimize } f(\mathbf{x}), \quad (2.3)$$

onde $\mathbf{x} \in \mathbb{R}^n$. Neste caso, como a solução ótima (\mathbf{x}^*) pode ser qualquer ponto de \mathbb{R}^n , diz-se que o problema é sem restrições. Neste tipo de problema, poderão estar presentes os limites inferiores e superiores para as variáveis de projeto, neste caso a solução estará em um subconjunto $X \subset \mathbb{R}^n$, sendo $X = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}_I \leq \mathbf{x} \leq \mathbf{x}_S\}$.

As restrições representam limitações no domínio da função, modelando o problema sem a presença delas, amplia-se o domínio do problema. Muitos problemas com restrições podem ser convertidos em problemas sem restrições. Por exemplo, a restrição $x_1 + x_2 = B$ pode ser eliminada substituindo-se $x_2 = B - x_1$ em todas as ocorrências de x_2 na formulação do problema [6].

2.3 Otimização com restrições

Supondo que o problema de otimização seja dado por

$$\left\{ \begin{array}{l} \text{Minimize } f(\mathbf{x}) \\ \text{Sujeito a } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q \\ \quad \quad \quad \mathbf{x}_I \leq \mathbf{x} \leq \mathbf{x}_S \end{array} \right., \quad (2.4)$$

que é equivalente ao problema da Equação (2.1). Grande parte dos problemas de otimização são modelados com a presença de restrições.

Uma solução ótima (\mathbf{x}^*) para o problema da Equação (2.4) deverá atender a todas as restrições impostas, além de gerar o menor valor possível para $f(\mathbf{x})$. Diz-se que um ponto viola uma restrição quando deixa de atendê-la, passando a ser considerado inviável para solucionar o problema [7, 6]. Segundo Friedlander [2], a dificuldade dos problemas de otimização com restrições depende fortemente da complexidade destas.

2.4 Otimização estrutural

Entre as áreas em que a otimização está fortemente presente estão as engenharias civil, mecânica, naval, aeronáutica, entre outras ciências capazes de trazer informações sobre a formulação do problema além daquelas de conhecimento do usuário.

A otimização estrutural consiste basicamente em encontrar projetos de engenharia com o menor custo possível ou ainda de maior desempenho (onde o custo pode ser irrelevante), garantindo as condições de segurança. Entre outras aplicações, pode-se mencionar: minimização da seção transversal de vigas de pontes, minimização do peso de estruturas de edificações, minimização dos custos e/ou materiais utilizados nas indústrias naval, automobilística, aeronáutica [7, 8]. Um exemplo bastante citado que demanda otimização estrutural é o projeto de treliças metálicas como torres de transmissão. Devido ao grande volume fabricado, torna-se importante que estas estruturas tenham baixo custo de produção.

Na otimização estrutural, comumente o problema será modelado com a presença de mais de uma função objetivo, caracterizando um problema de otimização multiobjetivo. Entre as restrições dos problemas de otimização estrutural estão presentes limites para as tensões internas nos elementos, deslocamentos máximos permitidos, entre outras. Algumas restrições presentes na otimização estrutural podem ser implícitas e para determiná-las é necessário o uso de simuladores. Em problemas complexos, essa característica pode demandar alto custo computacional.

2.5 Algoritmos de otimização

Grande parte dos problemas de otimização atuais são ou podem se tornar complexos, tornando impossível que sejam resolvidos de forma rápida. Com isso, o uso de recursos computacionais torna-se indispensável para a resolução de tais problemas. Basicamente os algoritmos de otimização são classificados de acordo com a característica de serem ou não determinísticos.

Os algoritmos determinísticos são aqueles que dada uma determinada entrada, sempre produzirão o mesmo resultado ao final do processo. Muitos algoritmos desta classe podem ser difíceis de serem utilizados, pois exigem diferenciação das funções, pelo menos de primeira ordem, que podem não estar disponível. Estes algoritmos também podem ter

limitações quanto ao tipo de problema em que podem ser aplicados [9]. Alguns exemplos de algoritmos dessa classe são Método Simplex, Gradiente Conjugado, Método de Newton, Método Quasi-Newton, entre outros [10, 11, 12, 6, 3].

Os algoritmos estocásticos são aqueles que tomam decisões probabilísticas durante o processo de busca. Aplicando tais algoritmos, dada uma determinada entrada, o resultado final poderá variar de acordo com as decisões tomadas. Devido a possibilidade de variação dos resultados, estes algoritmos são conhecidos como não-determinísticos. Estes algoritmos nem sempre produzem o melhor resultado possível, mas frequentemente encontram boas soluções. O desempenho dos algoritmos estocásticos normalmente é avaliada por métodos empíricos e não por meio de provas matemáticas formais.

A partir da observação de fenômenos da natureza foram propostos algoritmos que os imitam. Tais algoritmos podem ter inspiração em processos evolutivos, onde se parte de uma condição inicial e evolui-se para uma outra; mas podem também ter inspiração no comportamento inteligente de algumas espécies sociais, como é o caso de alguns insetos. Alguns algoritmos deste tipo são brevemente descritos nos próximos parágrafos.

O Algoritmo Genético (GA – do inglês *Genetic Algorithm*) foi proposto por Holland [13] e teve como inspiração natural a teoria de evolução das espécies proposta por Charles Darwin. É um dos algoritmos bio-inspirados mais utilizados na literatura e já foi aplicado em diversas classes de problemas de otimização. Baseando-se nos conceitos da genética biológica, os processos ocorridos entre as espécies imitados pelo GA são basicamente: a reprodução, a seleção natural e as possíveis mutações que as espécies sofrem durante o processo evolutivo.

O algoritmo Recozimento Simulado (SA – do inglês *Simulated Annealing*) foi proposto por Kirkpatrick, Gelatt e Vecchi [14]. O algoritmo imita o processo metalúrgico de recozimento, que consiste em aquecer e resfriar o metal com o objetivo de reduzir as imperfeições da sua estrutura cristalina. Entre as aplicações do SA, aquelas voltadas à resolução de problemas de otimização combinatória são as mais encontradas.

O algoritmo Otimização por Colônia de Formigas (ACO – do inglês *Ant Colony Optimization*) foi proposto em 1996 por Dorigo, Maniezzo e Colorni [15]. A inspiração natural que viabilizou o desenvolvimento do ACO foi a interação entre as formigas [16, 17]. Grande parte das aplicações do ACO estão relacionadas à resolução de problemas formulados em grafos.

Os Sistemas Imunológicos Artificiais (AIS – do inglês *Artificial Immune Systems*) foram propostos por [18] e tiveram inspiração no funcionamento do sistema imunológico biológico [19].

A imitação do movimento de pássaros em revoadas ou peixes em cardumes deu origem ao algoritmo Otimização por Enxame de Partículas (PSO – do inglês *Particle Swarm Optimization*), que foi proposto em 1995 por Eberhart e Kennedy [20].

O algoritmo Evolução Diferencial (DE – do inglês *Differential Evolution*) foi proposto por Storn e Price [21]. Este algoritmo estocástico não foi inspirado em fenômenos da natureza. Durante as iterações, o DE altera as coordenadas das soluções (vetores) candidatas a partir de um vetor gerado pela adição de um vetor à diferença ponderada entre outros dois vetores.

O algoritmo Otimização por Aranhas Sociais (SSO – do inglês *Social Spider Optimization*) foi proposto em 2013 por Cuevas et al. [22]. O fenômeno natural que serviu de inspiração para o desenvolvimento do SSO foi o comportamento durante o forrageamento de uma espécie de aranhas sociais. O SSO simula o uso da teia como forma de comunicação entre as aranhas na captura das presas.

O algoritmo Otimização Baseada em Biogeografia (BBO – do inglês *Biogeography Based Optimization*) foi proposto em 2008 por Simon [23]. A inspiração natural do BBO foi a biogeografia, que é uma área da ciência que estuda a distribuição geográfica de organismos biológicos. Basicamente o BBO simula o movimento migratório de espécies entre *habitats* geograficamente isolados.

Além dos algoritmos mencionados acima, existem outros bio-inspiradas, entre eles o Algoritmo Vagalume (FA – do inglês *Firefly Algorithm*) [24] que teve como inspiração as luzes dos vagalumes piscando, o algoritmo Pesquisa por Organismos Simbióticos (SOS – do inglês *Symbiotic Organisms Search*) [25], inspirado na interação entre espécies distintas em um habitat; entre outros algoritmos que podem ser encontradas na literatura.

O algoritmo Colônia de Abelhas Artificiais (ABC – do inglês *Artificial Bee Colony*) teve como inspiração natural o comportamento social observado no forrageamento de colônias de abelhas reais. Por se tratar de objeto de estudo neste trabalho, este algoritmo encontra-se descrito com maiores detalhes no Capítulo 3.

3 Algoritmo inspirado em colônias de abelhas

3.1 Introdução

Este capítulo descreve o algoritmo escolhido ser aplicado como ferramenta de busca para neste trabalho, bem como aspectos envolvidos no seu desenvolvimento.

Diversos algoritmos para resolução de problemas complexos de busca e otimização têm sido propostos tendo fenômenos da natureza como inspiração. Alguns desses algoritmos foram inspirados no comportamento observado em algumas espécies sociais como é o caso de alguns insetos.

O Algoritmo Colônia de Abelhas Artificiais (ABC - do inglês *Artificial Bee Colony*) consiste de um algoritmo com inspiração natural. Das classes de algoritmos bio-inspirados, o ABC se encaixa entre aqueles que atuam aplicando conceitos de inteligência coletiva.

Neste capítulo serão descritas algumas características de uma colônia de abelhas que influenciaram na concepção do ABC. Em seguida o ABC será descrito, além de algumas alterações propostas com intuito de melhorar o desempenho do algoritmo. Por fim, são apresentadas algumas aplicações já realizadas do algoritmo.

3.2 Inteligência coletiva

O comportamento social de algumas espécies tem sido o interesse de cientistas de diversas áreas nos últimos anos. Uma das características destas sociedades que chama a atenção é o fato de muitas atividades serem realizadas em conjunto para que seja atingido um determinado objetivo que beneficie toda a população. Tal comportamento é conhecido como inteligência coletiva.

O termo enxame geralmente aplica-se a qualquer forma de coleção de agentes (ou indivíduos) interagindo. Muitos enxames apresentam comportamento coletivo inteligente. Uma colônia de abelhas consiste de um exemplo clássico de enxame inteligente, além de outras coleções de agentes como formigas, vespas, células, entre outros [26, 27].

Para que um enxame seja classificado como inteligente, algumas características deverão estar presentes, como a auto-organização e a divisão do trabalho [26]:

- Auto-organização consiste da capacidade de se estabelecer uma estrutura global de um sistema por meio de interações entre os agentes de níveis específicos. Um sistema pode ser visto com um conjunto de elementos e as relações entre eles. Para que haja auto-organização, são estabelecidas regras básicas para a interação entre os agentes do sistema. Geralmente as interações ocorrem com base em informações locais dos agentes, sem qualquer conhecimento da condição num nível global que possa existir.
- Divisão do trabalho passa a existir quando num enxame diversas tarefas são desenvolvidas simultaneamente por indivíduos especializados. Acredita-se que desta forma há uma maior eficiência na realização das tarefas do que quando executadas sequencialmente por muitos indivíduos não especializados.

3.3 Enxame de abelhas – forrageamento

Como mencionado na Seção 3.2, num enxame de abelhas pode ser observado o comportamento coletivo inteligente. Uma colônia de abelhas pode ser vista como um sistema dinâmico que se ajusta de acordo com suas necessidades, ainda que seus agentes (abelhas), se observados de forma individual possuam capacidade e conhecimento limitados. O funcionamento do sistema não será prejudicado se uma abelha não apresentar o desempenho esperado [28].

Uma das atividades realizadas numa colônia de abelhas de extrema importância é o forrageamento. Com o objetivo de maximizar o estoque de alimento no ninho, as abelhas sempre buscam ter boas fontes para serem exploradas. Quando se fala do forrageamento numa colônia de abelhas, basicamente as atividades realizadas estão relacionadas à busca por novas fontes de alimento, exploração das fontes conhecidas, recrutamento de abelhas para intensificação da exploração de boas fontes e abandono de fontes com recursos esgotados. Para que estas atividades sejam realizadas, três elementos básicos estão presentes no forrageamento das colônias de abelhas [26, 28, 29]:

- Fontes de alimento;
- Abelhas campeiras;

- Abelhas não-campeiras.

Uma fonte de alimento será considerada atraente ou não dependendo de fatores como: sua proximidade do ninho, concentração de alimento, facilidade de extração, entre outros. Por simplicidade, a qualidade de uma fonte pode ser representada por um valor, embora isto dependa dos fatores mencionados.

As abelhas campeiras estão relacionadas com uma determinada fonte de alimento com o objetivo de explorá-la. De posse de informações sobre a fonte como distância, direção, rentabilidade, entre outras, ao retornarem para o ninho estas abelhas então as dividem com as outras.

Dois tipos de abelhas não-campeiras estão presentes numa colônia: *i*) as abelhas exploradoras que buscam por novas fontes de alimento ao redor do ninho e *ii*) as abelhas observadoras que se encontram no ninho e ao receberem das abelhas campeiras informações sobre as fontes, escolhem aquela que irão explorar. Durante as visitas à fonte conhecida, cada abelha poderá escolher uma outra fonte na região, com base em informações visuais da vizinhança.

Dentro de um ninho, a região mais importante quando se trata da troca de informações entre as abelhas sobre as fontes de alimento, é a área de dança. A interação entre as abelhas durante o forrageamento se dá por meio de uma dança realizada nesta área. Nesta dança as abelhas campeiras realizam movimentos que possibilitam a identificação a direção, distância, qualidade do alimento das fontes conhecidas, entre outras informações.

Evidentemente, na área de dança estarão presentes diversas abelhas campeiras compartilhando informações. Existe uma grande chance das fontes mais rentáveis serem escolhidas por um número maior de abelhas observadoras, visto que estarão circulando mais informações sobre estas fontes. Pode-se concluir então que o recrutamento é diretamente proporcional à rentabilidade da fonte de alimento [30, 26, 28, 27, 31].

Ao retornarem para o ninho e estocarem o alimento, as abelhas poderão agir de três maneiras: *i*) não recrutar outras abelhas e abandonar a fonte, *ii*) dançar e recrutar outras abelhas para intensificar a exploração da fonte ou *iii*) continuar explorando a fonte sem recrutar outras abelhas [26].

Quando uma fonte tem sua capacidade esgotada, uma abelha exploradora deve definir aleatoriamente uma substituta. Estas abelhas saem do ninho na busca por uma nova fonte sem nenhuma informação. Ao encontrar uma nova fonte, utilizando sua capacidade

de memorização, a abelha inicializa a exploração imediatamente e ao retornar para a colméia, compartilha as informações sobre a nova fonte encontrada [26, 28, 27, 31].

3.4 Algoritmo Colônia de Abelhas Artificiais

O algoritmo Colônia de Abelhas Artificiais (ABC - do inglês *Artificial Bee Colony*) foi proposto em 2005 por Karaboga [26], para a resolução de problemas de otimização multimodal e multidimensional. Consiste de um algoritmo populacional bio-inspirado. O fenômeno natural que serviu de inspiração para o desenvolvimento do ABC foi o comportamento social observado em colônias de abelhas reais, em particular a realização das atividades envolvidas no forrageamento.

No ABC as posições das fontes de alimento representam as possíveis soluções para o problema. A qualidade de uma fonte artificial equivale ao valor da função objetivo produzido pela solução. Artificialmente estão presentes as abelhas campeiras, observadoras e exploradoras. Inicialmente o enxame é composto por abelhas campeiras observadoras. Se a solução apontada por uma abelha campeira não melhorar durante as iterações do ABC, tal abelha será considerada exploradora até a substituição da solução.

No ABC existe uma fonte de alimento associada a cada abelha campeira. As abelhas observadoras artificiais escolherão as fontes com base nas aptidões das soluções (fontes) apontadas pelas abelhas campeiras. A abelha exploradora artificial gerará aleatoriamente uma nova solução candidata.

Para que um algoritmo de busca seja robusto, deverão estar presentes as fases de exploração e intensificação. No ABC a fase de intensificação é realizada pelas abelhas campeiras e observadoras enquanto que a fase de exploração fica por conta das abelhas exploradoras.

A taxa de recrutamento numa colônia real representa o quão rápido o enxame localiza e explora fontes de alimento. No ABC esta taxa representa o quão rápido é possível encontrar boas soluções para o problema. Numa colônia real a sobrevivência fica condicionada à eficiência das abelhas para encontrar e explorar fontes de alimento. No ABC, a resolução do problema está condicionada à capacidade de descobrir e operar sobre boas soluções.

O ABC pode ser descrito basicamente como um algoritmo que simula uma colônia de

abelhas praticando o forrageamento na busca por uma solução ótima para um problema de otimização. Os principais passos do ABC podem ser descritos no Algoritmo 1 [26, 28, 27, 31].

Algoritmo 1: ABC

Inicialize a população;

repita

 Posicione as abelhas campeiras em suas fontes de alimento;

 Calcule a probabilidade de escolha da fonte por abelhas observadoras;

 Posicione as abelhas observadoras nas fontes de alimento dependendo de seus valores;

 Interrompa exploração de fontes esgotadas;

 Envie abelhas exploradoras para buscar novas fontes de alimento;

 Memorize a melhor fonte de alimento encontrada até o momento;

até *Número máximo de ciclos*;

Na etapa de inicialização da população do Algoritmo 1, posições de fontes de alimento (soluções candidatas) serão geradas aleatoriamente dentro do espaço de busca do problema para que as abelhas campeiras artificiais sejam posicionadas. Com isto, a inicialização de cada solução candidata pode ser definida por [32]

$$x_{ij} = x_{ij}^{min} + rand(0, 1)(x_{ij}^{max} - x_{ij}^{min}), \quad (3.1)$$

onde min e max correspondem respectivamente aos limites inferior e superior do parâmetro j ($j = 1, 2, \dots, D$) da solução x_i ($i = 1, 2, \dots, SN/2$), $rand(0, 1)$ produz um valor aleatório entre $[0, 1]$. A população inicial será composta por $SN/2$ soluções (fontes de alimento), onde SN é o total de abelhas na população, sendo que metade delas já conhecem fontes de alimento e a outra metade fará a escolha de acordo com a qualidade da fonte. Cada solução x_i é um vetor D -dimensional, sendo D o número de variáveis de projeto do problema. Iniciada a população, serão repetidos ciclos $C = 1, 2, \dots, C_{max}$ do processo de busca das abelhas campeiras, observadoras e exploradoras [28, 31].

Com as fontes de alimento artificiais já relacionadas às abelhas campeiras, cada uma delas será avaliada e uma probabilidade de escolha será associada com base na qualidade da solução para resolver o problema. Ainda no Algoritmo 1, o posicionamento das

abelhas observadoras nas fontes, ocorrerá por um processo de seleção de acordo com as probabilidades atribuídas e compartilhadas pelas abelhas campeiras. Em consequência, as fontes com altas probabilidades terão um recrutamento maior de abelhas observadoras, intensificando assim a exploração em tais posições [26, 28].

Como já mencionado, durante as visitas às fontes, as abelhas poderão escolher memorizar uma nova fonte vizinha daquela conhecida. No ABC este evento se dá por meio de comparação entre soluções após realização de busca local nas fontes conhecidas [28]. Para tanto, a exploração da vizinhança de uma determinada fonte artificial x_i ocorre de acordo com a equação

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad (3.2)$$

onde v_{ij} é o novo valor candidato a substituir x_{ij} , $k \in \{1, 2, \dots, BN\}$ e $j \in \{1, 2, \dots, D\}$ são índices escolhidos aleatoriamente, sendo que $k \neq i$, $BN = SN/2$ é o número de abelhas campeiras ou observadoras e ϕ_{ij} é um valor aleatório entre $[-1, 1]$. Se o valor produzido pela Equação (3.2) violar os limites pré-definidos para a variável j da solução x_i , este deverá ser ajustado para um valor aceitável. Avaliada a qualidade da nova solução (v_i), e sendo esta melhor que a antiga (x_i), a solução será substituída, caso contrário será mantida. Todas as abelhas visitarão a fonte de alimento conhecida. Desta forma a Equação (3.2) será executada SN vezes [28, 31].

Cada abelha observadora escolherá uma fonte artificial com uma probabilidade de seleção calculada por

$$p_i = \frac{fit_i}{\sum_{n=1}^{BN} fit_n}, \quad (3.3)$$

onde p_i é a probabilidade de seleção da fonte x_i ($i = 1, 2, \dots, BN$), fit_i é o valor da aptidão da mesma solução x_i para resolver o problema. A aptidão fit_i de cada solução será determinada por [33]

$$fit_i = \begin{cases} 1/(1 + f_i), & \text{se } f_i \geq 0 \\ 1 + |f_i|, & \text{se } f_i < 0 \end{cases}, \quad (3.4)$$

onde f_i é o valor da função objetivo produzido com a solução x_i . Deve-se atentar que ao visitar a fonte escolhida, cada abelha observadora artificial fará uma busca local aplicando a Equação (3.2) [28].

Numa colônia real, uma fonte de alimento cujo néctar tenha esgotado será abandonada e deverá ser substituída por outra encontrada por uma abelha exploradora. No ABC a simulação deste evento ocorre quando abelhas campeiras artificiais tornam-se exploradoras e buscam de forma aleatória novas fontes (soluções). A seleção das abelhas campeiras que se tornarão exploradoras é orientada por um parâmetro de controle chamado *limite*. Se uma solução não tiver sua qualidade melhorada por um determinado número de ciclos (*limite*), esta será abandonada e a abelha campeira a ela relacionada passará a ser considerada exploradora. As abelhas exploradoras artificiais substituem uma determinada solução x_i abandonada por uma nova gerada com a Equação (3.1) [26, 28, 31].

Com base nos parágrafos anteriores, destaca-se que o ABC proporciona quatro processos de seleção [28, 31]

- Seleção global na fase das abelhas observadoras, que escolhem soluções de acordo com probabilidades.
- Seleção local ao definir uma solução vizinha (k) ao aplicar a Equação (3.2).
- Seleção local gulosa ao determinar se uma solução será substituída ou não de acordo com avaliação da solução gerada com a Equação (3.2).
- Seleção aleatória na fase das abelhas exploradoras.

Por fim, deve ser destacado que os únicos parâmetros a serem definidos para a execução do ABC são:

1. Número de fontes de alimento (SN);
2. Valor de *limite*;
3. Número máximo de ciclos (C_{max}).

3.5 Variações do ABC

O ABC vem recebendo a atenção de alguns pesquisadores e algumas variações para o algoritmo foram propostas tendo como objetivos estender suas aplicações ou melhorar seu desempenho. Esta seção apresenta algumas destas propostas organizadas de forma cronológica.

3.5.1 Período entre 2005 e 2007

Em 2005 o algoritmo ABC foi proposto como já mencionado na Seção 3.4.

Em 2007, Karaboga e Basturk [32] propuseram uma alteração para o ABC com o objetivo de estender sua aplicação para resolução de problemas com restrições. Para o tratamento das restrições, Karaboga e Basturk [32] adotaram o critério de seleção proposto por Deb, descrito em [34]. A idéia consiste em aplicar o critério de Deb na etapa de busca local das abelhas campeiras e observadoras, um dos momentos em que ocorre seleção no ABC.

Ainda voltando a atenção para a busca local das abelhas artificiais, Karaboga e Basturk [32] propuseram também uma alteração para o ABC que diz respeito à etapa de produção da nova solução candidata a substituir a antiga fonte memorizada. Neste caso, a proposta consiste em substituir a Equação (3.2) por

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), & \text{se } R_j < MR, j \in \{1, 2, \dots, D\} \\ x_{ij} & , \text{ caso contrário} \end{cases}, \quad (3.5)$$

onde R_j é um número real escolhido aleatoriamente entre $[0, 1]$, MR é um parâmetro a ser definido entre $[0, 1]$ e controla quais variáveis da solução x_i serão alteradas ou não.

As abelhas exploradoras promovem diversidade na população, sendo possível inclusive a inserção de novas soluções infactíveis. Com este argumento Karaboga e Basturk [32] propuseram ainda que a fase destas abelhas ocorra apenas a cada período pré-determinado de ciclos. Assim, no ABC proposto em [32] a cada *SSP* (do inglês *Scout Production Period*) ciclos deverá ser determinado se existem ou não soluções que deverão ser substituídas aleatoriamente.

3.5.2 Período entre 2008 e 2009

Em 2008, Karaboga e Basturk [27] propuseram uma nova alteração para a Equação (3.2), como foi feito em [32]. Tal alteração diz respeito ao parâmetro ϕ_{ij} que deixa de ser um valor aleatório entre $[-1, 1]$ passando a ser -1 ou 1 exclusivamente.

Também em 2008, Quan e Shi [35] propuseram algumas alterações para o ABC, uma delas diz respeito à Equação (3.2) onde é sugerido que o valor de ϕ_{ij} seja escolhido entre $[0, 1]$ e não entre $[-1, 1]$.

Quan e Shi [35] defendem que calcular o número de abelhas observadoras relacionadas às fontes artificiais acelera a intensificação da exploração daquelas de boa qualidade. Quan e Shi [35] propuseram então que o número de abelhas observadoras associadas a cada uma das fontes de alimento pode ser calculado por

$$N_i = p_i \times NW, \quad (3.6)$$

onde N_i é o total de abelhas observadoras que escolhem a fonte x_i ($i = 1, 2, \dots, SN/2$), p_i é calculado pela Equação (3.3) e NW é o número total de abelhas observadoras ($NW = SN/2$). Com esta informação, a Equação (3.2) (observando o valor de ϕ_{ij}) será aplicada N_i vezes sobre cada solução x_i .

3.5.3 Período entre 2010 e 2011

Em 2010, Zhu e Kwong [36] constataram que apesar da eficiência do ABC relatada na literatura, existe uma limitação na Equação (3.2). Zhu e Kwong [36] observaram que a Equação (3.2) é boa em exploração por escolher aleatoriamente a solução x_k , fazendo com que a chance de serem escolhidas soluções boas ou não tão boas seja a mesma. Porém a Equação (3.2) é pobre em intensificação da exploração na busca local. Com isso, inspirados no algoritmo PSO [20], Zhu e Kwong [36] propuseram substituir a Equação (3.2) por

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}), \quad (3.7)$$

onde a terceira parcela do lado direito é chamada *gbest*, y_j é o parâmetro j da melhor solução (y), ψ_{ij} é um número aleatório entre $[0, C]$, sendo C uma constante não negativa.

Como no PSO [20, 37], a Equação (3.7) traz a vantagem da informação da melhor solução global para guiar a busca. O parâmetro C é importante no balanceamento entre exploração e intensificação da busca local. Se $C = 0$, a Equação (3.7) se torna idêntica à Equação (3.2). Quando C cresce a partir de 0 até um certo valor (Zhu e Kwong [36] recomendam $0 \leq C \leq 2$), a intensificação da exploração cresce correspondentemente. Porém o valor de C não pode ser tão grande por duas razões:

- Causaria enfraquecimento da intensificação, aumentando a exploração;
- Poderia fazer com que o termo *gbest* dirigisse a busca sobre a melhor solução.

Com a alteração proposta, Zhu e Kwong [36] deram o nome *Gbest-guided ABC* (GABC) ao novo algoritmo.

Em 2010, ao aplicarem o ABC na resolução de problemas de otimização estrutural, Hadidi, Azad e Azad [38] propuseram que na etapa de busca local, para cada parâmetro j ($j = 1, 2, \dots, D$) um número aleatório entre $[0, 1]$ seja escolhido e sendo menor que P_a , a Equação (3.2) será aplicada, caso contrário, j será mantido. P_a é um parâmetro a ser definido entre $[0, 1]$.

Outra alteração proposta por Hadidi, Azad e Azad [38] diz respeito à fase das abelhas observadoras, onde foi empregada Seleção por Torneio. Esta estratégia também foi adotada por Pan et. al [39]. Na Seleção por Torneio aplicada, M fontes são escolhidas aleatoriamente e será considerada vencedora aquela que tiver a melhor qualidade, sendo esta a que a abelha observadora explorará. Aplicando tal seleção, a probabilidade associada a cada solução foi removida.

Hadidi, Azad e Azad [38] constataram ainda que a busca aleatória por novas soluções pelas abelhas exploradoras pode ser benéfico nas iterações iniciais, mas pode não ser eficiente no final do processo de busca. Hadidi, Azad e Azad [38] propuseram então uma alteração para a fase das abelhas exploradoras aplicando uma mutação Gaussiana baseada em [40], dada por

$$v_{ij} = x_{ij} + N(0, \sigma_j), \quad j = 1, 2, \dots, D, \quad (3.8)$$

onde $N(0, \sigma_j)$ é um número aleatório de uma distribuição normal com média 0 e desvio padrão σ_j associado a cada uma das variáveis de projeto j , a ser definido a cada iteração do algoritmo. Em [38] o valor de σ_j foi definido como sendo o desvio padrão de cada parâmetro j no conjunto de soluções candidatas, como sugerido por Koohestani e Azad [40]. Desta forma o valor de σ_j tende a decrescer com o aumento do número de iterações. As abelhas exploradoras realizam busca global no início do ABC e busca local no final.

O nome dado por Hadidi, Azad e Azad [38] ao novo ABC por eles proposto foi *Modified Artificial Bee Colony* (MABC).

Em 2011 Karaboga e Akay [33] aplicaram a Equação (3.5) proposta em [32] na etapa de busca local das abelhas campeiras e observadoras artificiais. Porém, Karaboga e Akay [33] propuseram garantir que pelo menos um parâmetro da solução, escolhido aleatoriamente, seja alterado caso nenhum seja selecionado por meio do parâmetro R_j .

Quanto à fase das abelhas exploradoras, Karaboga e Akay [33] aplicaram em seu ABC

alterado a proposta de Karaboga e Basturk [32]. A fase das abelhas exploradoras ocorre então apenas a cada SSP ciclos.

Karaboga e Akay [33] experimentaram 9 valores para o parâmetro MR da Equação (3.5) (de 0,1 a 0,9 com passos de 0,1), 5 valores para $limite$ ($0,1 \times SN \times D$; $0,5 \times SN \times D$; $SN \times D$; $2 \times SN \times D$ e $4 \times SN \times D$) e 5 valores para SSP ($0,1 \times SN \times D$; $0,5 \times SN \times D$; $SN \times D$; $2 \times SN \times D$ e $4 \times SN \times D$), obtendo como as melhores atribuições: $MR \in \{0,3;0,8\}$, $limite \in \{0,5 \times SN \times D; SN \times D\}$ e $SSP \in \{0,1 \times SN \times D; 2 \times SN \times D\}$.

3.5.4 Período entre 2012 e 2013

Em 2012, com base em [41] e [42], Gao e Liu [43] propuseram que a inicialização da população no ABC seja feita combinando um Sistema Caótico com um Método de Aprendizagem baseado em Oposição. O Algoritmo 2 descreve os passos da inicialização proposta em [43].

Algoritmo 2: Nova inicialização do ABC proposta em [43]

– Sistema Caótico –

```

para  $i = 1 : SN$  faça
  |
  | para  $j = 1 : D$  faça
  | | Inicializar aleatoriamente  $ch_j \in (0, 1)$ ;
  | | para  $k = 1 : K$  faça
  | | |  $ch_j = \text{sen}(\pi ch_j)$ ;
  | | fim
  | |  $x_{ij} = x_{ij}^{min} + ch_j(x_{ij}^{max} - x_{ij}^{min})$ ;
  | fim
fim

```

– Método de Aprendizagem baseado em Oposição –

```

para  $i = 1 : SN$  faça
  |
  | para  $j = 1 : D$  faça
  | |  $x_{ij} = x_{ij}^{min} + x_{ij}^{max} - x_{ij}$ ;
  | fim
fim

```

Usando o mesmo operador de mutação do algoritmo DE [21], Gao e Liu [43],

propuseram também substituir a Equação (3.2) por

$$v_{ij} = x_{bestj} + \phi_{ij}(x_{r1j} - x_{r2j}), \quad (3.9)$$

onde $r1$ e $r2$ são índices distintos e diferentes de i , escolhidos aleatoriamente, \mathbf{x}_{best} é a melhor solução da população, ϕ_{ij} fica mantido como sendo um valor aleatório entre $[-1, 1]$.

Assim como Zhu e Kwong [36], Gao e Liu [43] constataram que a Equação (3.2) é boa para exploração, mas pobre para intensificação na busca local, causando convergência lenta ao ABC, diferentemente do que ocorre no algoritmo DE. Por outro lado a Equação (3.9) pode causar convergência prematura. Gao e Liu [43] propuseram então um novo mecanismo de busca local, cujos passos podem ser descritos no Algoritmo 3

Algoritmo 3: Mecanismo de busca local do ABC adotado em [43]

```

para  $i = 1 : SN$  faça
  Escolha  $X_{r1}$  e  $X_{r2} \in \{1, 2, \dots, SN\} - \{i\}$ ;
  Escolha  $j \in \{1, 2, \dots, D\}$ ;
  Produza  $\phi_{ij} \in [-1, 1]$ ;
  Aplique a Equação (3.9);
  se  $f(\mathbf{V}_i) < f(\mathbf{X}_i)$  então
    |  $\mathbf{X}_i = \mathbf{V}_i$ ;
  senão
    | se  $rand(0, 1) < P$  então
      | Escolha  $k \in \{1, 2, \dots, SN\} - \{i\}$ ;
      | Escolha  $j \in \{1, 2, \dots, D\}$ ;
      | Produza  $\phi_{ij} \in [-1, 1]$ ;
      | Aplique a Equação (3.2);
      | se  $f(\mathbf{V}_i) < f(\mathbf{X}_i)$  então
        | |  $\mathbf{X}_i = \mathbf{V}_i$ ;
      | fim
    | fim
  fim
fim

```

No Algoritmo 3 o parâmetro P balanceia exploração e busca local. Quando $P = 0$

apenas a Equação (3.9) será aplicada. Quando P cresce de 0 a 1, a exploração da Equação (3.2) cresce correspondentemente. P não pode ser tão grande pois pode enfraquecer a busca local do ABC [43].

Em 2012, Gao, Liu e Huang [44] apresentaram a Equação (3.10) para o cálculo da diversidade da população de soluções, dada por

$$Diversidade = \frac{1}{SN} \sum_{i=1}^{SN} \sqrt{\frac{1}{D} \sum_{j=1}^D (x_{ij} - \bar{x}_j)^2}, \quad (3.10)$$

onde SN é o número de fontes de alimento (soluções), D é o número de variáveis de projeto do problema e \bar{x}_j é a média do valor da variável j na colônia. Quanto maior o valor de *Diversidade*, melhor.

Gao, Liu e Huang [44] substituíram a Equação (3.2) pela Equação (3.9), mas também usaram a Equação (3.11), a mesma usada no DE, para a busca local das abelhas campeiras e observadoras, dada por

$$v_{ij} = x_{bestj} + \phi_{ij}(x_{r1j} - x_{r2j}) + \phi_{ij}(x_{r3j} - x_{r4j}) \quad (3.11)$$

onde os índices $r1, r2, r3$ e $r4$ são escolhidos aleatoriamente em $\{1, 2, \dots, SN\}$, devendo serem mutuamente exclusivos e diferentes de i , \mathbf{x}_{best} é a melhor solução da população e $j \in \{1, 2, \dots, D\}$ é um índice escolhido aleatoriamente. Como na Equação (3.2), ϕ_{ij} é um valor aleatório entre $[-1, 1]$. Assim como na Equação (3.9), na Equação (3.11) a busca permanece em torno da melhor solução, aumentando assim a intensificação da exploração no ABC. Em [44] os melhores resultados foram obtidos quando usada a Equação (3.9).

Para a fase das abelhas exploradoras, Gao, Liu e Huang [44] propuseram substituição da busca aleatória pelos passos do Algoritmo 2.

Em 2013, Xiang e An [45] propuseram uma nova forma de inicialização da população para o ABC. Com o objetivo de obter diversidade e alta qualidade das soluções iniciais, como fizeram Gao e Liu [43], Xiang e An [45] adotam o Sistema Caótico na inicialização da população. A diferença é que o Sistema Caótico proposto por Xiang e An [45] gera o termo caótico (ch) usando a equação

$$ch_{k+1} = \mu ch_k(1 - ch_k), \quad 0 \leq ch_0 \leq 1, \quad (3.12)$$

onde k é uma de um total de K iterações caóticas, ch_k é o termo caótico na iteração k , sendo que $ch_0 \notin \{0; 0, 25; 0, 5; 0, 75; 1\}$, μ é um parâmetro de controle que proporciona dinamismo caótico especialmente se $\mu = 4$ segundo Xiang e An [45].

Xiang e An [45] propuseram o uso da Equação (3.12) com base nos estudos de Alatas, Akin e Ozer [46] que constataram que o mapeamento caótico logístico propicia melhor desempenho que o mapeamento caótico senoidal.

Outra alteração para o ABC proposta por Xiang e An [45] diz respeito ao cálculo das probabilidades de seleção das fontes de alimento. Com o objetivo de manter a diversidade da população, Xiang e An [45] propuseram a substituição da Equação (3.3) por

$$p_i = \frac{\frac{1}{fit_i}}{\sum_{j=1}^{SN} \frac{1}{fit_j}}, \quad (3.13)$$

onde fit_i corresponde à aptidão da solução i , ainda calculada pela Equação (3.4).

Para a fase das abelhas observadoras, Xiang e An [45] propuseram a geração de duas novas soluções candidatas e não apenas uma. Com base nos estudos de Banharnsakun, Achalakul e Sirinaovakul [47], para aumentar a velocidade de convergência do ABC, Xiang e An [45] substituem a Equação (3.2) por

$$v_{ij} = x_{ij} + \phi_{ij} [x_{ij} - x_{bj}] f_b, \quad (3.14)$$

onde aparecem os termos x_{bj} e f_b que correspondem respectivamente à variável j e o valor da função objetivo da melhor solução x_b , os demais termos ficam mantidos como na Equação (3.2). A segunda equação usada por Xiang e An [45] para substituir da Equação (3.2) é dada por

$$\bar{v}_{ij} = x_{ij} + \phi_{ij} [x_{ij} - x_{bj}] \exp\left(\frac{t}{MCN}\right), \quad (3.15)$$

onde aparecem t que é o número da iteração atual e MCN que é o número máximo de ciclos, as outras variáveis assumem os mesmos papéis que na Equação (3.14).

Por fim, tentando evitar que a busca fique presa num mínimo local, baseados na Equação (3.12), Xiang e An [45] ainda propuseram alteração para a fase das abelhas exploradoras. O Algoritmo 4 descreve os passos de cada abelha exploradora ao serem definidas as soluções (x_i no Algoritmo 4) que deverão ser substituídas.

Algoritmo 4: Busca Caótica das abelhas exploradoras aplicada em [45]

```

para  $j = 1 : D$  faça
  |  $ch_j = (x_{ij} - x_j^{min}) / (x_j^{max} - x_j^{min});$ 
fim
 $k = 1;$ 
enquanto  $k < K$  faça
  |
  | para  $j = 1 : D$  faça
  | |  $ch_j = 4 \times ch_j \times (x_j^{max} - x_j^{min});$ 
  | |  $v_j = x_j^{min} + ch_j \times (x_j^{max} - x_j^{min});$ 
  | | // $v_j$  deve estar dentro dos valores limites definidos para a variável  $j$ .
  | fim
  | se  $f(\mathbf{V}) < f(\mathbf{X}_i)$  então
  | |  $\mathbf{X}_i = \mathbf{V};$ 
  | | Interromper busca por solução que substituirá  $\mathbf{X}_i;$ 
  | fim
  |  $k = k + 1$ 
fim

```

O nome dado ao novo algoritmo proposto por Xiang e An [45] foi ERABC.

3.6 Aplicações do ABC

Desde que foi proposto, o ABC vem sendo aplicado na resolução de problemas de otimização de classes distintas. Os parágrafos seguintes descrevem algumas destas aplicações.

Em 2009, Singh [48] aplicou o ABC no problema de busca de árvore geradora mínima em um grafo contendo n nós, com conexões ponderadas não direcionadas. O problema pode ser descrito basicamente como a busca por uma árvore contendo pelo menos l ($2 \leq l < n - 1$) folhas, que possua a menor soma possível de pesos entre as conexões. Este tipo de problema está relacionado, por exemplo, ao projeto de circuitos, redes de computadores etc

Para modelar o problema no ABC, Singh [48] tomou como base os operadores propostos por Julstrom [49]. Singh [48] aplicou o algoritmo ABC-LCMST (Nome dado ao

ABC proposto em [48].) e também os algoritmos ACO-LCMST e TS-LCMST, propostos em [50], além do algoritmo SCGA proposto em [49]. Em seguida os resultados gerados pelos quatro algoritmos foram comparados.

Singh [48] concluiu que os melhores resultados foram obtidos com o algoritmo ABC-LCMST. Também foi constatado em [48] que entre os algoritmos experimentados, o ABC-LCMST exigiu o menor custo computacional para resolver o problema.

Em 2010, ao aplicarem o ABC na resolução de problemas de otimização estrutural, além de proporem alterações para o ABC descritas na Seção 3.5.3, Hadidi, Azad e Azad [38] introduziram ao algoritmo um método de penalização adaptativo (Classe de métodos de penalização descrita na Seção 4.2.3 deste trabalho.) para tratamento das restrições.

Como função de penalização, Hadidi, Azad e Azad [38] adotaram a equação proposta por Rajeev e Krishnamoorthy [51], definida por

$$\begin{aligned} P(\mathbf{X}) &= W(\mathbf{X})KC \\ C &= \sum_{r=1}^{NG} g_r(\mathbf{X}) \end{aligned} \quad , \quad (3.16)$$

onde \mathbf{X} é um vetor D -dimensional, $W(\mathbf{X})$ é o peso da estrutura, K é uma constante de penalização e g_r é o valor de violação à restrição r de um total NG entre restrições de tensão e deslocamento.

Para adaptação de K , Hadidi, Azad e Azad [38] usaram a Equação (3.17), proposta por Koohestani e Azad [40]. K inicia com valor pequeno e a cada iteração do ABC sofre modificação dada por

$$\begin{cases} K(t) = K(t-1) + \Delta K & , \text{ se melhor solução é infactível} \\ K(t) = K(t-1) - \Delta K/2, & \text{ se melhor solução é factível} \end{cases} \quad , \quad (3.17)$$

onde $K(t)$ é o valor de K na iteração t e ΔK é o tamanho do passo a ser definido pelo usuário.

Hadidi, Azad e Azad [38] aplicaram os algoritmos MABC (Nome dado ao ABC proposto em [38]) e ABC proposto em [26] e em seguida compararam os resultados obtidos. O algoritmo MABC gerou melhores resultados que o ABC.

Em 2011, Karaboga e Ozturk [52] aplicaram o ABC na resolução de problema de classificação e agrupamento de dados. O agrupamento é o processo de reconhecimento

de padrão num conjunto de dados para então agrupá-los de acordo com medidas de similaridade num espaço multidimensional. Medidas de distância são geralmente usadas para avaliação de similaridade. O problema de otimização passa a ser então minimizar a soma das distâncias entre cada objeto e o centro do grupo ao qual ele pertence [52].

O problema de agrupamento pode ser descrito por

$$J(w, z) = \sum_{i=1}^N \sum_{j=1}^K w_{ij} \|x_i - z_j\|^2, \quad (3.18)$$

onde K é o número de grupos, N é o número de dados, x_i é a localização D -dimensional do dado i e z_j é o centro do grupo j , dado por

$$z_j = \frac{1}{N_j} \sum_{i=1}^N w_{ij} x_i, \quad (3.19)$$

onde N_j é o número de dados no grupo j , w_{ij} é o peso de associação do dado x_i com o grupo j , que será 1 (se o dado i está alocado no grupo j) ou 0 (se o dado i não está alocado no grupo j).

Karaboga e Ozturk [52] aplicaram também um algoritmo PSO [53] e em seguida compararam os resultados obtidos. Foi constatado melhor desempenho do ABC em 92% dos problemas analisados.

Em 2011, para resolver problemas de otimização estrutural, Sonmez [54] combinou ao ABC o método de penalização adaptativo (Classe de métodos de penalização descrita na Seção 4.2.3 deste trabalho.) para tratar as restrições. Para converter o problema com restrições para um problema sem restrições, Sonmez [54] aplicou a equação proposta por Hasançebi [55], dada por

$$W_c = W_i \left[1 + r \left(\sum_{j=1}^m g_j \right) \right], \quad (3.20)$$

onde W_c é a nova função objetivo sem restrições, W_i é a função objetivo gerada com a solução x_i , g_j ($j = 1, 2, \dots, m$) correspondem às violações às restrições de tensão e deslocamento cometidas pela solução x_i . r é o parâmetro de penalização que foi ajustado a cada iteração do ABC por

$$r_i(t) = \begin{cases} 1/f \times r_i(t-1), & \text{se } W_{ci}(t-1) \text{ é factível} \\ f \times r_i(t-1) & , \text{ se } W_{ci}(t-1) \text{ é infactível} \end{cases}, \quad (3.21)$$

onde $r_i(t)$ e $r_i(t - 1)$ são os parâmetros de penalização relacionados à solução x_i nas iterações t e $t - 1$ respectivamente, $W_{ci}(t - 1)$ corresponde ao valor de função objetivo da solução x_i na iteração $t - 1$ e f é uma constante arbitrária, que denota a taxa de aprendizado para $r(t)$. O valor recomendado para f na literatura é 1,1, porém Sonmez [54] ainda propôs que o valor de f seja dado por

$$f = 1 + \frac{1}{n_c}, \quad (3.22)$$

onde $f > 1,01$, n_c é o número total de restrições. Com isto, o ajuste do parâmetro de penalização $r(t)$ fica relacionado ao número de restrições do problema, quanto maior mais rígida será a penalização.

Sonmez [54] aplicou os algoritmos ABC-AP (Nome dado ao ABC proposto em [54]) e ABC proposto em [26] e em seguida comparou os resultados obtidos. O algoritmo ABC-AP gerou melhores resultados que o ABC.

Em 2012, Akay e Karaboga [29] ressaltaram que problemas de otimização de projetos de engenharia são geralmente de grande escala ou não-lineares ou possuem restrições e então aplicam o ABC na resolução destes problemas.

Para resolução de problemas de grande escala, porém sem restrições, Akay e Karaboga [29] aplicaram o ABC proposto em [26].

Para resolução de problemas com restrições, Akay e Karaboga [29] explicam que devido ao fato do ABC não possuir uma função de penalização, a aplicação deste método para o tratamento de restrições exigiria ajuste fino de parâmetros e da própria função de penalização. Com este argumento, Akay e Karaboga [29] aplicam o ABC combinado ao critério de seleção proposto por Deb, descrito em [34], assim como foi feito em [32], [52] e [33].

No algoritmo avaliado por Akay e Karaboga [29] também foi aplicado o parâmetro *SSP*, proposto em [32], fazendo com que a fase das abelhas exploradoras ocorra apenas a cada *SSP* ciclos. Os passos do ABC aplicado em [32], [33] e [29] podem ser descritos no Algoritmo 5.

Algoritmo 5: ABC Modificado aplicado em [32], [33] e [29]

Inicialize a população;

Avalie a população;

repita

 Produza uma nova solução (v_i) usando a Equação (3.5) para cada abelha campeira;

 Selecione entre v_i e x_i baseado no critério de Deb [34];

 Calcule os valores de probabilidade das soluções usando a Equação (??);

 Posicione as abelhas observadoras nas fontes dependendo de seus valores;

 Produza uma nova solução (v_i) usando a Equação (3.5) para cada abelha observadora;

 Selecione entre v_i e x_i baseado no critério de Deb [34];

se *SSP* ciclos completados **então**

 Determinar soluções abandonadas usando *limite*;

 Envie abelhas exploradoras para buscar novas fontes usando a Equação (3.1);

fim

 Memorize a melhor fonte de alimento encontrada até o momento;

até *Número máximo de ciclos*;

Em 2014, Bullinaria e AlYahya [56] aplicaram o ABC no processo de treinamento supervisionado de uma rede neuronal. Neste caso o espaço de busca multi-dimensional corresponde aos pesos das conexão entre os neurônios da rede. A função objetivo é uma medida de desempenho da rede, como a soma dos erros gerados na comparação entre dos dados obtidos e os dados esperados.

Bullinaria e AlYahya [56] avaliaram o ABC sob distintas configurações alterando os parâmetros envolvidos no algoritmo (*SN* e *limite*), além de definirem limites inferiores e superiores permitidos para os pesos das conexões da rede.

Bullinaria e AlYahya [56] concluíram que com a aplicação dos algoritmos ABC ou BP (do inglês *Back-Propagation* - algoritmo tradicionalmente aplicado em treinamento de redes neurais) foi possível obter desempenho similar da rede, mas o ABC gerou melhores resultados.

4 Tratamento de restrições

4.1 Introdução

Em problemas de otimização com restrições uma solução será dita factível se atender as restrições, caso contrário será considerada infactível. Nos algoritmos populacionais, manter um conjunto de soluções candidatas exclusivamente factíveis pode se tornar uma tarefa complexa. Manter também soluções infactíveis exige que seja adotada alguma estratégia para o tratamento das restrições do problema. É importante observar que a técnica escolhida para tratar as restrições pode influenciar no desempenho do algoritmo.

Quando se fala da aplicação de algoritmos bio-inspirados para a resolução de problemas de otimização com restrições, uma estratégia bastante utilizada na literatura para o tratamento das restrições, é adotar o critério de seleção proposto por Deb [34]. Este critério pode ser considerado um método simples e de baixo custo computacional. O critério consiste em promover um torneio entre duas soluções candidatas, onde a vencedora será definida aplicando três regras básicas [34]:

- Qualquer solução factível será preferida contra qualquer solução infactível.
- Entre duas soluções factíveis, aquela com o melhor valor de função objetivo será preferida.
- Entre duas soluções infactíveis, aquela com a menor violação às restrições será preferida.

Por esta estratégia existe uma grande chance de material de qualidade presente entre as soluções infactíveis da população deixar de ser usado nas operações do algoritmo de busca. Dependendo das operações realizadas no algoritmo, a chance deste tipo de solução ser totalmente descartada também pode ser alta.

Neste trabalho, a técnica escolhida para tratamento das restrições foi o método de penalização. Este capítulo traz a descrição do método de penalização, como se aplica e algumas alternativas para manipulação dos parâmetros envolvidos. Por fim são listadas algumas limitações identificadas em alguns dos métodos de penalização discutidos, para os quais são apresentadas sugestões de alteração.

4.2 Métodos de penalização

Os métodos de penalização têm como objetivo transformar um problema de otimização com restrições em um problema sem restrições. Na sequência o problema poderá ser resolvido por meio de um algoritmo para resolução de problemas sem restrições. Os métodos de penalização podem ser aplicados a problemas de otimização contendo restrições de igualdade e/ou desigualdade.

A penalização se dá na forma de um acréscimo ou amplificação ao valor da função objetivo caso a solução seja infactível. O valor da penalização será determinado em função das violações às restrições, cometidas pelas soluções infactíveis. As violações estarão envolvidas na penalização por meio de uma função de penalização.

Uma função de penalização deve ser contínua e também gerar valores positivos para pontos infactíveis e 0 ou 1 (dependendo do tipo de penalização) para pontos factíveis. Dados problemas de otimização formulados como na Equação (2.4), uma função de penalização adequada pode ser definida por

$$\alpha(\mathbf{x}) = \sum_{i=1}^m [\max(0, g_i(\mathbf{x}))]^p + \sum_{j=1}^q |h_j(\mathbf{x})|^p, \quad (4.1)$$

onde p é um inteiro positivo (usualmente 1 ou 2), g_i ($i = 1, 2, \dots, m$) e h_j ($j = 1, 2, \dots, q$) são as próprias funções das restrições do problema [12, 57, 58, 6].

4.2.1 Penalização estática

Dado o problema de otimização da Equação (2.4), aplicando o método de penalização, ao ser convertido em um problema sem restrições o mesmo poderá ser reescrito na forma

$$\begin{cases} \text{Minimize} & f(\mathbf{x}) + k\alpha(\mathbf{x}) \\ \text{Sujeito a} & \mathbf{x} \in \mathbb{R}^n \end{cases} \quad (4.2)$$

onde $\alpha(\mathbf{x})$ é definida pela Equação (4.1), k é chamado de parâmetro de penalização, devendo ser $k > 0$ e suficientemente grande. O parâmetro k define a severidade da penalização. A penalização será dita estática quando o valor de k se mantém constante durante o processo de busca pela solução do problema [12, 58, 6].

Na Equação (4.2) o termo $k\alpha(\mathbf{x})$ está sendo somado ao valor da função objetivo $f(\mathbf{x})$,

caracterizando assim uma penalização do tipo **Aditiva**. Pelo **Método de Penalização Aditivo** a penalização consiste de um acréscimo ao valor da função objetivo. Neste caso, o valor da penalização será > 0 para as soluções infactíveis e 0 para as soluções factíveis [12, 58, 6].

Uma outra alternativa para reescrever o problema da Equação (2.4) de forma a convertê-lo em um problema sem restrições é dada por

$$\begin{cases} \text{Minimize} & f(\mathbf{x}) \times (1 + k\alpha(\mathbf{x})) \\ \text{Sujeito a} & \mathbf{x} \in \mathbb{R}^n \end{cases}, \quad (4.3)$$

onde $\alpha(\mathbf{x})$ continua sendo definida pela Equação (4.1) e k continua sendo o parâmetro de penalização.

Na Equação (4.3) o termo $(1 + k\alpha(\mathbf{x}))$ amplifica o valor da função objetivo $f(\mathbf{x})$, o que caracteriza uma penalização do tipo **Multiplicativa** [59]. Pelo **Método de Penalização Multiplicativo** a penalização será 1 para as soluções factíveis e > 1 para as soluções infactíveis [60].

4.2.2 Penalização dinâmica

Ao transformar o problema da Equação (2.4) em um problema sem restrições, quando aplicada a penalização do tipo dinâmica, poderão ser aplicadas as formas dadas pelas Equações (4.2) ou (4.3).

Pelo Método de Penalização Dinâmico o valor do parâmetro de penalização k presente nas Equações (4.2) e (4.3) será alterado (usualmente crescer) durante as iterações do processo de busca pela solução do problema. Tipicamente a atualização do valor do parâmetro k é dada por

$$k = (C \times t)^\eta, \quad (4.4)$$

onde C é um valor real positivo, cujo valor sugerido na literatura é $0,5$, para η o valor sugerido é 2 . Com a Equação (4.4) o valor de k crescerá com o aumento do número de iterações t . Desenvolver uma penalização dinâmica eficiente pode se tornar uma tarefa tão difícil quanto encontrar um parâmetro de penalização que funcione bem na penalização estática [57, 61, 59].

4.2.3 Penalização adaptativa

Claramente o ajuste do parâmetro k presente nas Equações (4.2) e (4.3) não consiste de uma tarefa trivial [62]:

- Se k for pequeno, pode provocar convergência para uma região não atrativa (infectível).
- Se k for grande, boas características de soluções infectíveis não terão a chance de serem consideradas, provocando convergência prematura para uma solução não tão boa.

Na penalização adaptativa, como na penalização dinâmica, o valor do parâmetro de penalização (k) será alterado durante o processo de busca pela solução do problema. A diferença entre os dois métodos é que na penalização adaptativa, para atualizar o valor de k são usadas informações geradas durante o processo de busca [59]. Fica restando então definir quais serão e como as informações serão trabalhadas para o ajuste do parâmetro de penalização.

Diversas formas de atualização do parâmetro de penalização já foram propostas. Uma das primeiras foi proposta em 1992 por Bean e Hadj-Alouane [63], que usando um Algoritmo Genético como ferramenta de busca, sugeriram que o parâmetro de penalização λ fosse alterado a cada N_f gerações. Em [63] a atualização de λ depende da condição do melhor indivíduo da população nas últimas N_f gerações do algoritmo. Assim, ao longo da execução do algoritmo o valor de λ é definido por

$$\lambda_{k+1} = \begin{cases} \beta_1 \lambda_k, & \text{se o melhor indivíduo por } N_f \text{ gerações foi infectível} \\ \frac{\lambda_k}{\beta_2}, & \text{se o melhor indivíduo por } N_f \text{ gerações foi factível} \\ \lambda_k, & \text{caso contrário} \end{cases} \quad (4.5)$$

onde λ_{k+1} é o valor do parâmetro a ser usado na próxima geração ($k + 1$), já que k é a geração corrente. O valor de λ para a primeira geração ($k = 1$) deve ser definido pelo usuário. β_1 e β_2 são parâmetros que correspondem respectivamente à taxa de crescimento e decrescimento do parâmetro de penalização. Para β_1 e β_2 deve ser observado que $\beta_1 > \beta_2 > 1$ objetivando melhoria das soluções candidatas no início do algoritmo.

Um método adaptativo foi proposto por Coit, Smith e Tate [64], que também adotaram o Algoritmo Genético como ferramenta de busca ao tratar de um problema

de maximização. Pelo método proposto em [64] a função objetivo penalizada será dada por

$$F_p(\mathbf{x}) = F(\mathbf{x}) - (F_{all} - F_{feas}) \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i} \right]^{k_i}, \quad (4.6)$$

onde $F_p(\mathbf{x})$ corresponde à função objetivo penalizada, $F(\mathbf{x})$ é a função objetivo não penalizada, F_{all} é a função objetivo não penalizada da melhor solução da população e F_{feas} denota a função objetivo não penalizada da melhor solução factível da população. Não havendo soluções factíveis para determinação de F_{feas} , Coit, Smith e Tate [64] definiram que uma solução infactível que viole pouco as restrições é equivalente a uma solução factível. Também na Equação (4.6), $d(\mathbf{x})$ é uma função que indica a distância entre \mathbf{x} e a região factível. Desta forma $d_i(\mathbf{x})$ pode ser vista como a medida de violação da solução \mathbf{x} a cada uma das restrições i do problema. Uma definição adequada para $d_i(\mathbf{x})$ ($i = 1, 2, \dots, m + q$) pode ser dada por

$$d_i(\mathbf{x}) = \begin{cases} \max(0, g_i(\mathbf{x})), & \text{se a restrição } i \text{ for do tipo } g_i(\mathbf{x}) \leq 0 \\ |h_i(\mathbf{x})| & , \text{ se a restrição } i \text{ for do tipo } h_i(\mathbf{x}) = 0 \end{cases}. \quad (4.7)$$

Ainda na Equação (4.6), NFT_i (do inglês *Near-Feasibility Threshold*) consiste de um limiar próximo à região factível referente à restrição i . Neste caso, pode-se considerar como uma boa escolha para NFT_i o valor da menor violação da restrição i no conjunto de soluções candidatas.

Ao exemplificar a aplicação do método proposto em [64] na resolução de problemas de minimização, Coit, Smith e Tate [64] propuseram substituir a Equação (4.6) por

$$F_p(\mathbf{x}) = F(\mathbf{x}) + (F_{feas} - F_{all}) \left(\frac{\eta}{NFT} \right)^k, \quad (4.8)$$

em que F_{all} e F_{feas} tiveram posições invertidas, η corresponde ao número de restrições violadas por \mathbf{x} e NFT foi escolhido como sendo a violação total cometida pela solução \mathbf{x} .

Yokota et al. [65] propuseram um método de penalização multiplicativo adaptativo ao aplicarem o Algoritmo Genético na resolução de um problema de maximização com restrições. Pelo método proposto em [65], a função objetivo penalizada $F(\mathbf{x})$ será dada por

$$F(\mathbf{x}) = f(\mathbf{x}) \left[1 - \frac{1}{m+q} \sum_{i=1}^{m+q} d_i(\mathbf{x}) \right], \quad (4.9)$$

onde $f(\mathbf{x})$ é a função objetivo não penalizada, m e q são os totais de restrições de desigualdade e igualdade respectivamente, $d_i(\mathbf{x})$ é a violação cometida pela solução \mathbf{x} em cada uma das restrições i ($i = 1, 2, \dots, m + q$). Nestas condições $d_i(\mathbf{x})$ pode ser definida pela Equação (4.7).

Ao propor um método para o tratamento de restrições que independe de parâmetros, Deb [34] argumentou que a partir do momento que se sabe que uma solução candidata é infactível, não faz sentido calcular sua função objetivo. Tal solução não poderá ser implementada na prática. Com base neste argumento, ao resolver o problema da Equação (2.4), Deb [34] propôs uma função para a avaliação das soluções candidatas definida por

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se } \mathbf{x} \text{ é factível} \\ f_{max} + \alpha(\mathbf{x}), & \text{ caso contrário} \end{cases}, \quad (4.10)$$

onde $f(\mathbf{x})$ é a função objetivo definida no problema, f_{max} corresponde ao valor da função objetivo da pior solução factível presente no conjunto de soluções, $\alpha(\mathbf{x})$ é definida pela Equação (4.1), adotando $p = 1$. Segundo Deb [34], não existindo soluções factíveis, $f_{max} = 0$. Pela Equação (4.10) nenhuma solução infactível terá valor de função objetivo melhor que a pior factível.

Também adotando o Algoritmo Genético como ferramenta de busca, Montemurro, Vincenti e Vannucci [66] propuseram uma estratégia de penalização adaptativa intitulada ADP (do inglês *Automatic Dynamic Penalisation*), na qual estão presentes parâmetros de penalização relacionados a cada uma das restrições do problema. A idéia básica por traz do ADP é que algumas soluções infactíveis podem ser importantes para dirigir a busca para regiões interessantes do espaço factível. Por esta razão, no ADP as soluções infactíveis não são descartadas e são usadas no ajuste dos parâmetros de penalização.

Quando aplicado o método ADP, ao transformar o problema da Equação (2.4) em um problema sem restrições, a função objetivo será dada por

$$F_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se } \mathbf{x} \text{ é factível} \\ f(\mathbf{x}) + \sum_{i=1}^m c_i d_i(\mathbf{x}) + \sum_{j=1}^q r_j d_j(\mathbf{x}), & \text{ caso contrário} \end{cases}, \quad (4.11)$$

onde c_i ($i = 1, 2, \dots, m$) e r_j ($j = 1, 2, \dots, q$) são os parâmetros de penalização relacionados a cada uma das restrições $g_i(\mathbf{x})$ e $h_j(\mathbf{x})$, $d_i(\mathbf{x})$ e $d_j(\mathbf{x})$ indicam as violações cometidas pela

solução \mathbf{x} e são definidas por

$$\begin{aligned} d_i(\mathbf{x}) &= \max(0, g_i(\mathbf{x})) \\ d_j(\mathbf{x}) &= |h_j(\mathbf{x})| \end{aligned} \quad , \quad (4.12)$$

onde $g_i(\mathbf{x})$ ($i = 1, 2, \dots, m$) são as restrições do tipo $g(\mathbf{x}) \leq 0$ e $h_j(\mathbf{x})$ ($j = 1, 2, \dots, q$) são as restrições do tipo $h(\mathbf{x}) = 0$.

O ajuste dos parâmetros c_i e r_j da Equação (4.11) será realizado a cada iteração do algoritmo de busca. Para isto a população deverá ser dividida em dois grupos distintos, um composto por soluções factíveis e o outro composto por soluções infactíveis. Na sequência, a melhor solução de cada grupo deverá ser considerada uma candidata em potencial para resolver o problema. Com base nestas informações, o ajuste dos parâmetros de penalização é definido por

$$\begin{aligned} c_i &= \frac{|f_{best}^F - f_{best}^{NF}|}{(G_i)_{best}^{NF}} \\ r_j &= \frac{|f_{best}^F - f_{best}^{NF}|}{(H_j)_{best}^{NF}} \end{aligned} \quad , \quad (4.13)$$

onde F e NF significam factível e não-factível respectivamente. f_{best}^F e f_{best}^{NF} são os valores da função objetivo das melhores soluções factível e infactível respectivamente, enquanto $(G_i)_{best}^{NF}$ e $(H_j)_{best}^{NF}$ representam as violações da melhor solução infactível em relação a cada uma das restrições. No caso de não existirem soluções factíveis no conjunto, aquelas com a menor violação em relação a uma determinada restrição serão reunidas num sub-grupo de soluções “virtualmente factíveis”, de onde deverá sair o valor f_{best}^F . O restante será considerado realmente infactível. Montemurro, Vincenti e Vannucci [66] definiram que o número de soluções “virtualmente factíveis” corresponde a 10% do conjunto de soluções candidatas.

4.2.4 Método de Penalização Adaptativo – APM

O APM é um método de penalização adaptativo proposto por Barbosa e Lemonge [67] em 2002 e formalizado por Barbosa e Lemonge [68] em 2003 com o nome *Adaptive Penalty Method* (APM).

Uma característica bastante atrativa do APM é que o método não depende de qualquer tipo de parâmetro a ser definido pelo usuário. Aplicando o APM as informações

necessárias para ajuste dos valores dos parâmetros de penalização são basicamente:

- Média do valor da função objetivo.
- Nível de violação de cada restrição.

Pelo APM, ao transformar o problema da Equação (2.4) em um problema sem restrições, a função objetivo será dada por

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se } \mathbf{x} \text{ é factível} \\ \bar{f}(\mathbf{x}) + \sum_{j=1}^m k_j v_j(\mathbf{x}), & \text{ caso contrário} \end{cases}, \quad (4.14)$$

onde

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \text{ se } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle \\ \langle f(\mathbf{x}) \rangle, & \text{ caso contrário} \end{cases}, \quad (4.15)$$

em que $\langle f(\mathbf{x}) \rangle$ é a média do valor da função objetivo na população corrente.

No APM não existe a ideia de que uma solução factível é superior a uma solução infactível. Na Equação (4.14), pode ser observado que além de ser penalizada, uma solução infactível não poderá ter valor de função objetivo menor que a média deste valor no conjunto de soluções candidatas como define a Equação (4.15). A Figura 4.1 ilustra a definição de $\bar{f}(\mathbf{x})$ dada pela Equação (4.15). A figura apresenta um conjunto de 10 soluções (factíveis e infactíveis). Entre as soluções infactíveis (1, 2, ..., 6), 1, 2, 3 e 4 têm seus valores de função objetivo (representados por círculos abertos) menores que a média da população. De acordo com o APM, o valor de $\bar{f}(\mathbf{x})$ para estas soluções, deverá ser $\langle f(\mathbf{x}) \rangle$. Já os indivíduos 5 e 6 têm valor de função objetivo pior que a média da população, então estes valores serão mantidos ($\bar{f}(\mathbf{x}) = f(\mathbf{x})$).

Na Equação (4.14) v_j é o valor da violação a cada uma das restrições j , cometida pela solução \mathbf{x} . Além disso, na Equação (4.14) estão presentes os parâmetros de penalização k_j relacionados a cada uma das restrições. A ideia é que os parâmetros de penalização sejam distribuídos de forma que aqueles relacionados a restrições mais difíceis de serem atendidas tenham valores relativamente mais altos [67, 69, 59]. A atualização de cada um desses parâmetros durante o processo de busca é dada por

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2}, \quad (4.16)$$

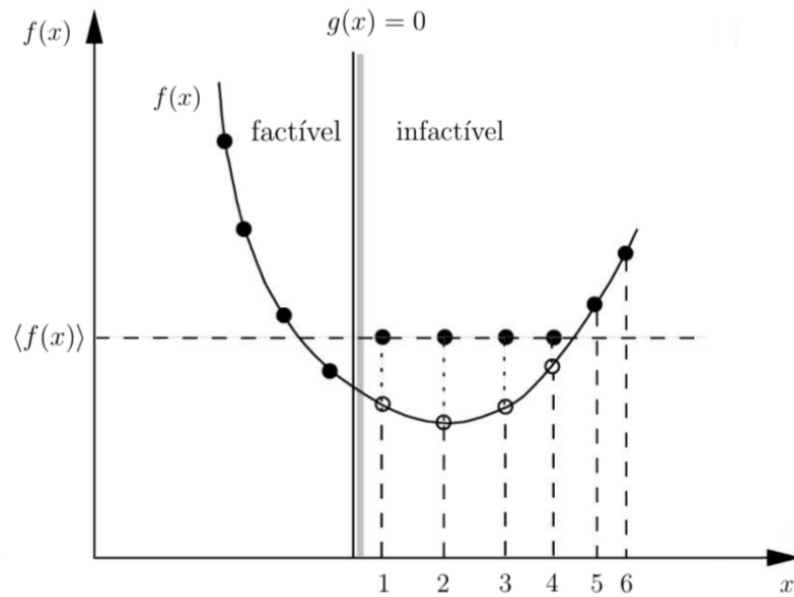


Figura 4.1: Definição de $\bar{f}(\mathbf{x})$ no APM, adaptada de [67].

onde $\langle v_j(\mathbf{x}) \rangle$ ($j = 1, 2, \dots, m$) e $\langle v_l(\mathbf{x}) \rangle$ são respectivamente as médias das violações às restrições j e l na população corrente.

Barbosa e Lemonge [69] observaram que quando se aplica o método APM junto a um Algoritmo Genético do tipo geracional, basta atualizar os parâmetros de penalização a cada geração. Porém, quando o Algoritmo Genético é do tipo *Steady-State*, em que a população sofre alterações durante as gerações, são necessárias algumas modificações no APM. Barbosa e Lemonge [69] propuseram então para o APM, que a função objetivo penalizada continue sendo dada pela Equação 4.14. Porém, o valor de $\bar{f}(\mathbf{x})$ passa a ser o valor h definido por

$$h = \begin{cases} f(\mathbf{x}_p), & \text{se não existe solução factível na população} \\ f(\mathbf{x}_b), & \text{caso contrário} \end{cases}, \quad (4.17)$$

onde \mathbf{x}_p é a pior solução do conjunto e \mathbf{x}_b é a melhor solução factível. Quanto a atualização dos parâmetros de penalização k_j , Barbosa e Lemonge [69] propuseram a equação

$$k_j = h \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2}, \quad (4.18)$$

onde $\langle v_j(\mathbf{x}) \rangle$ ($j = 1, 2, \dots, m$) e $\langle v_l(\mathbf{x}) \rangle$ continuam sendo respectivamente as médias das

violações às restrições j e l na população corrente. Quanto à frequência de atualização dos valores dos parâmetros de penalização e do valor de h , Barbosa e Lemonge [69] definiram que esta etapa se realiza apenas quando um novo indivíduo inserido na população se tratar do melhor factível ou quando for atingido um determinado nível (definido pelo usuário) de inserção de novos indivíduos na população. Além disso, Barbosa e Lemonge [69] definiram que ao ajustar os parâmetros de penalização todas as soluções devem ser avaliadas com os novos valores [69].

Apesar do bom desempenho apresentado pelo APM, em 2008 Barbosa e Lemonge [70] levantaram algumas questões a respeito do método:

- Possibilidade de redução das oscilações dos parâmetros k_j apresentadas em [68].
- Possibilidade de redução do custo computacional sem comprometer a qualidade dos resultados.

Na tentativa de avaliar tais possibilidades, Barbosa e Lemonge [70] propuseram algumas variantes para o APM, descritas por:

- APM Esporádico:
 - Calcular as médias de violações às restrições (v_j) na população corrente.
 - Atualizar os parâmetros k_j aplicando a Equação (4.16).
 - Manter os parâmetros de penalização k_j fixos por f gerações.

Fica claro o ganho no tempo computacional, mas visto que os parâmetros serão atualizados apenas a cada f gerações, deve ser observado se fica mantido o desempenho do método.

- APM Esporádico com acúmulo de violações:
 - Acumular violações às restrições (v_j) por f gerações.
 - Atualizar os parâmetros k_j aplicando a Equação (4.16), usando valores acumulados das violações.
 - Manter os parâmetros de penalização k_j fixos por f gerações.

O ganho computacional pode ser pequeno, mas as oscilações nos valores de k_j podem ser reduzidas.

- APM com parâmetros de penalização monotônicos:
 - Parâmetros de penalização devem crescer indefinidamente.
 - Variante consiste em não permitir que os parâmetros de penalização k_j tenham seus valores reduzidos ao serem atualizados ao longo do processo de busca.
- APM com amortecimento:
 - Reduzir oscilação dos valores de k_j usando uma média ponderada entre o antigo e o novo valor determinado.
 - Aplicar

$$k_j = \theta k_j^{novo} + (1 - \theta) k_j^{corrente}, \quad (4.19)$$

onde $\theta \in [0, 1]$.

Além de proporem as variantes para o APM descritas no parágrafo anterior, no mesmo trabalho Barbosa e Lemonge [70] ainda experimentaram a combinação de algumas delas:

- Acumular violações v_j por f gerações e em seguida amortecer a atualização dos parâmetros k_j aplicando a Equação (4.19).
- Acumular as violações v_j por f gerações, amortecer a atualização dos parâmetros k_j aplicando Equação (4.19) e considerar a forma monotônica não permitindo redução nos valores de k_j .

Em 2012, Lemonge, Barbosa e Bernardino [71] propuseram algumas variantes para a adaptação do APM proposta em [69], para aplicação do método em algoritmos do tipo *Steady-State*. Em todas as variantes propostas em [71] todos os parâmetros são mantidos fixos durante $r \times N$ inserções de novos indivíduos na população.

Em 2013, Garcia et al. propuseram novas variantes para o método APM em [72], além de mencionar aquelas propostas por Barbosa e Lemonge em [70]. As novas variantes propostas são:

- Variante 01:

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \quad \text{se } f(\mathbf{x}) > \lfloor f(\mathbf{x}) \rfloor \\ \lfloor f(\mathbf{x}) \rfloor & , \quad \text{outro caso} \end{cases}, \quad (4.20)$$

onde $\lfloor f(\mathbf{x}) \rfloor$ é a função objetivo da pior solução factível da população atual. Caso não existe solução factível, será usada a média da função objetivo.

- Variante 02:

$$k_j = \lfloor f(\mathbf{x}) \rfloor \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (4.21)$$

onde $\lfloor f(\mathbf{x}) \rfloor$ é o valor da função objetivo da pior solução factível da população atual. Caso não existe solução factível, será utilizada a média da função objetivo.

- Variante 03:

- Para iniciar o cálculo de $\langle v_j(\mathbf{x}) \rangle$ somar as violações de todos os indivíduos da população inactiváveis na restrição j .
- Para obtenção da média, dividir a soma do item anterior apenas entre os indivíduos inactiváveis na restrição j e não entre todos da população.

- Variante 04:

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (4.22)$$

onde $\langle f(\mathbf{x}) \rangle$ passa a ser a soma da função objetivo de todos os indivíduos dividida apenas entre os indivíduos inactiváveis.

- Variante 05:

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & , \quad \text{se } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle \\ \langle f(\mathbf{x}) \rangle & , \quad \text{outro caso} \end{cases} \quad (4.23)$$

onde $\langle f(\mathbf{x}) \rangle$ passa a ser a soma da função objetivo de todos os indivíduos dividida apenas entre os indivíduos inactiváveis.

A Tabela 4.1 apresenta os novos métodos de penalização criados a partir das variantes propostas para o método APM em [72]. Garcia et al. [72] definiram nomes para os novos métodos propostos e na Tabela 4.1 eles são apresentados na primeira coluna. Na Tabela 4.1, tomando como exemplo o método APM_worst_3, o valor de $\bar{f}(\mathbf{x})$ do APM passa a ser definido pela Equação (4.20) e não mais pela Equação (4.15) e também os parâmetros de penalização k_j passam a ser atualizados pela Equação (4.21) e não mais pela Equação (4.16). Já no método APM_Med_6, a única alteração proposta para o APM, consiste em substituir a Equação (4.16) pela Equação (4.22) para atualização dos parâmetros de penalização k_j .

Tabela 4.1: Novas variantes para o APM propostas em [72].

Métodos	Variantes				
	01	02	03	04	05
APM_worst	x				
APM_worst_2		x			
APM_worst_3	x	x			
APM_Med			x		
APM_Med_2			x	x	x
APM_Med_3			x	x	
APM_Med_4			x		x
APM_Med_5					x
APM_Med_6				x	
APM_Med_7				x	x
APM_Med_Worst	x		x		
APM_Med_Worst_2	x	x	x		

Garcia et al. [72] experimentaram suas variantes e também aquelas propostas por Barbosa e Lemonge [70] adotando o Algoritmo Genético como ferramenta de busca e constataram que os métodos APM com parâmetros de penalização monotônicos (APM_Mono), APM_Med_4 e APM_Med_5 proporcionaram maior robustez na resolução dos problemas analisados.

Carvalho et al. [73] experimentaram as variantes propostas por Garcia et al. [72] e Barbosa e Lemonge [70] e ainda a variante APM com parâmetros de penalização monotônicos (APM_Mono), que foi proposta porém não avaliada em [70]. Carvalho et al. [73] obtiveram os melhores resultados aplicando os métodos APM_Mono e APM_Med_3.

4.3 Alterações propostas

O presente trabalho tem como objetivo avaliar o desempenho do algoritmo ABC, descrito no Capítulo 3, na resolução de problemas de otimização com restrições, adotando como medida de tratamento de restrições o método de penalização. Tendo em vista tal objetivo e considerando a inspiração natural do algoritmo, alguns cuidados devem ser tomados ao combinar com o ABC métodos de penalização adaptativos.

No ABC, ao simular o forrageamento de uma colônia de abelhas, com o aumento do número de iterações existe a possibilidade das abelhas campeiras e observadoras

artificiais optarem por explorar fontes muito próximas ou até as mesmas. Somadas a essa característica do ABC, as equações aplicadas por alguns métodos de penalização adaptativos para ajuste dos valores dos parâmetros de penalização, podem gerar resultados indesejados. Isso se deve às operações e/ou informações sobre o conjunto de soluções candidatas envolvidas nestas equações.

Entre as Seções 4.3.1 e 4.3.3 são levantadas algumas questões sobre alguns dos métodos de penalização adaptativos discutidos anteriormente. Também são apresentadas algumas sugestões de alteração para estes métodos objetivando melhoria de desempenho quando combinados ao ABC. Nas Seções 4.3.1 e 4.3.2 são apresentadas sugestões derivadas de observações sobre os métodos de penalização propostos em [64] e [66], apresentados na Seção 4.2.3. Já a Seção 4.3.3 traz a proposta de uma nova variação para o método APM descrito na Seção 4.2.4.

4.3.1 *Coit Mod*

Os métodos de penalização propostos por Coit, Smith e Tate [64] podem gerar valores negativos nos termos $(F_{all} - F_{feas})$ e $(F_{feas} - F_{all})$ das Equações (4.6) e (4.8) respectivamente, como constatado em Duarte, Fonseca e Lemonge [74]. Num problema de minimização, uma penalização negativa aumenta a chance de uma solução infactível ser vista como uma boa solução.

Duarte, Fonseca e Lemonge [74] consideraram também a possibilidade dos termos $(F_{all} - F_{feas})$ e $(F_{feas} - F_{all})$ das Equações 4.6 e 4.8 resultarem em 0. Este resultado faz com que soluções infactíveis não sejam penalizadas, ainda que tenham valores altíssimos de violação. Com base em tais argumentos, Duarte, Fonseca e Lemonge [74] propuseram uma alteração para o método de penalização proposto em [64], descrita nos próximos parágrafos.

O método proposto com a alteração sugerida em Duarte, Fonseca e Lemonge [74] foi intitulado **Coit Mod** e aplicando tal método a função objetivo de uma solução infactível passa a ser dada por

$$F_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) + |F_{all} - F_{feas}| \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i} \right]^{k_i}, & \text{se } F_{all} \neq F_{feas} \\ f(\mathbf{x}) + |F_{all} \sum_{i=1}^{m+q} NFT_i - F_{feas}| \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i} \right]^{k_i}, & \text{se } F_{all} = F_{feas} \end{cases}, \quad (4.24)$$

onde $f(\mathbf{x})$ é a função objetivo não penalizada, as definições de F_{all} , F_{feas} , $d_i(\mathbf{x})$ e NFT_i ficam mantidas como na Equação (4.6). A alteração aparece na substituição de $(F_{all} - F_{feas})$ da Equação (4.6) por $|F_{all} - F_{feas}|$ ou $|F_{all} \sum_{i=1}^{m+q} NFT_i - F_{feas}|$ presentes na Equação (4.24). Esta alteração garante que os parâmetros de penalização se mantenham positivos ao serem ajustados.

4.3.2 ADP Log

Experimentos realizados por Duarte, Fonseca e Lemonge [74] mostraram que o método de penalização adaptativo ADP, proposto em [66], ao ser usado junto do ABC pode gerar o valor 0 para os parâmetros de penalização c_i ($i = 1, 2, \dots, m$) e r_j ($j = 1, 2, \dots, q$). Este resultado ocorre quando $f_{best}^F = f_{best}^{NF}$ na Equação (4.25). Parâmetros de penalização com valor 0 fazem com que soluções infactíveis sejam vistas como factíveis durante o processo de busca. Esta incorreção pode inclusive prejudicar o resultado final obtido, como apresentado em [74].

Propõe-se neste trabalho uma alteração para o método ADP [66], tendo em vista evitar incorreções na penalização e na definição da condição de factibilidade de soluções candidatas. Pelo novo método aqui proposto, intitulado **ADP Log**, os parâmetros de penalização passam a ser ajustados por

$$c_i = \frac{|f_{best}^F - f_{best}^{NF} \times \log((G_i)_{best}^{NF})|}{(G_i)_{best}^{NF}}, \quad (4.25)$$

$$r_j = \frac{|f_{best}^F - f_{best}^{NF} \times \log((H_j)_{best}^{NF})|}{(H_j)_{best}^{NF}}$$

onde os termos envolvidos têm as mesmas definições daqueles presentes na Equação (4.13).

A diferença entre os métodos ADP e ADP Log está na adição dos fatores $\log((G_i)_{best}^{NF})$ e $\log((H_j)_{best}^{NF})$ que aparecem na Equação (4.25). Estes fatores garantem que o valor dos parâmetros de penalização c_i e r_j sejam positivos ainda que $f_{best}^F = f_{best}^{NF}$. Ainda, estes fatores promovem alteração no valor de f_{best}^{NF} de forma a não prejudicar a ideia geral do método ADP, tendo em vista que os valores produzidos pela operação \log são relativamente pequenos. Ao aplicar o método ADP Log, a função objetivo penalizada fica definida como na Equação (4.11), proposta em [66].

Resultados e constatações quanto ao método ADP Log quando combinado ao ABC,

podem ser consultados no Capítulo 5 deste trabalho.

4.3.3 *APM Sum*

O método de penalização adaptativo APM, proposto por Barbosa e Lemonge [67], descrito na Seção 4.2.4 deste trabalho, tem-se mostrado bastante eficiente e robusto em diversas aplicações na literatura.

Sabe-se que pelo APM o ajuste dos parâmetros de penalização relacionados a cada uma das restrições do problema ocorre na forma dada pela Equação (4.16). Tal ajuste faz com que cada parâmetro k_j seja uma parte do valor médio da função objetivo no conjunto de soluções candidatas. A distinção entre os parâmetros k_j ($j = 1, 2, \dots, m$) fica dependendo então da proporção do valor médio de violação de cada uma das restrições j quando comparado com o mesmo valor das demais restrições.

Tomando por base a informação de que parâmetros de penalização devem ter valores suficientemente grandes [12, 58, 6], propõe-se neste trabalho uma nova variação para o método APM. Pelo novo método, intitulado **APM Sum**, o ajuste dos parâmetros de penalização será dado por

$$k_j = |\langle f(\mathbf{x}) \rangle| + \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2}. \quad (4.26)$$

onde $\langle f(\mathbf{x}) \rangle$ continua sendo a média do valor da função objetivo na população corrente, assim como $\langle v_l(\mathbf{x}) \rangle$ ainda é a média da violação à restrição l , como proposto em [67].

A diferença entre os métodos APM e APM Sum é que a proporção da violação média de cada uma das restrições na Equação (4.26) torna-se uma parcela a ser somada a $|\langle f(\mathbf{x}) \rangle|$, não sendo mais uma taxa como na Equação (4.16).

O objetivo com o método APM Sum é manter os parâmetros de penalização com os maiores valores possíveis, se comparados com aqueles obtidos com o método APM. No método APM Sum, a função objetivo penalizada fica definida pela Equação (4.14).

Resultados e constatações quanto ao método APM Sum quando combinado ao ABC, podem ser consultados no Capítulo 5 deste trabalho.

5 Experimentos computacionais

5.1 Introdução

Este capítulo apresenta a metodologia adotada para a realização dos experimentos voltados ao objetivo principal deste trabalho. São listadas as escolhas feitas quanto ao algoritmo ABC e também quanto aos métodos de penalização a ele combinados. Os valores adotados para os parâmetros do ABC e dos métodos de penalização são analisados. São discutidas algumas constatações a respeito da combinação de alguns métodos de penalização com o algoritmo ABC. Neste caso, também são apresentadas as medidas tomadas neste trabalho para que fosse possível avaliar o ABC combinado a métodos de penalização de classes distintas.

Este capítulo traz ainda a descrição de algumas técnicas aplicadas para análise dos resultados obtidos. Por fim, os problemas submetidos ao algoritmo proposto são agrupados em três classes distintas e descritos individualmente.

5.2 Implementação do algoritmo

Com o objetivo de avaliar a eficiência do algoritmo ABC combinado a métodos de penalização, foi desenvolvido um código-fonte na linguagem de programação PythonTM versão 2.7.5. Devido ao grande número de testes computacionais realizados, o algoritmo proposto foi executado em equipamentos possuindo distintas configurações de *hardware* (diferentes processadores, quantidade de memória RAM etc).

Quanto aos parâmetros a serem definidos para a execução do algoritmo ABC, para todos os testes realizados os valores atribuídos foram:

- $SN = 50$;
- $limite = 0,5 \times SN \times D$, onde D corresponde à dimensão do problema [33];
- $C_{max} = 10000$.

Pode-se observar que com esta configuração, o algoritmo realiza um total de 500000 ($SN \times C_{max}$) avaliações da função objetivo, independente do problema tratado. Este

custo computacional máximo foi adotado como estratégia de avaliação de desempenho e também como referência para a comparação do algoritmo quando executado com um número menor de avaliações.

Neste trabalho buscou-se avaliar a aplicação do algoritmo ABC com o mínimo de alterações possíveis na forma como foi proposto por Karaboga em [26]. Esta decisão se deve à ausência de trabalhos na literatura combinando ao ABC grande parte dos métodos de penalização aqui avaliados.

Como apresentado na Seção 3.5 deste trabalho, alguns pesquisadores vêm tentando melhorar a capacidade de intensificação na busca local do ABC. Com base em tais estudos, além do ABC proposto em [26], foi implementado outro algoritmo em que a Equação (3.2) foi substituída pela Equação (3.9) proposta por Gao e Liu em [43]. Este segundo algoritmo é identificado neste trabalho como **ABC G-best**. O algoritmo ABC G-best gerou melhores resultados que o ABC proposto em [26] após testes realizados com ambos. O Algoritmo 6 descreve os passos do ABC G-best.

Algoritmo 6: ABC G-best

Inicialize a população composta por $SN/2$ fontes de alimento;

repita

 Posicione as $SN/2$ abelhas campeiras nas fontes de alimento;

 Realize a busca local de cada abelha campeira com o Algoritmo 3;

 Calcule a probabilidade de escolha das abelhas observadoras para cada fonte;

 Posicione as $SN/2$ abelhas observadoras nas fontes de alimento selecionadas;

 Realize a busca local das $SN/2$ abelhas observadoras com o Algoritmo 3;

 Descarte soluções que não melhoraram por *limite* ciclos;

 Substitua soluções descartadas por outras aleatórias com as abelhas exploradoras;

 Memorize a melhor fonte de alimento encontrada até o momento;

até *Número máximo de ciclos*;

Diversos métodos de penalização foram combinados ao ABC G-best a fim de possibilitar a identificação de qual deles contribuiria melhor com o algoritmo. Todos os métodos de penalização analisados neste trabalho encontram-se descritos nas Seções 4.2 e 4.3. Os métodos escolhidos para avaliação, bem como os valores atribuídos aos seus respectivos parâmetros estão descritos na Tabela 5.1. Com o objetivo de facilitar

a identificação dos métodos, a cada um deles foi associado um nome, apresentado na primeira coluna da Tabela 5.1.

Tabela 5.1: Métodos de penalização analisados com o ABC G-best.

Nome	Tipo do Método de Penalização	Referência
A(k=1E5)	Método aditivo estático com $k = 10^5$ e $p = 2$	Eq. (4.1) e (4.2)
A(k=1E7)	Método aditivo estático com $k = 10^7$ e $p = 2$	Eq. (4.1) e (4.2)
A(k=1E9)	Método aditivo estático com $k = 10^9$ e $p = 2$	Eq. (4.1) e (4.2)
D(C=1.0)	Método aditivo dinâmico com $C = 1, 0$, $\eta = 2$ e $p = 2$	Eq. (4.1), (4.2) e (4.4)
D(C=0.5)	Método aditivo dinâmico com $C = 0, 5$, $\eta = 2$ e $p = 2$	Eq. (4.1), (4.2) e (4.4)
Coit Mod	Método aditivo adaptativo	Seção 4.3.1
Deb	Método aditivo adaptativo	Eq. (4.10)
ADP Log	Método aditivo adaptativo	Seção 4.3.2
APM	Método aditivo adaptativo	Seção 4.2.4
APM Mono	Método aditivo adaptativo	Seção 4.2.4
APM Damp	Método aditivo adaptativo com $\theta = 0, 5$	Seção 4.2.4
APM Worst	Método aditivo adaptativo	Seção 4.2.4
APM Worst 2	Método aditivo adaptativo	Seção 4.2.4
APM Med	Método aditivo adaptativo	Seção 4.2.4
APM Med 3	Método aditivo adaptativo	Seção 4.2.4
APM Med 4	Método aditivo adaptativo	Seção 4.2.4
APM Med 5	Método aditivo adaptativo	Seção 4.2.4
APM Sum	Método aditivo adaptativo	Seção 4.3.3

Entre as diversas variantes do método APM (Seção 4.2.4), aquelas aqui avaliadas foram escolhidas com base nos estudos de Carvalho et al. [73] e Garcia et al. [72]. As variantes de natureza esporádica [70], que atualizam os parâmetros de penalização ao decorrer das gerações, não puderam ser avaliadas em combinação com o ABC G-best. O motivo é uma característica do algoritmo ABC, descrita no parágrafo seguinte.

No ABC, existe a possibilidade de cada abelha campeira ou observadora alterar a fonte memorizada ao efetuar a busca local. Caso ocorram estas substituições, o conjunto de soluções candidatas poderá sofrer diversas alterações em cada ciclo do algoritmo. Esta característica implica em cuidados a serem tomados ao combinar métodos de penalização adaptativos com o ABC.

Quando aplicados métodos de penalização adaptativos, dependendo da frequência de atualização dos parâmetros de penalização, estes poderão se tornar incompatíveis com o conjunto de soluções. Por exemplo, suponha um conjunto de soluções a partir do qual foram definidos os parâmetros de penalização. Se o conjunto sofrer alterações, os parâmetros poderão não penalizar corretamente as novas soluções candidatas.

No caso do ABC, ainda que uma única abelha tenha alterado a fonte memorizada, deve-se considerar um novo conjunto de soluções candidatas. Um exemplo é considerar a aplicação do método APM integrado ao ABC. Ao atualizar um parâmetro de penalização k_j (Equação (4.16)) relacionado a uma determinada restrição $j \in [1, m]$, será determinado o valor $\langle v_j(\mathbf{x}) \rangle$. Caso nenhuma solução do conjunto viole a restrição j , o valor de k_j será 0. Se uma abelha substituir a fonte memorizada e a nova solução candidata violar justamente a restrição j , esta solução infactível não será penalizada nesta restrição. Um outro erro possível de acontecer é se uma nova solução violar justamente apenas a restrição j , tal solução será vista como factível ao ser avaliada.

No ABC a atualização dos parâmetros de penalização não pode ocorrer apenas a cada iteração do algoritmo, como sugere grande parte dos métodos adaptativos. Por este motivo deve-se tomar cuidado ao integrar este tipo de método com o ABC e aqueles de natureza esporádica [70] não podem ser utilizados.

A solução encontrada neste trabalho para a aplicação de métodos de penalização adaptativos foi simular a entrada no conjunto de soluções, de cada nova solução gerada na busca local das abelhas. A partir daí parâmetros “temporários” são calculados e esta nova solução será penalizada (se infactível). Se confirmada a entrada da nova solução no

conjunto, os parâmetros de penalização passam a ter os valores dos “temporários”, caso contrário serão mantidos.

Nota:

Quanto ao número de avaliações da função objetivo realizadas durante a execução do ABC, diferente de muitos algoritmos bio-inspirados, tal valor não é determinado apenas multiplicando o número de soluções presentes no conjunto pelo número de iterações definido. Tal distinção se deve à presença das abelhas exploradoras artificiais. A fase das abelhas exploradoras pode aumentar o número total de avaliações por gerar uma nova solução para o conjunto que deverá ser avaliada. Porém o acréscimo é um número incerto, dado que é possível que tal operação não seja realizada nenhuma vez durante a execução do ABC ou ainda que sejam geradas diversas soluções aleatórias. Recentemente, em [75], foi apresentada uma discussão sobre a comparação do algoritmo ABC com outros algoritmos bio-inspirados, focando na comparação de algoritmos que realizam números de avaliações iguais. Com isso, em [75] foi apresentado pseudo-código que sugere a implementação do ABC com a presença de um contador de avaliações. Devido ao fato de tal discussão ser recente e ter iniciado após andamento do estudo aqui apresentado, neste trabalho fica mantida a totalização do número de avaliações baseado nos números de soluções no conjunto e iterações realizadas.

5.3 Análise de resultados

Esta seção traz a descrição de algumas das técnicas aplicadas neste trabalho para análise dos resultados obtidos.

5.3.1 *Perfis de desempenho*

Perfis de Desempenho (do inglês *Performance Profiles*) é uma ferramenta gráfica proposta por Dolan e More [76] em 2002. A técnica tem como objetivo facilitar a interpretação e a visualização de resultados obtidos em experimentos com grande volume de dados. Os próximos parágrafos descrevem com maiores detalhes a técnica proposta em [76].

Suponha que se tenha um conjunto S com n_s algoritmos e um outro conjunto P com n_p problemas. Suponha ainda que existe o interesse em usar o tempo computacional como

medida de desempenho de cada algoritmo $s \in S$. Para cada problema $p \in P$ e algoritmo s , tem-se

$$t_{p,s} = \text{tempo computacional gasto para resolver } p \text{ com } s. \quad (5.1)$$

Compara-se o desempenho de s aplicado a p com o melhor desempenho entre os algoritmos de S quando aplicados ao mesmo problema p . Surge então a taxa de desempenho, dada por

$$r_{p,s} = \frac{t_{p,s}}{\text{menor } \{t_{p,s} : s \in S\}}. \quad (5.2)$$

Assume-se ainda a existência de um parâmetro $r_M \geq r_{p,s}$ para todo p , de forma que $r_{p,s} = r_M$ se e somente se o algoritmo s não resolve o problema p .

O objetivo com a técnica é obter uma avaliação global do desempenho de s . Define-se então

$$\rho_s(\tau) = \frac{1}{n_p} |p \in P : r_{p,s} \leq \tau|, \quad (5.3)$$

onde $\rho_s(\tau)$ é a probabilidade do algoritmo $s \in S$ estar dentro de uma faixa $\tau \in \mathbb{R}$ de taxa de desempenho. Em outras palavras, $\rho_s(\tau)$ corresponde à fração de problemas em P que o algoritmo s consegue resolver até uma taxa de desempenho limite τ .

A função $\rho_s : \mathbb{R} \rightarrow [0, 1]$ é não-decrescente, constante em partes e contínua à direita de cada ponto de interrupção. A Figura 5.1 exemplifica o gráfico de $\rho_s(\tau)$, onde o algoritmo C tem o menor valor de τ tal que $\rho(\tau) = 1$ e portanto é considerado o mais robusto.

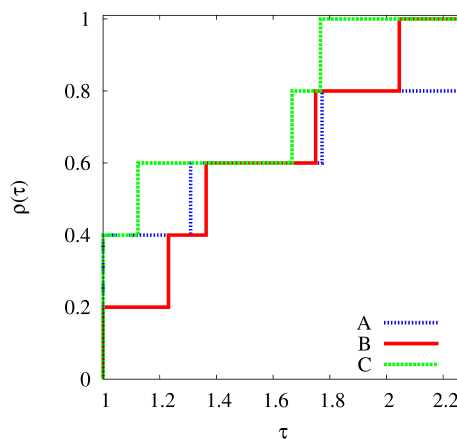


Figura 5.1: Exemplo de gráfico da técnica Perfis de Desempenho.

Uma importante característica dos Perfis de Desempenho, destacada por Dolan e More [76], é que a técnica não é sensível ao número de problemas em P , podendo ser pequeno. Outras propriedades que devem ser destacadas são:

- O valor $\rho_s(1)$ é a fração de problemas em que s apresenta o melhor desempenho entre todos os algoritmos de S ;
- Quando $\tau \rightarrow r_M^-, \rho_s(\tau)$ representa a fração de problemas que o algoritmo s consegue resolver.

Aplicações da técnica Perfis de Desempenho podem ser observadas em [77] e [78]. Barreto, Bernardino e Barbosa [79] propuseram uma extensão para a técnica, intitulada Perfis de Desempenho Probabilístico, visando sua aplicação com a presença de algoritmos estocásticos em S .

Em [78], Barbosa, Bernardino e Barreto destacaram que a área sob a curva ρ_s ($AUC_s = \int \rho_s(\tau) d\tau$) é um indicador do desempenho global do algoritmo s no conjunto de problemas P . Quanto maior AUC_s , maior a eficiência de s .

5.3.2 Escalonamento Multidimensional

Entre os métodos de penalização podem ocorrer similaridades ou dissimilaridades de comportamento quando combinados ao ABC G-best. A fim de possibilitar a avaliação da distinção entre os métodos aqui analisados, aplica-se a técnica Escalonamento Multidimensional (MDS – do inglês *Multidimensional Scaling*).

O MDS é uma técnica que mede a similaridade ou dissimilaridade entre objetos de um conjunto, produzindo uma distribuição visual em um espaço multidimensional. Ao fim do processo, graficamente os objetos similares estarão próximos e aqueles dissimilares estarão distantes entre si. Medidas sobre os objetos são utilizadas como entradas no MDS, que em seguida são transformadas em distâncias geométricas entre tais objetos [80, 81].

Neste trabalho, por meio da biblioteca `sklearn.manifold` [82] para a linguagem de programação PythonTM, aplica-se a técnica MDS usando o algoritmo SMACOF (do inglês *Scaling by Majorizing a Convex Function*). Pelo SMACOF, um conjunto de n pontos m -dimensionais é passado como entrada e o algoritmo retorna um outro conjunto de n pontos r -dimensionais, sendo $r \ll m$. O algoritmo SMACOF e suas variações encontra-se descrito em [83].

5.4 Problemas analisados

Para a avaliação do algoritmo proposto neste trabalho, diversos problemas de otimização encontrados na literatura foram considerados. Foram analisados problemas de Engenharia Mecânica, de Otimização Estrutural e um conjunto de 24 funções, todos descritos nas próximas seções.

Para cada um dos problemas analisados foram realizadas um total de 25 execuções independentes da combinação ABC G-best + Método de Penalização (cada um entre os avaliados, listados na Tabela 5.1). As restrições de igualdade foram convertidas em restrições de desigualdade na forma dada pela Equação (2.2), adotando a tolerância $\epsilon = 10^{-6}$.

Para a análise dos resultados foram considerados, dependendo do problema, até três valores máximos para o número de avaliações da função objetivo, 5000, 50000 e 500000. O objetivo com esta estratégia foi avaliar se haveria melhora dos resultados com o aumento do orçamento computacional.

Para cada problema são apresentadas tabelas com informações estatísticas relacionadas aos resultados obtidos com os métodos de penalização avaliados junto do ABC G-best. Nestas tabelas, na coluna **Melhor**, que lista os melhores resultados obtidos, o menor (minimização) deles aparece destacado em negrito. As colunas como **Mediana**, **Média**, **DP** (Desvio Padrão), são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna **TF** (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas **ME** (Número mínimo de execuções) e **MA** (Número mínimo de avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna **Melhor**.

Nos Perfis de Desempenho os valores atribuídos a $t_{p,s}$ foram a média e o valor mínimo entre as funções objetivo das soluções encontradas nas 25 execuções. Já para a técnica MDS, as cinco métricas (dimensões) escolhidas para cada um dos métodos de penalização foram: o menor valor de função objetivo, a quantidade de soluções factíveis, a média, o desvio padrão e a mediana, todos definidos entre os resultados das 25 execuções e normalizados entre 0 e 1. Ainda quanto ao MDS, as similaridades tiveram redução de 5 para 2 dimensões.

5.5 Experimento 01 – Problemas clássicos da Engenharia Mecânica

Este experimento trata-se da resolução de alguns problemas de otimização da Engenharia Mecânica largamente utilizados na literatura. Seis problemas desta classe foram impostos ao algoritmo proposto neste trabalho. Cada um dos problemas é descrito entre as Seções 5.5.1 e 5.5.6, onde também encontram-se discussões sobre os resultados obtidos.

5.5.1 MTC – Mola sob Tração/Compressão

Este problema de otimização consiste em minimizar o volume V de uma mola [84] ilustrada na Figura 5.2 sujeita a tração ou compressão .

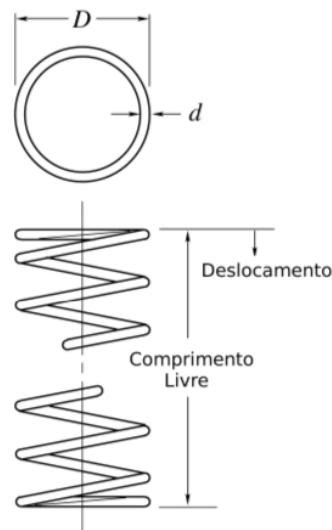


Figura 5.2: MTC – Mola sob Tração/Compressão.

As variáveis de projeto envolvidas no problema, bem como seus valores aceitos são: o diâmetro do fio ($d = x_1 \in [0,05; 2,0]$), o diâmetro de cada volta ($D = x_2 \in [0,25; 1,3]$) e o número de espirais ativos na mola ($N = x_3 \in [2,0; 15,0]$).

O problema é modelado por

$$\text{MTC} = \left\{ \begin{array}{l} \text{Minimize} \quad V(\mathbf{x}) = (x_3 + 2)x_2x_1^2 \\ \text{Sujeito a} \quad g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\ \quad \quad \quad g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\ \quad \quad \quad g_3(\mathbf{x}) = 1 - \frac{140,45x_1}{x_2^2x_3} \leq 0 \\ \quad \quad \quad g_4(\mathbf{x}) = \frac{x_2 + x_1}{1,5} - 1 \leq 0 \\ \quad \quad \quad 0,05 \leq x_1 \leq 2,0 \\ \quad \quad \quad 0,25 \leq x_2 \leq 1,3 \\ \quad \quad \quad 2,0 \leq x_3 \leq 15,0 \end{array} \right. . \quad (5.4)$$

A Tabela 5.2 mostra que sob o custo computacional máximo de 50000 avaliações da função objetivo, com o método ADP Log foi obtido o maior valor apresentado na coluna Melhor. Além disso, o método ADP Log produziu valores para as métricas Mediana e Média consideravelmente maiores se comparados com tais valores dos demais métodos. Em contrapartida, o menor valor para a métrica Mediana foi obtido com os métodos APM e APM Damp. Para a métrica Média, o menor valor foi obtido com o método APM Sum, proposto neste trabalho. Destaca-se que com todos os métodos analisados foram encontradas soluções factíveis nas 25 execuções.

Tabela 5.2: Resultados obtidos para o problema Mola sob Tração/Compressão (MTC) com cada método de penalização combinado ao ABC G-best, adotando 50000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
A(k=1E5)	0.012667	0.012724	0.012672	0.012677	1.430320e-05	1.000000	25	25	14050	7
A(k=1E7)	0.012666	0.012728	0.012676	0.012691	2.390380e-05	1.000000	25	25	40000	25
A(k=1E9)	0.012667	0.012733	0.012672	0.012680	1.996711e-05	1.000000	25	25	25450	14
ADP Log	0.013162	0.085782	0.033735	0.033168	0.016895	1.000000	25	25	12650	25
APM	0.012665	0.012720	0.012666	0.012677	2.122546e-05	1.000000	25	25	40100	17
APM Damp	0.012665	0.012838	0.012666	0.012687	4.230402e-05	1.000000	25	25	37800	23
APM Med	0.012665	0.012723	0.012671	0.012684	2.280150e-05	1.000000	25	25	37600	19
APM Med 3	0.012665	0.012722	0.012668	0.012684	2.456012e-05	1.000000	25	25	34750	14
APM Med 4	0.012666	0.013101	0.012667	0.012701	8.536842e-05	1.000000	25	25	37300	23
APM Med 5	0.012665	0.012951	0.012678	0.012702	5.731170e-05	1.000000	25	25	33500	11
APM Mono	0.012669	0.012865	0.012720	0.012722	4.409366e-05	1.000000	25	25	40350	5
APM Sum	0.012666	0.012722	0.012668	0.012674	1.456034e-05	1.000000	25	25	49400	1
APM Worst	0.012665	0.013193	0.012667	0.012708	1.021292e-04	1.000000	25	25	37700	17
APM Worst 2	0.012665	0.012720	0.012668	0.012678	2.080548e-05	1.000000	25	25	35100	6
Coit Mod	0.012674	0.012772	0.012722	0.012725	1.491085e-05	1.000000	25	25	28650	13
D(C=0.5)	0.012666	0.012740	0.012723	0.012707	2.681649e-05	1.000000	25	25	47600	11
D(C=1.0)	0.012666	0.012741	0.012671	0.012693	2.908418e-05	1.000000	25	25	40900	15
Deb	0.012666	0.012732	0.012670	0.012690	2.660468e-05	1.000000	25	25	37100	21

Tabela 5.3: Resultados obtidos para o problema Mola sob Tração/Compressão (MTC) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
A(k=1E5)	0.012666	0.012669	0.012667	0.012667	9.546501e-07	1.000000	25	25	229600	25
A(k=1E7)	0.012666	0.012668	0.012667	0.012667	6.124476e-07	1.000000	25	25	461550	23
A(k=1E9)	0.012666	0.012668	0.012667	0.012667	6.266846e-07	1.000000	25	25	67750	14
ADP Log	0.012852	0.027393	0.013364	0.016511	0.004735	1.000000	25	25	389800	18
APM	0.012665	0.012719	0.012665	0.012676	2.143944e-05	1.000000	25	25	67700	15
APM Damp	0.012665	0.012719	0.012665	0.012678	2.289033e-05	1.000000	25	25	306900	15
APM Med	0.012665	0.012719	0.012667	0.012682	2.347897e-05	1.000000	25	25	37600	19
APM Med 3	0.012665	0.012720	0.012666	0.012683	2.504203e-05	1.000000	25	25	79050	3
APM Med 4	0.012665	0.012719	0.012666	0.012676	2.144427e-05	1.000000	25	25	333400	16
APM Med 5	0.012665	0.012719	0.012666	0.012681	2.401915e-05	1.000000	25	25	317050	6
APM Mono	0.012666	0.012686	0.012668	0.012669	4.099313e-06	1.000000	25	25	399150	2
APM Sum	0.012665	0.012719	0.012666	0.012668	1.046896e-05	1.000000	25	25	262900	14
APM Worst	0.012665	0.012719	0.012665	0.012668	1.049442e-05	1.000000	25	25	149750	15
APM Worst 2	0.012665	0.012719	0.012665	0.012674	1.967219e-05	1.000000	25	25	467450	18
Coit Mod	0.012668	0.012722	0.012719	0.012715	1.395260e-05	1.000000	25	25	115600	13
D(C=0.5)	0.012665	0.012669	0.012667	0.012667	1.091867e-06	1.000000	25	25	282650	14
D(C=1.0)	0.012666	0.012669	0.012667	0.012667	8.553330e-07	1.000000	25	25	403050	13
Deb	0.012665	0.012719	0.012666	0.012673	1.719377e-05	1.000000	25	25	482600	3

Na Tabela 5.3 observa-se que com o aumento do número máximo de avaliações da função objetivo para 500000, o método ADP Log fica mantido como aquele que apresenta o maior valor na coluna Melhor da tabela, além deste método apresentar os maiores valores para as métricas Mediana e Média. Os métodos APM, APM Damp, APM Worst e APM Worst 2 apresentam os melhores resultados encontrados para o problema também para a métrica Mediana.

A Tabela 5.4 traz comparação do resultado obtido neste trabalho para o problema MTC com outros encontrados na literatura. Observa-se que com o algoritmo aqui proposto, foi obtido resultado igual ao de outros trabalhos entre os comparados, que corresponde ao melhor resultado encontrado para o problema. Como pode ser observado na Tabela 5.4, o problema MTC possui múltiplas soluções ótimas.

Tabela 5.4: Comparação do resultado obtido para o problema Mola sob Tração/Compressão (MTC) com outros trabalhos da literatura.

Trabalho	V	d	D	N	Avaliações
Bernardino et al. [85]	0,01267	0,05143	0,35053	11,6612	N/A
Coello e Becerra [86]	0,01272	0,05	0,3174	14,0318	50000
He e Wang [87]	0,01267	0,05173	0,35764	11,2445	N/A
Runarsson e Yao [88]	0,01268	0,05164	0,35549	11,3758	N/A
Gandomi et al. [89]	0,01267	0,05169	0,35673	11,2885	20000
Akay e karaboga [29]	0,01267	0,05216	0,368159	10,648442	30000
Brajevic e Tuba [90]	0,01267	0,051691	0,356769	11,285988	15000
Brajevic, Tuba e Subotic [91]	0,01267	0,051871	0,361108	11,03686	240000
Este estudo	0,01267	0,051662	0,356072	11,327060	37600

5.5.2 *RV – Redutor de Velocidade*

Este problema corresponde a minimizar o peso W de um redutor de velocidade ilustrado na Figura 5.3 [7, 84, 4, 92, 93].

As sete variáveis envolvidas neste projeto, bem como seus valores aceitos 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 ($z = x_3 \in [17; 28]$), os tamanhos das hastes 1 ($l_1 = x_4 \in [7, 3; 8, 3]$) e 2 ($l_2 = x_5 \in [7, 3; 8, 3]$) entre os rolamentos e os diâmetros dos mastros ($d_1 = x_6 \in [2, 9; 3, 9]$) e ($d_2 = x_7 \in [5, 0; 5, 5]$). A variável x_3 é inteira e o restante são contínuas. Entre as

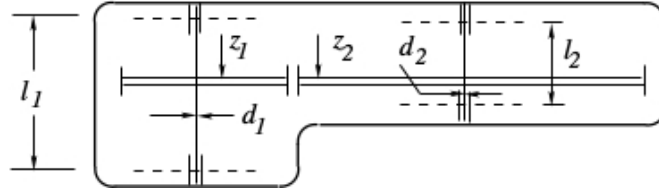


Figura 5.3: RV – Redutor de Velocidade.

restrições estão: limitações de tensão e deflexão da superfície da engrenagem dos dentes, deslocamentos transversais das hastes 1 e 2 originados da força transmitida e limites de tensões nas hastes 1 e 2.

O problema é dado por

$$\text{RV} = \left\{ \begin{array}{l} \text{Minimize} \quad W(\mathbf{x}) = 0,7854x_1x_2^2(3,3333x_3^2 + 14,9334x_3 - 43,0934) \\ \quad \quad \quad -1,508x_1(x_6^2 + x_7^2) + 7,477(x_6^3 + x_7^3) \\ \quad \quad \quad +0,7854(x_4x_6^2 + x_5x_7^2) \\ \text{Sujeito a} \quad g_1(\mathbf{x}) = 27x_1^{-1}x_2^{-2}x_3^{-1} - 1 \leq 0 \\ \quad \quad \quad g_2(\mathbf{x}) = 397,5x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0 \\ \quad \quad \quad g_3(\mathbf{x}) = 1,93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} - 1 \leq 0 \\ \quad \quad \quad g_4(\mathbf{x}) = 1,93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} - 1 \leq 0 \\ \quad \quad \quad g_5(\mathbf{x}) = \left[\left(\frac{745x_4}{x_2x_3} \right)^2 + (16,9)10^6 \right]^{0,5} / 0,1x_6^3 - 1100 \leq 0 \\ \quad \quad \quad g_6(\mathbf{x}) = \left[\left(\frac{745x_5}{x_2x_3} \right)^2 + (157,5)10^6 \right]^{0,5} / 0,1x_7^3 - 850 \leq 0 \\ \quad \quad \quad g_7(\mathbf{x}) = x_2x_3 - 40 \leq 0 \\ \quad \quad \quad g_8(\mathbf{x}) = 5 - \frac{x_1}{x_2} \leq 0 \\ \quad \quad \quad g_9(\mathbf{x}) = \frac{x_1}{x_2} - 12 \leq 0 \\ \quad \quad \quad g_{10}(\mathbf{x}) = (1,5x_6 + 1,9)x_4^{-1} - 1 \leq 0 \\ \quad \quad \quad g_{11}(\mathbf{x}) = (1,1x_7 + 1,9)x_5^{-1} - 1 \leq 0 \\ \quad \quad \quad 2,6 \leq x_1 \leq 3,6 \\ \quad \quad \quad 0,7 \leq x_2 \leq 0,8 \\ \quad \quad \quad 17 \leq x_3 \leq 28 \\ \quad \quad \quad 7,3 \leq x_4, x_5 \leq 8,3 \\ \quad \quad \quad 2,9 \leq x_6 \leq 3,9 \\ \quad \quad \quad 5,0 \leq x_7 \leq 5,5 \end{array} \right. \quad (5.5)$$

Tabela 5.5: Resultados obtidos para o problema Redutor de Velocidade (RV) com cada método de penalização combinado ao ABC G-best, adotando 50000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	2994.341	2994.341	2994.341	2994.341	0.000e+00	1.000	25	25	8650	6
APM Damp	2994.341	2994.341	2994.341	2994.341	3.638e-13	1.000	25	25	8300	2
APM Med 4	2994.341	2994.341	2994.341	2994.341	1.162e-07	0.960	25	24	17150	1
ADP Log	2994.341	2994.341	2994.341	2994.341	0.000e+00	1.000	25	25	7850	15
Coit Mod	2994.341	2994.341	2994.341	2994.341	1.464e-07	1.000	25	25	13400	3
APM Med	2994.341	2998.266	2994.341	2994.498	0.769	1.000	25	25	9950	3
APM Sum	2994.341	2994.341	2994.341	2994.341	0.000e+00	1.000	25	25	7650	5
APM	2994.341	2994.341	2994.341	2994.341	6.698e-08	1.000	25	25	8600	23
APM Worst 2	2994.341	2994.341	2994.341	2994.341	1.909e-08	1.000	25	25	9200	25
APM Med 5	2994.341	2994.341	2994.341	2994.341	2.728e-13	1.000	25	25	19500	13
A(k=1E7)	2994.343	2994.412	2994.357	2994.361	0.016	1.000	25	25	2600	25
APM Worst	2994.341	2994.342	2994.341	2994.341	1.362e-04	1.000	25	25	8800	17
D(C=1.0)	2994.573	3037.805	2996.423	2999.234	8.368	1.000	25	25	31450	4
A(k=1E5)	2994.441	2999.370	2995.156	2995.769	1.324	1.000	25	25	1850	23
APM Med 3	2994.341	3087.197	2994.341	2998.056	18.196	1.000	25	25	9250	6
Deb	3325.060	5267.469	4451.204	4347.911	796.342	0.120	25	3	200	20
A(k=1E9)	2994.341	2994.343	2994.342	2994.342	4.212e-04	1.000	25	25	3950	1
D(C=0.5)	2995.357	3043.684	3000.668	3004.008	10.477	1.000	25	25	23550	15

A Tabela 5.5 mostra que com 70% dos métodos de penalização analisados no trabalho, foi possível obter o melhor resultado encontrado para o problema RV. A Tabela 5.5 apresenta os resultados obtidos sob o custo computacional de 50000 avaliações da função objetivo. Porém, muitos dos métodos precisaram de um número menor de avaliações para gerarem seus resultados, como pode ser observado na coluna MA. Com o método Deb foi obtido o maior resultado entre aqueles apresentados na coluna Melhor. O método também apresenta o menor número de resultados factíveis entre as 25 execuções, como pode ser observado entre as colunas Exec. e Fac..

A Tabela 5.6 compara o melhor resultado obtido neste trabalho para o problema RV com outros encontrados na literatura. A tabela mostra que o algoritmo proposto neste trabalho encontrou um resultado melhor que aqueles obtidos com os algoritmos dos trabalhos comparados.

Tabela 5.6: Comparação do resultado obtido para o problema Redutor de Velocidade (RV) com outros trabalhos da literatura.

Trabalho	W	b	m	z	l_1	l_2	d_1	d_2	Avaliações
Mezura, Coello e Becerra [84]	3025,005	3,506	0,701	17	7,460	7,962	3,363	5,309	36000
Bernardino, Barbosa e Lemonge [94]	2996,349	3,500	0,700	17	7,300	7,800	3,350	5,287	36000
Bernardino, Barbosa e Lemonge [94]	2996,348	3,500	0,700	17	7,300	7,800	3,350	5,287	36000
Bernardino et al. [85]	2996,348	3,500	0,700	17	7,300	7,800	3,350	5,287	36000
Runarsson e Yao [88]	2996,348	3,500	0,700	17	7,300	7,800	3,350	5,287	N/A
Aragon et al. [95]	2996,348	3,500	0,700	17	7,300	7,800	3,350	5,287	36000
Akay e Karaboga [29]	2997,058	3,500	0,700	17	7,300	7,800	3,350	5,288	30000
Brajevic e Tuba [90]	2994,471	3,500	0,700	17	7,300	7,715	3,350	5,287	15000
Brajevic, Tuba e Subotic [91]	2996,348	3,500	0,700	17	7,300	7,800	3,350	5,287	240000
Este estudo	2994,341	3,500	0,700	17	7,300	7,715	3,350	5,287	3950

5.5.3 T3B – Treliça de Três Barras

Este problema consiste do projeto de uma treliça plana composta por três barras [93], ilustrada na Figura 5.4.

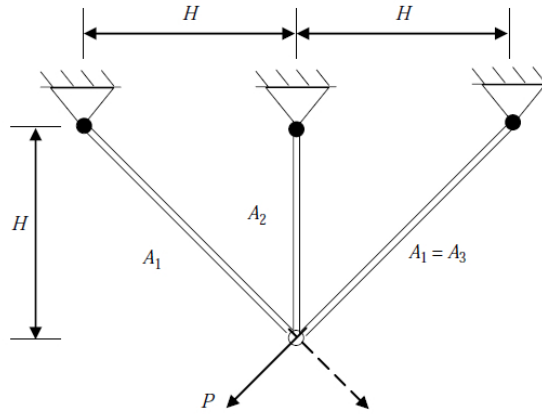


Figura 5.4: T3B – Treliça de Três Barras.

O objetivo do problema é reduzir o peso da estrutura buscando as dimensões que minimizam as áreas das seções transversais das barras. Deve-se considerar que a treliça estará sujeita a carregamento como mostra a Figura 5.4. As restrições envolvidas no problema referem-se aos limites de tensões em cada barra da estrutura.

A formulação matemática do problema é dada por

$$\mathbf{T3B} = \begin{cases} \text{Minimize} & f(\mathbf{x}) = (2\sqrt{2}x_1 + x_2) \times l \\ \text{Sujeito a} & g_1(\mathbf{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \\ & g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0, \\ & g_3(\mathbf{x}) = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0 \\ & 0 \leq x_1, x_2 \leq 1 \end{cases}, \quad (5.6)$$

onde $l = 100\text{cm}$, $P = 2\text{kN/cm}^2$ e $\sigma = 2\text{kN/cm}^2$.

A Tabela 5.7 mostra que com 5000 avaliações da função objetivo, entre os melhores resultados obtidos para o problema T3B com cada método, com o ADP Log foi gerado o maior deles. Os demais métodos apresentaram resultados consideravelmente próximos, sendo que o melhor resultado encontrado para o problema foi obtido com os métodos APM Sum, A(k=1E7), Deb e A(k=1E9), sendo que os menores valores para as métricas Mediana e Média foram produzidos pelo método Deb.

Tabela 5.7: Resultados obtidos para o problema Treliza de Três Barras (T3B) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	263.897	264.019	263.911	263.918	0.024	1.000	25	25	3100	9
APM Damp	263.897	264.015	263.914	263.922	0.027	1.000	25	25	2050	21
APM Med 4	263.897	265.744	263.936	264.086	0.452	1.000	25	25	2500	3
ADP Log	264.919	288.172	276.508	275.181	7.089	1.000	25	25	50	18
Coit Mod	263.898	263.948	263.908	263.911	0.011	1.000	25	25	4600	16
APM Med	263.909	278.149	264.987	266.641	3.597	1.000	25	25	1750	15
APM Sum	263.896	264.029	263.906	263.918	0.034	1.000	25	25	3550	14
APM	263.899	264.053	263.931	263.946	0.046	1.000	25	25	3400	13
APM Worst 2	263.897	264.016	263.925	263.934	0.032	1.000	25	25	4850	14
APM Med 5	263.902	263.982	263.921	263.925	0.023	1.000	25	25	5000	9
A(k=1E7)	263.896	263.953	263.909	263.913	0.017	1.000	25	25	4900	1
APM Worst	263.897	263.992	263.910	263.918	0.022	1.000	25	25	4650	23
D(C=1.0)	263.943	265.658	264.220	264.295	0.400	1.000	25	25	4650	11
A(k=1E5)	263.906	263.989	263.926	263.935	0.025	1.000	25	25	3050	22
APM Med 3	263.947	278.244	266.296	267.214	3.187	1.000	25	25	1400	15
Deb	263.896	263.920	263.897	263.899	0.005	1.000	25	25	4500	13
A(k=1E9)	263.896	263.943	263.908	263.912	0.013	1.000	25	25	4650	16
D(C=0.5)	263.911	268.393	264.797	265.083	1.059	1.000	25	25	4850	23

Tabela 5.8: Resultados obtidos para o problema Treliça de Três Barras (T3B) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	263.896	263.896	263.896	263.896	1.611e-04	1.000	25	25	401850	25
APM Damp	263.896	263.896	263.896	263.896	5.393e-05	1.000	25	25	55500	23
APM Med 4	263.896	263.897	263.896	263.896	1.821e-04	1.000	25	25	440400	13
ADP Log	263.969	265.649	264.499	264.564	0.486	1.000	25	25	280200	25
Coit Mod	263.896	263.896	263.896	263.896	9.094e-05	1.000	25	25	406600	20
APM Med	263.897	263.937	263.907	263.909	0.011	1.000	25	25	415800	18
APM Sum	263.896	263.896	263.896	263.896	2.899e-05	1.000	25	25	36350	15
APM	263.896	263.896	263.896	263.896	1.652e-04	1.000	25	25	293750	4
APM Worst 2	263.896	263.896	263.896	263.896	1.255e-04	1.000	25	25	375750	20
APM Med 5	263.896	263.896	263.896	263.896	7.041e-05	1.000	25	25	160000	5
A(k=1E7)	263.896	263.896	263.896	263.896	1.195e-04	1.000	25	25	331300	2
APM Worst	263.896	263.896	263.896	263.896	3.978e-05	1.000	25	25	217700	17
D(C=1.0)	263.896	263.897	263.896	263.896	1.561e-04	1.000	25	25	481500	11
A(k=1E5)	263.896	263.898	263.896	263.896	4.008e-04	1.000	25	25	291200	6
APM Med 3	263.896	263.929	263.903	263.906	0.010	1.000	25	25	363200	15
Deb	263.896	263.896	263.896	263.896	1.125e-04	1.000	25	25	408950	21
A(k=1E9)	263.896	263.897	263.896	263.896	1.523e-04	1.000	25	25	383400	1
D(C=0.5)	263.896	263.896	263.896	263.896	1.462e-04	1.000	25	25	121600	17

Na Tabela 5.8, nota-se que com 90% dos métodos de penalização foi obtido o mesmo valor de função objetivo em todas as execuções. Apenas com os métodos ADP Log e APM Med não foi obtido o melhor resultado encontrado neste trabalho para o problema T3B. A coluna MA da Tabela 5.8 mostra que alguns dos métodos de penalização convergiram com um número menor que 500000 avaliações.

A Tabela 5.9 apresenta resultados encontrados na literatura para o problema T3B comparados àquele aqui obtido. Observa-se que o menor resultado encontrado entre os trabalhos comparados também foi obtido com o algoritmo proposto.

Tabela 5.9: Comparação do resultado obtido para o problema Treliça de Três Barras (T3B) com outros trabalhos da literatura.

Trabalho	$f(\mathbf{x})$	x_1	x_2	Avaliações
Yang e Gandomi [93]	263,896	0,789	0,408	15000
Park, Chang e Lee [96]	263,897	0,789	0,408	N/A
Ray e Saini [97]	264,300	0,795	0,395	N/A
Brajevic e Tuba [90]	263,896	0,789	0,401	15000
Este estudo	263,896	0,789	0,401	36350

5.5.4 VP – Vaso de Pressão

Este problema trata-se da minimização do peso W de um vaso de pressão cilíndrico com duas tampas esféricas [7, 84, 4, 92], ilustrado na Figura 5.5.

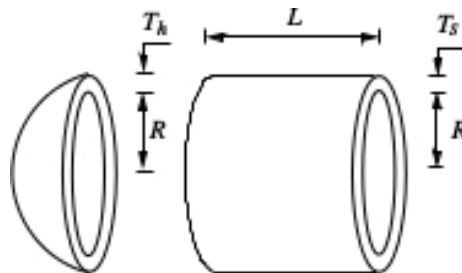


Figura 5.5: VP – Vaso de Pressão.

O problema envolve quatro variáveis de projeto medidas em polegadas: a espessura do vaso de pressão ($T_s = x_1$), a espessura da tampa ($T_h = x_2$), o raio interno do vaso ($R = x_3$) e a altura do componente cilíndrico ($L = x_4$). As variáveis T_s e T_h são discretas,

tendo como valores aceitos $0,0625 \leq T_s, T_h \leq 5,0$ em passos constantes de $0,0625$. Já as variáveis R e L são contínuas, tendo os valores aceitos no intervalo $10,0 \leq R, L \leq 200,0$. Trata-se então de um problema não-linear e com variáveis mistas (discretas e contínuas).

O problema é modelado por

$$\mathbf{VP} = \left\{ \begin{array}{l} \text{Minimize} \quad W(\mathbf{x}) = 0,6224x_1x_2x_3 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \\ \text{Sujeito a} \quad g_1(\mathbf{x}) = 0,0193x_3 - x_1 \leq 0 \\ \quad \quad \quad g_2(\mathbf{x}) = 0,00954x_3 - x_2 \leq 0 \\ \quad \quad \quad g_3(\mathbf{x}) = 1296000 - \pi x_3^2x_4 - 4/3\pi x_3^3 \leq 0 \\ \quad \quad \quad g_4(\mathbf{x}) = x_4 - 240 \leq 0 \\ \quad \quad \quad 0,0625 \leq x_1, x_2 \leq 5 \text{ (em passos constantes de } 0,0625) \\ \quad \quad \quad 10 \leq x_3, x_4 \leq 200 \end{array} \right. . \quad (5.7)$$

A Tabela 5.10 mostra que apenas com o método APM Sum foi possível obter o melhor resultado encontrado para o problema VP sob o custo computacional de 5000 avaliações da função objetivo. Na coluna Melhor da tabela nota-se que com o método Coit Mod foi obtido resultado próximo do melhor encontrado, já que os demais métodos apresentam diferença considerável nos resultados. O método Coit Mod também apresenta os menores valores para as métricas Mediana e Média. Os métodos ADP Log, D(C=1.0) e D(C=0.5) foram aqueles com os quais foram obtidos os maiores resultados.

Tabela 5.10: Resultados obtidos para o problema Vaso de Pressão (VP) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	6535.276	7562.775	6887.453	6842.876	226.342	1.000	25	25	2850	6
APM Damp	6419.887	7556.749	6649.929	6766.841	308.879	1.000	25	25	3300	10
APM Med 4	6398.716	7100.257	6578.427	6690.528	245.628	1.000	25	25	1000	5
ADP Log	9544.481	6.657e+04	2.697e+04	2.824e+04	1.321e+04	1.000	25	25	150	21
Coit Mod	6390.801	7143.032	6475.969	6556.826	188.532	1.000	25	25	4350	23
APM Med	6425.155	8064.563	6841.784	6864.346	365.103	1.000	25	25	2450	6
APM Sum	6390.286	7511.166	6622.392	6748.137	289.434	1.000	25	25	1900	4
APM	6402.717	7083.364	6596.873	6668.476	210.888	1.000	25	25	4950	10
APM Worst 2	6406.141	8303.548	6943.508	6905.181	372.546	1.000	25	25	3200	20
APM Med 5	6400.090	7692.142	6608.128	6745.716	291.541	1.000	25	25	2750	8
A(k=1E7)	6420.472	7609.133	6713.035	6751.035	289.824	1.000	25	25	4050	3
APM Worst	6397.573	7080.395	6604.658	6668.396	215.802	1.000	25	25	2550	23
D(C=1.0)	7929.329	3.216e+04	1.823e+04	1.768e+04	5096.067	1.000	25	25	500	2
A(k=1E5)	6526.156	8647.131	7126.941	7302.894	600.437	1.000	25	25	1400	14
APM Med 3	6399.892	8319.868	6619.206	6834.812	456.872	1.000	25	25	3950	23
Deb	6401.397	7387.701	6883.163	6811.224	230.135	1.000	25	25	1950	13
A(k=1E9)	6399.843	7053.301	6614.022	6643.452	175.091	1.000	25	25	500	13
D(C=0.5)	8413.795	3.413e+04	1.525e+04	1.666e+04	6524.478	1.000	25	25	700	2

Tabela 5.11: Resultados obtidos para o problema Vaso de Pressão (VP) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	6390.204	6424.816	6398.168	6399.074	8.084	1.000	25	25	460650	11
APM Damp	6388.399	6398.473	6392.309	6392.288	2.202	1.000	25	25	55700	1
APM Med 4	6388.025	6395.196	6389.860	6390.337	1.702	1.000	25	25	154550	2
ADP Log	7337.096	1.200e+04	1.047e+04	1.008e+04	1191.600	1.000	25	25	493100	21
Coit Mod	6388.243	6391.691	6389.109	6389.416	0.991	1.000	25	25	42450	7
APM Med	6425.155	7527.909	6512.595	6695.176	272.702	1.000	25	25	2450	6
APM Sum	6389.483	6412.349	6396.456	6397.813	5.606	1.000	25	25	234750	4
APM	6389.543	6412.238	6395.064	6396.051	5.175	1.000	25	25	126900	5
APM Worst 2	6389.278	6411.647	6396.114	6396.479	5.265	1.000	25	25	121850	13
APM Med 5	6388.992	6397.593	6391.883	6391.955	2.115	1.000	25	25	395350	1
A(k=1E7)	6388.237	6404.080	6389.795	6391.135	3.349	1.000	25	25	76550	19
APM Worst	6389.635	6413.589	6396.756	6398.264	6.512	1.000	25	25	91300	11
D(C=1.0)	6388.453	6403.925	6391.541	6392.375	3.665	1.000	25	25	165500	1
A(k=1E5)	6418.721	6513.867	6470.760	6468.259	19.445	1.000	25	25	409250	25
APM Med 3	6399.892	7570.717	6553.769	6656.516	285.073	1.000	25	25	3950	23
Deb	6389.314	6534.577	6432.010	6441.389	40.761	1.000	25	25	499000	17
A(k=1E9)	6388.082	6414.309	6394.742	6396.560	7.477	1.000	25	25	334250	17
D(C=0.5)	6388.467	6405.316	6390.997	6393.048	4.428	1.000	25	25	464900	21

Na Tabela 5.11 observa-se que se o custo computacional máximo for aumentado para 500000 avaliações da função objetivo, o resultado do método APM Sum é reduzido, mas o método APM Med 4 passa a gerar o melhor resultado encontrado neste trabalho. Os métodos $D(C=0.5)$ e $D(C=1.0)$, que na Tabela 5.10 não apresentam bons desempenhos, na Tabela 5.11 têm resultados consideravelmente próximos do melhor encontrado. O método Coit Mod continua sendo aquele que apresentou os menores valores para as métricas Mediana e Média na resolução do problema VP. Já o método ADP Log se mantém como o que produziu o maior resultado.

A Tabela 5.12 compara o resultado obtido para o problema VP com outros algoritmos propostos na literatura. Observa-se que foi obtido resultado melhor que alguns entre os comparados, mas que existem também outros melhores.

Tabela 5.12: Comparação do resultado obtido para o problema Vaso de Pressão (VP) com outros trabalhos da literatura.

Trabalho	W	T_s	T_h	R	L	Avaliações
Coello e Cortes [98]	6061,123	0,813	0,438	42,087	176,779	150000
Bernardino et al. [94]	6059,855	0,813	0,438	42,097	176,651	80000
Runarsson e Yao [88]	6832,584	1,125	0,563	58,127	44,594	N/A
Akay e Karaboga [29]	6059.715	0,813	0,438	42,098	176.637	30000
Brajevic e Tuba [90]	6059,714	0,813	0,438	42,098	176,637	15000
Este estudo	6388,025	0,813	0,438	42,095	190,737	42450

5.5.5 VEL – Viga Engastada e Livre

Este problema trata-se da minimização do volume V de uma viga engastada e livre sujeita a uma carga $P = 50000N$, como mostra a Figura 5.6 [99, 7, 4, 92].

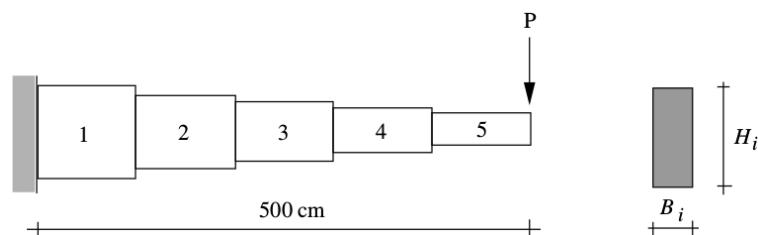


Figura 5.6: VEL – Viga Engastada e Livre.

O problema envolve dez variáveis de projeto que correspondem à altura (H_i) e à largura (B_i) da seção transversal retangular de cada um dos cinco elementos que compõem a viga. As variáveis B_1 e H_1 são inteiras, sendo que $1 \leq B_1, H_1 \leq 64$. As variáveis B_2 e B_3 , assumem valores discretos do conjunto $\{2, 4; 2, 6; 2, 8; 3, 1\}$. Já as variáveis H_2 e H_3 que também são discretas, admitem valores do conjunto $\{45, 0; 50, 0; 55, 0; 60, 0\}$. Por fim as variáveis B_4 , H_4 , B_5 e H_5 são contínuas tendo valores aceitos no intervalo $[1, 60]$. As variáveis são dadas em centímetros e o módulo de elasticidade do material é 200 GPa .

A formulação matemática do problema é dada por

$$\mathbf{VEL} = \left\{ \begin{array}{l} \text{Minimize} \quad V(H_i, B_i) = 100 \sum_{i=1}^5 H_i B_i \\ \text{Sujeito a} \quad g_i(H_i, B_i) = \sigma - 14000 \leq 0 \quad i = 1, 2, \dots, 5 \\ \quad \quad \quad g_{i+5}(H_i, B_i) = H_i/B_i - 20 \leq 0 \quad i = 1, 2, \dots, 5 \\ \quad \quad \quad g_{11}(H_i, B_i) = \delta - 2, 7 \leq 0 \\ \quad \quad \quad 1 \leq H_1 \leq 64 \\ \quad \quad \quad H_2 \text{ e } H_3 \in \{45, 0; 50, 0; 55, 0; 60, 0\} \\ \quad \quad \quad 1 \leq H_4, H_5 \leq 60 \\ \quad \quad \quad 1 \leq B_1 \leq 64 \\ \quad \quad \quad B_2 \text{ e } B_3 \in \{2, 4; 2, 6; 2, 8; 3, 1\} \\ \quad \quad \quad 1 \leq B_4, B_5 \leq 60 \end{array} \right. , \quad (5.8)$$

onde δ é o deslocamento da extremidade da viga na direção vertical.

A Tabela 5.13 mostra que com 5000 avaliações da função objetivo, os piores resultados para o problema VEL foram encontrados com os métodos Deb e D(C=0.5). O melhor resultado foi obtido com o método APM Worst. Entre as colunas Exec. e Fac. da Tabela 5.13 observa-se que nem todos os métodos produziram soluções factíveis nas 25 execuções, sendo que o método Deb foi o que resolveu o problema no menor número de execuções. Com isto, o método Deb é o menos indicado para ser integrado ao ABC G-best para resolução do problema VEL, seguido do método D(C=0.5). Destaca-se que os métodos APM Damp e A(k=1E5) geraram os menores valores para as métricas Mediana e Média respectivamente.

Tabela 5.13: Resultados obtidos para o problema Viga Engastada e Livre (VEL) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	7.890e+04	1.375e+05	9.364e+04	9.826e+04	1.440e+04	1.000	25	25	3000	16
APM Damp	7.425e+04	1.132e+05	8.690e+04	9.031e+04	1.115e+04	1.000	25	25	5000	23
APM Med 4	8.544e+04	1.720e+05	1.229e+05	1.200e+05	1.971e+04	1.000	25	25	5000	18
ADP Log	7.439e+04	2.058e+05	1.127e+05	1.224e+05	3.319e+04	1.000	25	25	3000	23
Coit Mod	7.813e+04	1.146e+05	9.363e+04	9.515e+04	1.006e+04	1.000	25	25	4450	12
APM Med	7.373e+04	1.082e+05	9.008e+04	9.079e+04	9058.374	1.000	25	25	4800	22
APM Sum	8.450e+04	1.201e+05	9.416e+04	9.764e+04	9997.060	1.000	25	25	4900	25
APM	7.580e+04	1.273e+05	9.312e+04	9.295e+04	1.302e+04	0.960	25	24	5000	23
APM Worst 2	7.348e+04	4.129e+05	8.972e+04	1.046e+05	6.662e+04	0.920	25	23	4600	11
APM Med 5	7.787e+04	1.763e+05	1.120e+05	1.174e+05	2.268e+04	0.920	25	23	3250	23
A(k=1E7)	8.230e+04	1.143e+05	9.416e+04	9.535e+04	8285.955	1.000	25	25	5000	23
APM Worst	7.165e+04	1.155e+05	9.607e+04	9.416e+04	1.189e+04	0.960	25	24	4950	23
D(C=1.0)	9.113e+04	2.003e+05	1.413e+05	1.470e+05	3.442e+04	0.960	25	24	2200	20
A(k=1E5)	7.670e+04	9.842e+04	8.880e+04	8.880e+04	5614.735	1.000	25	25	3650	10
APM Med 3	7.759e+04	1.114e+05	9.334e+04	9.449e+04	1.108e+04	1.000	25	25	3500	25
Deb	1.500e+05	7.608e+05	3.676e+05	4.055e+05	1.812e+05	0.440	25	11	1050	3
A(k=1E9)	7.429e+04	1.174e+05	9.713e+04	9.570e+04	9981.481	1.000	25	25	4300	13
D(C=0.5)	1.098e+05	2.914e+05	1.530e+05	1.613e+05	4.627e+04	0.960	25	24	950	21

Tabela 5.14: Resultados obtidos para o problema Viga Engastada e Livre (VEL) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	6.767e+04	7.852e+04	7.392e+04	7.372e+04	2676.130	1.000	25	25	324650	20
APM Damp	6.563e+04	7.153e+04	6.873e+04	6.841e+04	1408.751	1.000	25	25	342950	15
APM Med 4	6.780e+04	7.650e+04	7.226e+04	7.196e+04	2161.379	1.000	25	25	207250	10
ADP Log	6.652e+04	7.296e+04	6.928e+04	6.930e+04	1669.900	1.000	25	25	156950	9
Coit Mod	6.845e+04	7.355e+04	7.101e+04	7.099e+04	1465.213	1.000	25	25	208850	15
APM Med	6.586e+04	7.418e+04	6.891e+04	6.899e+04	1855.863	1.000	25	25	248300	9
APM Sum	6.878e+04	7.519e+04	7.181e+04	7.191e+04	1728.860	1.000	25	25	446250	15
APM	6.548e+04	7.130e+04	6.824e+04	6.808e+04	1478.328	1.000	25	25	208450	4
APM Worst 2	6.565e+04	7.272e+04	6.907e+04	6.933e+04	1850.070	1.000	25	25	284800	13
APM Med 5	6.757e+04	7.548e+04	7.236e+04	7.202e+04	1806.185	1.000	25	25	123900	3
A(k=1E7)	6.703e+04	7.720e+04	7.168e+04	7.204e+04	1973.384	1.000	25	25	253950	13
APM Worst	6.475e+04	6.957e+04	6.628e+04	6.634e+04	1111.290	1.000	25	25	287950	11
D(C=1.0)	6.655e+04	7.668e+04	7.282e+04	7.243e+04	2171.690	1.000	25	25	34800	23
A(k=1E5)	6.653e+04	7.510e+04	7.002e+04	7.050e+04	1946.796	1.000	25	25	79900	11
APM Med 3	6.496e+04	7.527e+04	6.875e+04	6.891e+04	2810.178	1.000	25	25	173050	19
Deb	1.265e+05	2.474e+05	1.844e+05	1.753e+05	2.910e+04	1.000	25	25	188700	18
A(k=1E9)	7.008e+04	7.709e+04	7.388e+04	7.370e+04	2253.988	1.000	25	25	312500	21
D(C=0.5)	6.575e+04	7.659e+04	7.137e+04	7.177e+04	2631.649	1.000	25	25	181000	7

A Tabela 5.14 mostra que aumentando o custo computacional máximo para 500000 avaliações da função objetivo, o método APM Worst se mantém como aquele que gerou o melhor resultado para o problema VEL. O método APM Worst também produziu os menores valores para as métricas Mediana e Média. Todos os métodos de penalização geraram soluções factíveis nas 25 execuções, como pode ser observando entre as colunas Exec. e Fac. da Tabela 5.14. Com o método Deb foi produzido o maior resultado entre os melhores encontrados, além disto, tal resultado ficou muito distante dos demais.

A Tabela 5.15 traz a comparação do melhor resultado obtido com outros encontrados da literatura. Observa-se que a maioria dos algoritmos comparados geram resultados melhores que o obtido aqui, mas existem também resultados piores.

Tabela 5.15: Comparação do resultado obtido para o problema Viga Engastada e Livre (VEL) com outros trabalhos da literatura.

Trabalho	V	H_1	B_1	H_2	B_2	H_3	B_3	H_4	B_4	H_5	B_5	Avaliações
Lemonge e Barbosa [59]	64698,560	60	3	55,0	3,1	50,0	2,6	45,626	2,289	34,593	1,793	35000
Bernardino, Barbosa e Lemonge [94]	65559,600	60	3	55,0	3,1	50,0	2,6	44,395	2,235	32,879	2,004	35000
Erbatur et al. [99]	64834,380	60	3	55,0	3,1	50,0	2,6	45,431	2,308	34,731	1,814	N/A
Runarson e Yao [88]	64599,650	60	3	55,0	3,1	50,0	2,6	45,551	2,284	35,063	1,753	N/A
Carvalho [4]	64578,229	60	4	55,0	3,1	50,0	2,6	44,091	2,204	34,995	1,749	35000
Este estudo	64750,000	60	3	55,0	3,1	50,0	2,6	43,858	2,243	38,609	2,006	173050

5.5.6 VS – Viga Soldada

Neste problema o objetivo é minimizar o custo $C(h, l, t, b)$ de uma viga soldada, ilustrada na Figura 5.7. Considera-se $h \in [0, 125; 10, 0]$ e $0, 1 \leq l, t, b \leq 10, 0$ [84, 7, 4, 92].

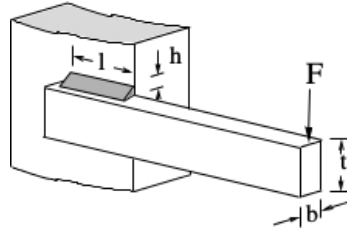


Figura 5.7: VS – Viga Soldada.

A formulação matemática do problema é dado por

$$\text{VS} = \left\{ \begin{array}{l} \text{Minimize} \quad C(h, l, t, b) = 1,10471h^2l + 0,04811tb(14,0 + l) \\ \text{Sujeito a} \quad g_1(\tau) = \tau - 13600 \leq 0 \\ \quad \quad \quad g_2(\sigma) = \sigma - 30000 \leq 0 \\ \quad \quad \quad g_3(h, b) = h - b \leq 0 \\ \quad \quad \quad g_4(P_c) = 6000 - P_c \leq 0 \\ \quad \quad \quad g_5(\delta) = \delta - 0,25 \leq 0 \\ \quad \quad \quad 0,125 \leq h \leq 10; \quad 0,1 \leq l, t, b \leq 10 \end{array} \right. , \quad (5.9)$$

onde

$$\begin{aligned} \tau' &= \frac{6000}{\sqrt{2hl}}, & \tau'' &= \frac{6000(14 + 0,5l)\alpha}{2\{0,707hl[l^2/12 + 0,25(h+t)^2]\}} \\ \alpha &= \sqrt{0,25[l^2 + (h+t)^2]}, & \tau &= \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'\tau''/\alpha} \end{aligned} \quad (5.10)$$

$$\sigma = \frac{504000}{t^2b}, \quad P_c = 64746,022(1 - 0,0282346t)tb^3, \quad \delta = \frac{2,1952}{t^3b}$$

A Tabela 5.16 mostra que o maior resultado apresentado na coluna Melhor foi obtido com o método ADP Log. Esta informação fica ainda mais evidente se observado o valor na coluna Média relacionado a este método, tal valor tem diferença considerável se comparado com os demais da mesma coluna. Já o método APM Med apresentou os menores valores para as métricas Mediana e Média.

Tabela 5.16: Resultados obtidos para o problema Viga Soldada (VS) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	2.546	6.640	3.598	3.919	1.074	1.000	25	25	2050	13
APM Damp	2.520	5.283	3.815	3.646	0.817	1.000	25	25	1000	17
APM Med 4	2.415	5.226	3.562	3.445	0.739	1.000	25	25	4800	24
ADP Log	2.856	20.169	6.706	8.421	4.740	1.000	25	25	150	14
Coit Mod	2.491	6.527	3.222	3.344	0.902	1.000	25	25	2900	14
APM Med	2.474	4.770	2.861	3.144	0.658	1.000	25	25	4950	21
APM Sum	2.579	5.765	3.497	3.751	0.787	1.000	25	25	450	13
APM	2.540	6.119	3.650	3.828	1.037	1.000	25	25	4100	14
APM Worst 2	2.478	6.071	3.327	3.497	0.782	1.000	25	25	2850	14
APM Med 5	2.610	5.325	3.516	3.710	0.741	1.000	25	25	400	14
A(k=1E7)	2.555	5.781	3.643	3.824	0.868	1.000	25	25	1800	13
APM Worst	2.504	5.611	3.212	3.489	0.828	1.000	25	25	2350	13
D(C=1.0)	2.558	5.232	3.372	3.439	0.561	1.000	25	25	2800	19
A(k=1E5)	2.750	6.851	3.731	3.976	1.046	1.000	25	25	1250	19
APM Med 3	2.524	5.529	3.320	3.582	0.866	1.000	25	25	950	14
Deb	2.478	4.705	3.510	3.540	0.751	1.000	25	25	3050	1
A(k=1E9)	2.555	5.302	3.745	3.944	0.812	1.000	25	25	1800	13
D(C=0.5)	2.590	5.462	3.566	3.774	0.728	1.000	25	25	4350	8

Tabela 5.17: Resultados obtidos para o problema Viga Soldada (VS) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	2.409	2.570	2.478	2.485	0.045	1.000	25	25	173200	23
APM Damp	2.384	2.403	2.393	2.393	0.005	1.000	25	25	331050	22
APM Med 4	2.384	2.418	2.403	2.402	0.009	1.000	25	25	203050	1
ADP Log	2.856	6.706	4.905	5.036	1.098	1.000	25	25	150	14
Coit Mod	2.386	2.497	2.410	2.417	0.023	1.000	25	25	464850	23
APM Med	2.382	2.396	2.384	2.385	0.003	1.000	25	25	181650	4
APM Sum	2.399	2.525	2.476	2.471	0.032	1.000	25	25	479200	4
APM	2.383	2.397	2.389	2.389	0.004	1.000	25	25	433450	24
APM Worst 2	2.383	2.400	2.391	2.391	0.004	1.000	25	25	350650	18
APM Med 5	2.393	2.481	2.415	2.420	0.019	1.000	25	25	422400	23
A(k=1E7)	2.396	2.653	2.467	2.480	0.052	1.000	25	25	175300	7
APM Worst	2.383	2.409	2.391	2.393	0.007	1.000	25	25	253050	24
D(C=1.0)	2.404	2.548	2.443	2.449	0.034	1.000	25	25	119450	7
A(k=1E5)	2.410	2.614	2.480	2.486	0.046	1.000	25	25	283900	23
APM Med 3	2.383	2.393	2.386	2.387	0.003	1.000	25	25	290550	11
Deb	2.390	2.544	2.455	2.458	0.036	1.000	25	25	245250	22
A(k=1E9)	2.415	2.593	2.479	2.486	0.054	1.000	25	25	316350	19
D(C=0.5)	2.412	2.533	2.440	2.449	0.028	1.000	25	25	389550	23

Na Tabela 5.17 observa-se que o melhor resultado encontrado para o problema VS passa a ser obtido com o método APM Med se o custo computacional máximo for aumentado para 500000 avaliações da função objetivo. Este método também produziu os menores valores para as métricas Mediana e Média. Já o método ADP Log foi aquele com o qual obteve-se o maior entre os melhores resultados encontrados com os métodos.

A Tabela 5.18 apresenta resultados para o problema VS encontrados na literatura que são comparados àquele aqui obtido. Observa-se que o resultado gerado com o ABC G-best, tem diferença quase ignorável daqueles obtidos nos trabalhos comparados. O melhor resultado entre os comparados foi obtido no trabalho de Hwang e He [100].

Tabela 5.18: Comparação do resultado obtido para o problema Viga Soldada (VS) com outros trabalhos da literatura.

Trabalho	C	h	l	t	b	Avaliações
Lemonge e Barbosa [59]	2,382	0,244	6,212	8,302	0,244	200000
Bernardino et al. [85]	2,381	0,244	6,218	8,291	0,244	320000
Hedar e Fukushima [101]	2,381	0,244	6,216	8,294	0,244	N/A
Liu [102]	2,381	0,244	6,218	8,292	0,244	N/A
Hwang e He [100]	2,250	0,223	1,582	12,847	0,225	N/A
Lee e Geem [103]	2,381	0,244	6,223	8,292	0,244	110000
Zhang, Luo e Wang [104]	2,381	0,244	6,218	8,292	0,244	4800
Este estudo	2,382	0.245	6.211	8.296	0.245	181650

5.5.7 *Discussão sobre os resultados dos problemas clássicos da Engenharia Mecânica*

As Tabelas 5.19 e 5.20 trazem os valores da função objetivo das melhores soluções encontradas para cada um dos problemas, obtidas com cada um dos métodos de penalização analisados. Em alguns dos problemas como MTC, VP, VEL e VS foi possível obter melhora do resultado com o aumento do número de iterações do algoritmo. Já para os problemas RV e T3B, os melhores resultados foram obtidos com apenas 5000 avaliações da função objetivo adotado como custo computacional máximo. Entre os resultados de cada problema, havendo maior distinção, com os métodos APM Worst, APM Sum, APM Med e APM Med 4 foram obtidos os melhores resultados. Entre estes métodos, o **APM**

Worst e o **APM Med 4** aparecem em destaque nas duas tabelas, e se destacam nesta análise. Com isto estes dois métodos são os mais indicados para serem integrados ao ABC G-best na resolução dos problemas da Engenharia Mecânica.

Tabela 5.19: Melhores resultados obtidos para os problemas da Engenharia Mecânica adotando 5000 avaliações da função objetivo. Entre as colunas da tabela RV = Redutor de Velocidade, MTC = Mola sob Tração/Compressão, VEL = Viga Engastada e Livre, VP = Vaso de Pressão, T3B = Treliça de Três Barras, VS = Viga Soldada

	RV	MTC	VEL	VP	T3B	VS
APM Mono	2994.341	0.012686	7.890e+04	6535.276	263.897	2.546
APM Damp	2994.341	0.012675	7.425e+04	6419.887	263.897	2.520
APM Med 4	2994.387	0.012729	8.544e+04	6398.716	263.897	2.415
ADP Log	2994.341	0.016759	7.439e+04	9544.481	264.919	2.856
Coit Mod	2994.341	0.012725	7.813e+04	6390.801	263.898	2.491
APM Med	2994.341	0.012688	7.373e+04	6425.155	263.909	2.474
APM Sum	2994.341	0.012689	8.450e+04	6390.286	263.896	2.579
APM	2994.341	0.012681	7.580e+04	6402.717	263.899	2.540
APM Worst 2	2994.341	0.012687	7.348e+04	6406.141	263.897	2.478
APM Med 5	2994.435	0.012684	7.787e+04	6400.090	263.902	2.610
A(k=1E7)	2994.343	0.012686	8.230e+04	6420.472	263.896	2.555
APM Worst	2994.341	0.012687	7.165e+04	6397.573	263.897	2.504
D(C=1.0)	3002.647	0.012701	9.113e+04	7929.329	263.943	2.558
A(k=1E5)	2994.441	0.012691	7.670e+04	6526.156	263.906	2.750
APM Med 3	2994.341	0.012711	7.759e+04	6399.892	263.947	2.524
Deb	3325.060	0.012675	1.500e+05	6401.397	263.896	2.478
A(k=1E9)	2994.341	0.012690	7.429e+04	6399.843	263.896	2.555
D(C=0.5)	3037.711	0.012696	1.098e+05	8413.795	263.911	2.590

Tabela 5.20: Melhores resultados obtidos para os problemas da Engenharia Mecânica adotando 500000 avaliações da função objetivo. Entre as colunas da tabela RV = Redutor de Velocidade, MTC = Mola sob Tração/Compressão, VEL = Viga Engastada e Livre, VP = Vaso de Pressão, T3B = Treliça de Três Barras, VS = Viga Soldada

	RV	MTC	VEL	VP	T3B	VS
APM Mono	2994.341	0.012666	6.767e+04	6390.204	263.896	2.409
APM Damp	2994.341	0.012665	6.563e+04	6388.399	263.896	2.384
APM Med 4	2994.341	0.012665	6.780e+04	6388.025	263.896	2.384
ADP Log	2994.341	0.012852	6.652e+04	7337.096	263.969	2.856
Coit Mod	2994.341	0.012668	6.845e+04	6388.243	263.896	2.386
APM Med	2994.341	0.012665	6.586e+04	6425.155	263.897	2.382
APM Sum	2994.341	0.012665	6.878e+04	6389.483	263.896	2.399
APM	2994.341	0.012665	6.548e+04	6389.543	263.896	2.383
APM Worst 2	2994.341	0.012665	6.565e+04	6389.278	263.896	2.383
APM Med 5	2994.341	0.012665	6.757e+04	6388.992	263.896	2.393
A(k=1E7)	2994.343	0.012666	6.703e+04	6388.237	263.896	2.396
APM Worst	2994.341	0.012665	6.475e+04	6389.635	263.896	2.383
D(C=1.0)	2994.357	0.012666	6.655e+04	6388.453	263.896	2.404
A(k=1E5)	2994.441	0.012666	6.653e+04	6418.721	263.896	2.410
APM Med 3	2994.341	0.012665	6.496e+04	6399.892	263.896	2.383
Deb	3262.073	0.012665	1.265e+05	6389.314	263.896	2.390
A(k=1E9)	2994.341	0.012666	7.008e+04	6388.082	263.896	2.415
D(C=0.5)	2994.347	0.012665	6.575e+04	6388.467	263.896	2.412

As Figuras 5.8, 5.9, 5.10 e 5.11 trazem resultados obtidos com a técnica Perfis de Desempenho. Porém as figuras destacam as regiões dos gráficos em que é possível identificar os menores valores de τ tal que $\rho(\tau) = 1$. Observa-se que independente da métrica usada para $t_{p,s}$ ou o número máximo de avaliações, as variantes do método APM estão sempre entre os métodos mais robustos. O método **APM Worst** merece destaque visto que aparece entre os melhores métodos em todas as figuras mencionadas. Portanto, o método APM Worst é considerado o método mais robusto para ser combinado ao ABC G-best na resolução deste grupo de problemas.

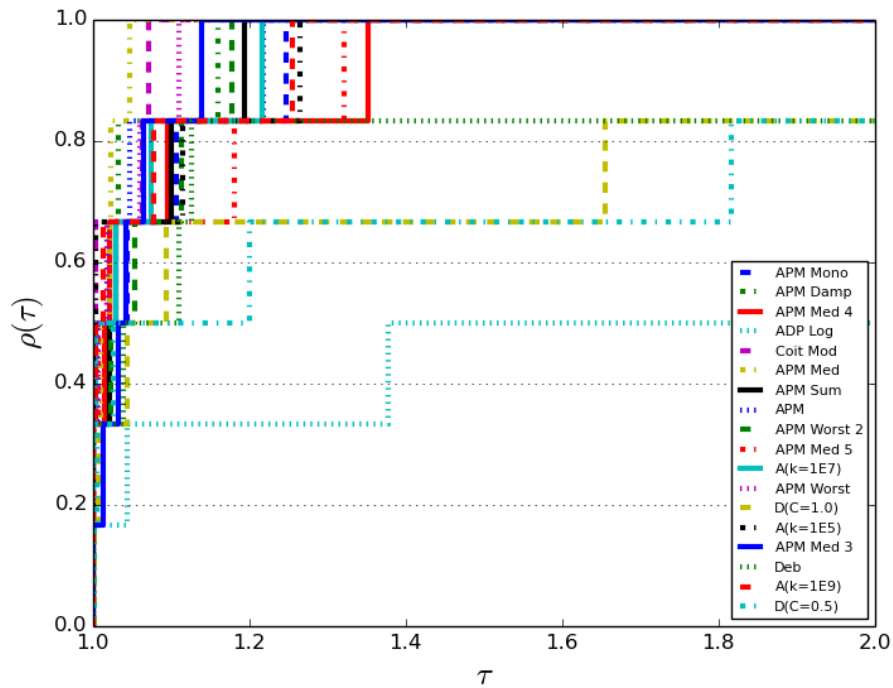


Figura 5.8: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

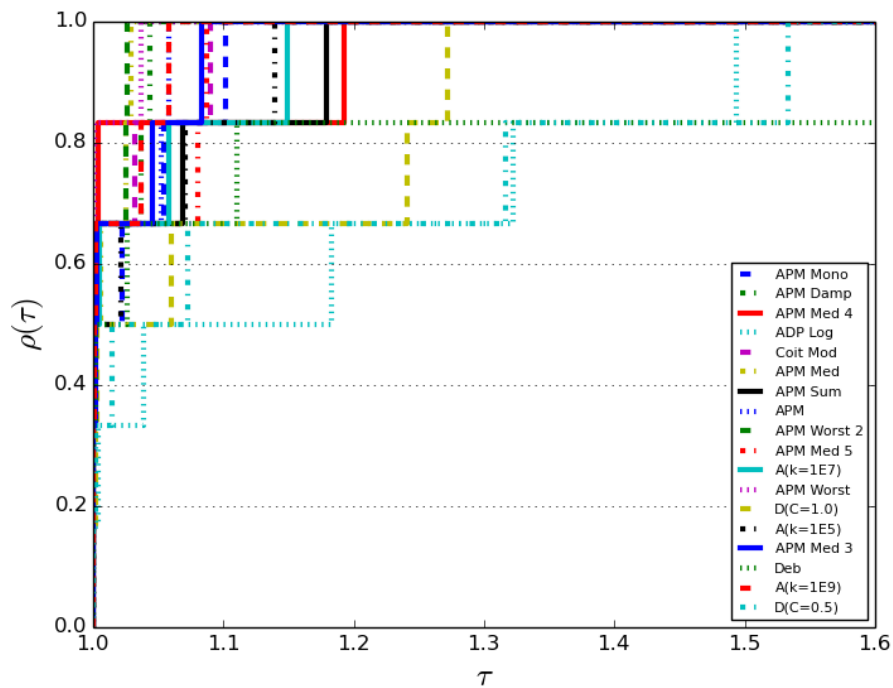


Figura 5.9: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

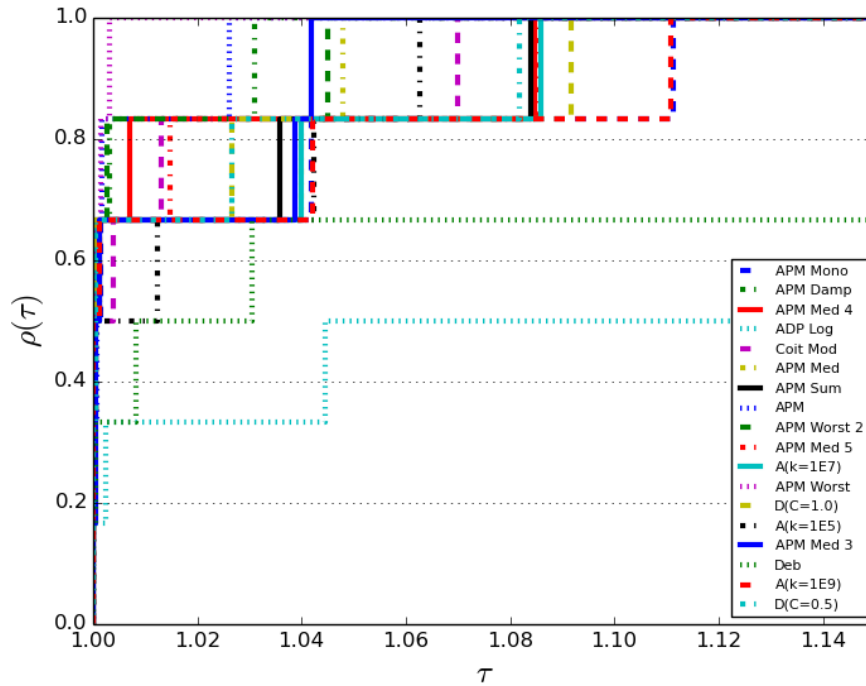


Figura 5.10: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

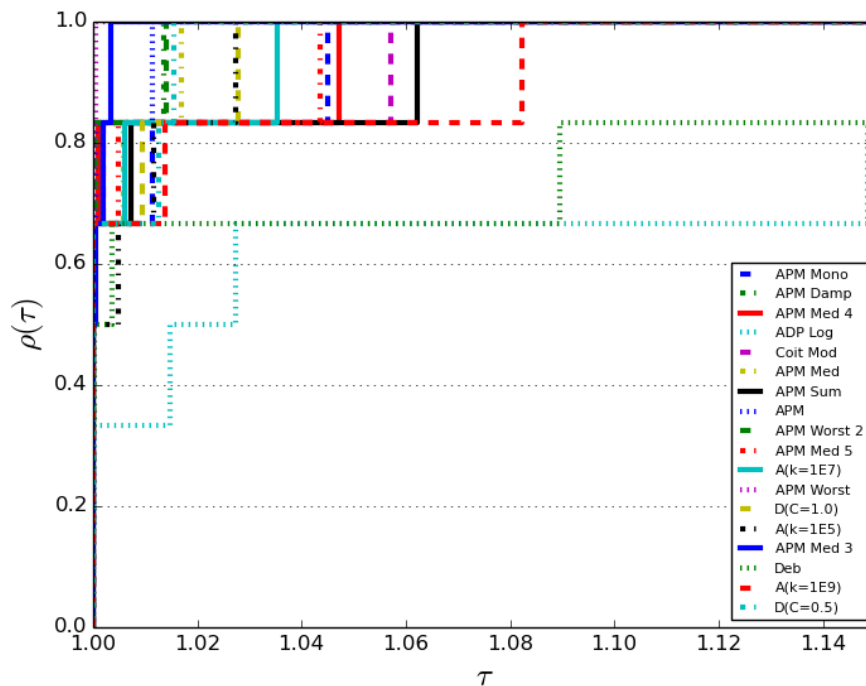


Figura 5.11: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas da Engenharia Mecânica adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

5.6 Experimento 02 – Problemas de otimização estrutural

Este experimento consiste da resolução de alguns problemas de Otimização Estrutural encontrados na literatura. Quatro problemas desta classe foram impostos ao algoritmo proposto neste trabalho. Cada um dos problemas é descrito entre as Seções 5.6.1 e 5.6.4, onde também encontram-se discussões sobre os resultados obtidos.

5.6.1 T10C – Treliça de 10 Barras – Caso Contínuo

Este problema de Otimização Estrutural é bastante utilizado na literatura e consiste da minimização do peso de uma treliça composta por 10 barras, ilustrada na Figura 5.12 [95, 7, 85, 4, 59]. No problema a treliça está sujeita a dois carregamentos $P = 100 \text{ kips}$ nos nós 2 e 4, como mostra a Figura

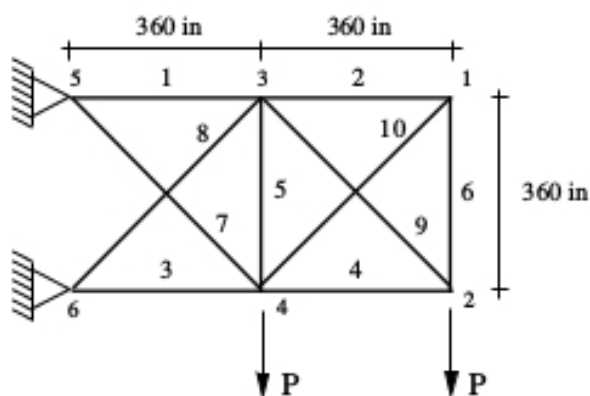


Figura 5.12: T10C – Treliça de 10 Barras – Caso Contínuo.

As 10 variáveis de projeto correspondem às áreas A_i ($i = 1, 2, \dots, 10$) das seções transversais de cada uma das barras, sendo que $0,1 \leq A_i \leq 33,5$. A função objetivo é linear e é definida por

$$W = \sum_{i=1}^n \rho A_i L_i, \quad (5.11)$$

onde n é o número de barras, L_i é o comprimento da i -ésima barra e ρ é a densidade do material. O problema está sujeito a restrições de limites de tensão em cada membro, que

normalizadas são definidas por

$$\frac{\sigma_i}{\bar{\sigma}} - 1 \leq 0, \quad i = 1, 2, \dots, n, \quad (5.12)$$

onde n é o número de barras e $\bar{\sigma}$ é o limite máximo de tensão aceito. Também estão presentes restrições de limites de deslocamentos aceitos, que normalizadas são definidas por

$$\frac{u_j}{\bar{u}} = -1 \leq 0, \quad j = 1, 2, \dots, m, \quad (5.13)$$

onde m é o grau de liberdade da estrutura e \bar{u} é o deslocamento máximo aceito.

Para este problema considera-se $\bar{\sigma} = \pm 25 \text{ ksi}$ o limite máximo de tensão permitido. Os deslocamentos são limitados em $\bar{u} = 2 \text{ in}$ nas direções x e y . A densidade do material é $\rho = 0,1 \text{ lb/in}^3$ e o módulo de elasticidade é $E = 10^4 \text{ ksi}$.

Considera-se a aplicação das Equações (5.11), (5.12) e (5.13) também nas formulações dos problemas descritos entre as Seções 5.6.2 e 5.6.4.

A Tabela 5.21 mostra que para o problema T10C, com o método APM Med foi gerado o melhor resultado encontrado sob o custo computacional de 5000 avaliações da função objetivo. Os métodos D(C=0.5), D(C=1.0) e Deb foram aqueles com os quais foram obtidos os maiores resultados apresentados na coluna Melhor. Entre eles os métodos D(C=0.5) e D(C=1.0) apresentam as maiores diferenças de resultados entre os demais. Observa-se que o método A(k=1E7) apresentou os menores valores para as métricas Mediana e Média.

Tabela 5.21: Resultados obtidos para o problema Treliça de 10 Barras – Caso Contínuo (T10C) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	5103.011	5686.846	5269.938	5318.855	146.244	1.000	25	25	4050	19
APM Damp	5147.591	6050.609	5299.235	5382.298	210.837	1.000	25	25	2750	8
APM Med 4	5192.006	6417.211	5562.480	5598.842	308.233	1.000	25	25	1050	13
ADP Log	5102.058	8868.591	7266.009	6882.086	1234.023	0.960	25	24	2450	7
Coit Mod	5132.942	5892.186	5243.540	5290.010	162.455	1.000	25	25	3900	9
APM Med	5091.848	7350.100	5328.966	5552.261	509.229	1.000	25	25	2350	25
APM Sum	5106.494	5995.875	5260.998	5337.527	207.194	1.000	25	25	3150	25
APM	5117.249	5871.366	5337.390	5346.803	177.430	1.000	25	25	4800	7
APM Worst 2	5108.477	6183.789	5362.664	5425.888	229.184	1.000	25	25	4900	14
APM Med 5	5173.263	6342.519	5405.606	5471.528	285.825	1.000	25	25	2600	8
A(k=1E7)	5128.497	5485.674	5219.634	5239.848	94.165	1.000	25	25	3150	21
APM Worst	5117.135	6296.745	5406.931	5444.831	257.479	1.000	25	25	4350	25
D(C=1.0)	7036.117	9227.551	7906.937	8047.060	592.763	0.920	25	23	0	24
A(k=1E5)	5093.080	5651.548	5229.260	5264.631	123.425	1.000	25	25	1650	15
APM Med 3	5096.019	6647.533	5223.817	5454.419	425.982	1.000	25	25	4950	1
Deb	5300.181	8868.591	6690.643	6804.588	827.434	0.920	25	23	1150	18
A(k=1E9)	5116.547	6191.989	5258.091	5353.660	262.486	1.000	25	25	2000	13
D(C=0.5)	7036.117	9227.551	7942.170	8079.317	557.244	0.920	25	23	0	24

Tabela 5.22: Resultados obtidos para o problema Treliça de 10 Barras – Caso Contínuo (T10C) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	5084.582	5126.265	5095.843	5097.449	9.980	1.000	25	25	142150	23
APM Damp	5085.479	5122.683	5096.285	5097.597	9.255	1.000	25	25	415950	12
APM Med 4	5071.531	5107.990	5091.573	5090.899	8.223	1.000	25	25	348700	4
ADP Log	5073.019	5130.013	5093.763	5095.818	12.093	1.000	25	25	390250	9
Coit Mod	5075.115	5109.966	5092.671	5092.655	7.744	1.000	25	25	443150	13
APM Med	5080.283	5104.066	5088.769	5089.637	6.314	1.000	25	25	144350	1
APM Sum	5067.223	5082.550	5078.808	5077.767	3.993	1.000	25	25	295100	1
APM	5078.896	5115.280	5093.017	5094.771	8.416	1.000	25	25	125050	7
APM Worst 2	5079.361	5112.339	5093.906	5094.938	9.444	1.000	25	25	45550	1
APM Med 5	5077.647	5111.492	5098.514	5097.455	8.243	1.000	25	25	371300	15
A(k=1E7)	5082.401	5150.006	5122.171	5120.551	18.279	1.000	25	25	81350	14
APM Worst	5074.957	5113.559	5097.985	5098.073	9.366	1.000	25	25	470200	11
D(C=1.0)	5083.283	5134.060	5098.787	5099.876	12.014	1.000	25	25	473800	25
A(k=1E5)	5067.438	5117.066	5083.255	5084.716	14.513	1.000	25	25	168450	3
APM Med 3	5079.352	5103.481	5089.115	5089.579	6.496	1.000	25	25	265800	12
Deb	5300.181	7084.630	6552.302	6453.286	433.684	1.000	25	25	1150	18
A(k=1E9)	5085.854	5128.842	5105.473	5104.686	12.316	1.000	25	25	150400	1
D(C=0.5)	5071.378	5122.471	5096.363	5097.231	11.630	1.000	25	25	40600	16

Na Tabela 5.22 observa-se que com o custo máximo aumentado para 500000 avaliações da função objetivo, houve redução do resultado obtido com a maioria dos métodos. Nota-se que o método com o qual foi obtido o melhor resultado para o problema T10C, passa a ser o APM Sum. Este método também apresentou os menores valores para as métricas Mediana e Média. Deve-se notar que entre as Tabelas 5.21 e 5.22 o resultado obtido com o método Deb não sofreu alteração, além de ser o maior valor presente na coluna Melhor da Tabela 5.22.

A Tabela 5.23 traz outros resultados para o problema T10C encontrados na literatura comparados com o melhor resultado obtido neste trabalho. A tabela mostra que com o algoritmo proposto neste trabalho foi obtido resultado melhor que alguns da literatura, mas que a maioria dos trabalhos comparados encontraram resultados melhores.

Tabela 5.23: Comparação do resultado obtido para o problema Treliça de 10 Barras – Caso Contínuo (T10C) com outros trabalhos da literatura.

Trabalho	W	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	Avaliações
Aragon et al. [95]	5142,300	31,238	0,317	23,611	14,507	0,316	0,316	8,135	21,618	21,222	0,316	350000
Lemonge e Barbosa [59]	5069,086	29,226	0,100	24,182	14,947	0,100	0,395	7,496	21,925	21,291	0,100	280000
Bernardino, Barbosa e Lemonge [94]	5064,670	29,781	0,100	22,551	15,505	0,100	0,524	7,529	21,157	22,214	0,100	280000
Bernardino et al. [85]	5061,160	30,527	0,100	22,916	15,483	0,100	0,546	7,476	21,016	21,554	0,100	280000
Runarsson e Yao [88]	5061,710	30,014	0,100	26,145	15,293	0,100	0,556	7,440	21,006	21,939	0,100	N/A
Silva [105]	5060,876	30,543	0,100	23,191	15,197	0,100	0,543	7,458	21,032	21,548	0,100	280000
Hadidi, Azad e Azad [38]	5060,970	30,657	0,100	23,043	15,282	0,100	0,563	7,472	21,008	21,509	0,100	N/A
Sonmez [54]	5060,888	30,548	0,100	23,180	15,218	0,100	0,551	7,463	21,058	21,501	0,100	500000
Este estudo	5067,223	31,897	0,100	23,842	14,778	0,100	0,100	8,570	20,330	20,755	0,100	295100

5.6.2 T10D – Treliça de 10 Barras – Caso Discreto

Este problema trata-se daquele descrito na Seção 5.6.1, que consiste da minimização do peso de uma treliça composta por 10 barras, sujeita a dois carregamentos $P = 100$ kips como mostra a Figura 5.12. A diferença entre os dois problemas é que neste caso as variáveis de projeto A_i ($i = 1, 2, \dots, 10$) passam a ser discretas. Sendo assim, as áreas das seções transversais das barras da treliça deverão ser escolhidas entre os valores 1,62; 1,80; 1,99; 2,13; 2,38; 2,62; 2,63; 2,88; 2,93; 3,09; 3,13; 3,38; 3,47; 3,55; 3,63; 3,84; 3,87; 3,88; 4,18; 4,22; 4,49; 4,59; 4,80; 4,97; 5,12; 5,74; 7,22; 7,97; 11,50; 13,50; 13,90; 14,20; 15,50; 16,00; 16,90; 18,80; 19,90; 22,00; 22,90; 26,50; 30,00; 33,50.

A Tabela 5.24 mostra que sob o custo computacional máximo de 5000 avaliações da função objetivo, o método A(k=1E7) foi aquele com o qual foi obtido o melhor resultado para o problema T10D. Já os menores valores para as métricas Mediana e Média foram obtidos com os métodos APM Worst 2 e A(k=1E5). Métodos com os quais não foi possível obter soluções factíveis, as colunas são preenchidas com o sinal “-”. A Tabela 5.24 é seguida da Figura 5.13 que traz um gráfico dendrograma gerado com o resultado obtido com a aplicação da técnica Escalonamento Multidimensional (MDS). O gráfico dendrograma tem a forma de uma árvore onde é possível visualizar os níveis de similaridade entre objetos comparados. O eixo vertical apresenta o nível de similaridade entre os objetos, sendo que estes são listados no eixo horizontal. As linhas verticais partindo dos objetos agrupados têm altura correspondente ao nível de semelhança com os demais objetos. Nota-se que com os métodos APM Mono e A(k=1E9) poderiam ser gerados resultados próximos daquele obtido com o método A(k=1E7). Como pode ser observado na Tabela 5.24, com o método ADP Log foram obtidos resultados de qualidade inferior aos dos outros métodos. Esta informação fica clara também no gráfico da Figura 5.13, já que este método aparece isolado dos demais a uma distância considerável. Os resultados apresentados na Tabela 5.24 referentes aos métodos A(k=1E5) e APM Sum são bons na comparação entre todos e merecem destaque, além desses métodos aparecerem posicionados num mesmo nível no gráfico da Figura 5.13. Observa-se ainda na Tabela 5.24 que com os métodos APM Med 4, APM Med, D(C=1.0), Deb e D(C=0.5) o problema T10D não foi resolvido sob o custo de 5000 avaliações.

Tabela 5.24: Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	5549.204	7024.978	5875.223	5911.152	284.985	0.960	25	24	1300	23
APM Damp	5608.160	7155.815	6025.993	6201.716	488.265	0.400	25	10	2400	25
APM Med 4	–	–	–	–	–	0.000e+00	25	0	–	–
ADP Log	7619.681	7619.681	7619.681	7619.681	0.000e+00	0.040	25	1	50	13
Coit Mod	5610.341	7268.922	5772.298	5914.450	395.989	1.000	25	25	4750	24
APM Med	–	–	–	–	–	0.000e+00	25	0	–	–
APM Sum	5538.341	5957.410	5716.189	5716.709	103.950	1.000	25	25	2800	17
APM	6217.369	7276.725	6292.615	6595.570	482.628	0.120	25	3	4750	25
APM Worst 2	5590.770	7204.365	5609.580	6134.905	756.261	0.120	25	3	5000	25
APM Med 5	5616.380	6928.191	5815.390	6064.252	465.882	0.200	25	5	3800	12
A(k=1E7)	5505.138	6877.706	5794.497	5855.563	254.190	1.000	25	25	1000	13
APM Worst	5728.500	7204.365	6173.031	6319.732	555.592	0.160	25	4	4950	22
D(C=1.0)	–	–	–	–	–	0.000e+00	25	0	–	–
A(k=1E5)	5532.121	5900.352	5693.834	5698.089	96.319	1.000	25	25	3950	15
APM Med 3	5873.860	5873.860	5873.860	5873.860	0.000e+00	0.040	25	1	4200	19
Deb	–	–	–	–	–	0.000e+00	25	0	–	–
A(k=1E9)	5548.457	6445.966	5868.159	5886.598	202.604	1.000	25	25	600	13
D(C=0.5)	–	–	–	–	–	0.000e+00	25	0	–	–

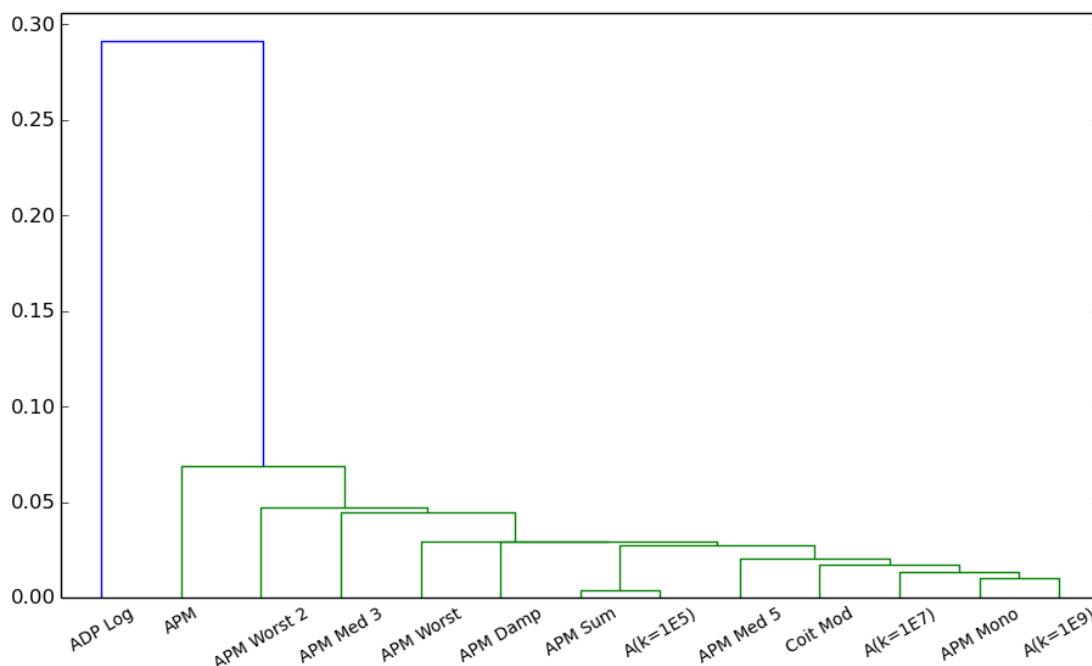


Figura 5.13: Dendrograma do resultado da técnica MDS para o problema Treliça de 10 Barras – Caso Discreto (T10D) adotando 5000 avaliações da função objetivo.

Na Tabela 5.25 observa-se que quando aumentado o número máximo de avaliações da função objetivo para 50000, cresce o número de métodos com os quais o ABC G-best resolve o problema T10D. Nota-se também que sob o este custo, com 7 dos 18 métodos analisados neste trabalho foi obtido o melhor resultado encontrado para o problem T10D. O ADP Log foi aquele com o qual foi gerado o maior valor apresentado na coluna Melhor da Tabela 5.25, devendo ser observada a diferença grande do resultado deste método quando comparado com os demais. Além disto, entre as colunas Exec. e Fac. da Tabela 5.25 constata-se que com o método ADP Log o ABC G-best convergiu em apenas 1 das 25 execuções.

Tabela 5.25: Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) com cada método de penalização combinado ao ABC G-best, adotando 50000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	5499.354	5768.468	5580.857	5595.329	68.262	1.000	25	25	12650	19
APM Damp	5491.717	5586.227	5538.341	5534.372	23.477	1.000	25	25	30200	20
APM Med 4	5532.121	8204.466	5694.940	6052.142	871.881	0.680	25	17	27450	24
ADP Log	7619.681	7619.681	7619.681	7619.681	0.000e+00	0.040	25	1	50	13
Coit Mod	5490.738	5656.224	5544.340	5557.781	42.320	1.000	25	25	37250	13
APM Med	5537.719	7167.789	5888.945	5928.707	422.294	0.440	25	11	9750	23
APM Sum	5491.717	5617.030	5538.341	5540.547	35.928	1.000	25	25	32500	7
APM	5490.738	5612.096	5538.341	5540.328	31.193	1.000	25	25	26300	14
APM Worst 2	5498.375	5655.201	5532.121	5540.764	33.964	1.000	25	25	12750	19
APM Med 5	5491.717	5727.552	5564.881	5562.172	48.710	1.000	25	25	13500	15
A(k=1E7)	5490.738	5668.965	5549.754	5558.223	48.146	1.000	25	25	36300	5
APM Worst	5490.738	5696.101	5545.435	5559.029	42.572	1.000	25	25	9850	17
D(C=1.0)	5491.717	5598.306	5527.836	5528.131	25.474	1.000	25	25	38250	12
A(k=1E5)	5490.738	5630.663	5523.138	5531.694	33.143	1.000	25	25	49250	5
APM Med 3	5572.339	6136.142	5861.317	5808.935	164.687	0.400	25	10	20800	2
Deb	–	–	–	–	–	0.000e+00	25	0	–	–
A(k=1E9)	5490.738	5836.968	5566.118	5573.956	68.825	1.000	25	25	14950	15
D(C=0.5)	5490.738	5628.341	5528.086	5534.405	29.451	1.000	25	25	38950	25

Tabela 5.26: Resultados obtidos para o problema Treliça de 10 Barras – Caso Discreto (T10D) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	5490.738	5558.184	5505.138	5512.649	17.562	1.000	25	25	231550	16
APM Damp	5490.738	5525.533	5499.354	5502.147	11.524	1.000	25	25	88450	9
APM Med 4	5491.717	5636.898	5537.719	5549.179	32.051	1.000	25	25	481100	13
ADP Log	5814.914	9218.873	7518.394	7524.336	1077.800	0.200	25	5	387150	3
Coit Mod	5490.738	5514.402	5499.354	5498.356	7.541	1.000	25	25	37250	13
APM Med	5490.738	5744.053	5542.117	5556.543	54.722	1.000	25	25	110650	5
APM Sum	5490.738	5498.375	5490.738	5491.701	2.021	1.000	25	25	111650	6
APM	5490.738	5523.906	5499.354	5500.756	8.759	1.000	25	25	26300	14
APM Worst 2	5490.738	5522.040	5504.158	5503.633	9.314	1.000	25	25	104400	25
APM Med 5	5490.738	5528.086	5499.354	5504.357	12.277	1.000	25	25	300450	18
A(k=1E7)	5490.738	5530.775	5499.354	5500.970	10.168	1.000	25	25	36300	5
APM Worst	5490.738	5535.036	5505.138	5508.062	13.650	1.000	25	25	9850	17
D(C=1.0)	5490.738	5522.040	5499.354	5499.856	8.404	1.000	25	25	56300	18
A(k=1E5)	5490.738	5505.138	5491.717	5492.700	3.536	1.000	25	25	49250	5
APM Med 3	5490.738	5813.027	5540.406	5554.755	59.377	1.000	25	25	394800	7
Deb	6561.310	8437.233	7593.816	7546.544	665.206	0.160	25	4	72900	13
A(k=1E9)	5490.738	5538.341	5505.138	5507.699	14.624	1.000	25	25	14950	15
D(C=0.5)	5490.738	5532.121	5499.354	5502.611	10.529	1.000	25	25	38950	25

A Tabela 5.26 confirma o crescimento do número de métodos de penalização que combinados ao ABC G-best produzem ótimos resultados para o problema T10D com o aumento do custo computacional. Entre os métodos analisados, o APM Med 4, o ADP Log e o Deb, foram os únicos que não geraram o melhor resultado encontrado para o problema neste trabalho. Com o método Deb, seguido do método ADP Log, foram obtidos os maiores resultados, além desses métodos apresentarem taxa de factibilidade consideravelmente baixa como pode ser observado entre as colunas Exec. e Fac.. Estas características tornam esses métodos os menos indicados para serem integrados ao ABC G-best para a resolução do problema T10D.

A Tabela 5.27 compara o melhor resultado obtido neste trabalho para o problema T10D com outros encontrados na literatura. Observa-se que o melhor resultado encontrado com o algoritmo proposto também foi obtido no trabalho [59], além de ser o melhor resultado entre os comparados.

Tabela 5.27: Comparação do resultado obtido para o problema Treliça de 10 Barras – Caso Discreto (T10D) com outros trabalhos da literatura.

Trabalho	W	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	Avaliações
Rajeev e Krishnamoorthy [51]	5613,580	33,50	1,62	22,00	15,50	1,62	1,62	14,20	19,90	19,90	2,62	N/A
Lemonge e Barbosa [59]	5619,662	26,50	1,62	26,50	16,90	1,80	1,62	13,50	22,00	19,90	1,99	N/A
Lemonge e Barbosa [59]	5572,600	30,00	1,62	22,00	16,90	1,62	1,62	11,50	22,00	22,00	1,80	24000
Lemonge e Barbosa [59]	5490,738	33,50	1,62	22,90	14,20	1,62	1,62	7,97	22,90	22,00	1,62	90000
Ghasemi, Hinton e Wood [106]	5493,360	33,50	1,62	22,00	14,20	1,62	1,62	7,97	22,90	22,00	1,62	40000
Este estudo	5490,738	33,50	1,62	26,50	15,50	1,62	1,62	7,97	22,00	19,90	1,62	26300

5.6.3 T25C – Treliça de 25 Barras – Caso Contínuo

Este problema consiste em minimizar o peso de uma treliça espacial composta por 25 barras, ilustrada na Figura 5.12 [54].

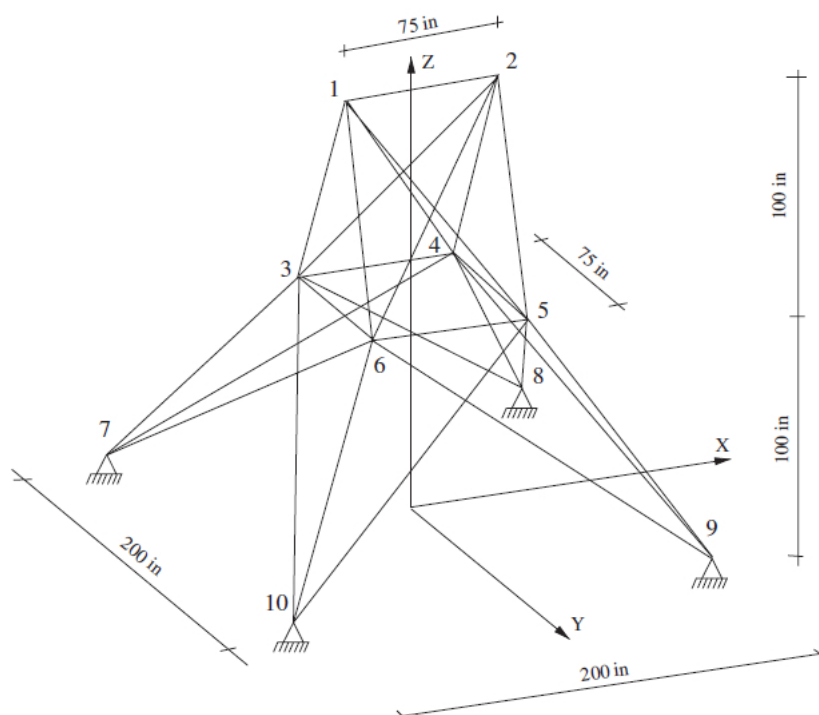


Figura 5.14: T25C – Treliça de 25 Barras – Caso Contínuo.

A treliça está sujeita a dois casos de carregamento, descritos na Tabela 5.28.

Tabela 5.28: Carregamentos da Treliça de 25 Barras – Caso Contínuo (T25C).

Carregamento	Nó	F_x	F_y	F_z
1	1	1,000	10,000	-5,000
	2	0,000	10,000	-5,000
	3	0,500	0,000	0,000
2	5	0,000	20,000	-5,000
	6	0,000	-20,000	-5,000

Neste problema supõe-se que toda a estrutura seja construída com material de módulo de elasticidade $E = 10^4 \text{ ksi}$ e que a densidade da massa seja $\rho = 0,1 \text{ lb/in}^3$. O

deslocamento em cada nó terá limite de $\pm 0,35$ in e a tensão máxima permitida em cada barra é 40 ksi.

Devido a simetria da estrutura nos eixos x e y , o problema envolve 8 variáveis de projeto A_i ($i = 1, 2, \dots, 8$), que correspondem às áreas das seções transversais dos 8 grupos de barras. Os valores aceitos para cada A_i variam entre $0,01$ in² e $5,0$ in². A distribuição das barras nos 8 grupos segue descrita na Tabela 5.29, que traz também os valores máximos permitidos de compressão nas respectivas barras.

Tabela 5.29: Agrupamento das barras da Treliza de 25 Barras – Caso Contínuo (T25C).

Área	Barras	Compressão
A_1	1	35,092
A_2	2, 3, 4 e 5	11,590
A_3	6, 7, 8 e 9	17,305
A_4	10 e 11	35,092
A_5	12 e 13	35,092
A_6	14, 15, 16 e 17	6,759
A_7	18, 19, 20 e 21	6,959
A_8	22, 23, 24 e 25	11,082

A Tabela 5.30 mostra que sob o custo computacional de 5000 avaliações da função objetivo, com o método D(C=0.5) foi obtido o maior resultado para o problema T25C. Além disto, o resultado produzido com o método D(C=0.5) tem valor consideravelmente elevado se comparado com os demais. Entre as colunas Exec. e Fac. da Tabela 5.30, observa-se que com todos os métodos foram obtidas soluções factíveis nas 25 execuções. O melhor resultado foi produzido com o método APM Med 3, que também gerou o menor valor para a métrica Média. Já para a métrica Mediana, o menor valor encontrado foi produzido com o método APM Med.

Tabela 5.30: Resultados obtidos para o problema Treliça de 25 Barras – Caso Contínuo (T25C) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	548.068	573.613	558.390	558.629	6.815	1.000	25	25	4650	11
APM Damp	546.963	596.998	551.587	554.184	9.799	1.000	25	25	4050	10
APM Med 4	546.807	586.786	556.185	558.003	8.965	1.000	25	25	4400	23
ADP Log	547.347	748.322	555.989	592.458	62.702	1.000	25	25	2550	3
Coit Mod	545.525	566.798	552.125	554.014	6.290	1.000	25	25	2850	5
APM Med	545.698	580.510	549.971	552.636	8.088	1.000	25	25	4250	5
APM Sum	545.684	573.915	551.674	552.952	5.449	1.000	25	25	4500	9
APM	545.948	571.790	550.414	552.797	6.566	1.000	25	25	4100	1
APM Worst 2	546.641	576.574	552.749	554.730	7.618	1.000	25	25	4650	8
APM Med 5	546.818	576.390	554.387	555.358	7.547	1.000	25	25	4500	6
A(k=1E7)	547.297	571.601	558.850	558.212	7.438	1.000	25	25	3950	3
APM Worst	546.781	588.449	555.349	557.406	9.734	1.000	25	25	3950	6
D(C=1.0)	573.897	827.716	677.707	693.090	67.408	1.000	25	25	4950	16
A(k=1E5)	547.104	594.976	554.053	555.630	9.503	1.000	25	25	3800	5
APM Med 3	545.504	571.510	550.790	552.479	6.589	1.000	25	25	4850	1
Deb	548.051	659.950	558.625	567.786	23.879	1.000	25	25	2200	23
A(k=1E9)	545.980	600.890	553.828	557.429	12.818	1.000	25	25	2350	12
D(C=0.5)	637.490	879.823	696.470	720.666	62.133	1.000	25	25	4950	9

Tabela 5.31: Resultados obtidos para o problema Treliça de 25 Barras – Caso Contínuo (T25C) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	545.384	546.442	546.007	545.964	0.243	1.000	25	25	300900	6
APM Damp	545.322	546.222	545.577	545.645	0.243	1.000	25	25	50550	17
APM Med 4	545.279	546.196	545.718	545.728	0.217	1.000	25	25	496800	5
ADP Log	545.219	546.319	545.469	545.578	0.282	1.000	25	25	125950	12
Coit Mod	545.436	546.795	545.858	545.986	0.405	1.000	25	25	297600	7
APM Med	545.250	546.077	545.506	545.508	0.204	1.000	25	25	370750	18
APM Sum	545.222	546.350	545.571	545.652	0.288	1.000	25	25	354950	1
APM	545.211	546.619	545.628	545.655	0.294	1.000	25	25	403450	22
APM Worst 2	545.235	546.651	545.738	545.732	0.315	1.000	25	25	210600	4
APM Med 5	545.332	546.576	545.746	545.831	0.351	1.000	25	25	360200	18
A(k=1E7)	545.200	547.154	545.835	545.927	0.420	1.000	25	25	145100	13
APM Worst	545.290	546.264	545.776	545.803	0.254	1.000	25	25	323800	9
D(C=1.0)	545.333	547.249	545.902	546.035	0.435	1.000	25	25	70100	25
A(k=1E5)	545.269	547.241	546.082	546.162	0.483	1.000	25	25	363550	11
APM Med 3	545.275	546.443	545.551	545.594	0.295	1.000	25	25	380150	12
Deb	548.051	635.472	558.625	566.807	20.316	1.000	25	25	2200	23
A(k=1E9)	545.201	548.284	545.971	546.031	0.595	1.000	25	25	357200	5
D(C=0.5)	545.400	546.844	546.035	546.060	0.374	1.000	25	25	238950	4

Na Tabela 5.31 nota-se que se aumentado para 500000 o número máximo de avaliações da função objetivo, o A(k=1E7) passa a ser o método com o qual foi obtido o melhor resultado para o problema T25C. Já o maior resultado foi produzido com o método Deb, que também apresenta os maiores valores para as métricas Mediana e Média. Grande parte dos métodos produziram como melhores resultados valores consideravelmente próximos.

A Tabel 5.32 compara o melhor resultado obtido para o problema T25C com outros encontrados na literatura. Observa-se que a diferença entre os resultados factíveis dos trabalhos comparados é praticamente ignorável, sendo que o melhor deles foi obtido no trabalho [107].

Tabela 5.32: Comparação do resultado obtido para o problema Treliça de 25 Barras – Caso Contínuo (T25C) com outros trabalhos da literatura.

Trabalho	W	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	Avaliações
Haftka e Gurdal [108]	545,220	0,010	1,987	2,991	0,010	0,012	0,683	1,679	2,664	N/A
Lee e Geem [109]	544,380 *	0,047	2,022	2,950	0,010	0,014	0,688	1,657	2,663	15000
Li et al. [110]	545,190	0,010	1,970	3,016	0,010	0,010	0,694	1,681	2,643	150000
Lamberti [107]	545,161	0,010	1,987	2,994	0,010	0,010	0,694	1,681	2,643	N/A
Sonmez [54]	545,193	0,011	1,979	3,003	0,010	0,010	0,690	1,679	2,652	400000
Este estudo	545,200	0,010	1,986	2,906	0,010	0,010	0,664	1,729	2,701	145100

* Solução viola restrições em 0,0122 [54]

5.6.4 T72C – Treliça de 72 Barras – Caso Contínuo

Este problema consiste da minimização do peso de uma treliça composta por 72 barras, ilustrada na Figura 5.15 [7, 4, 92, 54].

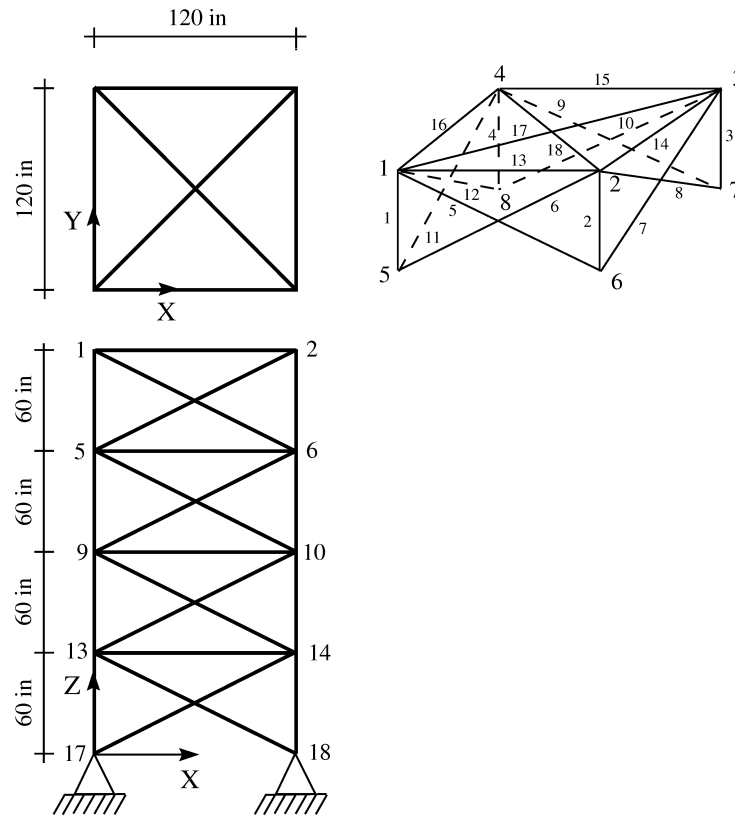


Figura 5.15: T72C – Treliça de 72 Barras – Caso Contínuo.

Dois casos de carregamento, descritos na Tabela 5.33, são definidos para esta estrutura.

Tabela 5.33: Carregamentos da Treliça de 72 Barras – Caso Contínuo (T72C).

Carregamento	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

No problema estão presentes 16 variáveis de projeto que correspondem às áreas A_i ($i = 1, 2, \dots, 16$) das seções transversais dos 16 grupos de barras. As áreas poderão variar

de $0,1 \text{ in}^2$ a $5,0 \text{ in}^2$. As 72 barras foram organizadas em 16 grupos, descritos na Tabela 5.34.

Tabela 5.34: Agrupamento das barras da Treliça de 72 Barras – Caso Contínuo (T72C).

Área	Barras
A_1	1, 2, 3 e 4
A_2	5, 6, 7, 8, 9, 10, 11 e 12
A_3	13, 14, 15 e 16
A_4	17 e 18
A_5	19, 20, 21 e 22
A_6	23, 24, 25, 26, 27, 28, 29 e 30
A_7	31, 32, 33 e 34
A_8	35 e 36
A_9	37, 38, 39 e 40
A_{10}	41, 42, 43, 44, 45, 46, 47 e 48
A_{11}	49, 50, 51 e 52
A_{12}	53 e 54
A_{13}	55, 56, 57 e 58
A_{14}	59, 60, 61, 62, 63, 64, 65 e 66
A_{15}	67, 68, 69 e 70
A_{16}	71 e 72

As restrições dizem respeito aos deslocamentos máximos aceites de $0,25 \text{ in}$ ao longo das direções x e y nos 16 nós. A estrutura também estará sujeita a restrições de tensão máxima permitida em cada barra, que terá limite de $\pm 25 \text{ ksi}$. Considera-se ainda a densidade do material $\rho = 0,1 \text{ lb/in}^3$ e o módulo de elasticidade $E = 10^4 \text{ ksi}$.

A Tabela 5.35 mostra que os métodos APM Med, D(C=1.0), APM Med 3 e D(C=0.5) foram os quatro com os quais obteve-se os melhores resultados para o problema T72C. Entre os quatro o D(C=1.0) se destaca, já que o melhor resultado apresentado na coluna Melhor está relacionado a este método. Os menores valores para as métricas Mediana e Média também foram produzidos com o método D(C=1.0), como pode ser observado na tabela. Entre as colunas Exec. e Fac. nota-se que com todos os métodos foram obtidas soluções factíveis nas 25 execuções.

Tabela 5.35: Resultados obtidos para o problema Treliça de 72 Barras – Caso Contínuo (T72C) com cada método de penalização combinado ao ABC G-best, adotando 5000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	415.578	641.778	504.604	509.799	57.397	1.000	25	25	4850	25
APM Damp	424.333	572.858	476.572	480.373	37.964	1.000	25	25	2000	23
APM Med 4	409.328	556.933	502.122	492.707	42.175	1.000	25	25	4700	6
ADP Log	429.881	589.849	489.670	495.041	39.978	1.000	25	25	2700	15
Coit Mod	407.818	489.262	453.346	456.511	20.867	1.000	25	25	4800	16
APM Med	397.651	550.399	464.582	466.660	41.594	1.000	25	25	4000	5
APM Sum	416.964	494.565	453.385	451.949	21.020	1.000	25	25	4700	14
APM	435.932	563.174	488.153	491.349	36.575	1.000	25	25	5000	9
APM Worst 2	426.719	565.053	482.810	485.752	41.225	1.000	25	25	4850	5
APM Med 5	428.000	571.174	502.122	500.280	35.957	1.000	25	25	4850	8
A(k=1E7)	410.951	544.280	484.284	478.364	32.332	1.000	25	25	5000	12
APM Worst	417.034	551.365	493.911	485.147	37.268	1.000	25	25	4600	21
D(C=1.0)	385.629	412.329	393.615	394.673	7.484	1.000	25	25	2900	14
A(k=1E5)	408.155	484.845	446.666	445.398	22.610	1.000	25	25	4950	12
APM Med 3	390.888	543.945	476.461	471.841	39.707	1.000	25	25	4850	14
Deb	437.773	683.791	493.180	513.660	66.083	1.000	25	25	2150	11
A(k=1E9)	440.385	628.147	498.965	505.971	43.386	1.000	25	25	4050	7
D(C=0.5)	386.813	567.290	415.017	437.237	48.722	1.000	25	25	3900	23

Tabela 5.36: Resultados obtidos para o problema Treliça de 72 Barras – Caso Contínuo (T72C) com cada método de penalização combinado ao ABC G-best, adotando 500000 avaliações da função objetivo. Na coluna Melhor, que lista os melhores resultados obtidos, o melhor deles aparece destacado em negrito. As colunas como Mediana, Média, DP (Desvio Padrão) são preenchidas com valores calculados entre as funções objetivo das soluções encontradas nas 25 execuções. A coluna TF (Taxa de Factibilidade) traz a porção das 25 execuções em que foram obtidas soluções factíveis. As colunas ME (Mínimo de Execuções) e MA (Mínimo de Avaliações) identificam respectivamente os números mínimos de execuções e avaliações da função objetivo necessários para que fosse obtido o resultado da coluna Melhor.

	Melhor	Pior	Mediana	Média	DP	TF	Exec.	Fac.	MA	ME
APM Mono	393.781	441.632	414.909	414.278	12.100	1.000	25	25	266500	18
APM Damp	388.170	437.508	414.604	414.331	11.084	1.000	25	25	175550	10
APM Med 4	391.589	427.401	411.472	412.807	10.745	1.000	25	25	456450	2
ADP Log	378.898	412.668	384.764	388.368	9.304	1.000	25	25	100800	7
Coit Mod	385.421	399.883	391.343	391.405	3.824	1.000	25	25	116150	15
APM Med	379.077	407.786	382.582	384.005	5.236	1.000	25	25	183250	5
APM Sum	381.439	394.361	387.971	387.887	3.365	1.000	25	25	498500	4
APM	392.415	429.269	403.644	404.306	9.405	1.000	25	25	225050	8
APM Worst 2	391.153	437.403	410.520	411.613	11.679	1.000	25	25	365800	22
APM Med 5	391.387	437.611	414.121	413.836	13.303	1.000	25	25	455000	1
A(k=1E7)	390.439	458.267	413.298	419.304	16.677	1.000	25	25	499300	15
APM Worst	384.908	444.907	407.729	406.710	15.512	1.000	25	25	62150	12
D(C=1.0)	383.820	396.211	386.832	387.886	3.176	1.000	25	25	23250	20
A(k=1E5)	383.231	390.042	385.607	385.972	1.942	1.000	25	25	492650	7
APM Med 3	380.230	394.387	386.044	385.929	3.557	1.000	25	25	465250	6
Deb	437.223	599.107	488.426	497.682	47.285	1.000	25	25	20150	15
A(k=1E9)	396.009	438.802	420.249	419.882	12.766	1.000	25	25	237850	20
D(C=0.5)	382.705	401.900	389.663	390.049	4.536	1.000	25	25	33900	4

A Tabela 5.36 mostra que independente do método de penalização, o resultado para o problema T72C melhora com o aumento do número máximo de avaliações da função objetivo para 500000. Observa-se na coluna MA da Tabela 5.36, que com muitos dos métodos analisados, foi necessário um número menor que 500000 avaliações da função objetivo para que fossem gerados os resultados apresentados na coluna Melhor. O melhor resultado encontrado para o problema T72C foi obtido com o método ADP Log. Em contrapartida o maior resultado apresentado na coluna Melhor foi obtido com o método Deb. Entre as Tabelas 5.35 e 5.36, os resultados na coluna Melhor referentes ao método Deb apresentam muito pouca redução. Já os menores valores para as métricas Mediana e Média foram obtidos com o método APM Med, que também apresenta como melhor resultado valor consideravelmente próximo daquele obtido com o método ADP Log.

As Figuras 5.16 e 5.17 apresentam gráficos gerados a partir dos resultados obtidos com a aplicação da técnica Escalonamento Multidimensional (MDS). A Figura 5.16 mostra que entre os métodos de penalização existe distinção de desempenho e os mesmos são agrupados com base na similaridade entre os resultados apresentados na Tabela 5.35. A Figura 5.17 mostra que mesmo aumentando o custo computacional máximo, fica mantida a distinção de desempenho entre os métodos de penalização. Os métodos são distribuídos basicamente em três grupos distintos, sendo que um deles é composto apenas pelo método Deb. O motivo deste método aparecer separado de todos os outros fica claro se comparados os resultados apresentados na coluna Melhor da Tabela 5.36, já que o maior valor está relacionado ao método Deb, além da grande diferença quando comparado com os demais.

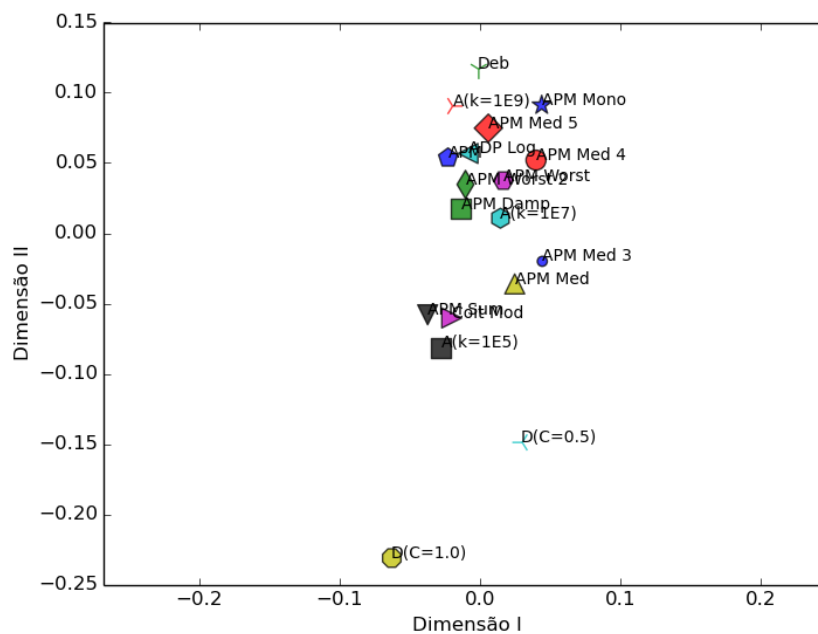


Figura 5.16: Resultado da técnica MDS para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 5000 avaliações da função objetivo.

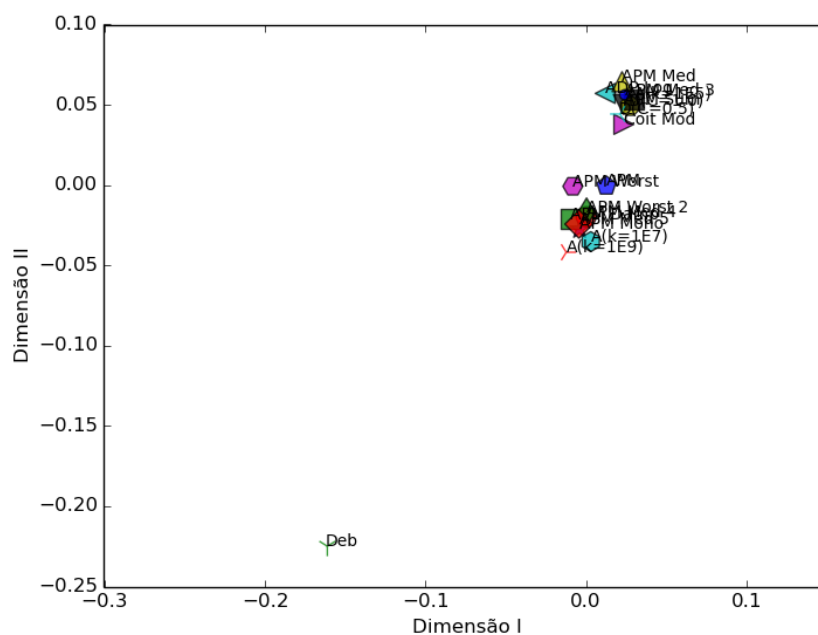


Figura 5.17: Resultado da técnica MDS para o problema Treliça de 72 Barras – Caso Contínuo (T72C) adotando 500000 avaliações da função objetivo.

A Tabela 5.37 traz o melhor resultado obtido para o problema T72C comparado a outros encontrados na literatura. O trabalho [54], único em que foi obtido resultado melhor que aquele aqui encontrado, adotou uma variação do algoritmo ABC como ferramenta de busca.

Tabela 5.37: Comparação do resultado obtido para o problema Treliça de 72 Barras – Caso Contínuo (T72C) com outros trabalhos da literatura.

Trabalho	W	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	Avaliações
Venkayya [111]	381,200	0,161	0,557	0,377	0,506	0,611	0,532	0,100	0,100	1,246	0,524	0,100	0,100	1,818	0,524	0,100	0,100	N/A
Erbatur et al. [99]	385,760	0,155	0,535	0,480	0,520	0,460	0,530	0,120	0,165	1,155	0,585	0,100	0,100	1,755	0,505	0,105	0,155	N/A
Lemonge et al. [59]	387,036	0,155	0,545	0,275	0,519	0,604	0,666	0,102	0,130	1,200	0,474	0,101	0,109	1,953	0,517	0,100	0,101	35000
Sonmez [54]	363,8392	0,168	0,535	0,444	0,580	0,521	0,518	0,010	0,105	1,297	0,519	0,010	0,010	1,891	0,517	0,010	0,010	800000
Este estudo	378,898	0,147	0,529	0,338	0,758	0,641	0,523	0,100	0,100	1,216	0,537	0,100	0,100	1,816	0,488	0,100	0,100	100800

5.6.5 Discussão sobre os Resultados dos Problemas de Otimização Estrutural

As Tabelas 5.38 e 5.39 trazem os valores de função objetivo das melhores soluções encontradas para cada um dos problemas de Otimização Estrutural, com cada um dos métodos de penalização analisados neste trabalho. Observa-se que com os métodos D(C=0.5), D(C=1.0), Deb, APM Med e APM Med 4 foi necessário um número maior de iterações do ABC G-best para resolver o problema T10D. Entre eles, o método Deb foi o que precisou do maior número de iterações. Nesta classe de problemas, o método Deb é o menos indicado para ser combinado ao algoritmo ABC G-best. Entre os resultados de cada problema, aqueles que apresentam maior distinção, com os métodos A(k=1E7), ADP Log, APM Sum, APM Med 3, D(C=1.0) e APM Med foram obtidos os melhores resultados.

Tabela 5.38: Melhores resultados obtidos para os problemas de Otimização Estrutural adotando 5000 avaliações da função objetivo. Entre as colunas da tabela T10D = Treliça de 10 Barras – Caso Discreto, T10C = Treliça de 10 Barras – Caso Contínuo, T72C = Treliça de 72 Barras – Caso Contínuo, T25C = Treliça de 25 Barras – Caso Contínuo.

	T10D	T10C	T72C	T25C
APM Mono	5549.204	5103.011	415.578	548.068
APM Damp	5608.160	5147.591	424.333	546.963
APM Med 4	–	5192.006	409.328	546.807
ADP Log	7619.681	5102.058	429.881	547.347
Coit Mod	5610.341	5132.942	407.818	545.525
APM Med	–	5091.848	397.651	545.698
APM Sum	5538.341	5106.494	416.964	545.684
APM	6217.369	5117.249	435.932	545.948
APM Worst 2	5590.770	5108.477	426.719	546.641
APM Med 5	5616.380	5173.263	428.000	546.818
A(k=1E7)	5505.138	5128.497	410.951	547.297
APM Worst	5728.500	5117.135	417.034	546.781
D(C=1.0)	–	7036.117	385.629	573.897
A(k=1E5)	5532.121	5093.080	408.155	547.104
APM Med 3	5873.860	5096.019	390.888	545.504
Deb	–	5300.181	437.773	548.051
A(k=1E9)	5548.457	5116.547	440.385	545.980
D(C=0.5)	–	7036.117	386.813	637.490

Tabela 5.39: Melhores resultados obtidos para os problemas de Otimização Estrutural adotando 500000 avaliações da função objetivo. Entre as colunas da tabela T10D = Treliça de 10 Barras – Caso Discreto, T10C = Treliça de 10 Barras – Caso Contínuo, T72C = Treliça de 72 Barras – Caso Contínuo, T25C = Treliça de 25 Barras – Caso Contínuo.

	T10D	T10C	T72C	T25C
APM Mono	5490.738	5084.582	393.781	545.384
APM Damp	5490.738	5085.479	388.170	545.322
APM Med 4	5491.717	5071.531	391.589	545.279
ADP Log	5814.914	5073.019	378.898	545.219
Coit Mod	5490.738	5075.115	385.421	545.436
APM Med	5490.738	5080.283	379.077	545.250
APM Sum	5490.738	5067.223	381.439	545.222
APM	5490.738	5078.896	392.415	545.211
APM Worst 2	5490.738	5079.361	391.153	545.235
APM Med 5	5490.738	5077.647	391.387	545.332
A(k=1E7)	5490.738	5082.401	390.439	545.200
APM Worst	5490.738	5074.957	384.908	545.290
D(C=1.0)	5490.738	5083.283	383.820	545.333
A(k=1E5)	5490.738	5067.438	383.231	545.269
APM Med 3	5490.738	5079.352	380.230	545.275
Deb	6561.310	5300.181	437.223	548.051
A(k=1E9)	5490.738	5085.854	396.009	545.201
D(C=0.5)	5490.738	5071.378	382.705	545.400

As Figuras 5.18, 5.19, 5.20 e 5.21 apresentam os resultados obtidos com a técnica Perfis de Desempenho, porém destacando as regiões dos gráficos em que os menores valores de τ tal que $\rho(\tau) = 1$ são identificados. Observa-se que independente da métrica usada para $t_{p,s}$ ou o número máximo de avaliações, os métodos A(k=1E5), Coit Mod, APM Sum, APM Med 3 e APM Med aparecem com frequência entre os mais robustos. Tendo por base a diferença entre os valores de τ e a frequência com que aparecem nos gráficos, os métodos **A(k=1E5)**, **APM Med 3** e **APM Sum** são vistos como os mais robustos para serem integrados ao ABC G-best classe para a resolução dos problemas desta classe.

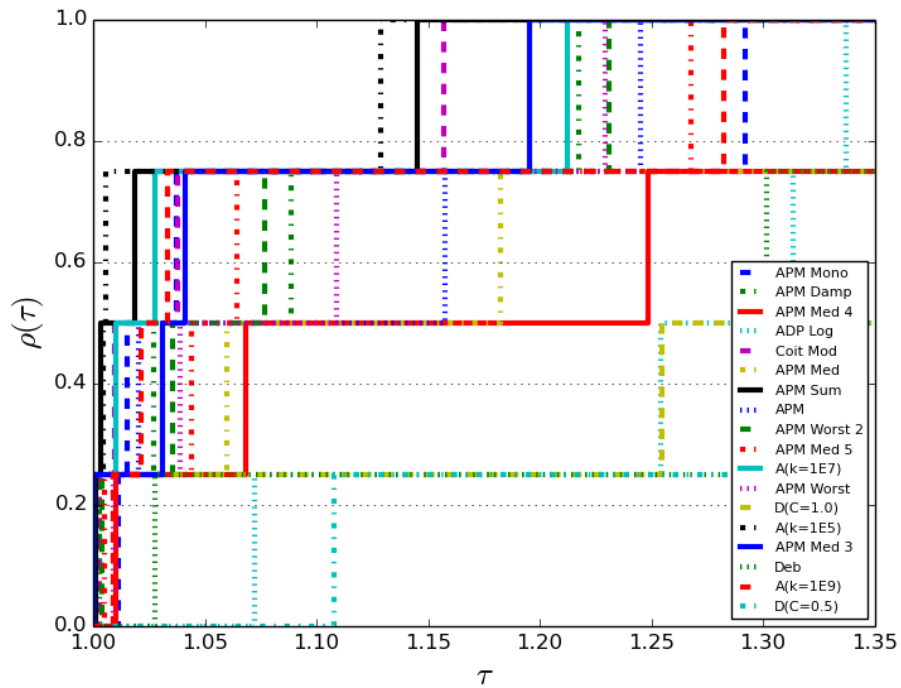


Figura 5.18: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

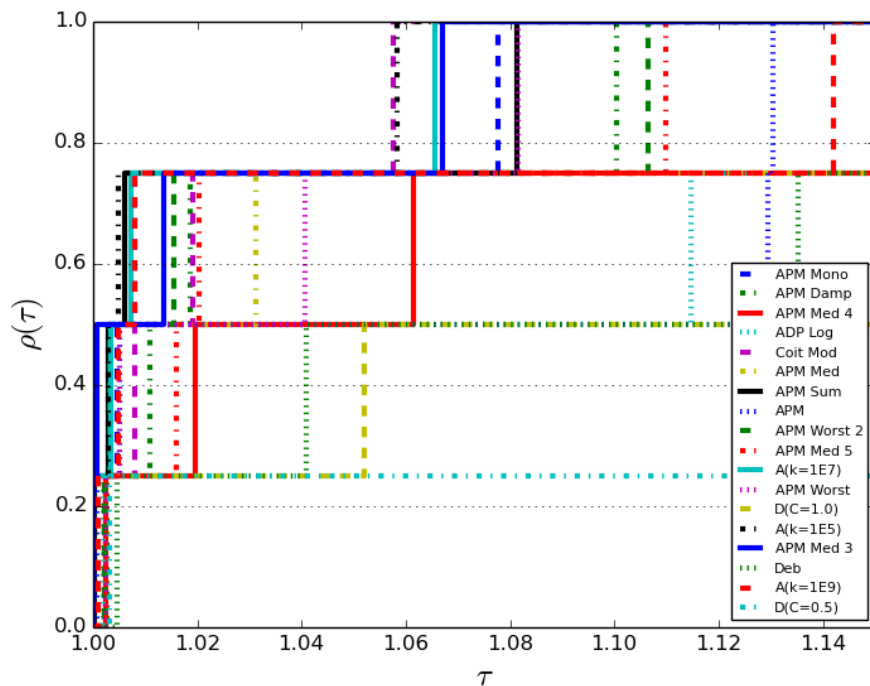


Figura 5.19: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

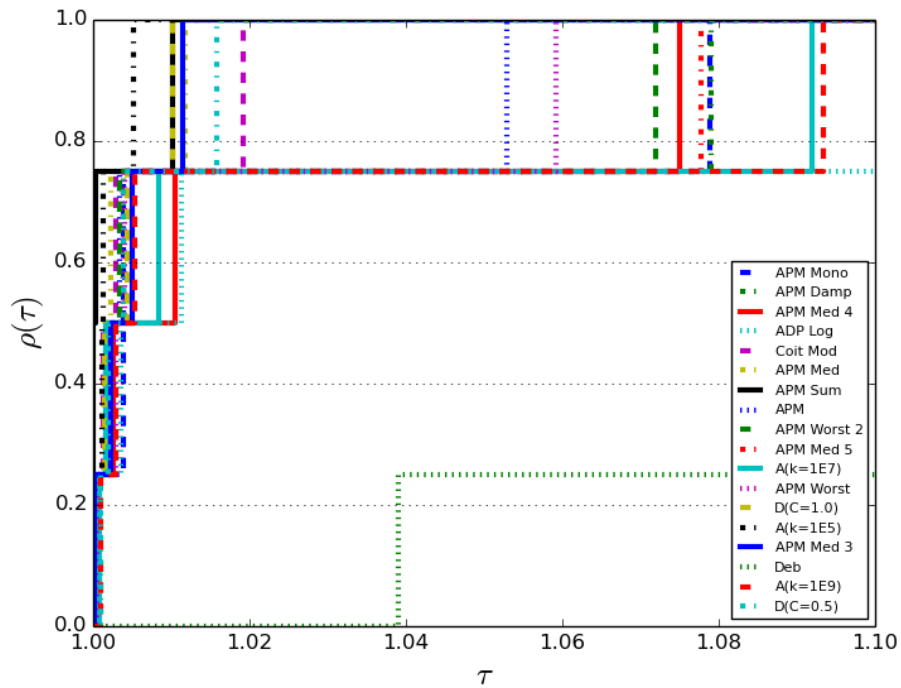


Figura 5.20: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

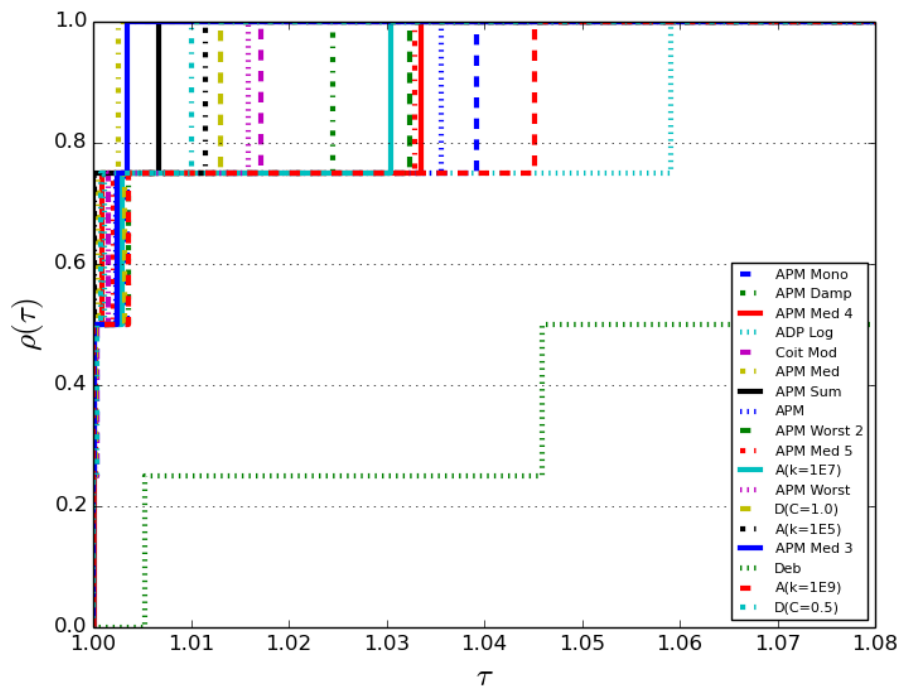


Figura 5.21: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para os problemas de Otimização Estrutural adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

5.7 Experimento 03 – *G-Suite*

Este experimento consiste da resolução de um conjunto composto por 24 problemas de otimização com restrições proposto em [112]. Todos os 24 problemas do conjunto foram impostos ao algoritmo proposto neste trabalho. O conjunto de problemas é descrito na Seção 5.7.1. Discussões sobre os resultados obtidos para o conjunto *G-Suite* são encontradas na Seção 5.7.2.

5.7.1 *G-Suite*

O *G-Suite* é um conjunto composto por 24 problemas de otimização com restrições, proposto em [112]. O conjunto é bastante usado na literatura quando se trata da avaliação de algoritmos estocásticos aplicados na resolução de problemas com restrições.

As definições detalhadas dos 24 problemas do conjunto, identificados como g01, g02, ..., g24, podem ser encontradas em [112]. A Tabela 5.40 apresenta algumas características dos problemas. Entre as colunas da Tabela 5.40 estão: $f(\mathbf{x}^*)$ que é a melhor solução conhecida para o problema, \mathbf{n} que é o número de variáveis de projeto, ρ que é a taxa estimada entre a região factível e o espaço de busca ($\rho = |F|/|S|$), **LD** que é o número de restrições de desigualdade lineares, **ND** que é o número de restrições de desigualdade não-lineares, **LI** que é o número de restrições de igualdade lineares, **NI** que é o número de restrições de igualdade não-lineares e **a** que é o número de restrições ativas. Deve-se destacar que a melhor solução conhecida para g20 é levemente infactível.

Tabela 5.40: Detalhes dos 24 problemas do conjunto *G-Suite*.

Prob.	$f(\mathbf{x}^*)$	n	Tipo Função	ρ	LD	ND	LI	NI	a
g01	-15,0000000000	13	Quadrática	0,0111%	9	0	0	0	6
g02	-0,8036191042	20	Não-linear	99,9971%	0	2	0	0	1
g03	-1,0005001000	10	Polinomial	0,0000%	0	0	0	1	1
g04	-30665,5386717834	5	Quadrática	52,1230%	0	6	0	0	2
g05	5126,4967140071	4	Cúbica	0,0000%	2	0	0	3	3
g06	-6961,8138755802	2	Cúbica	0,0066%	0	2	0	0	2
g07	24,3062090681	10	Quadrática	0,0003%	3	5	0	0	6
g08	-0,0958250415	2	Não-linear	0,8560%	0	2	0	0	0
g09	680,6300573745	7	Polinomial	0,5121%	0	4	0	0	2
g10	7049,2480205286	8	Linear	0,0010%	3	3	0	0	6
g11	0,7499000000	2	Quadrática	0,0000%	0	0	0	1	1
g12	-1,0000000000	3	Quadrática	4,7713%	0	1	0	0	0
g13	0,0539415140	5	Não-linear	0,0000%	0	0	0	3	3
g14	-47,7648884595	10	Não-linear	0,0000%	0	0	3	0	3
g15	961,7150222899	3	Quadrática	0,0000%	0	0	1	1	2
g16	-1,9051552586	5	Não-linear	0,0204%	4	34	0	0	4
g17	8853,5396748064	6	Não-linear	0,0000%	0	0	0	4	4
g18	-0,8660254038	9	Quadrática	0,0000%	0	13	0	0	6
g19	32,6555929502	15	Não-linear	33,4761%	0	5	0	0	0
g20	0,2049794002	24	Linear	0,0000%	0	6	2	12	16
g21	193,7245100700	7	Linear	0,0000%	0	1	0	5	6
g22	236,4309755040	22	Linear	0,0000%	0	1	8	11	19
g23	-400,0551000000	9	Linear	0,0000%	0	2	3	1	6
g24	-5,5080132716	2	Linear	79,6556%	0	2	0	0	2

5.7.2 *Discussão sobre os resultados do G-Suite*

Entre os 24 problemas do conjunto *G-Suite*, 18 (75%) foram resolvidos com o algoritmo proposto neste trabalho. Independente do número de avaliações da função objetivo necessárias, para 10 dos problemas foi obtido o valor mínimo conhecido na literatura [112]. Para os outros 8 problemas resolvidos, foram obtidos resultados próximos daqueles conhecidos, os resultados para os problemas g10, g15, g19 e g23 apresentaram as maiores diferenças.

Entre os problemas não resolvidos, que foram g05, g13, g17, g20, g21 e g22, os valores na coluna ρ da Tabela 5.40 são todos 0,0000%. Como pode ser consultado em [112], alguns dos problemas não resolvidos possuem funções complexas e/ou números altos de variáveis de projeto ou restrições de igualdade, tornando-os de difícil solução.

Entre as Tabelas 5.41 e 5.44 são apresentados os valores de função objetivo produzidos pelas melhores soluções encontradas para cada um dos problemas do conjunto *G-Suite*, com cada um dos métodos de penalização analisados neste trabalho. As Tabelas 5.41 e 5.42 mostram que para alguns dos problemas como g04, g06, g08, g11, g12, g16, g18 e g24 foram obtidos ótimos resultados sob o custo computacional de 5000 avaliações da função objetivo. Já para outros problemas como g01, g03, g07, g10, g14, g15, g19 e g23 foi necessário um número maior de iterações do ABC G-best para que fosse obtido o valor mínimo conhecido ou próximo dele, como pode ser observado entre as Tabelas 5.41 e 5.44. A Tabela 5.44 mostra que para o problema g14, sua solução foi obtida com o menor número de métodos de penalização entre aqueles analisados no trabalho. Na sequência, a Tabela 5.45 traz as melhores soluções encontradas para cada um dos problemas resolvidos.

Como as soluções de cada problema do conjunto *G-Suite* são conhecidas e apresentadas em [112], optou-se neste trabalho por não ser feita comparação dos resultados obtidos com outros da literatura. Avalia-se o algoritmo aqui proposto pela capacidade de solucionar os problemas do grupo.

Tabela 5.41: Melhores resultados obtidos para os problemas do *G-Suite* (g01 a g12) adotando 5000 avaliações da função objetivo.

	g01	g02	g03	g04	g06	g07	g08	g09	g10	g11	g12
APM Mono	-14.986	-0.690	-2.777e-04	-3.057e+04	-6809.220	28.128	-0.096	681.548	7597.172	0.789	-1.000
APM Damp	–	-0.690	-0.005	-3.060e+04	-6815.891	26.249	-0.095	682.076	1.023e+04	0.950	-1.000
APM Med 4	–	-0.690	–	-3.066e+04	-1847.545	27.413	-0.095	684.402	–	0.821	-1.000
ADP Log	-14.985	-0.690	-0.003	-3.048e+04	-6777.839	33.528	-0.096	681.693	9645.256	1.000	-1.000
Coit Mod	-14.997	-0.630	0.000e+00	-3.066e+04	-6926.002	28.929	-0.096	681.110	8892.691	1.000	-1.000
APM Med	–	-0.690	–	-3.057e+04	-6416.491	26.433	-0.091	681.979	–	0.998	-1.000
APM Sum	-14.585	-0.690	0.000e+00	-3.052e+04	-6933.605	27.439	-0.096	682.264	7480.229	0.813	-1.000
APM	–	-0.690	–	-3.052e+04	-5761.544	27.042	-0.096	683.124	–	0.839	-1.000
APM Worst 2	–	-0.690	–	-3.060e+04	-6706.677	26.435	-0.094	683.109	–	0.952	-1.000
APM Med 5	–	-0.690	–	-3.061e+04	-5586.731	26.956	-0.095	682.880	–	0.866	-1.000
A(k=1E7)	-14.998	-0.690	-0.002	-3.051e+04	-6940.219	27.354	-0.096	684.362	7478.236	1.000	-1.000
APM Worst	–	-0.690	–	-3.049e+04	-6918.472	26.714	-0.096	684.186	–	0.774	-1.000
D(C=1.0)	-14.646	-0.678	–	-3.065e+04	-6765.602	28.827	-0.096	684.194	9165.888	1.000	-1.000
A(k=1E5)	-14.996	-0.690	-0.002	-3.065e+04	-6909.021	25.064	-0.096	684.165	7265.220	0.805	-1.000
APM Med 3	–	-0.690	–	-3.062e+04	-6733.399	27.914	-0.096	683.660	–	0.839	-1.000
Deb	-14.995	-0.690	-1.858e-05	-3.067e+04	-6961.736	230.021	-0.096	1219.935	–	–	-1.000
A(k=1E9)	-14.993	-0.690	-0.001	-3.057e+04	-6940.219	26.264	-0.096	684.312	7214.911	0.882	-1.000
D(C=0.5)	-14.928	-0.655	–	-3.065e+04	-6770.051	31.544	-0.096	684.036	9473.571	1.000	-1.000

Tabela 5.42: Melhores resultados obtidos para os problemas do *G-Suite* (g13 a g24) adotando 5000 avaliações da função objetivo.

	g14	g15	g16	g18	g19	g23	g24
APM Mono	–	–	-1.707	-0.857	99.360	–	-5.506
APM Damp	–	–	-1.862	-0.804	97.913	–	-5.506
APM Med 4	–	–	-1.746	-0.771	256.674	–	-5.508
ADP Log	–	–	-1.851	-0.861	62.429	–	-5.495
Coit Mod	–	–	-1.851	-0.857	81.449	–	-5.508
APM Med	–	–	-1.893	-0.620	76.900	–	-5.508
APM Sum	–	–	-1.852	-0.843	107.287	–	-5.507
APM	–	–	-1.878	-0.829	93.262	–	-5.508
APM Worst 2	–	–	-1.854	-0.659	205.561	–	-5.504
APM Med 5	–	–	-1.875	-0.778	179.688	–	-5.508
A(k=1E7)	–	–	-1.769	-0.860	72.493	900.000	-5.507
APM Worst	–	–	-1.900	-0.653	141.788	–	-5.505
D(C=1.0)	–	–	-1.834	-0.745	98.741	–	-5.508
A(k=1E5)	–	–	-1.860	-0.859	72.493	900.000	-5.507
APM Med 3	–	–	-1.831	-0.576	141.080	–	-5.506
Deb	–	–	-1.834	-0.852	108.603	–	-5.506
A(k=1E9)	–	–	-1.806	-0.860	72.493	900.000	-5.507
D(C=0.5)	–	–	-1.824	-0.805	151.021	–	-5.508

Tabela 5.43: Melhores resultados obtidos para os problemas do *G-Suite* (g01 a g12) adotando 500000 avaliações da função objetivo.

	g01	g02	g03	g04	g06	g07	g08	g09	g10	g11	g12
APM Mono	-15.000	-0.693	-0.096	-3.066e+04	-6961.814	25.356	-0.096	681.136	7141.519	0.750	-1.000
APM Damp	-14.915	-0.693	-0.698	-3.066e+04	-6961.814	24.310	-0.096	680.809	9598.968	0.750	-1.000
APM Med 4	-14.615	-0.693	-0.966	-3.067e+04	-6319.304	24.366	-0.096	680.894	1.006e+04	0.750	-1.000
ADP Log	-15.000	-0.693	-0.210	-3.067e+04	-6777.839	28.599	-0.096	681.688	7402.362	1.000	-1.000
Coit Mod	-15.000	-0.724	-0.116	-3.067e+04	-6961.554	25.432	-0.096	680.643	7092.492	0.750	-1.000
APM Med	-13.286	-0.693	-0.701	-3.066e+04	-6672.248	24.458	-0.096	680.666	1.423e+04	0.750	-1.000
APM Sum	-15.000	-0.693	-0.292	-3.066e+04	-6961.814	24.323	-0.096	681.086	7173.619	0.751	-1.000
APM	-14.874	-0.693	-0.823	-3.066e+04	-6125.339	24.328	-0.096	680.666	9811.663	0.750	-1.000
APM Worst 2	-14.892	-0.693	–	-3.066e+04	-6961.814	24.318	-0.096	680.829	9019.097	0.750	-1.000
APM Med 5	-14.884	-0.693	-0.894	-3.066e+04	-5943.486	24.396	-0.096	680.912	9649.190	0.750	-1.000
A(k=1E7)	-15.000	-0.727	-0.626	-3.066e+04	-6961.814	24.500	-0.096	681.078	7160.846	0.750	-1.000
APM Worst	-14.920	-0.693	-0.774	-3.066e+04	-6961.814	24.370	-0.096	680.694	7742.619	0.750	-1.000
D(C=1.0)	-15.000	-0.700	-0.210	-3.066e+04	-6961.814	25.191	-0.096	681.079	7158.719	0.750	-1.000
A(k=1E5)	-15.000	-0.712	-0.348	-3.067e+04	-6961.809	24.637	-0.096	681.027	7175.996	0.750	-1.000
APM Med 3	-13.836	-0.693	-0.823	-3.066e+04	-6961.814	24.530	-0.096	680.741	–	0.750	-1.000
Deb	-15.000	-0.734	-0.058	-3.067e+04	-6961.814	154.124	-0.096	767.538	1.120e+04	–	-1.000
A(k=1E9)	-15.000	-0.693	-0.071	-3.066e+04	-6961.814	24.701	-0.096	680.887	7214.549	0.750	-1.000
D(C=0.5)	-15.000	-0.709	-0.323	-3.066e+04	-6961.814	24.974	-0.096	681.102	7104.687	0.750	-1.000

Tabela 5.44: Melhores resultados obtidos para os problemas do *G-Suite* (g13 a g24) adotando 500000 avaliações da função objetivo.

	g14	g15	g16	g18	g19	g23	g24
APM Mono	-47.369	967.521	-1.887	-0.865	76.476	0.000e+00	-5.508
APM Damp	-	967.521	-1.905	-0.848	50.874	0.000e+00	-5.508
APM Med 4	-	967.521	-1.746	-0.771	63.238	0.000e+00	-5.508
ADP Log	-47.369	967.521	-1.896	-0.864	35.630	0.000e+00	-5.505
Coit Mod	-47.369	-	-1.905	-0.866	37.268	0.000e+00	-5.508
APM Med	-	967.521	-1.904	-0.620	33.311	0.000e+00	-5.508
APM Sum	-47.559	967.521	-1.903	-0.864	61.893	0.000e+00	-5.508
APM	-	967.521	-1.905	-0.836	38.918	0.000e+00	-5.508
APM Worst 2	-	967.521	-1.905	-0.659	55.970	-	-5.508
APM Med 5	-	967.521	-1.902	-0.859	70.493	0.000e+00	-5.508
A(k=1E7)	-38.188	967.521	-1.902	-0.866	72.493	0.000e+00	-5.508
APM Worst	-	967.521	-1.905	-0.708	49.157	0.000e+00	-5.508
D(C=1.0)	-44.935	967.521	-1.902	-0.866	54.892	0.000e+00	-5.508
A(k=1E5)	-	967.521	-1.900	-0.866	72.493	0.000e+00	-5.508
APM Med 3	-	967.521	-1.858	-0.576	34.558	0.000e+00	-5.508
Deb	-43.720	-	-1.898	-0.866	88.807	0.000e+00	-5.508
A(k=1E9)	-44.045	967.521	-1.899	-0.866	72.493	900.000	-5.508
D(C=0.5)	-44.935	967.521	-1.903	-0.866	69.233	0.000e+00	-5.508

Tabela 5.45: Melhores soluções encontradas para os problemas do conjunto *G-Suite*.

Problema	Solução (\mathbf{x})	$f(\mathbf{x})$
g01	$\mathbf{x} = (1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 1, 000; 3, 000; 3, 000; 3, 000; 1, 000)$	-15, 000
g02	$\mathbf{x} = (3, 117; 3, 091; 3, 082; 3, 041; 3, 033; 2, 998; 2, 978; 0, 610; 2, 914; 2, 991; 0, 154; 1, 625; 0, 305; 0, 372; 0, 265; 0, 509; 0, 951; 0, 163; 0, 233; 0, 416)$	-0, 734
g03	$\mathbf{x} = (0, 320; 0, 296; 0, 376; 0, 424; 0, 334; 0, 290; 0, 345; 0, 287; 0, 256; 0, 165)$	-0, 966
g04	$\mathbf{x} = (78, 000; 33, 000; 29, 998; 45, 000; 36, 768)$	$-3, 067e + 04$
g06	$\mathbf{x} = (14, 095; 0, 843)$	-6961, 814
g07	$\mathbf{x} = (2, 278; 2, 096; 8, 712; 5, 045; 0, 936; 1, 460; 1, 530; 10, 000; 8, 496; 8, 224)$	24, 310
g08	$\mathbf{x} = (1, 228; 4, 245)$	-0, 096
g09	$\mathbf{x} = (2, 315; 1, 943; -0, 389; 4, 388; -0, 630; 1, 104; 1, 5955)$	680, 643
g10	$\mathbf{x} = (683, 253; 1000, 000; 5591, 771; 188, 899; 278, 450; 209, 202; 309, 335; 377, 564)$	7092, 492
g11	$\mathbf{x} = (0, 750; 0, 562)$	0, 750
g12	$\mathbf{x} = (5, 000; 5, 000; 5, 000)$	-1, 000
g14	$\mathbf{x} = (1e - 10; 1e - 10; 0, 768; 0, 305; 1e - 10; 0, 463; 0, 232; 1e - 10; 1e - 10; 1e - 10)$	-47, 559
g15	$\mathbf{x} = (4, 846; 1, 231; 0, 000e + 00)$	967, 521
g16	$\mathbf{x} = (705, 237; 68, 603; 102, 897; 282, 317; 37, 5981)$	-1, 905
g18	$\mathbf{x} = (0, 500; 0, 867; -0, 498; 0, 867; 0, 502; 0, 865; -0, 501; 0, 865; 0, 000e + 00)$	-0, 866
g19	$\mathbf{x} = (0, 000; 0, 000; 0, 000; 0, 000; 3, 995; 7, 176; 0, 000; 0, 000; 0, 000; 0, 000; 0, 693; 0, 000; 0, 607; 0, 056; 0, 465)$	33, 311
g23	$\mathbf{x} = (0, 000; 0, 000; 0, 000; 0, 000; 0, 000; 0, 000; 0, 000; 0, 000; 0, 000; 0, 030)$	$0, 000e + 00$
g24	$\mathbf{x} = (2, 330; 3, 178)$	-5.508

As Figuras 5.22 e 5.23 apresentam os resultados da técnica Perfis de Desempenho, porém destacando as regiões dos gráficos em que são detectados os menores valores de τ tal que $\rho(\tau) = 1$. Os gráficos foram gerados com base nos resultados da técnica quando aplicada sobre os dados da execução do algoritmo com o custo computacional máximo de 50000 avaliações da função objetivo. Este foi o custo máximo necessário para que os 75% dos problemas aqui resolvidos fossem solucionados, independente dos resultados obtidos. As Tabelas 5.41 e 5.42 mostram que sob o custo máximo de 5000 avaliações, 16 problemas do conjunto (67%) foram resolvidos. Sabendo que com o algoritmo aqui proposto não foi possível resolver todos os problemas do conjunto *G-Suite*, as Figuras 5.22 e 5.23 possibilitam a identificação dos métodos com os quais foi resolvido o maior número de problemas sob os menores custos possíveis.

Ainda que o valor $\rho(\tau) = 1$ não esteja disponível para o conjunto *G-Suite*, considera-se o método **APM Sum** o mais robusto para ser integrado ao ABC G-best na resolução deste grupo de problemas. Nas Figuras 5.22 e 5.23 este método apresenta vantagem considerável no valor de τ para produzir o maior valor possível para $\rho(\tau)$ (75), se comparado com os próximos melhores métodos.

Devido ao maior número de problemas neste grupo e considerando o número de métodos de penalização avaliados neste trabalho, para auxiliar na análise dos Perfis de Desempenho, as Tabelas 5.46, 5.47, 5.48, 5.49, 5.50 e 5.51 apresentam as áreas normalizadas entre 0 e 1, sob as curvas $\rho(\tau)$ de cada método. Como sugerido em [78], quanto maior a área, maior a eficiência do algoritmo. As Tabelas 5.46, 5.47, 5.48, 5.49, 5.50 e 5.51 só reforçam a robustez do método APM Sum na resolução do conjunto *G-Suite*.

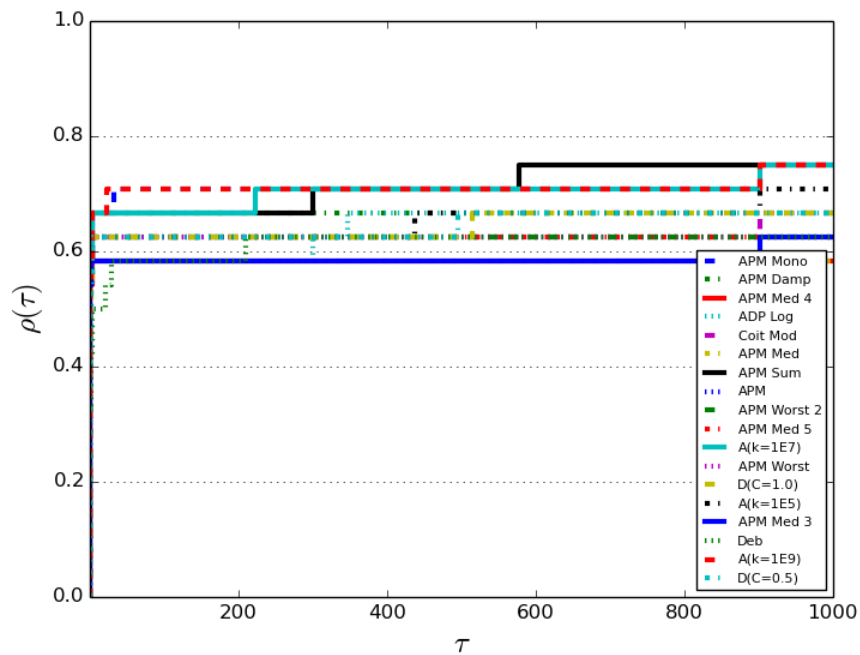


Figura 5.22: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para o *G-Suite* adotando a média da função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.

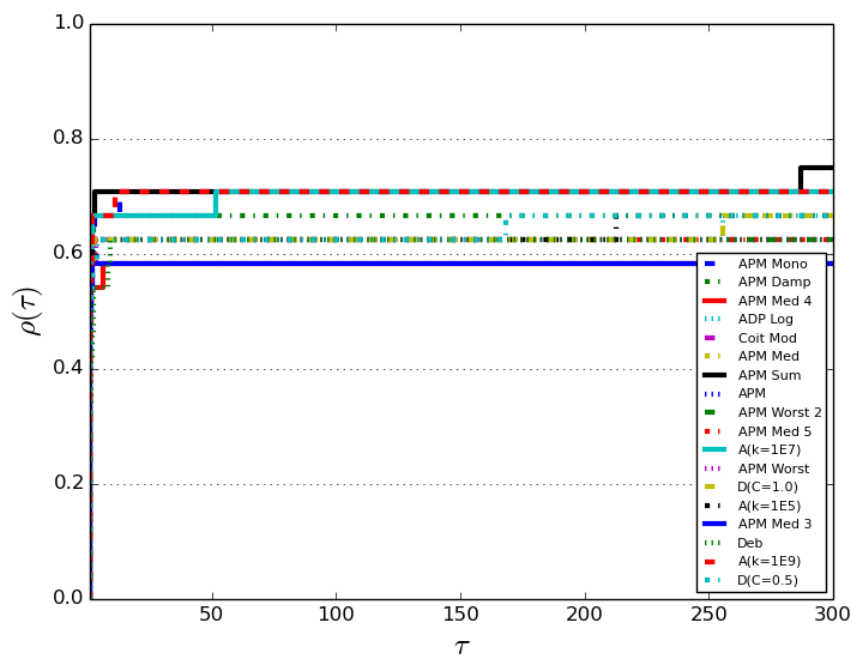


Figura 5.23: Perfis de Desempenho (destaque em $\rho(\tau) = 1$) para o *G-Suite* adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.

Tabela 5.46: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando a média da função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

Método	Área
A(k=1E5)	1.000000
A(k=1E7)	0.999182
A(k=1E9)	0.999111
ADP-Log	0.937458
APM-Mono	0.937411
APM-Sum	0.935375
Coit-Mod	0.915995
APM-Damp	0.876436
D(C=0.5)	0.861923
D(C=1.0)	0.858240
APM-Med-4	0.757680
APM-Med	0.754899
APM	0.753553
APM-Worst-2	0.752190
APM-Med-5	0.752157
APM-Med-3	0.752028
APM-Worst	0.744457
Deb	0.730079

Tabela 5.47: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 5000 avaliações da função objetivo.

Método	Área
A(k=1E5)	1.000000
A(k=1E9)	0.999015
A(k=1E7)	0.997376
APM-Mono	0.932361
APM-Sum	0.930668
ADP-Log	0.930384
Coit-Mod	0.930365
D(C=1.0)	0.865445
APM-Damp	0.864545
D(C=0.5)	0.857019
APM-Med	0.745674
APM	0.744662
APM-Med-4	0.742185
APM-Med-3	0.737874
APM-Worst	0.736683
APM-Med-5	0.735060
Deb	0.732336
APM-Worst-2	0.728134

Tabela 5.48: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando a média da função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.

Método	Área
APM-Sum	1.000000
A(k=1E9)	0.998506
APM-Mono	0.998232
A(k=1E7)	0.992054
APM-Damp	0.935014
A(k=1E5)	0.927894
D(C=0.5)	0.898130
D(C=1.0)	0.897475
ADP-Log	0.893158
Coit-Mod	0.884995
APM-Med-5	0.876975
APM-Worst	0.869055
APM	0.862641
APM-Worst-2	0.857177
Deb	0.848590
APM-Med-3	0.833424
APM-Med	0.819283
APM-Med-4	0.800012

Tabela 5.49: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 50000 avaliações da função objetivo.

Método	Área
APM-Sum	1.000000
A(k=1E9)	0.960836
APM-Mono	0.960674
A(k=1E7)	0.958298
APM-Damp	0.934821
D(C=0.5)	0.894255
A(k=1E5)	0.891612
ADP-Log	0.888901
D(C=1.0)	0.888889
APM	0.876452
APM-Worst	0.876426
APM-Med-5	0.876413
Coit-Mod	0.848396
APM-Worst-2	0.848384
Deb	0.847227
APM-Med	0.820082
APM-Med-3	0.820000
APM-Med-4	0.801766

Tabela 5.50: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando a média da função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

Método	Área
ADP-Log	1.000000
APM-Sum	0.999160
APM-Mono	0.973589
D(C=1.0)	0.955639
A(k=1E7)	0.955419
D(C=0.5)	0.954504
A(k=1E9)	0.951283
APM-Worst	0.937325
APM-Med-4	0.932874
APM	0.932003
APM-Damp	0.929469
APM-Med-5	0.926819
Coit-Mod	0.923463
APM-Med	0.913340
A(k=1E5)	0.899791
Deb	0.894103
APM-Med-3	0.869375
APM-Worst-2	0.839414

Tabela 5.51: Perfis de Desempenho – Áreas sob as curvas $\rho(\tau)$ para o conjunto *G-Suite*, adotando o valor mínimo de função objetivo como métrica e resultados obtidos com 500000 avaliações da função objetivo.

Método	Área
APM-Mono	1.000000
ADP-Log	0.999926
D(C=1.0)	0.999923
APM-Sum	0.999822
D(C=0.5)	0.999777
A(k=1E7)	0.999461
Coit-Mod	0.944483
A(k=1E9)	0.944474
A(k=1E5)	0.944189
APM-Med	0.943866
APM-Damp	0.943842
APM-Worst	0.943797
APM	0.943741
APM-Med-5	0.943608
APM-Med-4	0.943554
Deb	0.888517
APM-Med-3	0.888232
APM-Worst-2	0.833403

6 Conclusões

Neste trabalho foi feito um estudo sobre o desempenho do algoritmo Colônia de Abelhas Artificiais (ABC – do inglês *Artificial Bee Colony*) quando aplicado na resolução de problemas de otimização com restrições. Para o tratamento das restrições foram adotados métodos de penalização. Sobre o ABC destaca-se que é de fácil implementação e para a sua execução requer a definição de poucos parâmetros.

Diversos métodos foram integrados ao ABC com o objetivo de identificar aquele que melhor contribuiria com a convergência do algoritmo.

Buscou-se avaliar o ABC com o mínimo de alterações possíveis na forma como foi proposto. No entanto alguns pesquisadores vêm tentando melhorar a capacidade de intensificação na busca local do algoritmo. Com isto, foi implementado um segundo algoritmo identificado como ABC G-best neste trabalho. O ABC G-best gerou melhores resultados que o ABC.

Com a possibilidade do conjunto de soluções candidatas sofrer diversas alterações em cada iteração do ABC, neste trabalho os parâmetros de penalização foram ajustados a cada substituição de solução no conjunto.

Entre os métodos de penalização adaptativos aqui analisados, alguns deles apresentaram algumas características que provocaram erros na penalização. Para estes métodos foram propostas algumas alterações, dando origem neste trabalho a dois novos métodos intitulados Coit Mod e ADP Log. Já o método intitulado APM Sum, também proposto neste trabalho, consiste de uma variação para o método APM. O método APM não apresentou erros na penalização, mas com o APM Sum os parâmetros de penalização poderiam atingir valores que contribuiriam com o desempenho do ABC.

Diversos problemas de otimização encontrados na literatura foram impostos ao algoritmo. Os resultados confirmaram a competitividade do ABC. Em alguns casos foram obtidos resultados melhores que os relatados na literatura.

Conclui-se que combinar métodos de penalização ao algoritmo ABC para resolver problemas de otimização com restrições é uma alternativa confiável e eficiente.

Ainda sobre os resultados, quanto aos métodos de penalização, quando aplicados os adaptativos foram encontradas ótimas soluções. Métodos desse tipo aparecem com

frequência entre aqueles com os quais foram obtidos os melhores resultados. Sobre os métodos de penalização adaptativos, deve-se destacar que eles são de fácil aplicação, principalmente aqueles que não exigem a definição de nenhum parâmetro. Para o grupo de problemas da Engenharia Mecânica o método identificado como o mais robusto foi o APM Worst. Já entre os problemas de Otimização Estrutural, os métodos que se destacaram foram o A(k=1E5), o APM Med 3 e o APM Sum. Para a resolução dos problemas do conjunto *G-Suite*, o método APM Sum, proposto neste trabalho foi identificado como o mais robusto. Com base nos resultados obtidos e apresentados para cada um dos problemas analisados e resolvidos, entre os métodos que se destacaram o APM Sum é visto como a melhor escolha para ser combinado ao algoritmo ABC G-best.

Entre os métodos ADP Log, Coit Mod e APM Sum, propostos neste trabalho, quando combinados ao ABC G-best, o melhor desempenho foi obtido com o APM Sum, seguido do Coit Mod e por último, com o pior desempenho entre os três o método ADP Log. Deve-se destacar que o APM Sum foi identificado neste trabalho como o método mais indicado a ser integrado ao ABC G-best. Os resultados obtidos com o método Coit Mod estiveram entre os melhores métodos identificados neste trabalho. Já o método ADP Log, em diversos problemas analisados neste trabalho esteve presente entre aqueles com os quais foram obtidos os resultados menos atraentes.

Como sugestões de trabalhos futuros, pode-se considerar:

- Analisar a sensibilidade dos parâmetros do ABC com o objetivo de identificar a influência do ajuste de cada um deles na convergência do algoritmo;
- Realizar estudo comparativo do desempenho de variações propostas na literatura para o algoritmo ABC;
- Aplicar o algoritmo ABC na resolução de problemas de otimização multi-objetivo tratando as restrições com os métodos de penalização que se destacaram neste trabalho;
- Hibridizar o algoritmo ABC acoplando outras metaheurísticas;
- Realizar estudo cuidadoso sobre o método de penalização APM Sum, dado que a formulação do método soma parcelas de grandezas distintas ao ajustar os parâmetros de penalização e também que os resultados discutidos neste trabalho associados a este método foram obtidos de forma totalmente empírica.

REFERÊNCIAS

- [1] BORTOLOSSI, H. J., *Cálculo diferencial a várias variáveis: uma introdução à teoria de otimização*. 4th ed. Edições Loyola, Editora PUC-Rio: São Paulo - Brasil, Rio de Janeiro - Brasil, 2010.
- [2] FRIEDLANDER, A., *Elementos de programação não-linear. Série Manuais*, Editora da UNICAMP, 1994.
- [3] RAO, S. S., *Engineering optimization : Theory and Practice*. 4th ed. John Wiley & Sons, Inc, 2009.
- [4] CARVALHO, E. C. R., *Solução de problemas de otimização com restrições usando estratégias de penalização adaptativa e um algoritmo do tipo PSO*, Mestrado, PPGMC, Programa de Pós-graduação em Modelagem Computacional, UFJF, 2014.
- [5] GANDOMI, A., YANG, X., TALATAHARI, S., ALAVI, A., *Metaheuristic Applications in Structures and Infrastructures*. Elsevier Science, 2013.
- [6] LUENBERGER, D. G., YE, Y., *Linear and Nonlinear Programming*. 3rd ed. Springer Science+Business Media: New York - USA, 2008.
- [7] BERNARDINO, H. S., *Hibridização de Algoritmos Genéticos e Sistemas Imunológicos Artificiais para Problemas de Otimização com Restrições em Engenharia*, Mestrado, PPGMC, Programa de Pós-graduação em Modelagem Computacional, UFJF, 2008.
- [8] LEMONGE, A. C. C., *Aplicação de Algoritmos Genéticos em Otimização Estrutural*, Doutorado, COPPE, Programa de Engenharia Civil, UFRJ, 1999.
- [9] SILVA, M. M., *Otimização de Estruturas Reticuladas Incluindo Não-Linearidade Geométrica*, Mestrado, PPGMC, Programa de Pós-graduação em Modelagem Computacional, UFJF, 2011.
- [10] BORGWARDT, K. H., *The Simplex Method: A Probabilistic Analysis*. 1st ed. Springer-Verlag: Berlim - Alemanha, 1987.

- [11] BERTSEKAS, D. P., *Nonlinear Programming*. 2nd ed. Athena Scientific: Belmont - USA, 1999.
- [12] BAZARAA, M. S., SHERALI, H. D., SHETTY, C. M., *Nonlinear Programming - Theory and Algorithms*. 3rd ed. John Wiley & Sons, Inc: New Jersey - USA, 2006.
- [13] HOLLAND, J. H., “Genetic Algorithms and the Optimal Allocation of Trials”, *SIAM Journal on Computing*, v. 2, n. 2, pp. 88 – 105, 1973.
- [14] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P., “Optimization by simulated annealing”, *SCIENCE*, v. 220, n. 4598, pp. 671 – 680, 1983.
- [15] DORIGO, M., MANIEZZO, V., COLORNI, A., “Ant system: optimization by a colony of cooperating agents”, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, v. 26, n. 1, pp. 29–41, Feb 1996.
- [16] DORIGO, M., STUTZLE, T., *Ant Colony Optimization*. 1st ed. Massachusetts Institute of Technology: London, England, 2004.
- [17] GOLIATT, P. V. Z. C., ANGELO, J. S., BARBOSA, H. J. C., “Colônia de Formigas”, In: *Manual de Computação Evolutiva e Metaheurística*, chap. 5, pp. 87 – 105, Editora UFMG; Imprensa da Universidade de Coimbra: Belo Horizonte - Brasil; Coimbra - Portugal, 2013.
- [18] KEPHART, J. O., “A biologically inspired immune system for computers”. In: *Artificial Life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*, pp. 130 – 139, 1994.
- [19] BERNARDINO, H. S., BARBOSA, H. J., “Artificial Immune Systems for Optimization”, In: *Nature-Inspired Algorithms for Optimisation*, v. 193, pp. 389–411, *Studies in Computational Intelligence*, Springer Berlin Heidelberg, 2009.
- [20] EBERHART, R., KENNEDY, J., “A new optimizer using particle swarm theory”. In: *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pp. 39–43, IEEE, Oct 1995.

- [21] STORN, R., PRICE, K., “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces”, *Journal of Global Optimization*, v. 11, n. 4, pp. 341–359, 1997.
- [22] CUEVAS, E., CIENFUEGOS, M., ZALDÍVAR, D., PÉREZ-CISNEROS, M., “A swarm optimization algorithm inspired in the behavior of the social-spider”, *Expert Systems with Applications*, v. 40, n. 16, pp. 6374 – 6384, 2013.
- [23] SIMON, D., “Biogeography-Based Optimization”, *Evolutionary Computation, IEEE Transactions on*, v. 12, n. 6, pp. 702–713, Dec 2008.
- [24] YANG, X.-S., *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
- [25] CHENG, M.-Y., PRAYOGO, D., “Symbiotic Organisms Search: A new metaheuristic optimization algorithm”, *Computers & Structures*, v. 139, pp. 98 – 112, 2014.
- [26] KARABOGA, D., *An idea based on honey bee swarm for numerical optimization*, Tech. Rep. TR06, Erciyes University, Engineering Faculty, Kayseri, Turkiye, October 2005.
- [27] KARABOGA, D., BASTURK, B., “On the performance of artificial bee colony (ABC) algorithm”, *Applied Soft Computing*, v. 8, n. 1, pp. 687 – 697, 2008.
- [28] KARABOGA, D., BASTURK, B., “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *Journal of Global Optimization*, v. 39, n. 3, pp. 459–471, 2007.
- [29] AKAY, B., KARABOGA, D., “Artificial bee colony algorithm for large-scale problems and engineering design optimization”, *Journal of Intelligent Manufacturing*, v. 23, n. 4, pp. 1001–1014, 2012.
- [30] FREE, J. B., *A organização social das abelhas (Apis)*. EPU: Ed. da Univesidade de São Paulo, 1980.
- [31] KARABOGA, D., AKAY, B., “A comparative study of Artificial Bee Colony algorithm”, *Applied Mathematics and Computation*, v. 214, n. 1, pp. 108–132, 2009.

- [32] KARABOGA, D., BASTURK, B., “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems”, In: *Foundations of Fuzzy Logic and Soft Computing*, v. 4529, pp. 789–798, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2007.
- [33] KARABOGA, D., AKAY, B., “A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems”, *Applied Soft Computing*, v. 11, n. 3, pp. 3021 – 3031, 2011.
- [34] DEB, K., “An efficient constraint handling method for genetic algorithms”, *Computer Methods in Applied Mechanics and Engineering*, v. 186, n. 2–4, pp. 311 – 338, 2000.
- [35] QUAN, H., SHI, X., “On the Analysis of Performance of the Improved Artificial-Bee-Colony Algorithm”. In: *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, v. 7, pp. 654–658, Oct 2008.
- [36] ZHU, G., KWONG, S., “Gbest-guided artificial bee colony algorithm for numerical function optimization”, *Applied Mathematics and Computation*, v. 217, n. 7, pp. 3166 – 3173, 2010.
- [37] KENNEDY, J., EBERHART, R., “Particle swarm optimization”. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*, v. 4, pp. 1942–1948 vol.4, Nov 1995.
- [38] HADIDI, A., AZAD, S. K., AZAD, S. K., “Structural optimization using artificial bee colony algorithm”. In: *2nd international conference on engineering optimization*, 2010.
- [39] PAN, Q.-K., TASGETIREN, M. F., SUGANTHAN, P., CHUA, T., “A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem”, *Information Sciences*, v. 181, n. 12, pp. 2455 – 2468, 2011.
- [40] KOOHESTANI, K., AZAD, S. K., “An adaptive real-coded genetic algorithm for size and shape optimization of truss structures”. In: *The First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, 2009.

- [41] ALATAS, B., “Chaotic bee colony algorithms for global numerical optimization”, *Expert Systems with Applications*, v. 37, n. 8, pp. 5682 – 5687, 2010.
- [42] RAHNAMAYAN, S., TIZHOOSH, H., SALAMA, M., “Opposition-Based Differential Evolution”, *Evolutionary Computation, IEEE Transactions on*, v. 12, n. 1, pp. 64–79, Feb 2008.
- [43] GAO, W.-F., LIU, S.-Y., “A modified artificial bee colony algorithm”, *Computers & Operations Research*, v. 39, n. 3, pp. 687 – 697, 2012.
- [44] GAO, W., LIU, S., HUANG, L., “A global best artificial bee colony algorithm for global optimization”, *Journal of Computational and Applied Mathematics*, v. 236, n. 11, pp. 2741 – 2753, 2012.
- [45] XIANG, W.-L., AN, M.-Q., “An efficient and robust artificial bee colony algorithm for numerical optimization”, *Computers & Operations Research*, v. 40, n. 5, pp. 1256 – 1265, 2013.
- [46] ALATAS, B., AKIN, E., OZER, A. B., “Chaos embedded particle swarm optimization algorithms”, *Chaos, Solitons & Fractals*, v. 40, n. 4, pp. 1715 – 1734, 2009.
- [47] BANHARNSAKUN, A., ACHALAKUL, T., SIRINAOVAKUL, B., “The best-so-far selection in Artificial Bee Colony algorithm”, *Applied Soft Computing*, v. 11, n. 2, pp. 2888 – 2901, 2011, The Impact of Soft Computing for the Progress of Artificial Intelligence.
- [48] SINGH, A., “An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem”, *Applied Soft Computing*, v. 9, n. 2, pp. 625 – 631, 2009.
- [49] JULSTROM, B. A., “Codings and operators in two genetic algorithms for the leaf-constrained minimum spanning tree problem”, *International Journal of Applied Mathematics and Computer Science*, v. 14, pp. 385 – 396, 2004.
- [50] SINGH, A., BAGHEL, A. S., “New Metaheuristic Approaches For The Leaf-constrained Minimum Spanning Tree Problem”, *Asia-Pacific Journal of Operational Research*, v. 25, n. 04, pp. 575–589, 2008.

- [51] RAJEEV, S., KRISHNAMOORTHY, C., “Discrete Optimization of Structures Using Genetic Algorithms”, *Journal of Structural Engineering*, v. 118, n. 5, pp. 1233–1250, 1992.
- [52] KARABOGA, D., OZTURK, C., “A novel clustering approach: Artificial Bee Colony (ABC) algorithm”, *Applied Soft Computing*, v. 11, n. 1, pp. 652 – 657, 2011.
- [53] FALCO, I. D., CIOPPA, A. D., TARANTINO, E., “Facing classification problems with Particle Swarm Optimization”, *Applied Soft Computing*, v. 7, n. 3, pp. 652 – 658, 2007.
- [54] SONMEZ, M., “Artificial Bee Colony algorithm for optimization of truss structures”, *Applied Soft Computing*, v. 11, n. 2, pp. 2406 – 2418, 2011, The Impact of Soft Computing for the Progress of Artificial Intelligence.
- [55] HASANÇEBİ, O., “Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures”, *Computers & Structures*, v. 86, n. 1–2, pp. 119 – 132, 2008.
- [56] BULLINARIA, J. A., ALYAHYA, K., “Artificial Bee Colony Training of Neural Networks”, In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*, v. 512, pp. 191–201, *Studies in Computational Intelligence*, Springer International Publishing, 2014.
- [57] GASPAR-CUNHA, A., TAKAHASHI, R., ANTUNES, C. H., *Manual de Computação Evolutiva e Metaheurística*. Editora UFMG; Imprensa da Universidade de Coimbra: Belo Horizonte - Brasil; Coimbra - Portugal, 2013.
- [58] IZMAILOV, A., SOLODOV, M., *Otimização - volume 2: Métodos Computacionais*. 2nd ed., v. 2. IMPA: Rio de Janeiro - Brasil, 2012.
- [59] LEMONGE, A. C. C., BARBOSA, H. J. C., “An adaptive penalty scheme for genetic algorithms in structural optimization”, *International Journal for Numerical Methods in Engineering*, v. 59, n. 5, pp. 703–736, 2004.
- [60] HARRELL, L. J., RANJITHAN, S. R., “Evaluation of Alternative Penalty Function Implementations in a Watershed Management Design Problem”. In:

Proceedings of the Genetic and Evolutionary Computation Conference, v. 2, pp. 1551 – 1558, Morgan Kaufmann: Orlando, Florida, USA, July 1999.

- [61] JOINES, J., HOUCK, C., “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA’s”. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 579–584 vol.2, Jun 1994.
- [62] NANAKORN, P., MEESOMKLIN, K., “An adaptive penalty function in genetic algorithms for structural design optimization”, *Computers & Structures*, v. 79, n. 29 - 30, pp. 2527 – 2539, 2001.
- [63] BEAN, J., HADJ-ALOUANE, A., *A dual genetic algorithm for bounded integer programs*, Tech. Rep. 92-53, Department of Industrial and Operations Engineering, The University of Michigan, An Arbor, MI, 1992.
- [64] COIT, D. W., SMITH, A. E., TATE, D. M., “Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems”, *INFORMS Journal on Computing*, v. 8, n. 2, pp. 173–182, 1996.
- [65] YOKOTA, T., GEN, M., IDA, K., TAGUCHI, T., “Optimal design of system reliability by an improved genetic algorithm”, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, v. 79, n. 2, pp. 41 – 51, 1996.
- [66] MONTEMURRO, M., VINCENTI, A., VANNUCCI, P., “The Automatic Dynamic Penalisation method (ADP) for handling constraints with genetic algorithms”, *Computer Methods in Applied Mechanics and Engineering*, v. 256, n. 0, pp. 70 – 87, 2013.
- [67] BARBOSA, H. J., LEMONGE, A. C., “An adaptive penalty scheme in genetic algorithms for constrained optimization problems”. In: *In Proc. of the Genetic and Evolutionary Computation Conference*, v. 2, pp. 287 – 294, 2002.
- [68] BARBOSA, H. J., LEMONGE, A. C., “A new adaptive penalty scheme for genetic algorithms”, *Information Sciences*, v. 156, n. 3-4, pp. 215 – 251, 2003, Evolutionary Computation.

- [69] BARBOSA, H. J., LEMONGE, A. C., “An Adaptive Penalty Scheme for Steady-State Genetic Algorithms”, In: *Genetic and Evolutionary Computation — GECCO 2003*, v. 2723, pp. 718 – 729, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003.
- [70] BARBOSA, H. J., LEMONGE, A. C., “An adaptive penalty method for genetic algorithms in constrained optimization problems”, *Frontiers in Evolutionary Robotics*, v. 34, pp. 196, April 2008.
- [71] LEMONGE, A., BARBOSA, H., BERNARDINO, H., “A family of adaptive penalty schemes for steady-state genetic algorithms”. In: *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1 – 8, June 2012.
- [72] GARCIA, R. P., MONTA, B. G., CARVALHO, E. C. R., BARBOSA, H. J. C., BERNARDINO, H. S., LEMONGE, A. C. C., “Novas variantes para o método de penalização adaptativo (APM) para problemas de otimização com restrições”, In: *XLV Simpósio Brasileiro de Pesquisa Operacional - SBPO*, pp. 2017 – 2028, 2013.
- [73] CARVALHO, E. C. R., MONTA, B. G., GARCIA, R. P., BERNARDINO, H. S., LEMONGE, A. C. D. C., “Variantes para o método de penalização adaptativa (APM) para problemas de otimização com restrições aplicadas a problemas de engenharia”, In: *XXXIV Iberian Latin-American Congress on Computational Methods in Engineering - CILAMCE*, 2013.
- [74] DUARTE, G. R., DA FONSECA, L. G., LEMONGE, A. C. C., “Estudo de técnicas de penalização adaptativas no Algoritmo Colônia de Abelhas Artificiais aplicado a problemas de otimização em Engenharia”, In: *XXXV Iberian Latin-American Congress on Computational Methods in Engineering - CILAMCE*, 2014.
- [75] MERNIK, M., LIU, S.-H., KARABOGA, D., CREPINSEK, M., “On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation”, *Information Sciences*, v. 291, n. 0, pp. 115 – 127, 2015.

- [76] DOLAN, E. D., MORE, J. J., “Benchmarking optimization software with performance profiles”, *Mathematical Programming*, v. 91, n. 2, pp. 201 – 213, 2002.
- [77] BERNARDINO, H., BARBOSA, H., FONSECA, L., “Surrogate-assisted clonal selection algorithms for expensive optimization problems”, *Evolutionary Intelligence*, v. 4, n. 2, pp. 81 – 97, 2011.
- [78] BARBOSA, H., BERNARDINO, H., BARRETO, A., “Using performance profiles to analyze the results of the 2006 CEC constrained optimization competition”. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1 – 8, July 2010.
- [79] BARRETO, A. M., BERNARDINO, H. S., BARBOSA, H. J., “Probabilistic Performance Profiles for the Experimental Evaluation of Stochastic Algorithms”. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 10*, pp. 751 – 758, ACM: New York, NY, USA, 2010.
- [80] NEVO, S., NEVO, D., EIN-DOR, P., “Classifying information technologies: A multidimensional scaling approach”, *Communications of the Association for Information Systems*, v. 27, n. 1, pp. 45, 2010.
- [81] WICKELMAIER, F., *An introduction to MDS*, Tech. Rep. Reports from the Sound Quality Research Unit (SQRU), n. 7, Aalborg University, Denmark, May 2003.
- [82] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., DUCHESNAY, E., “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.
- [83] LEEUW, J., MAIR, P., “Multidimensional Scaling Using Majorization: SMACOF in R”, *Journal of Statistical Software*, v. 31, n. 3, pp. 1 – 30, 8 2009.

- [84] MEZURA-MONTES, E., COELLO, C. A. C., LANDA-BECERRA, R., “Engineering optimization using simple evolutionary algorithm”. In: *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*, pp. 149 – 156, Nov 2003.
- [85] BERNARDINO, H., BARBOSA, H., LEMONGE, A., FONSECA, L., “A new hybrid AIS-GA for constrained optimization problems in mechanical engineering.” In: *2008 IEEE congress on evolutionary computation (CEC 2008)*, pp. 1455 – 1462, IEEE Service Center, Piscataway: Hong-Kong, 2008.
- [86] COELLO, C., BECERRA, R., “Efficient evolutionary through the use of a cultural algorithm”, *Engineering Optimization*, v. 36, pp. 219 – 236, 2004.
- [87] HE, Q., WANG, L., “An effective co-evolutionary particle swarm optimization for constrained engineering design problems”, *Engineering Applications of Artificial Intelligence*, v. 20, n. 1, pp. 89 – 99, 2007.
- [88] RUNARSSON, T., YAO, X., “Stochastic ranking for constrained evolutionary optimization”, *Evolutionary Computation, IEEE Transactions on*, v. 4, n. 3, pp. 284–294, Sep 2000.
- [89] GANDOMI, A., YANG, X.-S., ALAVI, A., TALATAHARI, S., “Bat algorithm for constrained optimization tasks”, *Neural Computing and Applications*, v. 22, n. 6, pp. 1239–1255, 2013.
- [90] BRAJEVIC, I., TUBA, M., “An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems”, *Journal of Intelligent Manufacturing*, v. 24, n. 4, pp. 729–740, 2013.
- [91] BRAJEVIC, I., TUBA, M., SUBOTIC, M., “Improved Artificial Bee Colony Algorithm for Constrained Problems”. In: *Proceedings of the 11th WSEAS International Conference on Neural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems, NN’10/EC’10/FS’10*, pp. 185 – 190, World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, Wisconsin, USA, 2010.

- [92] SILVA, F. B. S., *Algoritmos genéticos para otimização de estruturas reticuladas baseadas em modelos adaptativos e lagrangeano aumentado*, Mestrado, PPGMC, Programa de Pós-Graduação em Modelagem Computacional, UFJF, 2011.
- [93] YANG, X., GANDOMI, A. H., “Bat algorithm: a novel approach for global engineering optimization”, *Engineering Computations*, v. 29, n. 5, pp. 464–483, 2012.
- [94] BERNARDINO, H., BARBOSA, I., LEMONGE, A., “A hybrid genetic algorithm for constrained optimization problems in mechanical engineering”. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 646–653, Sept 2007.
- [95] ARAGON, V., VICTORIA, S., ESQUIVEL, S., COELLO, C., “A modified version of a T-Cell Algorithm for constrained optimization problems”, *International Journal for Numerical Methods in Engineering*, v. 84, n. 3, pp. 351 – 378, 2010.
- [96] PARK, Y. C., CHANG, M. H., LEE, T. Y., “A new deterministic global optimization method for general twice-differentiable constrained nonlinear programming problems”, *Engineering Optimization*, v. 39, n. 4, pp. 397 – 411, 2007.
- [97] RAY, T., SAINI, P., “Engineering design optimization using a swarm with an intelligent information sharing among individuals”, *Engineering Optimization*, v. 33, n. 6, pp. 735–748, 2001.
- [98] COELLO, C. A. C., CORTES, N. C., “Hybridizing a genetic algorithm with an artificial immune system for global optimization”, *Engineering Optimization*, v. 36, n. 5, pp. 607 – 634, 2004.
- [99] ERBATUR, F., HASANÇEBI, O., TUTUNCU, I., KILIÇ, H., “Optimal design of planar and space structures with genetic algorithms”, *Computers & Structures*, v. 75, n. 2, pp. 209 – 224, 2000.
- [100] HWANG, S.-F., HE, R.-S., “A hybrid real-parameter genetic algorithm for function optimization”, *Advanced Engineering Informatics*, v. 20, n. 1, pp. 7 – 21, 2006.

- [101] HEDAR, A.-R., FUKUSHIMA, M., “Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization”, *Journal of Global Optimization*, v. 35, n. 4, pp. 521 – 549, 2006.
- [102] LIU, J.-L., “Novel orthogonal simulated annealing with fractional factorial analysis to solve global optimization problems”, *Engineering Optimization*, v. 37, n. 5, pp. 499 – 519, 2005.
- [103] LEE, K. S., GEEM, Z. W., “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice”, *Computer Methods in Applied Mechanics and Engineering*, v. 194, n. 36–38, pp. 3902 – 3933, 2005.
- [104] ZHANG, M., LUO, W., WANG, X., “Differential evolution with dynamic stochastic selection for constrained optimization”, *Information Sciences*, v. 178, n. 15, pp. 3043 – 3074, 2008, Nature Inspired Problem-Solving.
- [105] SILVA, A. F., *Análise de uma técnica de penalização adaptativa aplicada ao Algoritmo de Otimização por Enxame de Partículas*, Mestrado, COPPE, Programa de Pós-Graduação em Engenharia Civil, UFRJ, 2010.
- [106] GHASEMI, M., HINTON, E., WOOD, R., “Optimization of trusses using genetic algorithms for discrete and continuous variables”, *Engineering Computations*, v. 16, n. 3, pp. 272 – 303, 1999.
- [107] LAMBERTI, L., “An efficient simulated annealing algorithm for design optimization of truss structures”, *Computers & Structures*, v. 86, n. 19–20, pp. 1936 – 1953, 2008.
- [108] HAFTKA, R. T., GURDAL, Z., *Elements of Structural Optimization*. 3rd ed. Netherlands, 1992.
- [109] LEE, K. S., GEEM, Z. W., “A new structural optimization method based on the harmony search algorithm”, *Computers & Structures*, v. 82, n. 9–10, pp. 781 – 798, 2004.

- [110] LI, L., HUANG, Z., LIU, F., WU, Q., “A heuristic particle swarm optimizer for optimization of pin connected structures”, *Computers & Structures*, v. 85, n. 7–8, pp. 340 – 349, 2007.
- [111] VENKAYYA, V., “Design of optimum structures”, *Computers & Structures*, v. 1, n. 1–2, pp. 265 – 309, 1971.
- [112] LIANG, J., RUNARSSON, T. P., MEZURA-MONTES, E., CLERC, M., SUGANTHAN, P., COELLO, C. C., DEB, K., “Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization”, *Journal of Applied Mechanics*, v. 41, 2006.