

Érica da Costa Reis Carvalho

Solução de problemas de otimização com restrições usando estratégias de penalização adaptativa e um algoritmo do tipo PSO

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. Heder Soares Bernardino

Coorientadora: Prof.^a D.Sc. Patrícia Habib Hallak

Juiz de Fora

2014

Érica da Costa Reis Carvalho

Solução de problemas de otimização com restrições usando estratégias de penalização adaptativa e um algoritmo do tipo PSO

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 13 de Fevereiro de 2014.

BANCA EXAMINADORA

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

Prof. D.Sc. Heder Soares Bernardino - Coorientador
Universidade Federal de Juiz de Fora

Prof.^a D.Sc. Patrícia Habib Hallak - Coorientadora
Universidade Federal de Juiz de Fora

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

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

*Dedico esta dissertação a todos os meus familiares,
em especial meus pais Ana Paula e Fernando,
e ao meu namorado Bruno.*

AGRADECIMENTOS

Agradeço a Deus, em primeiro lugar, pelas bênçãos e por me dar a oportunidade de compartilhar este momento com as pessoas que amo.

Aos meus pais Ana Paula e Fernando, pelo apoio incondicional ao longo destes anos e em todas as etapas da minha vida. Amo vocês!

Ao meu namorado Bruno, que esteve ao meu lado durante todo o curso. Obrigada por cada gesto de carinho e apoio demonstrado.

Ao meu irmão Paulo Fernando, pelo carinho e alegria.

Ao vô Nilton e vó Nídia, pelo exemplo de sucesso e sabedoria, apoio pessoal e profissional.

À todos os meus familiares, em especial às tias Romilda, Helena, Vera e Maura e aos tio Victor e Lúcio.

Às amigas e amigos de Prados, que mesmo longe estão sempre presentes e aos amigos que conquistei ao longo do curso. Obrigada pelos finais de semana, pelas gargalhadas e por proporcionarem momentos únicos e especiais.

Aos meus orientadores, prof. Afonso Lemonge, prof. Heder Bernardino e prof^a. Patrícia Hallak, por todo o empenho, paciência e atenção. Obrigada por me conduzirem com toda a experiência, tornando possível a conclusão desta dissertação.

Aos colegas de mestrado, Daniele, Rafaela, Ana Amélia, Rafael, Jonathan, Maicon, Bernardo e Dênis. Sinto que nós percorremos este caminho juntos, nos complementando e nos fortalecendo. Obrigada pela rica troca e cumplicidade.

Aos professores, funcionários e colegas do programa de pós-graduação em Modelagem Computacional.

À CAPES, à Universidade Federal de Juiz de Fora e à FAPEMIG (Projeto TEC 528/11) pelo apoio financeiro concebido.

À todos que de forma direta ou indireta contribuíram para a realização deste trabalho.

Grandes mentes falam sobre ideias. Mentas medianas falam sobre coisas.

Mentas pequenas falam sobre pessoas.

Eleanor Roosevelt

RESUMO

Nos últimos anos, várias meta-heurísticas têm sido adotadas para a solução de problemas de otimização com restrições. Uma dessas meta-heurísticas que se torna cada vez mais popular é a Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO). O PSO é baseado na metáfora de como algumas espécies compartilham informações e, em seguida, usam essas informações para mover-se até os locais onde os alimentos estão localizados. A população é formada por um conjunto de indivíduos denominado partículas que representa possíveis soluções dentro de um espaço de busca multidimensional. Neste trabalho, são analisados problemas clássicos de otimização com restrições onde um algoritmo PSO os trata como sendo sem restrições através da introdução de um método de penalização adaptativa (*Adaptive Penalty Method* - APM). O APM adapta o valor dos coeficientes de penalização de cada restrição fazendo uso de informações coletadas da população, tais como a média da função objetivo e o nível de violação de cada restrição. Diversos experimentos computacionais são realizados visando avaliar o desempenho do algoritmo considerando vários problemas testes encontrados na literatura.

Palavras-chave: Otimização por Enxame de Partículas. Otimização com restrições. Métodos de penalização.

ABSTRACT

In recent years, several meta-heuristics have been adopted for the solution of constrained optimization problems. One of these meta-heuristic that is becoming increasingly popular is the Particle Swarm Optimization - PSO. PSO is based on the metaphor of how some species share information and then use this information to move to the places where food is located. The population is formed by a group of individuals called particles representing possible solutions within a space multidimensional search. In this thesis, classical problems of constrained optimization where a PSO algorithm treats them as being unconstrained by introducing a method of adaptive penalty (Adaptive Penalty Method - APM) are analyzed. The APM adjusts the value of the penalty coefficients of each constraint using the information collected from the population, such as the average of the objective function as well as the level of violation of each constraint. Several computational experiments are conducted to assess the performance the algorithm tests considering various problems found in the literature.

Keywords: Particle Swarm Optimization. Constrained optimization. Penalties methods.

SUMÁRIO

1	Introdução	18
2	Otimização	21
2.1	Introdução	21
2.2	A formulação do problema de otimização	22
2.3	Superfície de restrição em um espaço de projeto bidimensional	25
2.4	Otimização Estrutural	26
<i>2.4.1</i>	<i>Alguns exemplos de otimização estrutural</i>	<i>29</i>
<i>2.4.1.1</i>	<i>Problemas com restrições explícitas</i>	<i>29</i>
<i>2.4.1.2</i>	<i>Problemas com restrições implícitas</i>	<i>31</i>
2.5	Algoritmos de Otimização	32
<i>2.5.1</i>	<i>Determinísticos</i>	<i>32</i>
<i>2.5.2</i>	<i>Probabilísticos</i>	<i>32</i>
<i>2.5.2.1</i>	<i>Algoritmos Genéticos</i>	<i>33</i>
<i>2.5.2.2</i>	<i>Recozimento Simulado</i>	<i>34</i>
<i>2.5.2.3</i>	<i>Sistemas Imunológicos Artificiais</i>	<i>34</i>
<i>2.5.2.4</i>	<i>Colônia de Formigas</i>	<i>35</i>
<i>2.5.2.5</i>	<i>Colônia de Abelhas Artificial</i>	<i>35</i>
<i>2.5.2.6</i>	<i>Algoritmo Vaga-lume</i>	<i>36</i>
<i>2.5.2.7</i>	<i>Algoritmo Cuckoo Search</i>	<i>36</i>
<i>2.5.2.8</i>	<i>Enxame de Partículas</i>	<i>36</i>
3	Otimização por Enxame de Partículas	37
3.1	Introdução	37
3.2	Parâmetros	39
<i>3.2.1</i>	<i>Inércia</i>	<i>39</i>
<i>3.2.2</i>	<i>Fator de aceleração</i>	<i>41</i>
3.3	Atualização da velocidade	41
3.4	Topologia das partículas	43
3.5	Aplicações	44

4	Problemas com restrições	47
4.1	Introdução	47
4.2	Penalização Estática	48
4.3	Penalização Dinâmica	49
4.4	Penalização Adaptativa	50
4.4.1	<i>Estratégias de penalização aplicadas ao PSO</i>	53
4.5	Um Método de Penalização Adaptativa (APM)	55
4.5.1	<i>Variantes do APM</i>	57
5	Experimentos Numéricos	61
5.1	Implementação	61
5.2	Perfis de desempenho	62
5.3	Análise das variantes do APM	65
5.4	<i>Suite de Funções</i>	66
5.4.1	<i>Discussão da suite de funções</i>	67
5.5	Problemas clássicos da engenharia	73
5.5.1	<i>Mola Sob Tração/Compressão</i>	73
5.5.2	<i>Redutor de Velocidade</i>	76
5.5.3	<i>Viga Soldada</i>	78
5.5.4	<i>Vaso de Pressão</i>	79
5.5.5	<i>Viga Engastada e Livre</i>	81
5.5.6	<i>Discussão dos problemas de engenharia</i>	85
5.6	Problemas de otimização estrutural	86
5.6.1	<i>Treliça de 10 barras</i>	86
5.6.2	<i>Treliça de 25 barras</i>	88
5.6.3	<i>Treliça de 52 barras</i>	91
5.6.4	<i>Treliça de 60 barras</i>	92
5.6.5	<i>Treliça de 72 barras</i>	94
5.6.6	<i>Discussão dos problemas de otimização estrutural</i>	100
5.7	Análise dos Resultados	103
6	Conclusões	105
	REFERÊNCIAS	107

APÊNDICES	126
-----------------	-----

LISTA DE ILUSTRAÇÕES

2.1	Minimizar $f(\mathbf{x})$ é o mesmo que maximizar $-f(\mathbf{x})$	21
2.2	Classificação dos problemas de otimização, baseada na referência [1].	23
2.3	Classificação das técnicas de otimização numérica, reproduzida de [1].	24
2.4	Superfície de restrição em um espaço de projeto bidimensional, adaptada de [2].	26
2.5	Exemplo de um problema de otimização dimensional.	28
2.6	Exemplo de um problema de otimização dimensional e geométrica.	28
2.7	Exemplo de um problema de otimização dimensional, geométrica e topológica.	29
2.8	Pilar com seção transversal em forma de tubo.	29
3.1	Topologia das partículas: (a) Topologia global (b) Topologia local, baseada de [3].	44
4.1	Descrição da função \bar{f} , adaptada de [4].	56
5.1	Um exemplo do uso da ferramenta Perfis de desempenho.	64
5.2	Perfis de desempenho das variantes.	65
5.3	Perfis de desempenho utilizando o melhor valor para 5000 avaliações.	68
5.4	Perfis de desempenho utilizando a média para 5000 avaliações.	69
5.5	Perfis de desempenho utilizando o melhor valor para 50000 avaliações.	70
5.6	Perfis de desempenho utilizando a média para 50000 avaliações.	71
5.7	Perfis de desempenho utilizando o melhor valor para 500000 avaliações.	71
5.8	Perfis de desempenho utilizando a média para 500000 avaliações.	72
5.9	Mola sob tração/compressão.	74
5.10	Redutor de velocidade.	77
5.11	Viga Soldada.	78
5.12	Vaso de pressão.	81
5.13	Viga engastada e livre.	83
5.14	Perfis de desempenho utilizando o melhor valor.	85
5.15	Perfis de desempenho utilizando a média	86
5.16	Treliça de 10 barras.	87
5.17	Treliça de 25 barras.	90

5.18	Treliça de 52 barras.	93
5.19	Treliça de 60 barras.	96
5.20	Treliça de 72 barras.	99
5.21	Perfis de desempenho utilizando o melhor valor.	102
5.22	Perfis de desempenho utilizando a média	103

LISTA DE TABELAS

2.1	Alguns dos mais importantes registros históricos na área de otimização estrutural.	27
4.1	Variantes propostas por Garcia <i>et al.</i> [5] para o APM.	59
5.1	Desempenho dos cinco problemas obtidos através dos algoritmos A, B e C. . .	64
5.2	Área normalizada sob as curvas dos perfis de desempenho do algoritmos A, B e C.	64
5.3	Área normalizada sob as curvas dos perfis de desempenho para as variantes. .	66
5.4	Detalhes sobre o conjunto de 24 funções teste extraído de [6]. O número de variáveis de projeto é indicado por n , ρ é a taxa estimada entre a região factível e o espaço de busca, ni é o número de restrições de igualdade e ne é o número de restrições de desigualdade.	67
5.5	Área normalizada sob as curvas dos perfis de desempenho utilizando o melhor valor para 5000 avaliações.	68
5.6	Área normalizada sob as curvas dos perfis de desempenho utilizando a média para 5000 avaliações.	69
5.7	Área normalizada sob as curvas dos perfis de desempenho utilizando o melhor valor para 50000 avaliações.	69
5.8	Área normalizada sob as curvas dos perfis de desempenho utilizando a média para 50000 avaliações.	70
5.9	Área normalizada sob as curvas dos perfis de desempenho utilizando o melhor valor para 500000 avaliações.	72
5.10	Área normalizada sob as curvas dos perfis de desempenho utilizando a média para 500000 avaliações.	72
5.11	Resumo das áreas normalizadas para a <i>suíte</i> de funções envolvendo 5000, 50000 e 500000 avaliações da função objetivo com as métricas melhor e média . .	73
5.12	Resultados encontrados para o problema da mola sob tração/compressão. . . .	74
5.13	Comparação dos resultados para o problema da mola sob tração/compressão. .	75
5.14	Resultados encontrados para o problema do redutor de velocidade.	77
5.15	Variáveis de projeto e peso final para o problema do redutor de velocidade. . .	77

5.16	Resultados encontrados para o problema da viga soldada.	79
5.17	Melhores resultados encontrados para o problema da viga soldada usando diferentes métodos. Estes resultados foram extraídos da referência [7]. <i>^asistemas imunológicos artificiais; ^brecozimento simulado; ^calgoritmos evolucionários; ^dalgoritmo vaga-lume; ^eotimização de forrageamento bacteriana; ^fbusca harmônica; ^gbusca aleatória; ^hbusca em sistema carregado; ⁱmodelo sócio comportamental; ^jalgoritmo de sociedade e civilização; ^kprogramação geométrica; ^levolução diferencial; ^malgoritmo <i>T-cell</i>.</i>	80
5.18	Resultados encontrados para o problema do vaso de pressão.	81
5.19	Resultados estatísticos para o problema do vaso de pressão. Os resultados foram extraídos da referência [7].	82
5.20	Resultados encontrados para o problema da viga engastada e livre.	83
5.21	Problema da viga engastada e livre. Os resultados foram extraídos da Tabela 8 da referência [8]. <i>^aestratégia de evolução baseada em ranqueamento de nicho, ^bcontínuo/aproximado, ^cdiscreto preciso, ^daproximação discreta linear, ^eaproximação discreta conservativa, ^fmétodo de penalização adaptativa, ^gclareira.</i> Os autores são: Chen e Chen [9], Lamberti e Pappalettere [10], Thanedar e Vanderplaats [11], Erbatur <i>et. al</i> [12], Lemonge e Barbosa [13] e Bernardino <i>et. al</i> [14].	84
5.22	Área normalizada sob as curvas dos perfis de desempenho utilizando o melhor valor.	85
5.23	Área normalizada sob as curvas dos perfis de desempenho utilizando a média	86
5.24	Resultados encontrados para a treliça de 10 barras (caso contínuo).	88
5.25	Comparação entre os resultados da literatura e o resultado encontrado neste estudo para o problema da treliça de 10 barras (caso contínuo), com peso final em lb. Os autores são: Aragon <i>et. al</i> [15], Barbosa e Lemonge [13], Bernardino <i>et. al</i> [14] Bernardino <i>et. al</i> [16], Runarsson e Yao [17] e Silva <i>et al.</i> [18].	89
5.26	Resultados encontrados para a treliça de 10 barras (caso discreto).	89

5.27	Variáveis de projeto e peso total encontrados para o problema da treliça de 10 barras (caso discreto). Os autores são: Krishnamoorthy e Rajeev [19], Lemonge e Barbosa [13], Galante [20] e Ghasemi <i>et. al</i> [21].	90
5.28	Agrupamento para a Treliça de 25 barras.	91
5.29	Carregamento para a treliça de 25 barras (em <i>kips</i>).	91
5.30	Resultados encontrados para o problema da treliça de 25 barras.	91
5.31	Comparação com os resultados da literatura para a treliça de 25 barras, com peso final em lb. Os autores são: Krishnamoorthy e Rajeev [19], Lemonge e Barbosa [13], Zhu [22], Erbatur <i>et. al</i> [12], Wu e Chow [23] e Lemonge e Barbosa [13].	92
5.32	Agrupamento para a treliça de 52 barras.	94
5.33	Carregamento para a treliça de 52 barras (em <i>kN</i>).	94
5.34	Resultados encontrado para o problema da treliça de 52 barras.	94
5.35	Comparação dos resultados para o problema da treliça de 52 barras, com peso final em Kg. Os autores são: Lee e Geem [24], Li <i>et. al</i> [25] e Kaveh e Talatahari [26].	95
5.36	Comparação dos resultados para o problema da treliça de 52 barras, com peso final em Kg. As colunas 1 a 4 desta tabela referem-se aos resultados obtidos para as variantes de um algoritmo de colônia de formigas proposto por Capriles <i>et. al</i> [27].	95
5.37	Carregamento para a treliça de 60 barras (em <i>kips</i>).	96
5.38	Agrupamento para a treliça de 60 barras.	97
5.39	Resultados encontrados para o problema da treliça de 60 barras.	97
5.40	Comparação dos resultados para o problema da treliça de 60 barras, com peso final em lb. Os autores são: Barbosa e Lemonge [28], Bernardino [29], Silva <i>et al.</i> [30].	98
5.41	Agrupamento para a Treliça de 72 barras.	99
5.42	Carregamento para a treliça de W é o peso final em lb 72 barras (em <i>kips</i>).	100
5.43	Resultados encontrados para o problema da treliça de 72 barras.	100

5.44	Comparação dos resultados para o problema da treliça de 72 barras, com peso final em lb. As soluções marcadas com um asterisco não são rigorosamente factíveis. Os autores são: Venkaya [31], Gellatly e Berke [32], Schimit e Farshi [33], Erbatur <i>et. al</i> [12] e Lemonge e Barbosa [13].	101
5.45	Área normalizada sob as curvas dos perfis de desempenho utilizando o melhor valor.	102
5.46	Área normalizada sob as curvas dos perfis de desempenho utilizando a média .	102
5.47	Análise das variantes envolvendo todos os experimentos utilizando o melhor valor da função objetivo.	104
5.48	Análise das variantes envolvendo todos os experimentos utilizando o valor da média da função objetivo.	104
A.1	Resultados encontrados para a função G1. Melhor valor conhecido: -15.	128
A.2	Resultados encontrados para a função G2. Melhor valor conhecido: -0.803619.	129
A.3	Resultados encontrados para a função G3. Melhor valor conhecido: -1.000500.	130
A.4	Resultados encontrados para a função G4. Melhor valor conhecido: -30665.538671.	131
A.5	Resultados encontrados para a função G5. Melhor valor conhecido: 5126.496714.	132
A.6	Resultados encontrados para a função G6. Melhor valor conhecido: -6961.813875.	133
A.7	Resultados encontrados para a função G7. Melhor valor conhecido: 24.306209.	134
A.8	Resultados encontrados para a função G8. Melhor valor conhecido: -0.095825.	135
A.9	Resultados encontrados para a função G9. Melhor valor conhecido: 680.630057.	136
A.10	Resultados encontrados para a função G10. Melhor valor conhecido: 7049.248020.	137
A.11	Resultados encontrados para a função G11. Melhor valor conhecido: 0.7499.	138
A.12	Resultados encontrados para a função G12. Melhor valor conhecido: -1.	139
A.13	Resultados encontrados para a função G13. Melhor valor conhecido: 0.053941.	140
A.14	Resultados encontrados para a função G14. Melhor valor conhecido: -47.764885.	141
A.15	Resultados encontrados para a função G15. Melhor valor conhecido: 961.715022.	142
A.16	Resultados encontrados para a função G16. Melhor valor conhecido: -1.905155.	143
A.17	Resultados encontrados para a função G17. Melhor valor conhecido: 8853.539674.	144
A.18	Resultados encontrados para a função G18. Melhor valor conhecido: -0.866025.	145

A.19 Resultados encontrados para a função G19. Melhor valor conhecido: 32.655592.146

A.20 Resultados encontrados para a função G23. Melhor valor conhecido: -400.055099.147

A.21 Resultados encontrados para a função G24. Melhor valor conhecido: -5.508013.148

1 Introdução

Os recursos naturais que o homem utiliza para sua sobrevivência são limitados ao planeta onde vivemos. Por isso, a busca pela otimização deve estar presente a todo momento. Processos de otimização são amplamente utilizados para desenvolver soluções que exploram ao máximo a capacidade esperada de um objetivo. Nesse contexto, a pesquisa de algoritmos adequados para obter as soluções ótimas torna-se relevante.

A otimização pode ser definida como o processo em que se busca encontrar pontos de máximo ou de mínimo, descrita através de uma função chamada função objetivo que contém as variáveis de decisão ou variáveis de projeto. Cada vez mais a otimização tem sido utilizada em problemas de diversas áreas de engenharia, química, física, matemática, economia, entre outras áreas das ciências, onde os responsáveis pela tomada de decisões buscam na otimização ferramentas para maximizar desempenho e minimizar custos.

Uma variedade de meta-heurísticas têm sido desenvolvidas e aplicadas para resolver uma infinidade de problemas de otimização. Uma dessas meta-heurísticas que vem ganhando destaque é a Otimização por Enxame de Partículas (do inglês, *Particle Swarm Optimization* - PSO), que foi apresentada na década de 90 pelos pesquisadores Kennedy e Eberhart [34]. Eles desenvolveram uma técnica inspirada no comportamento social de enxames de partículas, no caso inicial, inspirando-se no vôo dos pássaros. O algoritmo não exige conhecimento prévio sobre o espaço de busca nem sobre a existência de derivadas da função objetivo e/ou das funções de restrição, bastando para sua aplicação a existência de uma função de avaliação dos indivíduos da população. Sua aplicação é bastante vasta, sendo utilizada em problemas em diversas áreas, tais como engenharia, finanças, medicina, biologia, dentre outras.

O algoritmo PSO apresenta duas principais vantagens: tem rápida convergência e usa poucos parâmetros de controle. Contudo, o desempenho do algoritmo depende fortemente dos seus parâmetros.

Na sua grande maioria os problemas de otimização apresentam restrições e uma das formas mais imediatas de tratar esses tipos de problemas é o uso de funções de penalização [35]. As funções de penalização consistem em transformar um problema com restrições em um problema sem restrições, de resolução mais fácil, cuja solução é igual ou se relaciona de

alguma forma com a solução do problema original. Essa mudança tem como base a adição ou subtração de algum valor na aptidão dos indivíduos. Contudo, a tarefa de se obter uma função de penalização adequada não é trivial e depende fortemente das estratégias de penalização adotadas.

Dentre as várias técnicas de penalização destacam-se [36]: estática, dinâmica e adaptativa. A penalização estática depende apenas da definição de um fator externo para a definição do valor a ser somado ou multiplicado à função objetivo. A penalização dinâmica, em geral, tem seus coeficientes de penalização diretamente relacionados ao número de gerações (no caso de um algoritmo genético ou o instante do processo evolutivo). A penalização adaptativa ajusta os valores dos coeficientes de penalização ao longo do processo de evolução.

O Método de Penalização Adaptativa (do inglês, *Adaptive Penalty Method* - APM) foi desenvolvido por Barbosa e Lemonge [37] para o tratamento de problemas de otimização com restrições via um algoritmo genético geracional. Tal estratégia tem demonstrado ser eficiente e robusta [30, 38, 39, 40, 41]. O método descarta a necessidade de qualquer parâmetro definido pelo usuário e baseia-se nas informações obtidas da população, tais como a média da função objetivo e o nível de violação de cada restrição. A ideia principal do método está na definição dos parâmetros de penalização que são mais rigorosos ou não, de acordo com o grau de dificuldade de cada restrição possa ser satisfeita.

Além do método original Barbosa e Lemonge [4] desenvolveram 4 novas variantes. Posteriormente, novas variantes foram desenvolvidas e as principais variações se dão no cálculo do valor da função objetivo e no cálculo das violações dos indivíduos. Uma combinação dessas variações com algumas características das variantes propostas por Barbosa e Lemonge [4] originaram doze novas variantes que foram apresentadas por Garcia *et al.* [5].

Essa dissertação tem como objetivo explorar a capacidade do APM e suas variantes acoplados ao PSO como ferramenta de otimização na busca de soluções para problemas de otimização com restrições, envolvendo funções matemáticas, problemas de engenharia mecânica e estrutural tradicionalmente encontrados na literatura. Pretende-se inicialmente fazer uma comparação entre o APM e suas variantes a fim de obter aquelas que apresentaram o melhor desempenho e utilizá-las na apresentação dos resultados e comparação com outros algoritmos encontrados na literatura.

Silva [18] apresentou uma dissertação que utiliza um PSO clássico como algoritmo evolucionário para o tratamento de problemas de otimização com restrições através do uso do APM e suas variantes preliminares.

Aqui, são destacados aspectos complementares aos do trabalho apresentado em [18], onde faz-se a introdução de novas variantes para o APM, um PSO modificado utilizando uma estratégia para melhorar seu desempenho através da introdução de um operador de loucura (CRPSO) [42], um número maior de problemas testes utilizados nos experimentos numéricos e a utilização dos perfis de desempenho [43] como ferramenta que possibilita obter uma análise mais detalhada e conclusiva de uma quantidade razoável de resultados.

A presente dissertação está dividida em 6 capítulos, incluindo esta introdução que apresenta uma visão geral sobre o tema que será tratado durante este texto. No capítulo 2 é apresentado todo o embasamento teórico necessário para o entendimento do conceito de otimização. Posteriormente, são apresentados conceitos da otimização estrutural, exemplos e aplicações. Ainda no capítulo 2, é apresentada uma descrição sucinta de alguns algoritmos de otimização encontrados na literatura. No capítulo 3 é apresentada a técnica de busca utilizada neste trabalho, conhecida como PSO. Alguns conceitos importantes do algoritmo são abordados, como: histórico, parâmetros, topologia e aplicações. O capítulo 4 apresenta estratégias para o tratamento de restrições em problemas de otimização com restrições encontradas na literatura. Em especial, é apresentado com detalhes o Método de Penalização Adaptativa (APM) e suas variantes. No capítulo 5 são apresentados e analisados os experimentos numéricos, que incluem problemas de otimização oriundos da matemática aplicada, engenharia mecânica e estrutural. Uma análise detalhada dos resultados é feita através de uma ferramenta gráfica conhecida como perfis de desempenho (*Performance Profiles*). Finalmente, no capítulo 6 são apresentadas as considerações finais, conclusões do trabalho e propostas para trabalhos futuros.

2 Otimização

2.1 Introdução

A otimização é um ramo da matemática que se preocupa com a obtenção das condições que dão o valor extremo de uma função (ou várias funções) sob determinadas circunstâncias. Em outras palavras, ela pode ser definida como o processo de encontrar o valor máximo ou mínimo de uma função. Como pode ser visto na Figura 2.1 reproduzida de [2], se um ponto \mathbf{x}^* corresponde ao valor mínimo da função $f(\mathbf{x})$, o mesmo ponto, também corresponde ao valor máximo do negativo da função, $-f(\mathbf{x})$.

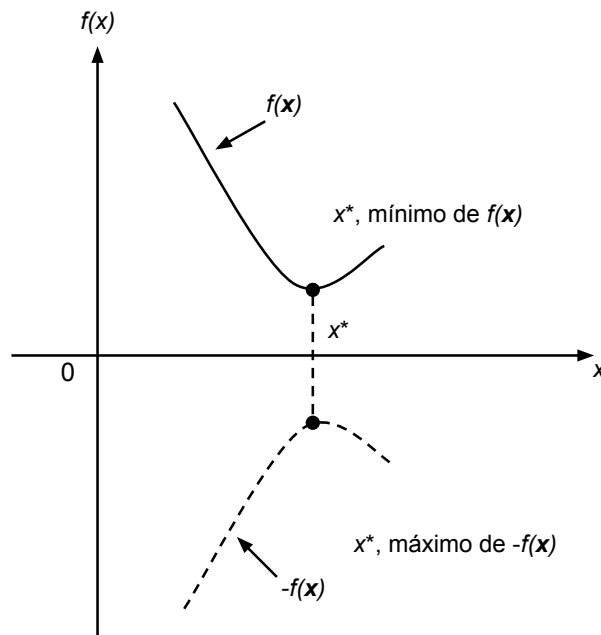


Figura 2.1: Minimizar $f(\mathbf{x})$ é o mesmo que maximizar $-f(\mathbf{x})$.

Para melhor entender a ideia dos problemas de otimização, pode-se fazer uma analogia com a subida de uma montanha. Suponha que a forma da montanha é a função em questão e se quer atingir o seu pico máximo. Em um ato intuitivo, a cada passo dado, inicialmente procura-se uma direção a seguir e depois é decidido o quanto andar. O quanto se anda em um passo, vai depender de quão bem comportado é o terreno da montanha no local. Por exemplo, se há um buraco adiante (ponto de mínimo local) não se quer passar por ele ao final do passo, então deve-se reduzir o passo, e escolher uma nova direção para seguir no passo seguinte, e assim por diante.

Segundo Rao [2], a existência de métodos de otimização foi possível por causa das contribuições de Newton e Leibnitz. Bernoulli, Euler, Lagrange e Weirstrass fundamentaram o cálculo das variações, que lida com a minimização de funções. A otimização para problemas com restrições tornou-se conhecida pelo nome de seu inventor Lagrange, que introduziu os “Multiplicadores de Lagrange” comumente usados em várias técnicas de otimização. Cauchy fez a primeira aplicação para o método da descida mais íngreme para resolver problemas de minimização sem restrições. Apesar destas contribuições, o progresso foi muito pouco até meados do século XX, quando os computadores digitais de alta velocidade foram utilizados para procedimentos de otimização, estimulando a pesquisa de outros métodos.

2.2 A formulação do problema de otimização

Um problema de otimização pode ser matematicamente definido como [44]:

$$\text{Encontrar } \mathbf{x} \text{ que minimize } f(\mathbf{x}) \quad (2.1)$$

sujeito a

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, n_g \quad (2.2)$$

$$h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, n_h \quad (2.3)$$

$$x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, n \quad (2.4)$$

onde \mathbf{x} é o vetor das n variáveis de projeto, $f_i(\mathbf{x})$ é a função objetivo e $g_j(\mathbf{x})$ e $h_k(\mathbf{x})$ são as restrições de desigualdade e igualdade, respectivamente. Os limites das variáveis são determinados através da Equação 2.4, onde x_m^L é o limite inferior e x_m^U é o limite superior da variável x_m . Nas outras expressões, n , n_g e n_h são o número de variáveis de projeto, número de restrições de desigualdade e igualdade, respectivamente.

Comumente, restrições de igualdade são transformadas em restrições de desigualdade da seguinte forma:

$$|h_k(\mathbf{x})| - \epsilon \leq 0 \quad (2.5)$$

onde ϵ é uma tolerância geralmente na casa de 10^{-4} .

Dependendo da escolha das variáveis de projeto, função objetivo e restrições, vários

tipos de problemas de otimização podem ser criados. A partir disso, obtêm-se uma classificação dos problemas de otimização, que pode ser observado com mais detalhes através da Figura 2.2, baseada na referência [1].

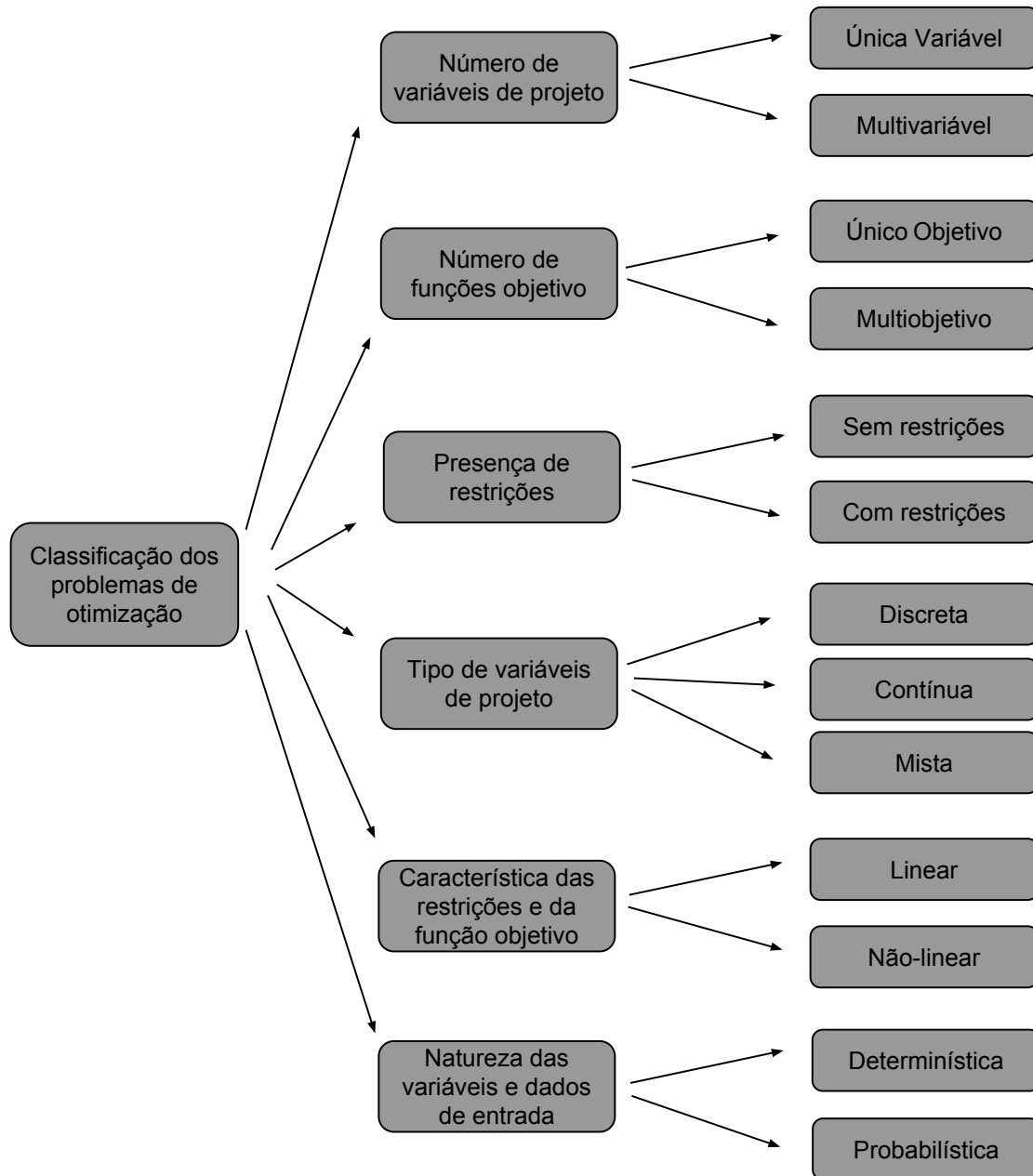


Figura 2.2: Classificação dos problemas de otimização, baseada na referência [1].

A Figura 2.3, reproduzida de [1], apresenta uma classificação completa e detalhada das técnicas de otimização. A técnica a ser usada nesta dissertação é a destacada em negrito.

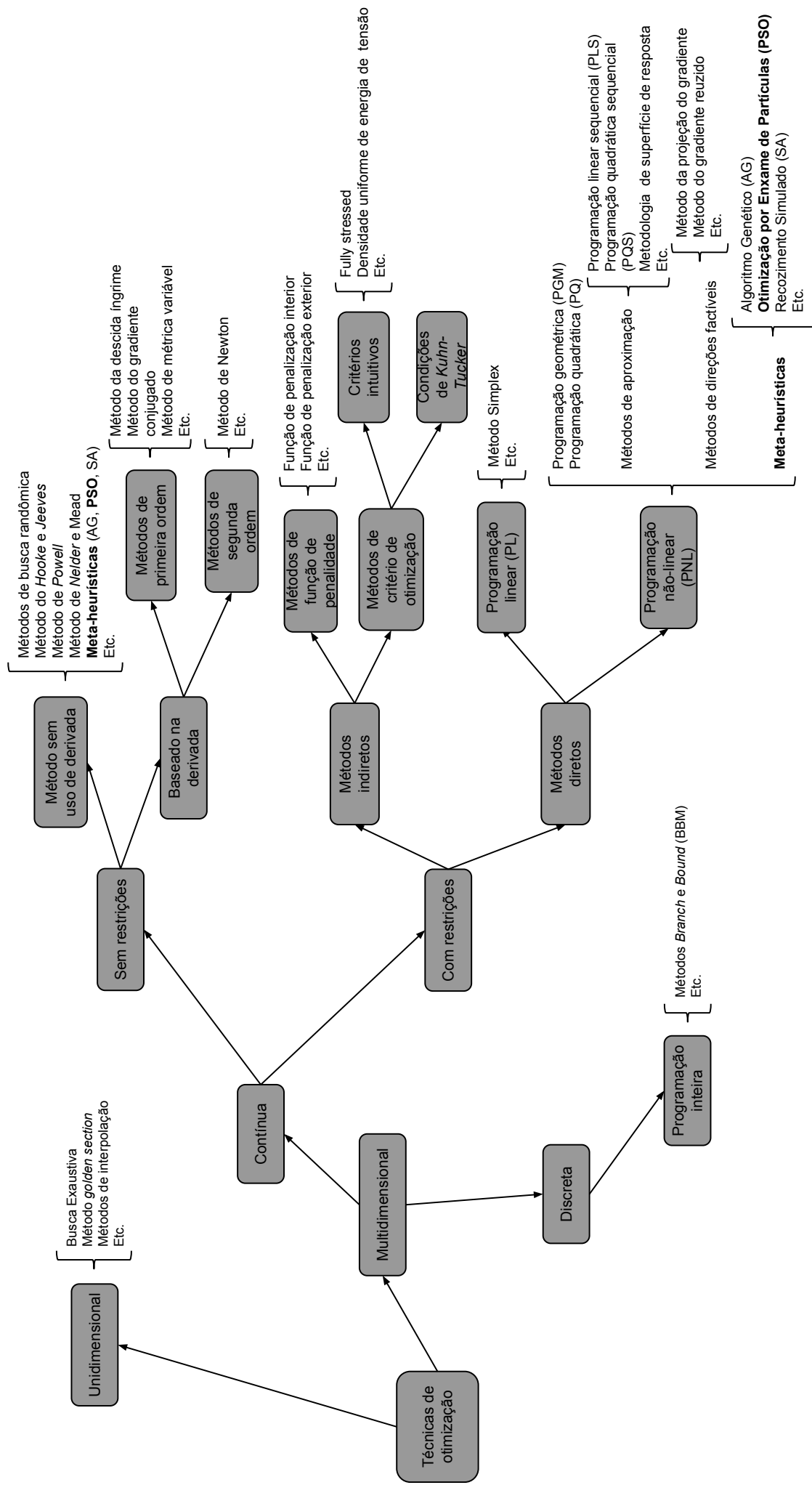


Figura 2.3: Classificação das técnicas de otimização numérica, reproduzida de [1].

2.3 Superfície de restrição em um espaço de projeto bidimensional

Na Figura 2.4, adaptada de [2], mostra-se um espaço de busca hipotético em duas dimensões onde percebe-se a definição de duas regiões sendo uma factível e outra infactível.

Algumas informações são encontradas nessa ilustração como, por exemplo, as definições das restrições do espaço de busca e as características das diversas soluções (pontos) quanto à sua factibilidade ou infactibilidade.

As restrições $g_4(\mathbf{x})$ e $g_5(\mathbf{x})$ são conhecidas como restrições de fronteira ou laterais que definem os limites das variáveis. Nesta ilustração nenhum ponto factível pode ter valor menor que $g_4(\mathbf{x})$ ou menor que $g_5(\mathbf{x})$. Além dessas restrições, presentes em todos os problemas de otimização, podem existir outras como as $g_1(\mathbf{x})$, $g_2(\mathbf{x})$ e $g_3(\mathbf{x})$ que muitas vezes restringem consideravelmente o espaço de busca dificultando significadamente a localização de um ponto factível. Um exemplo de ponto factível é o que se encontra no interior da região factível como mostra a Figura 2.4 e, por outro lado, mostra-se também um ponto infactível mais à esquerda desta ilustração.

Os pontos factíveis que encontram-se sobre as restrições são chamados de pontos de fronteiras admissíveis e que, também por outro lado, podem estar sobre as restrições e não serem factíveis, como também ilustra a figura. Quando uma solução factível está sobre uma fronteira diz-se que a restrição definida por esta fronteira está ativa e situações como esta também são ilustradas na Figura 2.4.

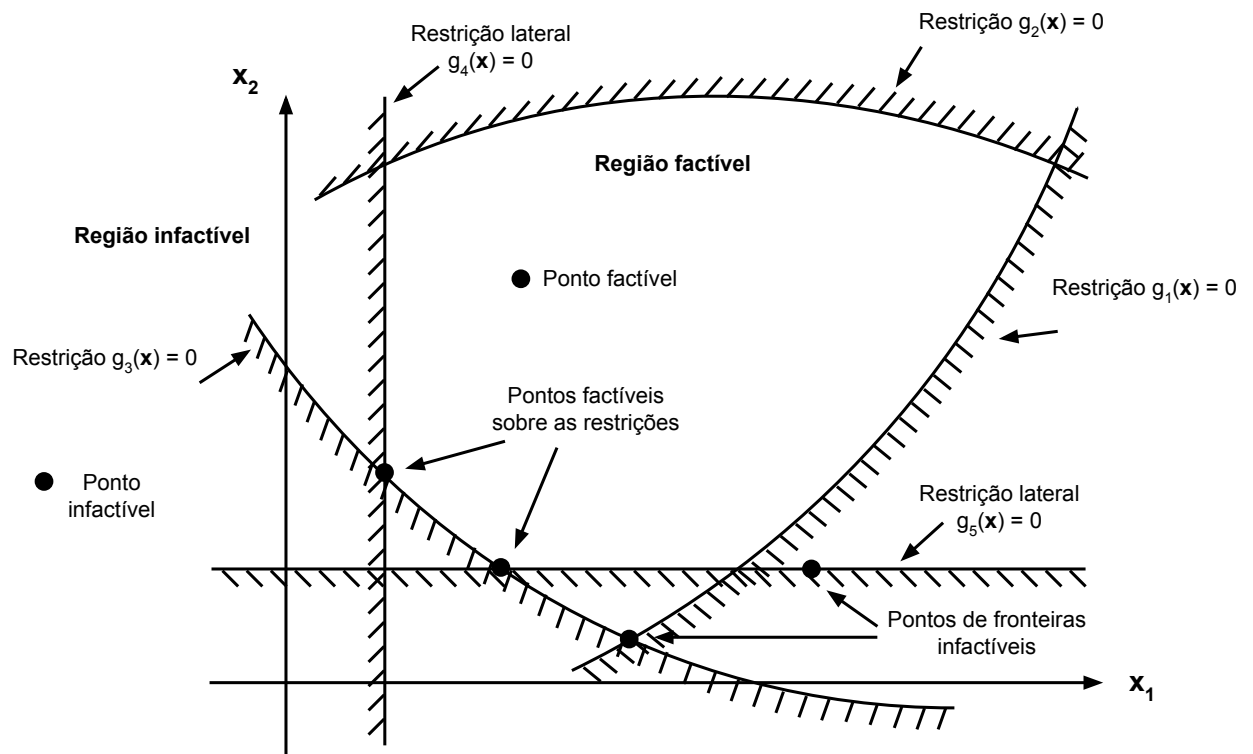


Figura 2.4: Superfície de restrição em um espaço de projeto bidimensional, adaptada de [2].

2.4 Otimização Estrutural

Espera-se da otimização estrutural uma ferramenta que automatiza todo o processo de criação de uma concepção estrutural de forma mais independente possível da experiência dos projetistas. Evidentemente, a experiência do projetista não deve ser desprezada, pois é de extrema importância e pode ser enriquecida com as soluções contra-intuitivas, exequíveis, apontadas pelos processos de otimização.

O conceito de otimização estrutural é mais antigo do que se pensa. De acordo com Gandomi *et al.* [1], o primeiro cientista a aplicar esse conceito foi Maxwell em 1869. Ele formulou uma teoria que estabeleceu a base para o desenvolvimento de projetos de minimização de peso de estruturas. Baseada na teoria de Maxwell, uma nova formulação para a minimização de peso de estruturas reticuladas foi proposta por Michell em 1904.

Historicamente, os problemas clássicos de otimização estrutural, ainda hoje estudados exaustivamente, datam do início da década de setenta e estes tiveram papel significativo no desenvolvimento das técnicas de otimização. Os problemas clássicos de otimização foram formulados envolvendo treliças e pórticos planos, grelhas, placas e cascas, geralmente

considerando o aço como material. As dimensões do problema quanto ao número de variáveis, complexidade da função objetivo e restrições dependia dos recursos computacionais disponíveis à época. As funções objetivo, em quase sua totalidade, diziam respeito a um único objetivo que era a minimização do peso da estrutura e as restrições relacionadas às tensões e deslocamentos para casos estáticos de carregamento [48].

Alguns, entre tantos trabalhos nesta área estão listados na Tabela 2.1 [48].

Tabela 2.1: Alguns dos mais importantes registros históricos na área de otimização estrutural.

Ano	Registro histórico
1950	Artigo de Heyman sobre projeto em regime plástico usando o mínimo de material (<i>Quarterly of Applied Mathematics</i>)
1954	Teorema do Peso Mínimo de Foulkes (<i>Proc. of the Royal Society of London</i>)
1973	Livro editado por Gallagher e Zienkiewicz sobre Projeto Ótimo de Estruturas
1979	Livro sobre Projeto Ótimo Aplicado do Arora (<i>Applied Optimal Design</i>)
1981	Relatório Técnico de Comitê sobre Otimização do ASCE: Recent Developments and Applications, Editado por Ovadia Lev
1997	Relatório Técnico do Comitê sobre Projeto Ótimo de Estruturas: Guia para Otimização Estrutural (incluindo a uma lista de temas, sistemas, e materiais), editado por Jasbir S. Arora
2002	Relatório Técnico do Comitê sobre Projeto Ótimo de Estruturas: Avanços Recentes Em Projeto Ótimo de Estruturas, editado por Scott A. Burns
2004	ATC-63 Projeto e Relatório sobre Quantificação do Desempenho de Edifícios e Parâmetros de Resposta (modelos de arquétipo para avaliação de desempenho)
2007	PEER Otimização de Edifícios Altos
2008	Otimização considerando aspectos de sustentabilidade, meio ambiente e economia de energia

Um problema de otimização estrutural pode ser formulado de diversas maneiras podendo-se variar os seus objetivos e suas restrições. A título de ilustração, na otimização de projetos de estruturas reticuladas como as treliças (barras conectadas por nós), existem três classificações:

- **Dimensional:** utiliza como variável de projeto um parâmetro de um elemento estrutural. No caso de treliças, a variável de projeto é a área da seção transversal. A Figura 2.5 ilustra a solução, onde é possível observar as diferentes áreas otimizadas das barras da estrutura.

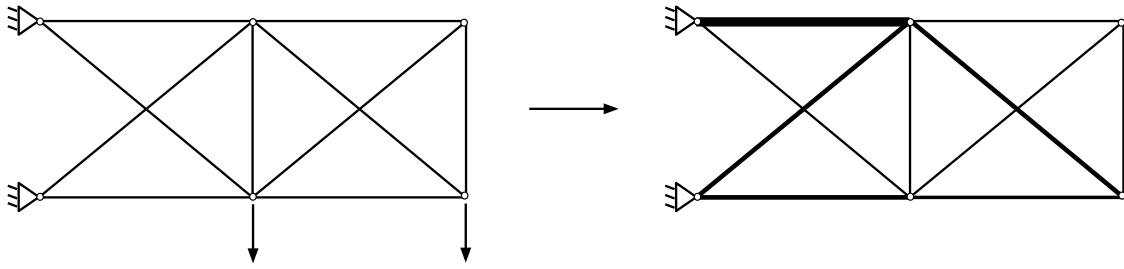


Figura 2.5: Exemplo de um problema de otimização dimensional.

- **Geométrica** (ou de forma): para esse tipo de otimização, a forma da estrutura pode ser alterada, porém o número de barras e suas conectividades devem ser mantidos. Um exemplo de otimização geométrica e dimensional pode ser observado através da Figura 2.6. Nota-se que os nós de apoio e os nós carregados mantiveram-se inalterados e ainda, as áreas das seções transversais também foram consideradas variáveis no processo de otimização. Neste caso, tem-se uma otimização geométrica e dimensional.

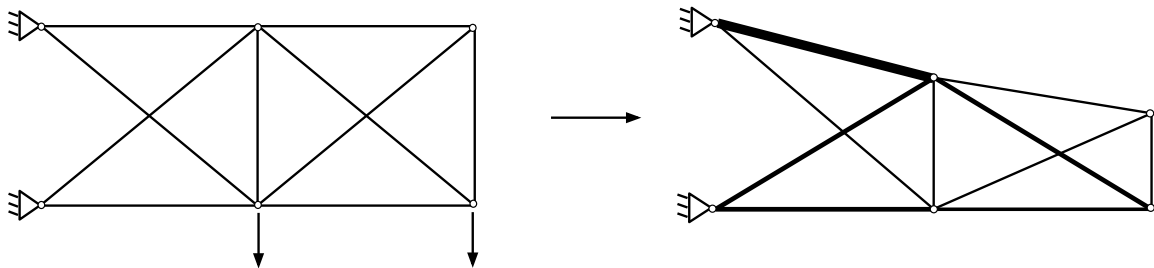


Figura 2.6: Exemplo de um problema de otimização dimensional e geométrica.

- **Topológica**: na otimização topológica permite-se alterar a conectividade bem como a quantidade de barras. A Figura 2.7 ilustra um exemplo de otimização dimensional, geométrica e topológica.

A otimização estrutural tem sido bastante aplicada ao longo dos anos. Lemonge *et al.* [49] utilizaram estruturas reticuladas envolvendo o dimensionamento das variáveis de projeto através do agrupamento de membros, Shimoda *et al.* [50] aplicaram métodos de otimização estrutural geométrica em estruturas de cobertura, Rong e Liang [51] apresentaram um conjunto de métodos de otimização para otimização topológica. Dentre outros exemplos que podem ser encontrados na literatura.

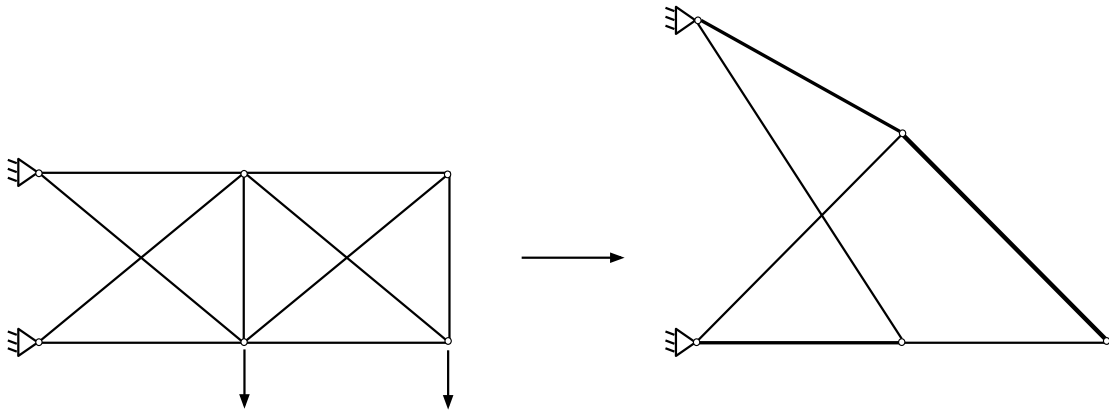


Figura 2.7: Exemplo de um problema de otimização dimensional, geométrica e topológica.

2.4.1 Alguns exemplos de otimização estrutural

Em problemas de otimização estrutural, as restrições podem ser explícitas ou implícitas dependendo do problema. Essas restrições, na maioria das vezes, estão associadas às tensões, deslocamentos, frequências de vibração, entre outras.

2.4.1.1 Problemas com restrições explícitas

Pilar em forma de tubo

O objetivo deste exemplo é dimensionar um pilar com seção transversal em forma de tubo [52] de comprimento L , conforme ilustrado na Figura 2.8. O pilar deve suportar uma carga normal de intensidade P sem provocar tensões normais superiores à máxima permitida e flambagem.

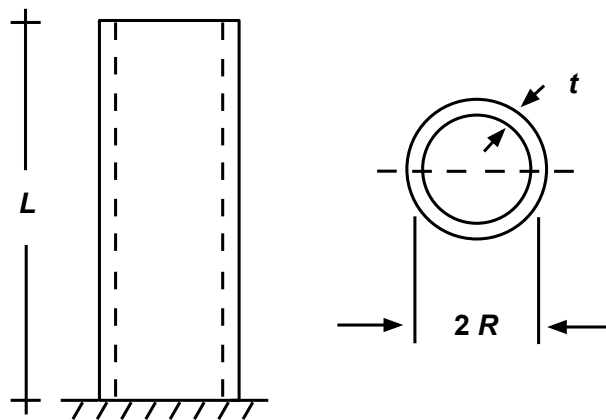


Figura 2.8: Pilar com seção transversal em forma de tubo.

As variáveis de projeto do problema são:

- a espessura da parede do tubo t ;
- o raio médio da seção transversal R ;
- a tensão normal, que é dada por $\sigma = \frac{P}{A}$, onde A é a área da seção transversal do pilar.
- a carga máxima de flambagem, dada por $\pi^2 \frac{EI}{4L^2}$, onde E é o módulo de elasticidade longitudinal do material e I é o momento de inércia da seção transversal.

A função objetivo é definida como:

$$f(R, t) = \rho AL = 2\rho\pi RLt$$

onde R é o raio médio da seção transversal, ρ é o peso específico do material e t a espessura da parede e assumindo que o pilar seja considerado um tubo com paredes finas ($R \gg t$), tem-se:

- área de seção transversal $A = 2\pi Rt$
- momento de inércia da seção transversal $I = \pi R^3 t$

A primeira restrição do problema é a tensão normal σ , que não pode ultrapassar o limite máximo σ_{max} :

$$\sigma = \frac{P}{2\pi RT} \leq \sigma_{max}$$

e a carga máxima aplicada não pode ultrapassar a carga máxima de flambagem:

$$\frac{\pi^3 ER^3 t}{4L^2} \geq P$$

Valores máximos e mínimos são definidos para as variáveis de projeto, como seguem:

$$R_{min} \leq R \leq R_{max} \quad t_{min} \leq t \leq t_{max}$$

O problema de minimização do pilar em forma de tubo é escrito formalmente como:

$$\min f(R, t) = \rho AL = 2\rho\pi RLt$$

sujeito a

$$\frac{P}{2\pi RT} \leq \sigma_{max}$$

$$\frac{\pi^3 ER^3 t}{4L^2} \geq P$$

$$R_{min} \leq R \leq R_{max}, \quad t_{min} \leq t \leq t_{max}$$

2.4.1.2 Problemas com restrições implícitas

Para a otimização de problemas de otimização estrutural em que as restrições são funções implícitas não-lineares das variáveis de projeto, por exemplo, aqueles que envolvem estruturas reticuladas, no caso dimensional, onde procura-se um conjunto x que representa a obtenção da solução de um sistema de equações.

Este sistema de equações, é formulado conforme a Equação 2.6, e representa a equação de equilíbrio da estrutura em sua forma discretizada:

$$K(x)u_l = f_l, \quad 1 \leq l \leq n_l \quad (2.6)$$

onde n_l é o número de casos de carregamento que a estrutura é submetida e $K(x)$ é a matriz de rigidez positiva definida e simétrica da estrutura derivada de uma formulação de elementos finitos [53], conforme Equação 2.7.

$$K = \mathop{\mathbf{A}}_{j=1}^n k_j \quad (2.7)$$

onde \mathbf{A} é o operador usado para acumular a matriz K_j da j -ésima barra e n é o número total de barras.

O vetor de deslocamento e tensão do l -ésimo caso de carregamento é representado por u_l e σ_l , respectivamente.

Para cada caso de carregamento, o sistema é solucionado no campo de deslocamento de acordo com a Equação 2.8.

$$u_l = [K(x)]^{-1} f_l \quad (2.8)$$

A tensão da j -ésima barra para o l -ésimo carregamento é calculada de acordo com a Lei de Hooke (Equação 2.9).

$$\sigma_{j,l} = E\delta(u_l) \quad (2.9)$$

onde δ é a deformação linear específica na direção longitudinal da barra.

Assim, com os deslocamentos dos nós e a tensão de cada barra, as restrições podem ser finalmente verificadas. Lembrando que, ilustrou-se aqui, apenas o caso em que as restrições implícitas envolveram somente deslocamentos e tensões. No caso mais geral, elas podem incluir tensões de flambagem, cargas críticas, frequências de vibração e, assim por diante.

2.5 Algoritmos de Otimização

Os algoritmos usados em um problema de otimização podem ser classificados baseados na natureza determinística ou não-determinística [54].

2.5.1 *Determinísticos*

Os algoritmos determinísticos são aqueles que dada uma determinada entrada, sempre irão produzir a mesma saída. A solução é dependente do ponto de partida fornecido e com isso, o algoritmo pode convergir para uma solução que não é necessariamente a solução ótima global.

Os problemas determinísticos podem ser classificados em duas classes:

- Programação linear: onde a função objetivo e as restrições são funções lineares das variáveis de projeto. Um exemplo clássico da programação linear é o Método Simplex [55].
- Programação não-linear: é quando a função objetivo e/ou pelo menos uma das restrições envolvidas não são funções lineares. Alguns exemplos são: Método de Programação Quadrática Sequencial, Método das Direções Viáveis, Método do Gradiente Reduzido, entre outros.

2.5.2 *Probabilísticos*

Os métodos de otimização baseados nos algoritmos probabilísticos em geral usam somente a avaliação da função objetivo e introduzem no processo de otimização parâmetros estocásticos. Essa classe de algoritmos possui vantagens em relação aos algoritmos determinísticos: a função objetivo e as restrições não precisam estar explicitadas; não

há restrição quanto ao ponto de partida dentro do espaço de busca da solução; trabalham adequadamente com variáveis contínuas e discretas ou uma combinação delas. Contudo, sua principal desvantagem é o custo computacional elevado.

Alguns exemplos de algoritmos probabilísticos, discutidos a seguir, são: Algoritmos Genéticos, Recozimento Simulado, Sistemas Imunológicos Artificiais, Colônia de Formigas, Colônia de Abelhas Artificial, Algoritmo Vaga-lume, Algoritmo *Cuckoo Search* e Enxame de Partículas.

2.5.2.1 Algoritmos Genéticos

O Algoritmo Genético (AG), foi desenvolvido por John Holland [56, 57], e seus colaboradores nos anos de 60 e 70. O algoritmo é classificado com um método probabilístico de busca inspirado na teoria de seleção natural de Charles Darwin e nos mecanismos da genética. Acreditava-se que seria possível a implementação de um algoritmo espelhado no processo de evolução encontrado na natureza.

Um AG usa uma nomenclatura muito próxima da utilizada em genética das populações para definir seus componentes e operadores. Uma população inicial de cromossomos é gerada e cada cromossomo representa uma possível solução para o problema. A população é avaliada, e o cromossomo recebe seu valor de *fitness* (aptidão). Os cromossomos são selecionados para a próxima geração, de acordo com a teoria de Darwin, e transferem suas características para seus descendentes através da reprodução (*crossover* e mutação). Todo esse processo é repetido até que seja encontrada uma solução satisfatória.

Existem várias formas de se representar uma possível solução para o problema a ser resolvido por um AG. A representação binária é uma das mais utilizadas pela sua simplicidade e por poder ser facilmente utilizada em uma grande quantidade de problemas. Outras formas de representação são: real, permutação de símbolos, árvores, entre outros.

Em relação a estratégia de reposição das populações nos AGs, destacam-se dois tipos: geracional, onde toda a população é substituída a cada geração, ocasionando grande variabilidade do material genético entre as gerações; e *steady-state*, onde somente uma parcela da população (geralmente 2 indivíduos) é substituída a cada iteração.

Os AGs são aplicados a uma grande diversidade de problemas [58, 59, 60], tais como: otimização estrutural, otimização de funções multimodais e com restrições, processamento de imagem, controle de sistemas, entre muitos outros.

2.5.2.2 Recozimento Simulado

O Recozimento Simulado ou *Simulated Annealing* (SA) é uma técnica de busca local probabilística, originária da termodinâmica que faz analogia entre o processo físico do resfriamento de um metal em estado de fusão e o problema de otimização. A técnica foi inicialmente desenvolvida por Kirkpatrick *et al.* [61] e é amplamente utilizada na metalurgia.

O algoritmo simula um processo térmico para obtenção de estados de baixa energia em um sólido. Primeiramente, a técnica se inicia com um aumento da temperatura do sólido para um valor máximo, fazendo com que ele se funda. Em um segundo momento, é realizado lentamente o resfriamento até que o resultado final seja uma massa homogênea.

Embora o método tenha sido desenvolvido para problemas discretos, ele pode ser usado para problemas contínuos da mesma maneira, como são usados os AGs. O algoritmo de recozimento simulado foi aplicado para diferentes tipos de problemas de otimização estrutural [62, 63, 64].

2.5.2.3 Sistemas Imunológicos Artificiais

Os Sistemas Imunológicos Artificiais (SIA) [65], são uma meta-heurística que tem constituído uma grande fonte de inspiração para a elaboração de soluções em sistemas computacionais e tem despertado bastante interesse nos pesquisadores da área de ciência da computação.

Vantagens dos SIAs em relação a outras abordagens evolutivas clássicas podem ser destacadas: são capazes de manter a diversidade da população sem nenhum mecanismo adicional; o tamanho da população a cada geração pode ser automaticamente definido de acordo com a demanda da aplicação; e as soluções ótimas tendem a ser preservadas.

Algumas propriedades dos sistemas imunológicos são de grande interesse e fazem com que o sistema seja escalável, robusto e flexível, tais como: unicidade, detecção distribuída, detecção imperfeita, detecção de anomalias e memória.

Existem muitas aplicações em que os sistemas imunológicos podem ser utilizados, como: segurança computacional, aprendizagem de máquina, reconhecimento de padrões, robótica, indústrias, entre outras.

2.5.2.4 Colônia de Formigas

O algoritmo de otimização conhecido como *Ant Colony Optimization* (ACO) foi introduzido por Dorigo e seus associados no começo da década de 90 [66, 67] e imita o comportamento cooperativo de uma colônia de formigas. As formigas utilizam o feromônio para guardar um caminho enquanto elas se movimentam. Isso acontece quando elas saem de sua colônia e caminham aleatoriamente na busca de alimento. Quando o alimento é encontrado, as formigas retornam à colônia deixando um rastro de feromônio para que as outras formigas da colônia possam encontrar o alimento.

Cada solução candidata do algoritmo ACO é representada por uma formiga. Cada formiga escolhe uma rota e deposita uma certa quantidade de feromônio em cada rota percorrida. As melhores rotas seriam identificadas pela quantidade de feromônio.

A otimização por colônia de formigas pode ser aplicada em diversos problemas [68, 69, 70], principalmente em problemas que envolvem procura de caminho em grafos.

2.5.2.5 Colônia de Abelhas Artificial

Artificial Bee Colony (ABC) ou Colônia Artificial de Abelhas foi inicialmente introduzida por Karaboga [71] em 2005 e é um algoritmo de inteligência coletiva baseado no comportamento de uma colônia de abelhas. As abelhas se organizam e desenvolvem uma inteligência coletiva que aumenta o seu desempenho no meio ambiente em que vivem. Três características dessas colônias são particularmente de interesse no desenvolvimento de algoritmos [72]: auto-organização, adaptação e divisão do trabalho.

O algoritmo ABC inicia com uma população de abelhas gerada aleatoriamente. As abelhas são classificadas em três grupos: abelhas operárias campeiras (*scout*), abelhas operárias (*employed*) e abelhas seguidoras (*recruited*). As campeiras são responsáveis por procurar fontes de alimentos. A partir do momento que uma abelha campeira explora uma fonte de alimento, ela se torna uma abelha operária. Quando a fonte de alimento se esgota, a abelha operária volta a ser uma abelha campeira. E as seguidoras são as abelhas que ajudam as operárias a explorar a fonte de alimento.

Algumas variações e aplicações do algoritmo ABC podem ser facilmente encontradas na literatura [73, 74, 75, 76].

2.5.2.6 Algoritmo Vaga-lume

O algoritmo *firefly* (FA) é um algoritmo de otimização baseado na observação da luz de vaga-lumes piscando. O algoritmo foi proposto em 2008 por Yang [77] e foi recentemente aperfeiçoado usando mapas caóticos [78].

O algoritmo baseado em vaga-lumes utiliza três regras: todos os vaga-lumes são considerados assexuados, de modo que um vaga-lume será atraído por outros vaga-lumes, independente do seu sexo; a atração é proporcional ao seu brilho, assim, para quaisquer dois vaga-lumes piscando, o menos brilhante irá se mover em direção ao mais brilhante. Se todos os brilhos forem iguais, os vaga-lumes se movem aleatoriamente. Por fim, o brilho de um vaga-lume é afetado ou determinado pela função objetivo.

O algoritmo *Firefly* é aplicado em diversos problemas, como por exemplo, a otimização de funções multimodais [79], problemas de otimização não-linear ruidosos [80], dentre outros.

2.5.2.7 Algoritmo *Cuckoo Search*

O algoritmo *Cuckoo Search* (CS) é baseado no comportamento reprodutivo de certas espécies de cucos e foi desenvolvido em 2009 por Yang e Deb [81].

O CS utiliza três regras idealizadas: cada cuco coloca um ovo de cada vez e o despeja em um ninho escolhido aleatoriamente; os melhores ninhos com alta qualidade de ovos irão transitar para as próximas gerações; o número de ninhos é fixo e o ovo colocado por um cuco é descoberto por um pássaro hospedeiro com uma probabilidade $p_a \in [0,1]$.

O método é considerado simples e eficiente [82]. Comparando-o com outras meta-heurísticas populacionais, o algoritmo possui apenas um parâmetro para ser determinado e pode ser aplicado com sucesso em problemas de otimização em engenharia [83].

2.5.2.8 Enxame de Partículas

A otimização através de enxame de partículas (PSO) é uma meta-heurísticas bastante conhecida e foi inicialmente proposta por Kennedy e Eberhart [34] em 1995. Essa meta-heurística é a técnica utilizada nessa dissertação e é apresentada com mais detalhes no Capítulo 3.

3 Otimização por Enxame de Partículas

3.1 Introdução

A Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*) é um algoritmo da área de otimização inspirado na natureza e introduzido por Kennedy e Eberhart [34] que se baseia no comportamento social de grupos de indivíduos, como os pássaros e os peixes.

A busca por alimentos e a relação entre pássaros ao longo do vôo podem ser modeladas como um mecanismo de otimização. Fazendo uma analogia, o termo partícula foi adotado para simbolizar os pássaros e representar as possíveis soluções do problema a ser resolvido. A região sobrevoada pelos pássaros é equivalente ao espaço de busca e encontrar o local com comida ou o ninho, corresponde a encontrar a solução ótima.

Para que o bando de pássaros sempre se aproxime do objetivo, ao invés de se perder ou nunca alcançar o alvo utiliza-se a *fitness* ou aptidão, função que avalia o desempenho das partículas. Para alcançar o alvo, sejam os alimentos ou os ninhos, os pássaros fazem uso de suas experiências e da experiência do próprio bando.

O PSO passou por muitas mudanças desde a sua introdução em 1995. Os pesquisadores aprendem sobre a técnica, derivam novas versões, desenvolvem novas aplicações e estudam sobre os efeitos dos vários parâmetros do algoritmo.

Comparado com outras técnicas de computação evolutiva, tais como os Algoritmos Genéticos, o PSO é de implementação fácil, convergência rápida e existem poucos parâmetros para serem ajustados [42], além de não possuir operadores de *crossover* e mutação. O algoritmo tem sido aplicado com sucesso em diversas áreas, tais como: otimização de funções, treinamento de redes neurais artificiais, controle de sistemas nebulosos e outras áreas onde os AG podem ser aplicados. Em contra-partida, o PSO, assim como as outras meta-heurísticas, pode apresentar uma convergência prematura alcançando-se apenas mínimos locais.

O algoritmo PSO possui uma população formada por partículas iniciadas

aleatoriamente, onde cada uma representa uma possível solução para o problema de otimização. Essas partículas são representadas por uma posição x_i e uma velocidade v_i . As partículas também possuem valores de aptidão e voam em um espaço do problema D dimensional, aprendendo com a informação histórica de todas as partículas. Utilizando a informação coletada no processo de busca, as partículas têm a tendência de voar em direção a melhor área ao longo deste processo. A velocidade v_i e a posição x_i da partícula i são apresentados nas Equações 3.1 e 3.2, respectivamente.

$$v_i(t+1) = v_i(t) + c_1 \cdot r_1(x_{pbest} - x_i) + c_2 \cdot r_2(x_{gbest} - x_i) \quad (3.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.2)$$

onde $v_i(t)$ é a velocidade de cada partícula, t é a iteração atual, c_1 e c_2 são constantes de aceleração para controlar a influência da informação cognitiva e social, respectivamente e r_1 e r_2 são números randômicos reais entre 0 e 1 com distribuição uniforme. O índice $pbest$ é a melhor posição de cada partícula até então e $gbest$ a melhor posição entre todas as partículas do enxame. A posição $x_i(t)$ é a posição atual da partícula, $x_i(t+1)$ é a nova posição desta partícula e $v_i(t+1)$ é a nova velocidade da partícula.

O conjunto inicial de partículas x_i é gerado de forma aleatória e espalhado pelo espaço de busca conforme a Equação 3.3

$$x_i^L \leq x_i \leq x_i^U, \quad i = 1, \dots, n \quad (3.3)$$

onde x_i^L e x_i^U contém os limites inferior e superior das posições das partículas, respectivamente e n é o número de partículas. As dimensões das partículas são determinadas pelo problema a ser otimizado, de acordo com o número de variáveis de projeto.

O algoritmo básico de otimização por enxame de partículas pode ser descrito brevemente utilizando os passos observados no Algoritmo 1.

Algoritmo 1: Pseudocódigo do algoritmo PSO.

```

para  $i \leftarrow 0$  até  $\text{tamanhoEnxame}$  faça
  | inicializa  $x_i$  e  $v_i$  aleatórios;
  | inicializa  $pBest_i$ ;
fim
inicializa  $gBest$ ;
enquanto não atingir condição de parada faça
  | para  $i \leftarrow 0$  até  $\text{tamanhoEnxame}$  faça
  | | calcula fitness da partícula  $i$ ;
  | | atualiza  $v_i$  de acordo com a Equação 3.1;
  | | atualiza  $x_i$  de acordo com a Equação 3.2;
  | |  $pBest_i \leftarrow$  melhor entre  $x_i$  e  $pBest_i$  ;
  | fim
  |  $gBest \leftarrow$  melhor entre  $pBest$  e  $gBest$  ;
fim
retorna  $gBest$ ;

```

3.2 Parâmetros

O algoritmo PSO básico possui um número pequeno de parâmetros para serem determinados pelo usuário. Os parâmetros c_1 e c_2 na Equação 3.1 podem mudar radicalmente o comportamento do algoritmo, provocando uma possível instabilidade. Eles são comumente definidos com o valor 2. O número máximo de iterações é normalmente usado como critério de parada.

A velocidade das partículas é limitada através de uma velocidade máxima denominada $vmax$. Se $vmax$ é grande, facilita a exploração global. Por outro lado, se $vmax$ for muito pequena facilitará a exploração local. Shi e Eberhart [84] apresentaram experimentos utilizando $vmax$ fixa e observou-se que o ajuste de velocidade máxima pode ser eliminado pois apresenta problemas: a velocidade máxima ideal é particular do problema e nenhuma regra específica é conhecida. Dessa forma, dois novos parâmetros denominados inércia e fator de aceleração, foram introduzidos com a finalidade de tentar amenizar tais problemas. Eles são discutidos nas subseções 3.2.1 e 3.2.2, respectivamente.

3.2.1 Inércia

Motivados pelo desejo de controlar melhor a exploração global e local, reduzir a importância da $vmax$ ou eliminá-la completamente, Shi e Eberhart [84, 85] propuseram um parâmetro denominado inércia w . A determinação adequada do valor de w proporciona

um equilíbrio entre a exploração global e local, e resulta em uma quantidade de iterações menores para encontrar uma boa solução.

O novo parâmetro w é introduzindo na Equação 3.1, conforme Equação 3.4 [84].

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1(x_{pbest} - x_i) + c_2 \cdot r_2(x_{gbest} - x_i) \quad (3.4)$$

Como o vetor de velocidades é inicializado aleatoriamente, começar o algoritmo com valores maiores para w gera uma busca mais abrangente. E ao longo das iterações, o valor de w vai sendo reduzido gradativamente possibilitando que as partículas encontrem a solução de forma mais rápida. A sugestão proposta por Eberhart e Shi [86] foi utilizar uma variação linear da inércia com um valor inicial de 0.9 e final de 0.4, conforme Equação 3.5.

$$w_i = (w_{ini} - w_{fim}) \frac{(N - i)}{N} + w_{fim} \quad (3.5)$$

onde w_{ini} e w_{fim} são os valores inicial e final, respectivamente, N é o número total de iterações e i é a iteração atual.

Posteriormente, Ratnaweera e Halgamuge [87] apresentaram uma extensão da proposta de Eberhar e Shi [86], onde os coeficientes c_1 e c_2 também variam ao longo das iterações. As Equações 3.6 e 3.7 apresentam a fórmula para o cálculo de c_1 e c_2 , respectivamente.

$$c_{1i} = (c_{1final} - c_{1inicial}) \frac{i}{N} + c_{1inicial} \quad (3.6)$$

$$c_{2i} = (c_{2final} - c_{2inicial}) \frac{i}{N} + c_{2inicial} \quad (3.7)$$

As modificações no cálculo dos coeficientes trouxeram ganhos significativos no desempenho do algoritmo [88, 87].

Naturalmente, outras estratégias podem ser adotadas para ajustar o valor da inércia. Chatterjee e Siarry [89] adotaram uma proposta que pretende melhorar a velocidade de convergência do algoritmo e refinar a busca no espaço multidimensional. A Equação 3.8 apresenta esse esquema, denominado variação não linear da inércia.

$$w_i = \left[\frac{(N - i)^q}{N^q} \right] (w_{ini} - w_{fim}) + w_{fim} \quad (3.8)$$

onde q é o expoente de não linearidade, w_{ini} e w_{fim} são os valores inicial e final, respectivamente, N é o número total de iterações e i é a iteração atual.

3.2.2 Fator de aceleração

Uma nova estratégia, inicialmente estudada por Kennedy [90], denominada fator de aceleração, foi introduzida com a finalidade de controlar a convergência das partículas e eliminar de vez o parâmetro $vmax$. Esse fator, denominado k , pode ser observado na Equação 3.9.

$$k = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (3.9)$$

onde $\phi > 4$.

É comum determinar ϕ com o valor de 4.1 [90] e, assim, a constante k é aproximadamente 0.7298.

Existem diferentes maneiras para a implementação desse fator. Clerc e Kennedy [91] utilizaram uma maneira muito simples, multiplicando k por todos os outros parâmetros, conforme mostra a Equação 3.10.

$$v_i(t+1) = k(v_i(t) + c_1 \cdot r_1(x_{pbest} - x_i) + c_2 \cdot r_2(x_{gbest} - x_i)) \quad (3.10)$$

3.3 Atualização da velocidade

Novas estratégias para o PSO com o objetivo de melhorar o desempenho e a competitividade do algoritmo têm sido propostas na literatura [92, 93]. Uma das modificações estudadas é a maneira com que as velocidades das partículas são atualizadas. A atualização da velocidade é um fator muito importante nos algoritmos PSO pois é ela que determina a velocidade de vôo das partículas no espaço de busca.

Um algoritmo PSO denominado *Craziness based Particle Swarm Optimization - CRPSO*, e introduzido por Kar *et al.* [42], tem como objetivo melhorar o comportamento do algoritmo no que diz respeito a convergência prematura e problemas de mínimos locais. A proposta introduz uma nova expressão para o cálculo da velocidade v_i associado com números aleatórios e um operador *craziness*. E ainda, um parâmetro r_2 é introduzido para tentar controlar o equilíbrio entre a busca global e local. A nova expressão para a

velocidade pode ser observada através da Equação 3.11.

$$v_i(t+1) = r_2 \cdot \text{sign}(r_3) \cdot v_i(t) + (1-r_2)c_1 \cdot r_1(x_{pbest} - x_i) + (1-r_2)c_2(1-r_1)(x_{gbest} - x_i) \quad (3.11)$$

onde r_1 , r_2 e r_3 são números aleatórios entre 0 e 1 gerados com distribuição uniforme e $\text{sign}(r_3)$ é uma função definida como:

$$\text{sign}(r_3) = \begin{cases} -1, & r_3 \leq 0.05 \\ 1, & r_3 > 0.05 \end{cases} \quad (3.12)$$

Em alguns casos raros, pode ocorrer que após o posicionamento das partículas, sua posição pode ser modificada pela Equação 3.2. Porém, um pássaro não pode, devido à inércia, ir em direção a uma região em que se pensa ser viável para a busca de alimento. Em vez disso, o pássaro pode ser conduzido para uma região que está na direção oposta que ele deveria “voar” a fim de alcançar as regiões viáveis. Com essa finalidade, $\text{sign}(r_3)$ é introduzido na expressão.

Em um cardume de peixes ou um bando de pássaros, um peixe ou um pássaro muitas vezes muda a sua direção. Essa característica pode ser descrita como sendo um fator de loucura e é modelada na técnica usando um operador denominado *craziness*. O operador *craziness* é introduzido na técnica para assegurar que uma partícula possa ter uma probabilidade pré-definida de loucura com o objetivo de manter a diversidade entre elas [42]. Desse modo, após a atualização da velocidade (Equação 3.11), o operador *craziness* é introduzido e a velocidade é novamente atualizada, conforme Equação 3.13.

$$v_i(t+1) = v_i(t+1) + P(r_4) \cdot \text{sign}(r_4) \cdot v_i^{\text{craziness}} \quad (3.13)$$

onde r_4 é um número aleatório real entre 0 e 1 gerado com distribuição uniforme, $v_i^{\text{craziness}}$ é um valor aleatório definido uniformemente dentro do intervalo $[v_i^{\text{min}}, v_i^{\text{max}}]$, $P(r_4)$ e $\text{sign}(r_4)$ são definidos, respectivamente, nas Equações 3.14 e 3.15.

$$P(r_4) = \begin{cases} 1, & r_4 \leq Pcr \\ 0, & r_4 > Pcr \end{cases} \quad (3.14)$$

$$\text{sign}(r_4) = \begin{cases} -1, & r_4 \geq 0.5 \\ 1, & r_4 < 0.5 \end{cases} \quad (3.15)$$

onde Pcr é uma probabilidade pré-definida de loucura.

O operador *craziness* possui uma similaridade com o operador de mutação dos algoritmos genéticos, uma vez que aumenta a diversidade do enxame [42]. Com esse operador, é possível explorar locais anteriormente nunca explorados, fazendo com que aumente as chances de encontrar o ótimo global.

Esta técnica de atualização da velocidade, denominada CRPSO, foi introduzida no algoritmo PSO desenvolvido para esta dissertação. O Algoritmo 2 descreve os passos do PSO com a introdução do operador *craziness*.

Algoritmo 2: Pseudocódigo do algoritmo PSO com o operador *craziness*.

```

para  $i \leftarrow 0$  até tamanhoEnxame faça
  | inicializa  $x_i$  e  $v_i$  aleatórios;
  | inicializa  $pBest_i$ ;
fim
inicializa  $gBest$ ;
enquanto não atingir condição de parada faça
  | para  $i \leftarrow 0$  até tamanhoEnxame faça
  | | calcula fitness da partícula  $i$ ;
  | | atualiza  $v_i$  de acordo com as Equações 3.11 e 3.13;
  | | atualiza  $x_i$  de acordo com a Equação 3.2;
  | |  $pBest_i \leftarrow$  melhor entre  $x_i$  e  $pBest_i$  ;
  | fim
  |  $gBest \leftarrow$  melhor entre  $pBest_i$  e  $gBest$  ;
fim
retorna  $gBest$ ;

```

3.4 Topologia das partículas

Outro componente importante que influencia diretamente no desempenho do algoritmo é a topologia de comunicação entre as partículas. A topologia de comunicação determina como as partículas do enxame trocam informações, o que influencia na avaliação da velocidade dessas partículas. As principais topologias utilizadas são: **global** e **local**.

Na topologia global o enxame está organizado de modo que todas as partículas estejam

conectadas entre si. A Figura 3.1 (a) ilustra essa topologia, denominada *gbest*. A topologia global proporciona uma convergência mais rápida, já que a informação da melhor posição é disseminada rapidamente entre todas as partículas do enxame. Em contrapartida, ela pode não garantir a qualidade da solução obtida, pois as partículas podem prender-se em mínimos locais.

Na topologia local, ilustrada pela Figura 3.1 (b), o enxame está organizado em formato de anel e cada partícula possui geralmente dois vizinhos. Embora a troca de informação entre as partículas seja mais lenta do que a global, este tipo de topologia, denominada *lbest*, provê uma melhor qualidade de soluções em comparação com a topologia global.

Entretanto, outros tipos de topologia podem ser utilizadas, tais como: *Von Neumann* [94], árvore [95], roda [96], grafos [3], entre outros.

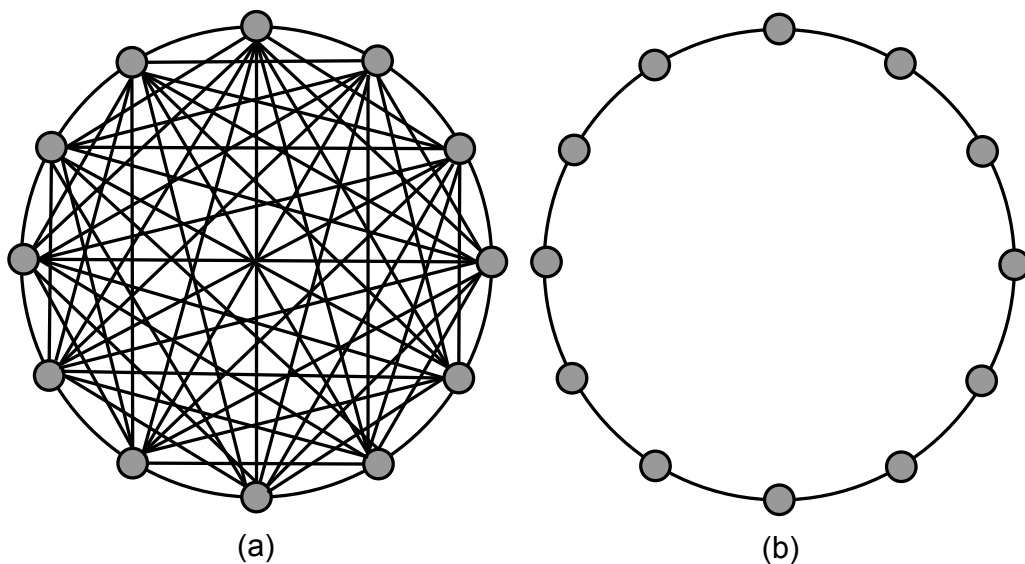


Figura 3.1: Topologia das partículas: (a) Topologia global (b) Topologia local, baseada de [3].

3.5 Aplicações

A Otimização por Enxame de Partículas tem sido aplicada em grande quantidade de problemas. A seguir, segue uma lista com alguns problemas que foram solucionados usando o algoritmo PSO, ou com uma combinação do algoritmo com outra técnica de busca.

Uma aplicação usando o PSO como o algoritmo de busca pode ser observada em [97]

onde aborda-se o problema de planejamento de caminho do robô em ambientes com muitos terrenos.

O volume crescente de dados a serem analisados em grandes redes impõe novos desafios para um sistema de detecção de invasão. Desde que os dados em redes de computadores cresceram rapidamente, a análise dessa grande quantidade de dados tem que ser feita dentro de um período razoável de tempo. Pensando nisso, Aljarahma e Ludwig [98] propuseram um sistema de detecção de invasão baseado em um algoritmo de clusterização paralelo através de um PSO utilizando a metodologia *MapReduce*. Os resultados experimentais em um conjunto de dados reais de invasão demonstraram que o sistema de detecção melhorou significativamente os aspectos de invasão e as taxas de falso alarme.

Um PSO híbrido foi desenvolvido por Yu *et al.* [99] envolvendo a combinação de um PSO com um AG para um modelo ideal de estimativa de demanda de energia (PSO-GA EDE) para a China. Os coeficientes das três formas do modelo (linear, exponencial e quadrática) são otimizados pelo PSO-GA usando fatores, tais como PIB, população, estrutura econômica, taxa de urbanização e estrutura de consumo de energia. O modelo PSO-GA EDE proposto alcançou maior precisão e confiabilidade do que métodos simples de otimização, tais como: AG, PSO, ACO.

Um novo método de otimização baseado na combinação de um PSO com um algoritmo de busca de vizinhança variável (*Variable Neighborhood Search - VNS*), proposto por Hamta *et al.* [100], foi desenvolvido para resolver problemas multi-objetivos de balanceamento de linhas de montagem. Os resultados computacionais indicaram que o algoritmo proposto foi superior, não só a qualidade da solução mas também no tempo de execução, quando comparado com algoritmos da literatura.

Uma aplicação de algoritmo de otimização por enxame de partículas em um sistema de modelagem de carga baseado em medição é apresentado por Rodriguez-Garcia *et al.* [101]. A modelagem de carga composta com base em medições é formulado como um problema de otimização, o que minimiza a diferença entre a medição e a resposta de um modelo simulado. O PSO é usado para estimativa de parâmetros do modelo de carga. Os resultados mostraram a capacidade do modelo de carga composta para representar o comportamento da carga após os distúrbios, e também as capacidades de otimização do PSO para a obtenção de estimativas de parâmetros adequados de carga.

Camp *et al.* [102] aplicaram um PSO híbrido em um projeto de baixo peso de treliças. A eficiência do algoritmo híbrido é demonstrada com vários exemplos e comparações com outros métodos clássicos de otimização. Gomes *et al.* [103] aplicaram um algoritmo PSO para a otimização de forma e dimensão com restrições de frequência. O método alcançou bons resultados comparado a outros métodos da literatura.

Kaveh e Talatahari [104] implementaram uma metodologia denominada HPSACO (*Heuristic Particle Swarm ACO*). A abordagem híbrida baseia-se no esquema de busca harmônica, um PSO e um ACO com o objetivo de encontrar modelos ótimos para diferentes tipos de estruturas. A eficiência do HPSACO foi demonstrada com testes em estruturas treliçadas com domínio de busca contínuo e discreto e em estruturas reticuladas com domínio de busca discreto.

Um algoritmo PSO híbrido (HPSO) baseado em um esquema de busca harmônica e em um PSO padrão foi desenvolvido por Li e Liu [105] e aplicado em estruturas espaciais com variáveis contínuas e discretas. A eficiência do algoritmo é testada em estruturas em conchas. Posteriormente, o HPSO foi testado em estruturas treliçadas com variáveis discretas por Li *et al.* [106].

Perez e Behdinan [107] apresentaram um PSO para problemas de otimização estrutural com restrições. O efeito de diferentes parâmetros de configuração e funcionalidade do algoritmo são apresentados pelos autores. A eficiência da abordagem é ilustrada com três problemas de otimização estrutural e os resultados mostraram a capacidade da metodologia proposta para a obtenção de melhores soluções para os problemas abordados.

4 Problemas com restrições

4.1 Introdução

A modelagem matemática tem sido utilizada para o estudo e compreensão de muitos problemas e fenômenos reais, na engenharia, economia, medicina, entre outras áreas. Os problemas de otimização com restrições são abundantes e a abordagem mais comum em algoritmos evolutivos para lidar com restrições é o uso de penalidades [108].

As funções de penalização foram originalmente propostas por Courant [109] na década de 40 e posteriormente expandida por Carroll e Fiacco [110, 111]. A ideia principal é transformar problemas de otimização com restrições em problemas sem restrições adicionando (ou subtraindo) um certo valor na função objetivo baseado na quantidade de violação das restrições presente em uma determinada solução. Assim, os métodos de penalização permitem a resolução de problemas com restrições por métodos tipicamente utilizados em problemas sem restrições.

A forma geral de uma função de penalização, segundo Mezura-Montes e Coello [35], pode ser definida como:

$$F(\mathbf{x}) = f(\mathbf{x}) + p(\mathbf{x}) \quad (4.1)$$

onde $F(\mathbf{x})$ é a função objetivo expandida a ser minimizada, $f(\mathbf{x})$ é a função objetivo original e $p(\mathbf{x})$ é o valor da penalização que pode ser calculada pela Equação 4.2.

$$p(x) = \sum_{j=1}^{n_g} r_j \cdot \max(0, g_j(\mathbf{x}))^2 + \sum_{k=1}^{n_h} c_k \cdot |h_k(\mathbf{x})| \quad (4.2)$$

onde r_j e c_k são constantes positivas chamadas de fatores de penalização, n_g é o número de restrições de desigualdade e n_h é o número de restrições de igualdade.

As técnicas de penalização podem ser classificadas como **multiplicativas** ou **aditivas**. No caso das multiplicativas, um fator de penalização positivo maior que 1 é utilizado com o intuito de aumentar o valor da *fitness* dos indivíduos ineficazes em problemas de minimização. No caso das aditivas, uma função de penalização é adicionada à função objetivo dos indivíduos ineficazes. O método mais usado para as penalizações aditivas é

definido pela Equação 4.1, tal que

$$p(\mathbf{x}) = \begin{cases} 0, & \text{se } \mathbf{x} \text{ factível} \\ k \sum_{j=1}^m (v_j(\mathbf{x}))^\beta, & \text{caso contrário} \end{cases} \quad (4.3)$$

O parâmetro β da Equação 4.3 usualmente definido como 2, m é o número total de restrições de desigualdade e igualdade e k é o parâmetro de penalização. Vale lembrar que as restrições de igualdade são transformadas em restrições de desigualdade. O valor do parâmetro k pode ser definido por um processo de tentativa e erro geralmente custoso, sendo difícil de ser ajustado adequadamente. A violação $v_j(\mathbf{x})$ é definido conforme a Equação 4.4.

$$v_j(\mathbf{x}) = \begin{cases} |h_j(\mathbf{x})|, & \text{para restrições de igualdade} \\ \max[0, g_j(\mathbf{x})], & \text{caso contrário.} \end{cases} \quad (4.4)$$

As técnicas de penalização aditivas podem ser divididas em dois tipos: **exterior** e **interior**. No caso das técnicas exteriores, inicia-se em uma região infactível e move-se em direção a uma região factível. Nas técnicas interiores, o termo de penalização é escolhido de modo que o seu valor seja pequeno em pontos distantes da fronteira e maiores em pontos perto da fronteira.

Ainda que a implementação de uma função de penalização seja considerada bastante simples, essas funções exigem um ajuste cuidadoso de seus fatores de penalização a fim de determinar o rigor das penalizações a serem aplicadas e, também, são altamente dependentes do problema [112]. Uma das dificuldades dos métodos de penalização é encontrar parâmetros de penalidade convenientes de forma que a solução \mathbf{x} corresponda, de alguma forma, ao mínimo (máximo) do respectivo problema com restrições.

As técnicas de manipulação de restrições baseadas em funções de penalização mostram muita diversidade na maneira de se definir os fatores de penalização. Coello [36] destaca algumas delas: **estática**, **dinâmica** e **adaptativa**.

4.2 Penalização Estática

As penalidades do tipo estática permanecem constantes durante todo o processo evolutivo. A função objetivo penalizada seria, então, a função objetivo da solução candidata mais uma “multa” (para problemas de minimização).

Em 1994, Homaifar [113] propôs a primeira abordagem para o uso de penalização estática, onde o usuário define vários níveis de violação e um coeficiente de penalização é escolhido para cada um de tal forma que esse coeficiente aumenta à medida que os níveis de violação também aumentam. A Equação 4.5 representa a função de penalização estática para um problema com m restrições (restrições de igualdade são transformadas em restrições de desigualdade).

$$F(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^m R_{k,i} \cdot \max[0, g_i(\mathbf{x})^2] \quad (4.5)$$

onde $F(x)$ é a função objetivo penalizada, $f(x)$ é a função objetivo sem penalização, $R_{k,i}$ são os coeficientes de penalização da restrição i e $k = 1, 2, \dots, l$, sendo l o número de níveis de violação das restrições definidas.

Outras abordagem foi proposta em 1996 por Hoffmeister e Sprave [114], e propõe o uso de uma função de penalização conforme a Equação 4.6, onde a aptidão $F(\mathbf{x})$ é dada por:

$$F(\mathbf{x}) = f(\mathbf{x}) + \sqrt{\sum_{i=1}^m H(-g_i(\mathbf{x}))g_i(\mathbf{x})^2} \quad (4.6)$$

onde

$$H(y) = \begin{cases} 1, & y > 0 \\ 0, & y \leq 0 \end{cases} \quad (4.7)$$

Algumas desvantagens podem ser observadas para a penalização estática:

- O número de parâmetros pode aumentar proporcionalmente ao número de restrições;
- Pode ocorrer de se penalizar uma determinada solução candidata com violações pequenas da mesma forma que se penaliza uma outra solução com violações maiores. Isto pode levar uma partícula próxima ao ótimo a se afastar, dificultando a busca da solução desejada.

4.3 Penalização Dinâmica

As penalizações dinâmicas vieram como alternativa de suprir a dificuldade do usuário em determinar os coeficientes de penalização, uma desvantagem das penalizações estáticas.

Nas penalizações dinâmicas, o período ou instante do processo evolutivo está diretamente envolvido com os coeficientes de penalização.

Um método pioneiro utilizando esse tipo de abordagem foi proposto por Joines e Houck [115] em 1994. Os indivíduos são avaliados conforme a Equação 4.8.

$$F(\mathbf{x}) = f(\mathbf{x}) + (C \cdot t)^\alpha \cdot SVC(\beta, \mathbf{x}) \quad (4.8)$$

onde as constantes C , α e β são definidas pelo usuário e $SVC(\beta, \mathbf{x})$ é definida como:

$$SVC(\beta, \mathbf{x}) = \sum_{i=1}^{n_g} D_i^\beta(\mathbf{x}) + \sum_{j=1}^{n_h} D_j(\mathbf{x}) \quad (4.9)$$

$D_i(\mathbf{x})$ e $D_j(\mathbf{x})$ podem ser definidos conforme as Equações 4.10 e 4.11, respectivamente.

$$D_i(\mathbf{x}) = \begin{cases} 0, & g_i(\mathbf{x}) \leq 0 \\ |g_i(\mathbf{x})|, & \text{caso contrário.} \end{cases} \quad 1 \leq i \leq n_g \quad (4.10)$$

$$D_j(\mathbf{x}) = \begin{cases} 0, & -\varepsilon \leq h_j(\mathbf{x}) \leq \varepsilon \\ |h_j(\mathbf{x})|, & \text{caso contrário.} \end{cases} \quad 1 \leq j \leq n_h \quad (4.11)$$

4.4 Penalização Adaptativa

Nas funções com penalização adaptativa, os parâmetros de penalização são atualizados ao longo do processo de evolução, de acordo com informações coletadas da população. Assim, eles podem ser alterados baseando-se nas gerações, no grau de violação das restrições do problema, na função objetivo, entre outros.

Um método de penalização adaptativa que tem seu fator de penalização alterado a cada geração foi proposto por Bean e Hadj-Alouane [116]. A função objetivo é dada conforme a Equação 4.12.

$$F(\mathbf{x}) = f(\mathbf{x}) + \lambda(t) \left[\sum_{j=1}^{n_g} g_j^2(\mathbf{x}) + \sum_{k=1}^{n_h} |h_k(\mathbf{x})| \right] \quad (4.12)$$

onde λ é atualizado a cada geração t da seguinte maneira:

$$\lambda(i+1) = \begin{cases} \frac{1}{\beta_1} \lambda(i), & \text{se } b^i \text{ for sempre factível} \\ \beta_2 \lambda(i), & \text{se } b^i \text{ nunca for factível} \\ \lambda(i), & \text{caso contrário.} \end{cases} \quad (4.13)$$

onde b^i é o melhor elemento da geração i , $\beta_1 \neq \beta_2$ e $\beta_1, \beta_2 > 1$. Neste método, o parâmetro de penalização da geração $i+1$ diminui quando os melhores indivíduos das últimas gerações são factíveis, aumenta se esses indivíduos são infactíveis ou não sofrem nenhuma alteração.

Em 1993, Smith e Tate [117] propuseram uma abordagem que mais tarde foi refinada por Coit e Smith [118] e Coit *et al.* [119], em que a magnitude da penalização é dinamicamente modificada de acordo com a *fitness* da melhor solução encontrada até o momento. A função objetivo penalizada é definida conforme a Equação 4.14.

$$F(\mathbf{x}) = f(\mathbf{x}) + (B_{factive} - B_{todos}) \sum_{i=1}^n \left(\frac{g_i(\mathbf{x})}{NFT(t)} \right)^k \quad (4.14)$$

onde $B_{factive}$ é o melhor valor da função objetivo da geração t , B_{todos} é o melhor valor geral da função objetivo não penalizada da geração t , $g_i(\mathbf{x})$ é o valor acumulado da violação da restrição i , k é uma constante que determina a severidade da penalização e NFT , denominado como Limiar de Viabilidade mais Próximo (*Near Feasibility Threshold*), é definido como o limiar da distância da região factível e infactível do espaço de busca.

Hamida e Schoenauer [120] propuseram uma abordagem que utiliza mecanismos de adaptação ao nível da população. Primeiro, uma função de penalização adaptativa cuida dos coeficientes de penalização de acordo com a proporção de indivíduos factíveis na população atual. Em seguida, uma estratégia de seleção/sedução é usada para combinar os indivíduos factíveis com os infactíveis e, portanto, explorar a região ao redor da fronteira do domínio factível. Por fim, a seleção é ajustada para favorecer um determinado número de indivíduos factíveis.

Foi desenvolvida por Lin e Wu [121] uma estratégia de penalização adaptativa para a manipulação de problemas de otimização com restrições sem a necessidade da busca por valores apropriados de fatores de penalização. O método é baseado na ideia de que a solução ótima é quase sempre localizado na fronteira entre os domínios factível e infactível. A estratégia ajusta automaticamente o valor do parâmetro de penalização usado para cada

restrição de acordo com a razão entre o número de violações de uma restrição específica e o número de restrições satisfeitas.

Uma função de penalização auto-adaptativa proposta por Tessema e Yen [122] para a solução de problemas de otimização com restrições via algoritmo genético utiliza uma nova função objetivo denominada valor de distância $d(x)$.

Inicialmente o método calcula o menor e o maior valor de *fitness* denominados f_{min} e f_{max} , respectivamente. Assim, o valor da *fitness* de cada indivíduo é atualizado subtraindo f_{min} conforme Eq. 4.15.

$$f'(\mathbf{x}) = f(\mathbf{x}) - f_{min} \quad (4.15)$$

onde $f(\mathbf{x})$ é o valor da função objetivo e $f'(\mathbf{x})$ é a nova função objetivo normalizada conforme a Equação 4.16

$$f''(\mathbf{x}) = \frac{f'(\mathbf{x})}{f_{max} - f_{min}} \quad (4.16)$$

onde $f''(\mathbf{x})$ é o valor da função objetivo normalizado.

Realizada essas duas transformações, a função aptidão vai estar entre 0 e 1. A violação da restrição de cada indivíduo inactivável é a média das violações normalizadas, conforme a Equação 4.17.

$$v(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^x \frac{c_j(\mathbf{x})}{c_{max_j}} \quad (4.17)$$

e $c_j(\mathbf{x})$ é definido como:

$$c_j(\mathbf{x}) = \begin{cases} \max(0, g(\mathbf{x})), & j = 1, \dots, k \\ \max(0, |h(\mathbf{x})| - \epsilon), & j = k + 1, \dots, m \end{cases} \quad (4.18)$$

onde ϵ é o valor de tolerância (10^{-4} ou 10^{-5}) e m é o número total de restrições.

Portanto, o valor de distância $d(x)$ utilizado para ordenar os indivíduos é formalizado conforme a Equação 4.19.

$$d(\mathbf{x}) = \begin{cases} v(\mathbf{x}), & \text{se } r_f = 0 \\ \sqrt{f''(\mathbf{x})^2 + v(\mathbf{x})^2}, & \text{caso contrário} \end{cases} \quad (4.19)$$

onde

$$r_f = \frac{\text{número de indivíduos factíveis}}{\text{tamanho da população}} \quad (4.20)$$

Uma nova abordagem denominada ADP (*Automatic Dynamic Penalisation*), foi desenvolvida por Montemurro *et al.* [123] para o tratamento de problemas de otimização com restrições. A ideia do método consiste na possibilidade de explorar a informação contida na população, na geração atual, a fim de orientar a busca por todo o domínio dando uma avaliação adequada aos coeficientes de penalização.

Recentemente, um livro intitulado “Constraint-Handling in Evolutionary Optimization”, que apresenta vários algoritmos e técnicas para resolver problemas de otimização com restrições em várias áreas da engenharia, matemática, química, física, e assim por diante, foi editado por Mezura-Montes [124]. Os temas apresentados neste livro mostram os destaques das pesquisas atuais e as tendências futuras na área de tratamento de restrições em computação evolucionista como, por exemplo: (i) A geração de mecanismos especiais que focam a busca das fronteiras do espaço de busca factível e a importância de soluções infactíveis nos processos evolutivos; (ii) Tratamento de restrições na otimização multi-objetivo; (iii) Controle de parâmetros ótimos para informar o usuário maneiras adequadas de ajustar os mesmos; (iv) Algoritmos híbridos, como busca global-local, combinação de heurísticas baseadas em abordagens e uso de métodos de programação matemática com o objetivo de melhorar o potencial da busca em espaços de busca com restrições; (v) A exploração de novas abordagens bioinspiradas como enxame de partículas, colônia de formigas, sistemas imunológicos artificiais, evolução diferencial, entre outros e (vi) Técnicas para tratamento de restrições com nível razoável de robustez com um pequeno número de avaliações da função objetivo.

4.4.1 Estratégias de penalização aplicadas ao PSO

Vários autores notaram o quão importante é conseguir o equilíbrio adequado entre a técnica de diversidade de controle, a técnica de manipulação de restrições e as características particulares dos algoritmos de busca. A principal desvantagem de um algoritmo PSO é sua convergência prematura, que pode ser aumentada por técnicas de manipulação de restrições que supervalorizam as partículas infactíveis [125]. Com o objetivo de tentar amenizar esses problemas, muitas abordagens são encontradas

na literatura combinando estratégias de manipulação da diversidade com técnicas de manipulação de restrições.

Em 2001, Ray e Liew [126] apresentaram uma estratégia de compartilhamento de informações multinível dentro de um enxame para a manipulação de problemas de otimização mono-objetivo com e sem restrições.

Foi apresentado por Hu *et al.* [127] um algoritmo PSO modificado para problemas de otimização com restrições em engenharia. O algoritmo é iniciado com um grupo de soluções factíveis e uma função de factibilidade é usada para verificar se as soluções recém-exploradas satisfazem todas as restrições. Assim, todas as partículas devem manter apenas as soluções factíveis.

Em 2005, Munoz *et al.* [128] introduziram o algoritmo PESO (*Particle Evolutionary Swarm Optimization*) para resolver problemas de otimização com restrições. Com o objetivo de controlar a convergência prematura e a baixa diversidade, dois novos operadores de perturbação foram introduzidos: “c-perturbação” e “m-perturbação”. O PESO obteve resultados superiores quando comparado a outros algoritmos PSO.

Aguirre *et al.* [125] desenvolveram em 2007 o algoritmo COPSO (*Constrained Optimization via Particle Swarm Optimization*) para a solução de problemas de otimização com restrições mono-objetivo. O COPSO inclui dois novos operadores de perturbação e uma estrutura de vizinhança em anel. Vários experimentos em problemas de engenharia mostraram que o COPSO é robusto, competitivo e rápido. No mesmo ano, Li *et al.* [129] apresentaram um algoritmo denominado *Heuristic Particle Swarm Optimization* (HPSO) para a otimização de estruturas conectadas por pinos. O algoritmo é baseado em um PSO com congregação passiva (PSOPC) [130] e um esquema de busca harmonia (HS) [131]. Os resultados mostraram que o algoritmo HPSO pôde efetivamente acelerar a taxa de convergência e atingir mais rapidamente a solução ótima comparado aos outros algoritmos.

Uma outra abordagem para a manipulação de restrições via PSO, denominada SiC-PSO (*Simple Constrained Particle Swarm Optimization*), adota uma estratégia de manipulação de restrições simples inspirada na estratégia do Deb [132]. As partículas são comparadas em pares no momento da escolha do *pBest*, *gBest* e *lBest*: (i) se duas partículas forem factíveis, é escolhida a que tem o melhor valor da função objetivo; (ii) se duas partículas forem infactíveis, é escolhida a que tiver menor grau de infactibilidade

e (iii) se uma partícula é factível e a outra infactível, é escolhida a partícula factível. Mais detalhes da abordagem podem ser observados em [133]. O esquema proposto se mostrou uma alternativa promissora pois obteve bons resultados com um número baixo de avaliações da função objetivo.

4.5 Um Método de Penalização Adaptativa (APM)

O método de penalização adaptativa, denominado APM (*Adaptative Penalty Method*), foi originalmente desenvolvido por Barbosa e Lemonge [37] para aplicação em problemas com restrições. O método não requer nenhum parâmetro definido pelo usuário, não demanda o conhecimento do problema e usa informação da população, tais como a média da função objetivo e o nível de violação de cada restrição para o cálculo dos coeficientes de penalização.

Em 2003, Barbosa e Lemonge [134] estudaram o comportamento dos parâmetros do APM através de testes utilizando um algoritmo genético. Finalmente em 2004, uma pequena mas importante modificação foi introduzida levando a sua forma atual. Assim, a função de aptidão para um problema de minimização é definida como [135]:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } \mathbf{x} \text{ é factível} \\ \bar{f}(\mathbf{x}) + \sum_{j=1}^m k_j v_j(\mathbf{x}), & \text{caso contrário.} \end{cases} \quad (4.21)$$

onde

$$\bar{f}(x) = \begin{cases} f(\mathbf{x}), & \text{se } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle \\ \langle f(\mathbf{x}) \rangle, & \text{caso contrário.} \end{cases} \quad (4.22)$$

e $\langle f(\mathbf{x}) \rangle$ é a média dos valores da função objetivo da população atual.

A Figura 4.1, adaptada de [4], apresenta soluções factíveis e infactíveis. Dentre o conjunto das 6 soluções infactíveis (1, 2, ..., 6), os indivíduos 1, 2, 3 e 4 (representados por círculos abertos), têm seus valores de função objetivo menores que a média da função objetivo da população e, de acordo com o método APM, o valor de $\bar{f}(\mathbf{x})$ é alterado para a média da função objetivo $\langle f(\mathbf{x}) \rangle$ para estas soluções. Os indivíduos 5 e 6 têm o valor da função objetivo pior do que o da média da população e, assim, $\bar{f}(\mathbf{x}) = f(\mathbf{x})$.

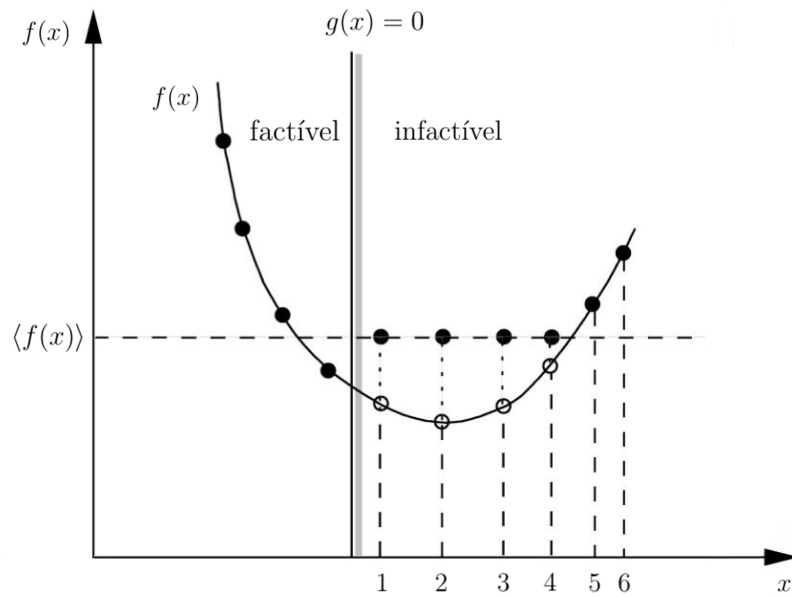


Figura 4.1: Descrição da função \bar{f} , adaptada de [4].

O parâmetro de penalização k_j é definido para cada geração como:

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (4.23)$$

onde $\langle v_j(\mathbf{x}) \rangle$ é o somatório das violações j de todos os indivíduos dividido pelo número de indivíduos da população (factíveis e infactíveis).

A ideia do método é que os valores dos coeficientes de penalização sejam distribuídos de tal forma que as restrições mais difíceis de serem satisfeitas tenham o valor do coeficiente de penalização relativamente mais alto.

O APM tem sido usado em vários trabalhos encontrados na literatura. Alguns deles são citados a seguir. Em [136], Young *et al.* propôs uma nova implementação com destaques para modificações na população corrente de um AG, introduzindo diversidade, para evitar convergência prematura.

Gallet *et al.* [137] desenvolveu um AG com o objetivo de pré-dimensionar um painel enrijecido da fuselagem de uma aeronave submetido a cargas de compressão. As restrições não lineares foram tratadas com o uso do APM.

A implementação de uma abordagem auto-adaptativa através de um AG para aplicações em problemas de otimização com restrições em eletromagnetismo foi apresentada por Rocha e Fernandes em [39] onde o APM foi modificado para tal uso.

O APM também foi acoplado a outros algoritmos evolucionistas como por exemplo

a Evolução Diferencial como máquina de busca acoplada ao APM e suas variantes apresentada por Krempser *et al.* em [30]. Em [40] Venter e Haftka apresentaram um algoritmo do tipo PSO usando uma formulação bi-objetivo usando o APM para o tratamento das restrições.

Em [138] Barbosa *et al.* mostraram um estudo comparativo do desempenho do APM e a técnica do ranqueamento estocástico, usando uma estratégia evolucionista, proposto por Runarsson e Yao [17]. Os métodos foram comparados usando-se a implementação dos mesmo em um AG simples básico sem muitos ingredientes. Um estudo comparativo envolvendo um número razoável de técnicas para tratamento de restrições com aplicação a um problema de otimização envolvendo rotas de dutos submarinos foi apresentado em [139] por Lucena *et al.* Algoritmos híbridos aplicados em problemas de otimização decorrentes da injeção de água para a recuperação de poços de petróleo foram apresentados por Oliveira *et al.* em [140] e o APM foi usado com sucesso para o tratamento das restrições.

O projeto ótimo de fundações onde a configuração desejada envolveu a busca dos melhores agrupamentos através de restrições de cardinalidade usadas por Barbosa e Lemonge em [141], foi proposto por Liu *et al.* em [142]. As restrições do problema foram tratadas com o APM. Yousefi *et al.* [41] usou o APM para tratar as restrições envolvidas em problemas de otimização de projetos de trocadores de calor compactos e o PSO foi o algoritmo de busca adotado. Os mesmos autores deste último trabalho usaram o algoritmo chamado Busca Harmônica com o APM acoplado para resolver problemas de otimização de elementos estruturais usados nestes mesmos trocadores em [143].

4.5.1 Variantes do APM

No ano de 2008 Barbosa e Lemonge [4] apresentaram 4 variações para o APM, que são descritas a seguir:

- *Sporadic* APM (APM_Spor): o método (i) calcula as violações das restrições v_j na população atual, (ii) atualiza os coeficientes de penalidade k_j e (iii) os mantém por uma quantidade fixa de gerações;
- *Sporadic Accumulation* APM (APM_Spor_Acum): o método (i) acumula as violações das restrições por um número fixo de gerações, (ii) atualiza os coeficientes de penalidade k_j e (iii) os mantém por uma quantidade fixa de gerações;

- *Sporadic Monotonic* APM (APM_Spor_Mono): nenhum coeficiente de penalização k_j pode ter seu valor reduzido ao longo das gerações, ou seja, se $k_j^{novo} < k_j^{atual}$ então $k_j^{novo} = k_j^{atual}$;
- *Damping* APM (APM_Damp) : o método é definido como $k_j^{novo} = \theta k_j^{novo} + (1 - \theta)k_j^{atual}$, onde $\theta \in [0, 1]$.

Outras variantes para o APM foram propostas por Garcia *et al.* [5] e posteriormente testadas por Carvalho *et al.* [144]. As variações são enunciadas a seguir:

- Variação 01: $\bar{f}(x)$ é modificada conforme a descrição a seguir:

$$\bar{f}(x) = \begin{cases} f(\mathbf{x}), & \text{se } f(\mathbf{x}) > \lfloor f(\mathbf{x}) \rfloor \\ \lfloor f(\mathbf{x}) \rfloor, & \text{caso contrário.} \end{cases} \quad (4.24)$$

em que $\lfloor f(\mathbf{x}) \rfloor$ é o valor da função objetivo do pior indivíduo factível da população atual. Caso não existam indivíduos factíveis, é utilizada a média da função objetivo;

- Variação 02: no cálculo do fator de penalização k_j , que originalmente utilizava a média da função objetivo, agora adota-se o valor da função objetivo do pior indivíduo factível da população atual, conforme a Equação que segue:

$$k_j = \lfloor f(\mathbf{x}) \rfloor \left| \frac{\langle v_j(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \right| \quad (4.25)$$

- Variação 03: $\langle v_j(\mathbf{x}) \rangle$, que originalmente representava a média das violações de todos os indivíduos em cada restrição, é então definida como o somatório das violações de todos os indivíduos na restrição j dividido pelo número de indivíduos que violam essa restrição;
- Variação 04: $\langle f(\mathbf{x}) \rangle$, que representava a média da função objetivo, agora representa o somatório da função objetivo de todos os indivíduos da população atual dividido pelo número de indivíduos infactíveis, dada pela Equação 4.26.

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

- Variação 05: a modificação ocorre na determinação de $\bar{f}(x)$. Utiliza-se o novo valor da média da função objetivo $\langle\langle f(\mathbf{x}) \rangle\rangle$, conforme descrito na Variação 04. A formulação pode ser observada pela Equação 4.27.

$$\bar{f}(x) = \begin{cases} f(\mathbf{x}), & \text{se } f(\mathbf{x}) > \langle\langle f(\mathbf{x}) \rangle\rangle \\ \langle\langle f(\mathbf{x}) \rangle\rangle, & \text{caso contrário.} \end{cases} \quad (4.27)$$

Na Tabela 4.1 são apresentadas as variantes desenvolvidas por Garcia *et al.* [5] a partir das variações 01 a 05, bem como algumas combinações entre elas.

Tabela 4.1: Variantes propostas por Garcia *et al.* [5] para o APM.

Métodos	Variações				
	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		

Portanto, quando referir-se ao método APM_Med_3, por exemplo, serão consideradas as seguintes modificações no método APM original:

- No cálculo do parâmetro de penalização k_j , $\langle v_j(\mathbf{x}) \rangle$ será o somatório das violações de todos os indivíduos na restrição j dividido pelo **número de indivíduos que violam** essa restrição;
- Ainda no cálculo do parâmetro de penalização k_j , $\langle f(\mathbf{x}) \rangle$, que antes representava a média da função objetivo, agora representa o somatório da função objetivo de todos os indivíduos da população atual dividido pelo **número de indivíduos inactíveis**.

Além dessas variantes, foi modificado o *Sporadic Monotonic* APM, já proposta em [4], com a finalidade de testá-la sem o acúmulo de violações. A nova variante, denominada

Monotonic APM (APM_Mono), funciona da mesma maneira que o APM original sem acumular violações, porém se $k_j^{novo} < k_j^{atual}$ então $k_j^{novo} = k_j^{atual}$.

5 Experimentos Numéricos

5.1 Implementação

Foi desenvolvido um algoritmo PSO em linguagem de programação C utilizando os mesmos parâmetros para todos os testes: codificação real das variáveis, quantidade de partículas igual a 50, $v_i^{craziness} = 0.001$, $c_1 = c_2 = 2.05$, $Pcr = 0.5$, topologia de vizinhança global e tolerância para restrições de igualdade $\varepsilon = 0.0001$;

Os parâmetros utilizados para as variantes do APM são: para a APM_Damp θ recebe o valor de 0.5 e para as variantes APM_Spor_Acum e APM_Spor_Mono o k é atualizado a cada 10 gerações.

Diversos experimentos computacionais foram realizados a fim de testar a eficiência do PSO juntamente com o APM e suas variantes. As funções utilizadas abrangem problemas de otimização em matemática, engenharia mecânica e estrutural. Para os problemas da matemática, foram utilizadas 24 funções conhecidas como *G-Suite*, ou *suite* de funções. Os problemas de engenharia utilizados são: Mola Sob Tração/Compressão, Redutor de Velocidade, Viga Soldada, Vaso de Pressão e Viga Engastada e Livre. Nos testes envolvendo problemas de otimização estrutural, foram utilizadas 5 tipos de treliças, planas e espaciais, sendo elas: treliças de 10, 25, 52, 60 e 72 barras.

Os testes foram feitos num total de 35 execuções independentes e as soluções consideradas foram apenas as factíveis. Os resultados dos problemas propostos são apresentados em tabelas e os melhores resultados são destacados em negrito. As tabelas são apresentadas da seguinte maneira:

- *método*: método utilizado para a busca da solução do problema;
- *melhor*: é o melhor valor encontrado pelo método;
- *mediana*: é o valor central da distribuição dos valores encontrados;
- *média*: é o valor da média aritmética dos valores encontrados;
- *pior*: é o pior valor encontrado pelo algoritmo dentre todos os valores;
- *dp*: desvio padrão;

- *nesf*: número total de execuções que foram encontradas soluções factíveis;
- *na*: número de avaliações da função objetivo.

Uma análise preliminar envolvendo as variantes do APM, enunciadas no capítulo 4, é proposta com o objetivo de encontrar aquelas que obtiveram o melhor desempenho em todos os experimentos e posteriormente utilizá-las na apresentação dos resultados e comparações com outros algoritmos encontrados na literatura. Foram testadas as 4 variantes propostas por Barbosa e Lemonge [4]: *Sporadic* APM (APM_Spor), *Sporadic Accumulation* APM (APM_Spor_Acum), *Sporadic Monotonic* APM (APM_Spor_Mono) e *Damping* APM (APM_Damp), as melhores variantes encontradas em Carvalho *et al.* [144] e a variante denominada APM_Worst.

Garcia *et al.* [5] propuseram 12 novas variantes para o APM, conforme introduzido no capítulo 4. Carvalho *et al.* [144] testaram cada variante proposta acopladas a um algoritmo genético geracional em um conjunto de problemas de engenharia mecânica e estrutural. As melhores variantes encontradas em [144] foram: APM_Med_3 e APM_Med. Portanto, as variantes que serão utilizadas na análise que será apresentada na seção 5.3 são: APM_Spor, APM_Spor_Acum, APM_Spor_Mono, APM_Damp, APM_Worst, APM_Med_3, APM_Med e o APM original.

5.2 Perfis de desempenho

A comparação dos resultados da otimização dos problemas testes é realizada através de uma ferramenta gráfica conhecida como perfis de desempenho (do inglês, *Performance Profiles*). Os perfis de desempenho foram propostos por Dolan e Moré [43] para facilitar a interpretação e visualização dos resultados obtidos em experimentos com grande quantidade de dados.

A avaliação experimental de algoritmos na prática não é nada trivial e apresenta algumas dificuldades, tais como Silva [145]: a definição de conjuntos de problemas é heterogênea, a decisão de como representar e interpretar os resultados obtidos nos experimentos e a determinação de medidas de desempenho para avaliar o algoritmo.

Considere um conjunto P de problemas teste p_j , com $j = 1, 2, \dots, n_p$, um conjunto de algoritmos a_i com $i = 1, 2, \dots, n_a$ e $t_{p,a} > 0$ uma métrica de desempenho (como, por

exemplo, tempo computacional, média, etc.). A razão de desempenho é definida como:

$$r_{p,a} = \frac{t_{p,a}}{\min\{t_{p,a} : a \in A\}}. \quad (5.1)$$

O perfil de desempenho do algoritmo é definido como:

$$\rho_a(\tau) = \frac{1}{n_p} |\{p \in P : r_{p,a} \leq \tau\}| \quad (5.2)$$

onde $\rho_a(\tau)$ é a fração de problemas resolvidos pelo algoritmo com desempenho dentro de um fator τ do melhor desempenho obtido, considerando todos os algoritmos.

Algumas propriedades em relação ao melhor desempenho do algoritmo podem ser observadas em Barbosa *et al.* [146]:

- Quando $\tau = 1$, $\rho_a(\tau)$ é a fração de problemas em que o algoritmo apresenta melhor desempenho quando comparado com os demais algoritmos;
- Quando $\tau = \infty$, $\rho_a(\tau)$ representa a fração de problemas que o algoritmo consegue resolver;
- $\rho_a(1)$ é a porcentagem de problemas em que o algoritmo a tem melhor desempenho. Considerando dois algoritmos E e F , se $\rho_E(1) > \rho_F(1)$ então, o algoritmo E resolve uma quantidade maior de problemas que o algoritmo F .

Uma extensão dessa ferramenta para trabalhar com algoritmos estocásticos foi proposta por Barreto *et al.* [147] e ficou conhecida como perfil de desempenho probabilístico. A ideia era de utilizar uma ferramenta que a princípio foi desenvolvida para ambientes determinísticos em algoritmos estocásticos, já que estes trazem muita incerteza devido aos diferentes desempenhos nas suas diversas execuções. Outras aplicações envolvendo os perfis de desempenho em problemas com restrições, podem ser observadas nas referências [146, 148].

A seguir é apresentado um exemplo do uso dos perfis de desempenho onde são analisados três algoritmos A, B e C para a solução de 5 problemas P1, P2, P3, P4 e P5. Os valores obtidos para cada problema em cada algoritmo é detalhado na Tabela 5.1.

A Figura 5.1 apresenta o gráfico dos perfis de desempenho onde os algoritmos A e C apresentaram o maior valor de $\rho(1)$, o que significa que os métodos obtiveram o melhor

Tabela 5.1: Desempenho dos cinco problemas obtidos através dos algoritmos A, B e C.

	P1	P2	P3	P4	P5
A	10.5	2.9	15.9	0.7	7.7
B	9.3	4.5	16.0	1.1	7.5
C	12.9	2.2	13.6	2.0	6.5

desempenho em um número maior de problemas. O algoritmo C apresentou o menor valor de τ , tal que $\rho(\tau) = 1$, e, portanto, é considerado o mais robusto. A Tabela 5.2 apresenta as áreas sob as curvas dos perfis de desempenho, onde o algoritmo C obteve o maior valor de área e é considerado o método com o melhor desempenho global.

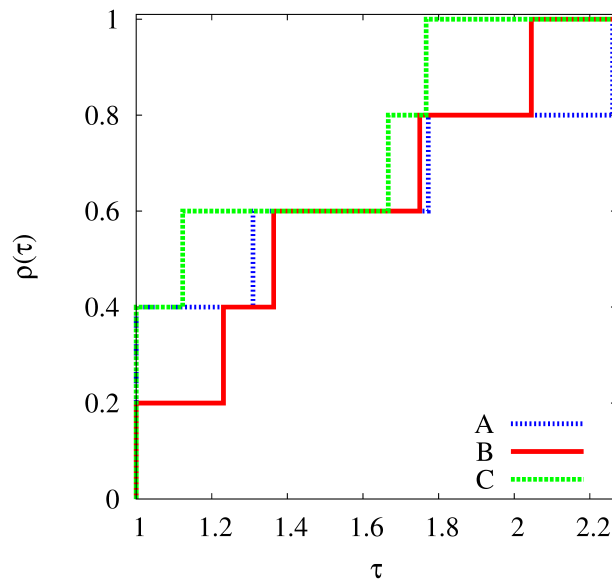


Figura 5.1: Um exemplo do uso da ferramenta Perfis de desempenho.

Tabela 5.2: Área normalizada sob as curvas dos perfis de desempenho do algoritmos A, B e C.

Algoritmo	C	A	B
Área	1	0.83449	0.82442

5.3 Análise das variantes do APM

A análise das variantes é feita utilizando todos os experimentos considerando o valor da **média** da função objetivo. Os perfis de desempenho, introduzido na seção anterior, é utilizado como ferramenta para obter uma análise mais detalhada e conclusiva do grande volume de resultados. A Figura 5.2(a) no intervalo $\tau \in [1; 1.0000005]$ apresenta o gráfico dos perfis de desempenho, onde o APM apresentou o maior valor de $\rho(1)$, o que significa que o método obteve o melhor desempenho em um número maior de problemas. A Figura 5.2(b) apresenta o gráfico dos perfis de desempenho onde o APM_Med_3 apresentou o menor valor de τ , tal que $\rho(\tau) = 1$, e, portanto, é considerado o mais robusto. A Tabela 5.3 apresenta as áreas sob as curvas dos perfis de desempenho, onde o APM_Med_3, APM_Worst, APM e APM_Spor_Mono obtiveram os maiores valores de área e são considerados os métodos com o melhor desempenho global. Portanto, esses métodos serão utilizados na apresentação e comparação dos resultados dos experimentos.

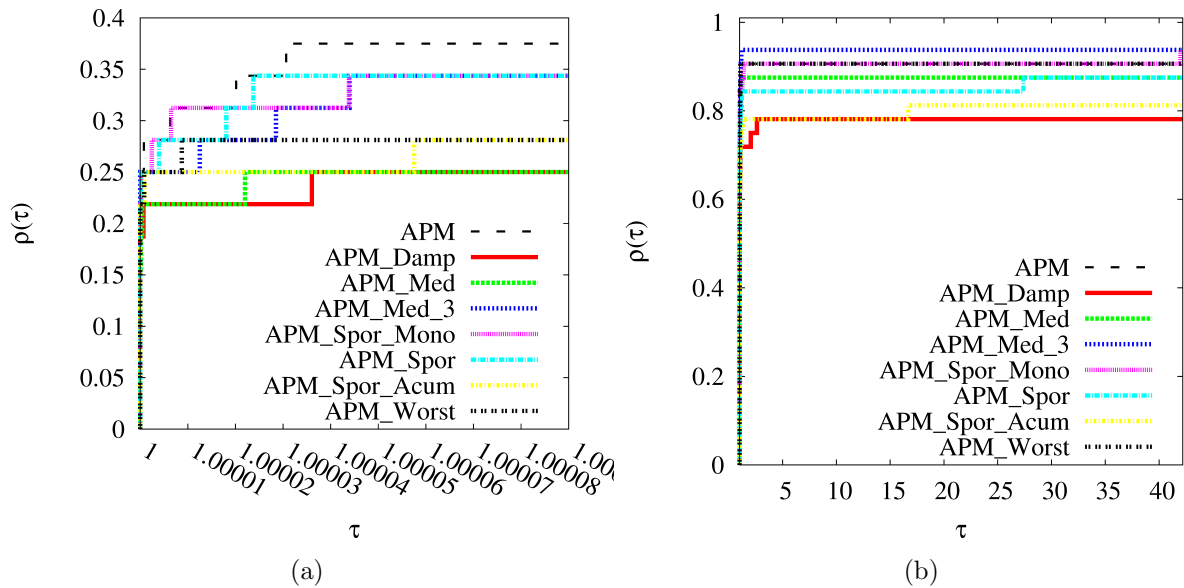


Figura 5.2: Perfis de desempenho das variantes.

Nas seções a seguir são apresentados os resultados dos experimentos computacionais dos quatro métodos divididos em três conjuntos: *suite* de funções, problemas clássicos da engenharia e problemas de otimização estrutural. Ao final de cada seção é apresentado uma análise considerando os melhores métodos para cada conjunto de problemas utilizando como métricas o melhor valor e a média da função objetivo. O volume de resultados apresentados é considerado grande. Dessa maneira, a seção 5.7 apresenta um

Tabela 5.3: Área normalizada sob as curvas dos perfis de desempenho para as variantes.

Método	Área
APM_Med_3	1
APM_Worst	0.9668
APM	0.9664
APM_Spor_Mono	0.9663
APM_Med	0.9333
APM_Spor	0.9118
APM_Spor_Acum	0.8536
APM_Damp	0.8313

resumo envolvendo todas as análises apresentadas nas seções 5.4, 5.5 e 5.6, bem como a variante que obteve o melhor desempenho em todos os experimentos analisados.

5.4 *Suite* de Funções

Um conjunto de problemas teste com 24 funções, denominado *G-Suite* e propostas por Liang *et. al* [6], foi utilizado para analisar a eficiência de cada método. O conjunto é constituído de diferentes tipos de funções que envolvem restrições de igualdade linear, de desigualdade linear e, também, de desigualdade não linear. Detalhes de cada uma das 24 funções testes são apresentados em [6] e podem ser observados na Tabela 5.4.

Os experimentos foram realizados considerando três níveis de avaliações da função objetivo: 5000, 50000 e 500000, comumente usados na literatura para estes experimentos. As três primeiras linhas de cada tabela apresentam os resultados encontrados para 5000 avaliações da função objetivo. As linhas 4, 5 e 6 apresentam os resultados para 50000 avaliações da função objetivo e as três últimas linhas apresentam os resultados para 500000 avaliações da função objetivo. As funções G20, G21 e G22 foram desconsideradas pois não foram encontradas soluções factíveis para os problemas.

Devido ao grande número de estudos encontrados na literatura com as funções *G-Suite* e a grande quantidade de resultados, optou-se por não fazer comparações com outros resultados da literatura. O Apêndice A apresenta os resultados encontrados para cada função nos três níveis de avaliação.

Os resultados com 5000 e 50000 avaliações para a função G5 na Tabela A não são apresentados pois nenhum dos métodos obtiverem soluções factíveis.

Tabela 5.4: Detalhes sobre o conjunto de 24 funções teste extraído de [6]. O número de variáveis de projeto é indicado por n , ρ é a taxa estimada entre a região factível e o espaço de busca, ni é o número de restrições de igualdade e ne é o número de restrições de desigualdade.

Problema	n	Tipo da função	$\rho(\%)$	ni	ne
G01	13	quadrática	0.0111	9	0
G02	20	não linear	99.9971	2	0
G03	10	polinomial	0.0000	0	1
G04	5	quadrática	52.1230	6	0
G05	4	cúbica	0.0000	2	3
G06	2	cúbica	0.0066	2	0
G07	10	quadrática	0.0003	8	0
G08	2	não linear	0.8560	2	0
G09	7	polinomial	0.5121	4	0
G10	8	linear	0.0010	6	0
G11	2	quadrática	0.0000	0	1
G12	3	quadrática	4.7713	1	0
G13	5	não linear	0.0000	0	3
G14	10	não linear	0.0000	0	3
G15	3	quadrática	0.0000	0	2
G16	5	não linear	0.0204	38	0
G17	6	não linear	0.0000	0	4
G18	9	quadrática	0.0000	12	0
G19	15	não linear	33.4761	5	0
G20	24	linear	0.0000	6	14
G21	7	linear	0.0000	1	5
G22	22	linear	0.0000	1	19
G23	9	linear	0.0000	2	4
G24	2	linear	79.6556	2	0

5.4.1 Discussão da suite de funções

A Figura 5.3(a) apresenta o gráfico dos perfis de desempenho no intervalo $\tau \in [1; 1.000005]$ para o *suite* de funções utilizando 5000 avaliações da função objetivo e o melhor valor da função objetivo. É possível verificar que as curvas do APM e da variante APM_Worst apresentaram o maior valor de $\rho(1)$, o que significa que os métodos obtiveram o melhor desempenho em um número maior de problemas. Na Figura 5.3(b) apresenta-se o gráfico dos perfis de desempenho onde os menores valores de τ , tal que $\rho(\tau) = 1$ é também do APM e APM_Worst, fazendo desses métodos os mais robustos. A Tabela 5.5 apresenta as áreas sob as curvas dos perfis de desempenho onde o APM e o APM_Worst novamente apresentam-se em primeiro e segundo lugar, respectivamente. Isso faz com

que eles sejam os métodos com o melhor desempenho global para este cenário.

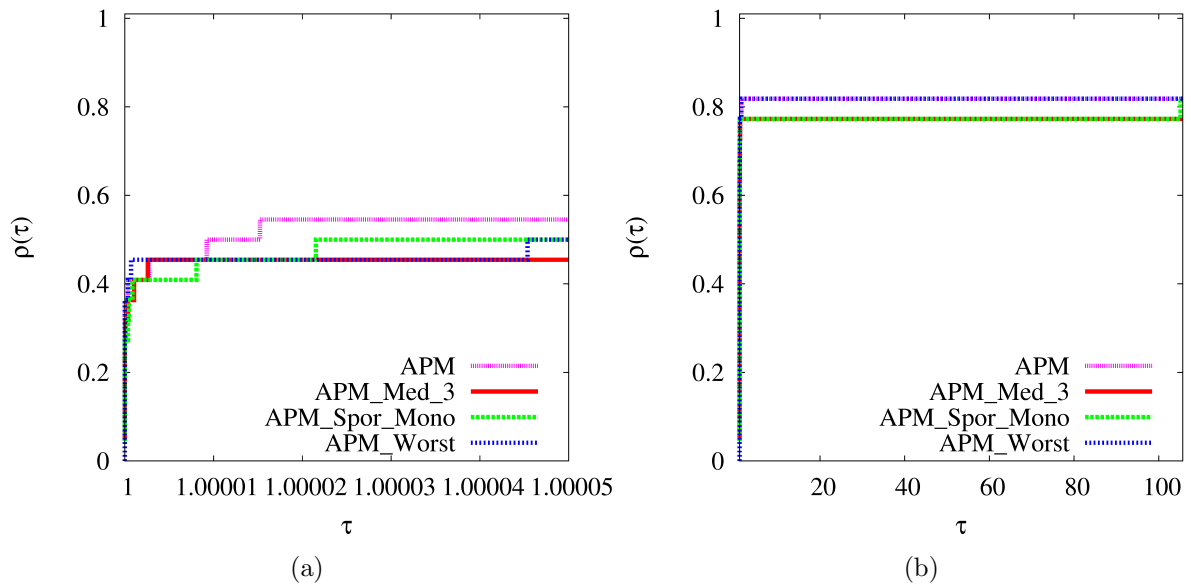


Figura 5.3: Perfis de desempenho utilizando o **melhor** valor para 5000 avaliações.

Tabela 5.5: Área normalizada sob as curvas dos perfis de desempenho utilizando o **melhor** valor para 5000 avaliações.

Método	APM	APM_Worst	APM_Spor_Mono	APM_Med_3
Área	1	0.99949	0.94474	0.94464

Utilizando agora o valor da média nas comparações, o gráfico dos perfis de desempenho é apresentado na Figura 5.4(a) no intervalo $\tau \in [1; 1.000005]$. A variante APM_Worst obteve o melhor desempenho em um número maior de problemas, ou seja, a curva de tal método apresentou o maior valor de $\rho(1)$. A variante APM_Worst e o APM obtiveram os menores valores de τ , tal que $\rho(\tau) = 1$ conforme Figura 5.4 (b), sendo considerados os mais robustos. A Tabela 5.6 apresenta as áreas sob as curvas dos perfis de desempenho com o APM_Worst em primeiro e o APM em segundo lugar. Portanto, para o primeiro nível de avaliações a variante APM_Worst e o APM foram os métodos mais robustos.

Para o segundo nível de avaliações e utilizando o melhor valor, a variante APM_Med_3 obteve o melhor desempenho em uma quantidade maior de problemas como pode ser observado na Figura 5.5(a) no intervalo $\tau \in [1; 1.00000005]$. As variantes que obtiveram os maiores valores para τ , tal que $\rho(\tau) = 1$ e são consideradas as mais robustas para

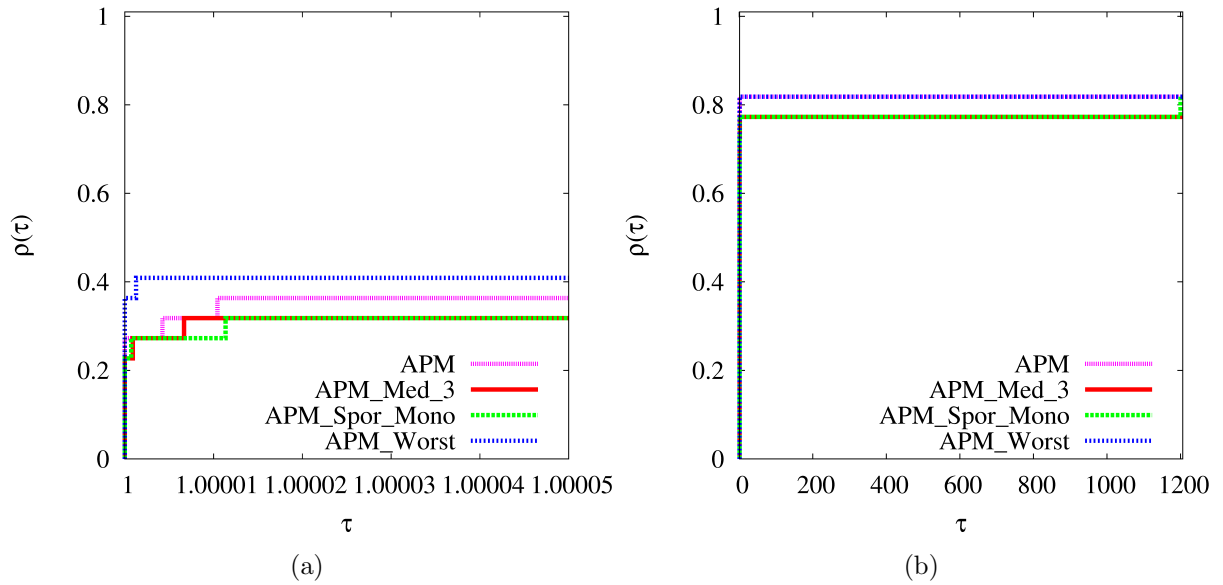


Figura 5.4: Perfis de desempenho utilizando a **média** para 5000 avaliações.

Tabela 5.6: Área normalizada sob as curvas dos perfis de desempenho utilizando a **média** para 5000 avaliações.

Método	APM_Worst	APM	APM_Spor_Mono	APM_Med_3
Área	1	0.99992	0.94439	0.94437

este cenário foram APM_Med_3 e APM_Spor_Mono, conforme Figura 5.5(b). A variante que obteve o maior valor de área, conforme a Tabela 5.7, foi APM_Med_3. Essa variante também resolveu um número maior de problemas, conforme mencionado anteriormente, e, portanto, é considerada a variante como o melhor desempenho global. Contudo, o APM_Spor_Mono ficou muito próximo alcançando o segundo lugar com um valor de área de 0.99999.

Tabela 5.7: Área normalizada sob as curvas dos perfis de desempenho utilizando o **melhor** valor para 50000 avaliações.

Método	APM_Med_3	APM_Spor_Mono	APM	APM_Worst
Área	1	0.99999	0.94736	0.94736

Utilizando ainda 50000 avaliações da função objetivo e nesse momento o valor da média nas comparações, a variante que apresentou o maior valor de $\rho(1)$ foi APM_Worst,

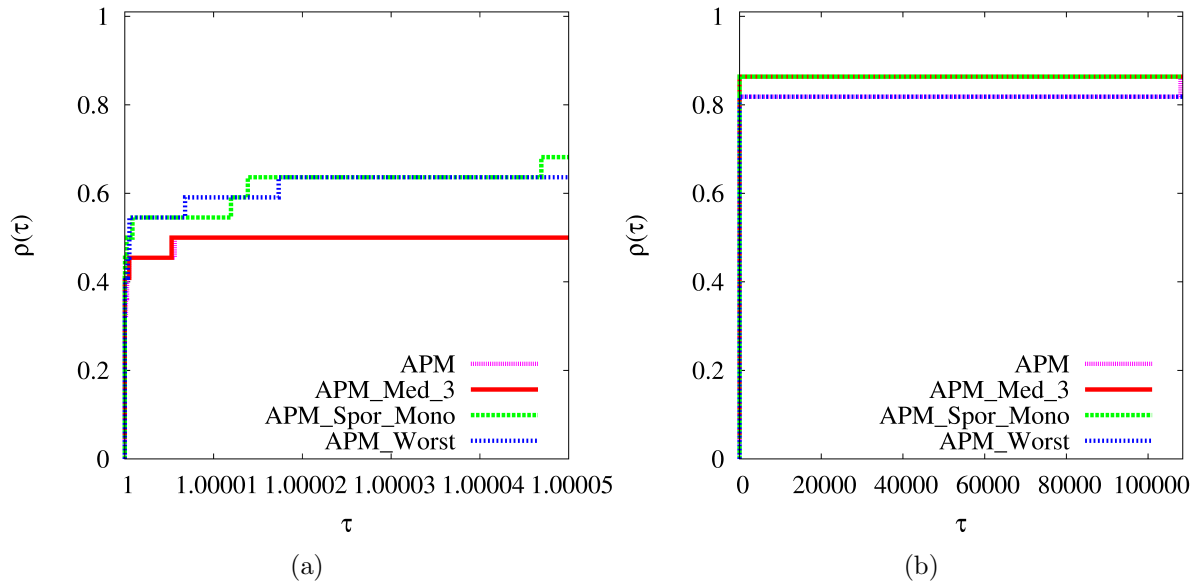


Figura 5.5: Perfis de desempenho utilizando o **melhor** valor para 50000 avaliações.

conforme Figura 5.6(a) no intervalo $\tau \in [1; 1.000005]$, seguida do APM_Med.3. A Figura 5.6(b) apresenta o gráfico dos perfis de desempenho onde o método com o menor valor de τ , tal que $\rho(\tau) = 1$, é considerado o mais robusto. Nesse cenário, os métodos APM_Med.3 e APM_Spor_Mono obtiveram os maiores valores e são considerados os mais robustos. A Tabela 5.8 reforça o que foi concluído a partir da Figura 5.6(b) e apresenta as variantes APM_Med.3 e APM_Spor_Acum com os maiores valores de área, 1 e 0.99995, respectivamente. Portanto, para o segundo nível de avaliações a variante considerada de melhor desempenho é a APM_Med.3.

Tabela 5.8: Área normalizada sob as curvas dos perfis de desempenho utilizando a **média** para 50000 avaliações.

Método	APM_Med.3	APM_Spor_Mono	APM_Worst	APM
Área	1	0.99995	0.894736	0.894736

No terceiro e último nível de avaliações para o *suite* de funções e utilizando o melhor valor, a variante APM_Med.3 obteve o melhor desempenho em um número maior de problemas conforme gráfico dos perfis de desempenho apresentado na Figura 5.7(a) no intervalo $\tau \in [1; 1.000005]$ e obteve o menor valor de τ , tal que $\rho(\tau) = 1$, conforme Figura 5.7(b). Portanto, ela é considerada a variante mais robusta. Analisando a área

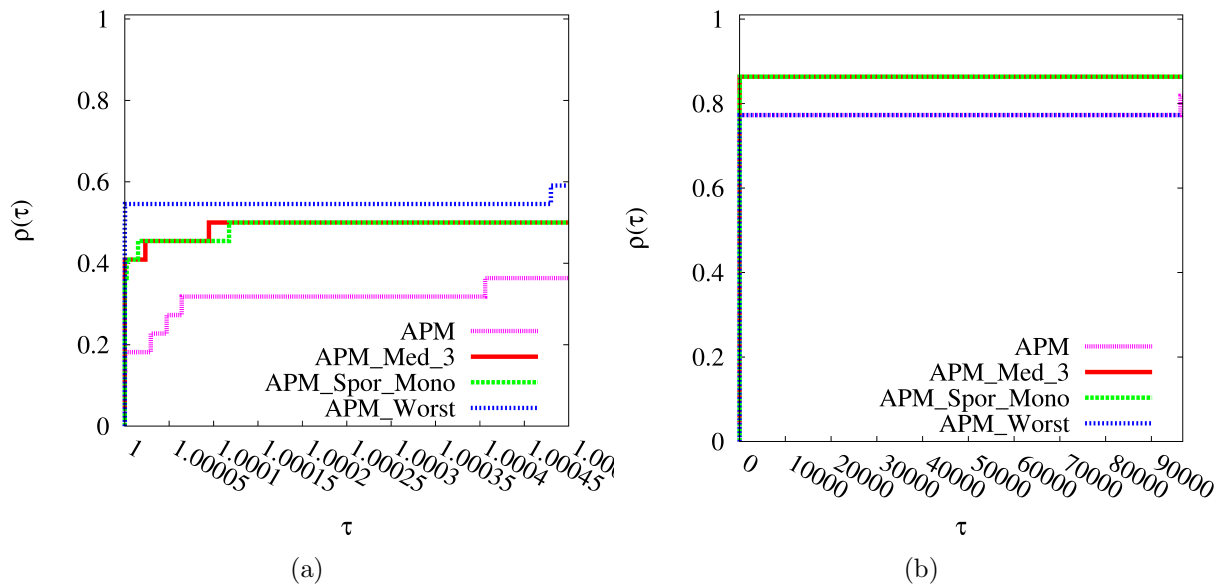


Figura 5.6: Perfis de desempenho utilizando a **média** para 50000 avaliações.

sob as curvas dos perfis de desempenho através da Tabela 5.9, as variáveis APM_Med_3 e APM_Spor_Mono obtiveram os maiores valores de área, respectivamente.

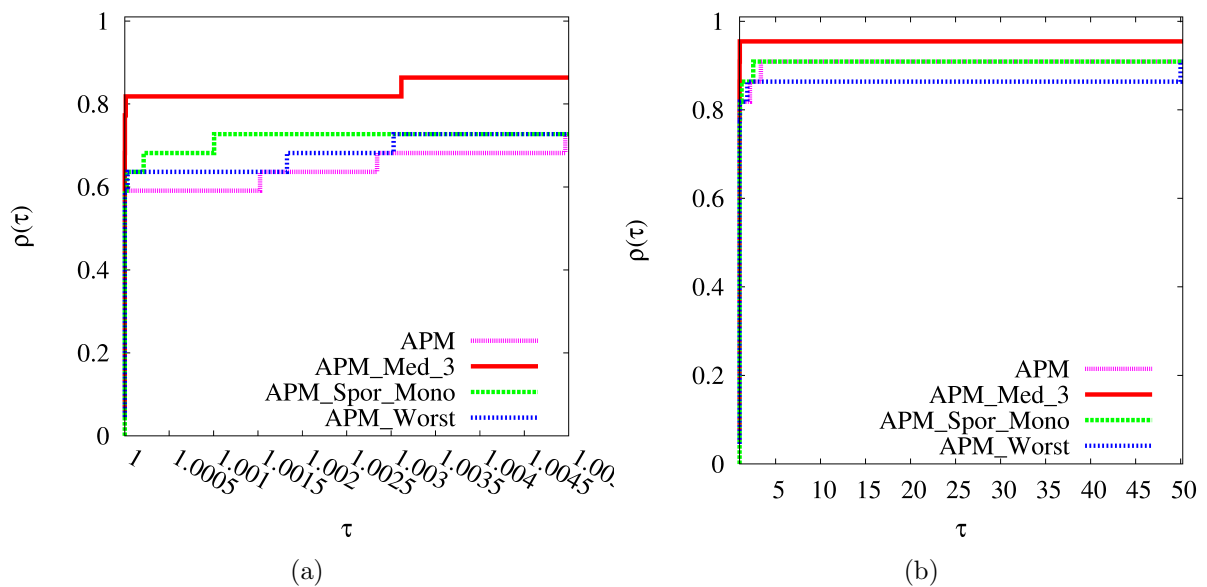


Figura 5.7: Perfis de desempenho utilizando o **melhor** valor para 500000 avaliações.

Por fim, utilizando o valor da média nas comparações, a Figura 5.8(a) no intervalo $\tau \in [1; 1.000000005]$ apresenta o método APM_Med_3 juntamente com o APM_Spor_Mono como aqueles que obtiveram o melhor desempenho em uma quantidade maior de problemas. Na Figura 5.8(b) o APM_Med_3 mostrou melhor desempenho apresentando

Tabela 5.9: Área normalizada sob as curvas dos perfis de desempenho utilizando o **melhor** valor para 500000 avaliações.

Método	APM_Med_3	APM_Spor_Mono	APM	APM_Worst
Área	1	0.949340	0.94902	0.90404

o menor valor de τ , tal que $\rho(\tau) = 1$ e é considerável o método mais robusto. A Tabela 5.10 apresenta os valores das áreas sob as curvas dos perfis de desempenho e comprova a qualidade do método APM_Med_3 para esse nível de avaliação, com um valor de área igual a 1. Essa variável é portanto considerada a que possui o melhor desempenho global para esse cenário.

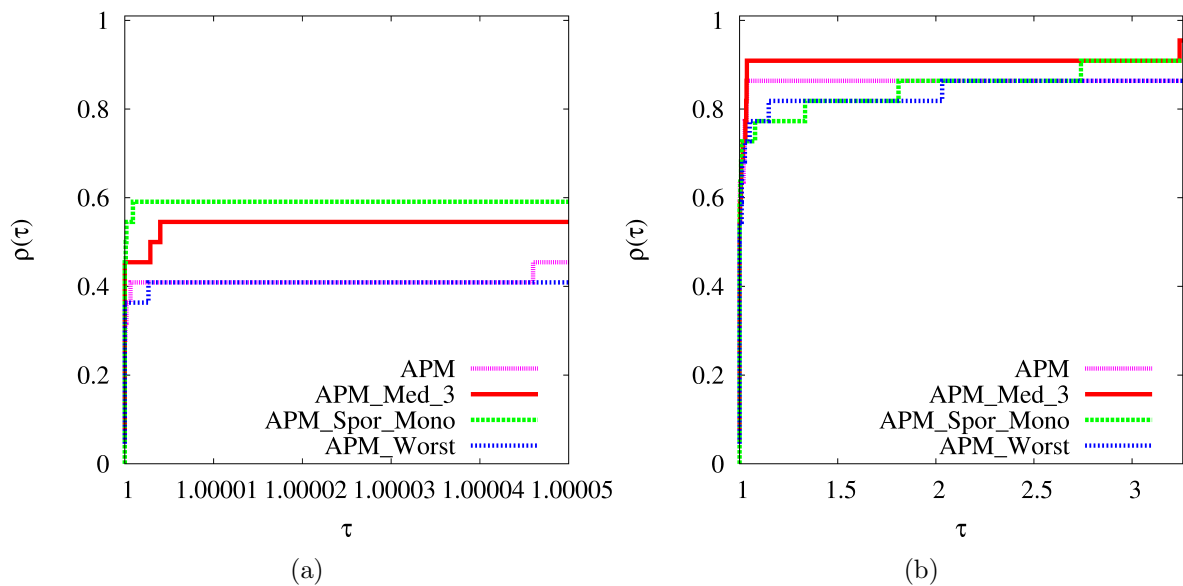


Figura 5.8: Perfis de desempenho utilizando a **média** para 500000 avaliações.

Tabela 5.10: Área normalizada sob as curvas dos perfis de desempenho utilizando a **média** para 500000 avaliações.

Método	APM_Med_3	APM	APM_Worst	APM_Spor_Mono
Área	1	0.95185	0.92695	0.90824

Um resumo envolvendo as análises da *suite* de funções para os três níveis de avaliação da função objetivo 5000, 50000 e 500000 com as métricas **melhor** e **média** é apresentado na Tabela 5.11. Observa-se que a variante APM_Med_3 não obteve um bom desempenho para o primeiro nível de avaliação, contudo, ela obteve o primeiro lugar nas outras 4 análises e é considerada a variante com o melhor desempenho global para essa classe de problemas.

Tabela 5.11: Resumo das áreas normalizadas para a *suite* de funções envolvendo 5000, 50000 e 500000 avaliações da função objetivo com as métricas **melhor** e **média**.

Área	5000		50000		500000	
	Melhor	Média	Melhor	Média	Melhor	Média
APM	1	0.99992	0.94736	0.89736	0.94902	0.95185
APM_Med_3	0.94464	0.94437	1	1	1	1
APM_Worst	0.99949	1	0.94736	0.894736	0.90404	0.92695
APM_Spor_Mono	0.94474	0.94439	0.99999	0.99995	0.949340	0.90824

5.5 Problemas clássicos da engenharia

5.5.1 Mola Sob Tração/Compressão

O objetivo do problema é minimizar o volume V da mola sob tração ou compressão [149], conforme ilustrada na Figura 5.9. As variáveis de projeto são o número de espirais ativos da mola ($N = x_1 \in [2, 15]$), o diâmetro de cada volta ($D = x_2 \in [0.25, 1.3]$) e o diâmetro do arame ($d = x_3 \in [0.05, 2]$). O volume e as restrições mecânicas são dadas por:

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

onde

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

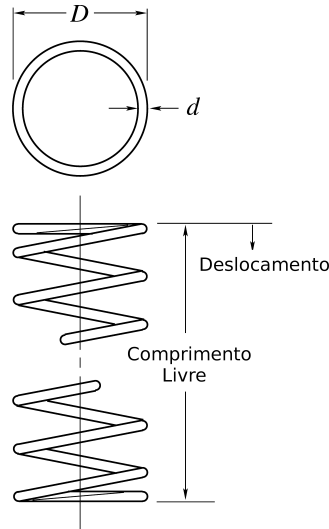


Figura 5.9: Mola sob tração/compressão.

Adotou-se um valor de 36000 para o número de avaliações da função objetivo. A Tabela 5.12 apresenta os resultados encontrados para o problema. A variante APM_Med_3 e APM_Spor_Mono obtiveram o melhor volume final igual a 0.01266. Contudo, os outros dois métodos encontraram valores similares.

Tabela 5.12: Resultados encontrados para o problema da mola sob tração/compressão.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	0.01267	0.01311	0.01354	6.9509e-03	0.01742	35/35
APM_Med_3	0.01266	0.01312	0.01389	9.1731e-03	0.01777	35/35
APM_Worst	0.01267	0.01288	0.01393	9.6090e-03	0.01777	35/35
APM_Spor_Mono	0.01266	0.01306	0.01392	8.6313e-03	0.01734	35/35

Na Tabela 5.13 e nas tabelas seguintes “Este estudo” significa o melhor resultado obtido entre o APM e as variantes APM_Med_3, APM_Worst e APM_Spor_Mono. A Tabela 5.13 apresenta uma comparação dos valores encontrados para as variáveis de projeto com diferentes resultados da literatura.

Tabela 5.13: Comparação dos resultados para o problema da mola sob tração/compressão.

Autores	V	d	D	N
Aragon <i>et al.</i> [15]	0.01267	0.05162	0.35511	11.3845
Barbosa and Lemonge [150]	0.01268	0.05117	0.3443	12.0707
Belegundu [151]	0.01283	0.05	0.3159	14.25
Bernardino <i>et al.</i> [14]	0.01267	0.05166	0.35603	11.3296
Bernardino <i>et al.</i> [16]	0.01267	0.05143	0.35053	11.6612
Coello [152]	0.0127	0.05148	0.35166	11.6322
Coello [153]	0.0127	0.05148	0.35166	11.6322
Coello and Becerra [154]	0.01272	0.05	0.3174	14.0318
Coello and Montes [155]	0.01268	0.05199	0.36397	10.8905
Dos Santos Coelho [156]	0.01267	0.05151	0.35253	11.5389
He and Wang [157]	0.01267	0.05173	0.35764	11.2445
He <i>et al.</i> [158]	0.01267	0.05169	0.35675	11.2871
Hedar and Fukushima [159]	0.01267	0.05174	0.358	11.2139
Hu <i>et al.</i> [160]	0.01267	0.05147	0.35138	11.6087
Huang <i>et al.</i> [161]	0.01267	0.05161	0.35471	11.4108
Montes and Coello [162]	0.01270	0.05164	0.35536	11.3979
Montes and Ocana [163]	0.01267	0.05183	0.35994	11.1071
Parsopoulos and Vrahatis [164]	0.01312	–	–	–
Ray and Liew [165]	0.01267	0.05216	0.36816	10.6484
Ray and Saini [166]	0.01306	0.05042	0.32153	13.9799
Runarsson and Yao [17]	0.01268	0.05164	0.35549	11.3758
Zhang <i>et al.</i> [167]	0.01267	0.05169	0.35672	11.289
Gandomi <i>et al.</i> [7]	0.01267	0.05169	0.35673	11.2885
Este estudo	0.01266	0.05406	0.41655	8.48436

5.5.2 Redutor de Velocidade

O objetivo é minimizar o peso W de um redutor de velocidade [149], ilustrado na Figura 5.10. As variáveis de projeto são a largura da face ($b = x_1 \in [2.6, 3.6]$), o módulo dos dentes ($m = x_2 \in [0.7, 0.8]$), o número de dentes ($n = x_3 \in [17, 28]$), o tamanho da haste 1 entre os suportes ($l_1 = x_4 \in [7.3, 8.3]$) e o tamanho da haste 2 entre os suportes ($d_2 = x_7$). A variável x_3 é inteira e as demais são contínuas. As restrições incluem limitações da tensão de flexão e de superfície da engrenagem de dentes, deslocamento transversal das hastes 1 e 2 gerado pela força transmitida e as tensões nas hastes 1 e 2. O peso e as restrições mecânicas podem ser dadas por:

$$W = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3)$$

$$g_1(x) = 27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1$$

$$g_2(x) = 397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1$$

$$g_3(x) = 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1$$

$$g_4(x) = 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1$$

$$g_5(x) = \frac{1}{0.1x_6^3} \left[\left(\frac{745x_4}{x_2x_3} \right)^2 + \{16.9\}10^6 \right]^{0.5} \leq 1100$$

$$g_6(x) = \frac{1}{0.1x_7^3} \left[\left(\frac{745x_5}{x_2x_3} \right)^2 + \{157.5\}10^6 \right]^{0.5} \leq 850$$

$$g_7(x) = x_2x_3 \leq 40$$

$$g_8(x) = x_1/x_2 \geq 5$$

$$g_9(x) = x_1/x_2 \leq 12$$

$$g_{10}(x) = (1.5x_6 + 1.9)x_4^{-1} \leq 1$$

$$g_{11}(x) = (1.1x_7 + 1.9)x_5^{-1} \leq 1$$

onde

$$2.6 \leq x_1 \leq 3.6 \quad 0.7 \leq x_2 \leq 0.8 \quad 17 \leq x_3 \leq 28$$

$$7.3 \leq x_4 \leq 8.3 \quad 7.8 \leq x_5 \leq 8.3 \quad 2.9 \leq x_6 \leq 3.9$$

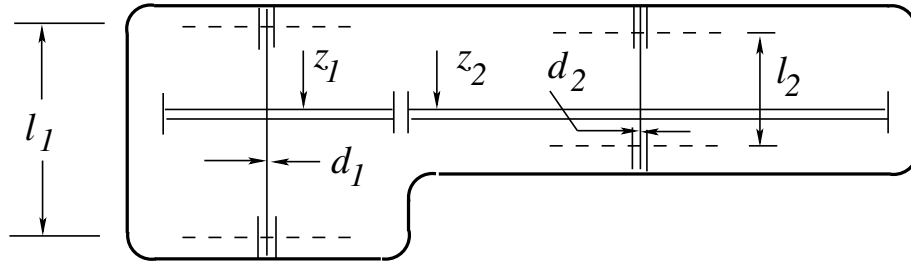


Figura 5.10: Redutor de velocidade.

Para o redutor de velocidade também foram utilizadas 36000 avaliações da função objetivo. A Tabela 5.14 apresenta os resultados encontrados para o problema, onde o APM obteve o melhor peso igual a 2996.3592.

Tabela 5.14: Resultados encontrados para o problema do redutor de velocidade.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	2996.3592	2996.3837	2998.8105	2.6529e+01	3007.4698	35/35
APM_Med.3	2996.3622	2996.3780	2999.6083	3.4911e+01	3016.7808	35/35
APM_Worst	2996.3654	2996.3927	3002.1440	4.2637e+01	3016.7988	35/35
APM_Spor_Mono	2996.3631	3005.7006	3003.6417	4.4687e+01	3016.7882	35/35

A Tabela 5.15 apresenta os valores encontrados para cada variável de projeto e o seu peso final. Os resultados usando os métodos APM e SR (*Stochastic Ranking*) não são fornecidos nas referências [150] e [17] utilizadas nesta tabela, mas eles foram obtidos usando as técnicas nestas referências.

Tabela 5.15: Variáveis de projeto e peso final para o problema do redutor de velocidade.

Método	b	m	n	l_1	l_2	d_1	d_2	Peso
ES [168]	3.5061	0.7008	17	7.4601	7.9621	3.3629	5.3089	3025.0051
AIS-GA [14]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3494
AIS-GA ^C [14]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3484
AIS-GA [16]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3483
APM [150]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3482
SR [17]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3481
T-Cell <i>et al.</i> [15]	3.5000	0.7000	17	7.3000	7.8000	3.3502	5.2866	2996.3481
Este estudo	3.5000	0.7000	17	7.3009	8.2999	3.3502	5.2868	2996.3592

5.5.3 Viga Soldada

O objetivo deste problema é minimizar o custo $C(h, l, t, b)$ de uma viga soldada [149], onde $h \in [0.125, 10]$ e $0.1 \leq l, t, b \leq 10$. A viga é ilustrada pela Figura 5.11. A função objetivo e as restrições são como segue:

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

$$g_1(\tau) = 13,600 - \tau \geq 0$$

$$g_2(\sigma) = 30,000 - \sigma \geq 0$$

$$g_3(b, h) = b - h \geq 0$$

$$g_4(P_c) = P_c - 6,000 \geq 0$$

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

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

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

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

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

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

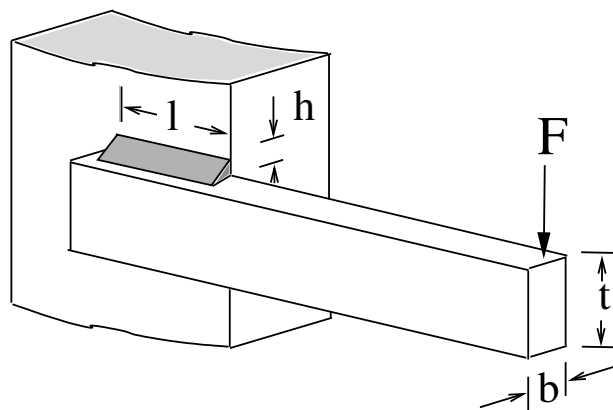


Figura 5.11: Viga Soldada.

O número de avaliações da função objetivo utilizada para esse problema foi de 320000. Os resultados obtidos podem ser visualizados na Tabela 5.16. O APM obteve o melhor custo final 2.38113, seguido do APM_Med_3, APM_Worst e APM_Spor_Mono, respectivamente.

Tabela 5.16: Resultados encontrados para o problema da viga soldada.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	2.38113	2.77504	2.81474	2.2005e+00	3.69870	35/35
APM_Med_3	2.38114	2.43315	2.67102	2.0656e+00	3.46638	35/35
APM_Worst	2.38115	2.66995	2.70019	1.8796e+00	3.32986	35/35
APM_Spor_Mono	2.38118	2.51280	3.14079	1.3686e+01	16.30031	35/35

Na Tabela 5.17, reproduzida de [7], são apresentadas as variáveis de projeto e o custo final para o problema usando diferentes métodos. O PSO se mostrou bastante competitivo se comparado aos resultados apresentados na tabela.

5.5.4 Vaso de Pressão

O problema tem por objetivo a minimização do peso W de um vaso de pressão cilíndrico com duas tampas esféricas [149], ilustrado na Figura 5.12. São quatro as variáveis de projeto (em *in*): a espessura do vaso de pressão (T_s), a espessura da tampa (T_h), o raio interno do vaso (R) e a altura do componente cilíndrico (L). Dentre essas, duas são variáveis discretas (T_s e T_h) e duas são contínuas (R e L). É um problema com restrições não-linear e com variáveis mistas (discretas e contínuas). O peso a ser minimizado e as restrições são dadas por:

$$W(T_s, T_h, R, L) = 0.6224T_sT_hR + 1.7781T_hR^2 + 3.1661T_s^2R$$

$$g_1(T_s, R) = T_s - 0.0193R \geq 0$$

$$g_2(T_h, R) = T_h - 0.00954R \geq 0$$

$$g_3(R, L) = \pi R^2L + 4/3\pi R^3 - 1,296,000 \geq 0$$

$$g_4(L) = -L + 240 \geq 0$$

Tabela 5.17: Melhores resultados encontrados para o problema da viga soldada usando diferentes métodos. Estes resultados foram extraídos da referência [7]. ^asistemas imunológicos artificiais; ^brecozimento simulado; ^calgoritmos evolucionários; ^dalgoritmo vaga-lume; ^eotimização de forrageamento bacteriana; ^fbusca harmônica; ^gbusca aleatória; ^hbusca em sistema carregado; ⁱmodelo sócio comportamental; ^jalgoritmo de sociedade e civilização; ^kprogramação geométrica; ^levolução diferencial; ^malgoritmo *T-cell*.

Autores	Método	h	l	t	b	Custo
Leite and Topping [169]	GA	0.2489	6.1097	8.2484	0.2485	2.4000
Deb [170]	GA	-	-	-	-	2.38
Deb [171]	GA	0.2489	6.1730	8.1789	0.2533	2.4331
Lemonge and Barbosa [13]	GA	0.2443	6.2117	8.3015	0.2443	2.3816
Barbosa and Lemonge [150]	GA	0.2442	6.2231	8.2915	0.2444	2.3814
Bernardino <i>et al.</i> [14]	GA-AIS ^a	0.2443	6.2202	8.2915	0.2444	2.3812
Bernardino <i>et al.</i> [16]	GA-AIS	0.2444	6.2183	8.2912	0.2444	2.3812
Atiqullah and Rao [172]	SA ^b	0.2471	6.1451	8.2721	0.2495	2.4148
Hedar and Fukushima [159]	SA	0.2444	6.2158	8.2939	0.2444	2.3811
Liu [173]	SA	0.2444	6.2175	8.2915	0.2444	2.3810
Hwang and He [174]	SA-GA	0.2231	1.5815	12.8468	0.2245	2.2500
He <i>et al.</i> [158]	PSO	0.2444	6.2175	8.2915	0.2444	2.3810
Zhang <i>et al.</i> [175]	EA ^c	0.2443	6.2201	8.2940	0.2444	2.3816
Runarsson and Yao [17]	EA	0.2758	5.0053	8.6261	0.2758	2.5961
Montes and Ocana [163]	BFO ^e	0.2057	3.4711	9.0367	0.2057	2.3868
Lee and Geem [24]	HS ^f	0.2442	6.2231	8.2915	0.2443	2.3807
Siddall [176]	RS ^g	0.2444	6.2819	8.2915	0.2444	2.3815
Akhtar <i>et al.</i> [177]	SBM ⁱ	0.2407	6.4851	8.2399	0.2497	2.4426
Ray and Liew [165]	SCA ^j	0.2444	6.2380	8.2886	0.2446	2.3854
Ragsdell and Phillips [178]	GP ^k	0.2536	7.1410	7.1044	0.2536	2.3398
Silva <i>et al.</i> [18]	PSO	-	-	-	-	2.3975
Zhang <i>et al.</i> [167]	DE ^l	0.2444	6.2175	8.2915	0.2444	2.3810
Aragon <i>et al.</i> [15]	TCA ^m	0.2444	6.2186	8.2915	0.2444	2.3811
Este estudo	PSO	0.2443	6.2186	8.2914	0.2443	2.3811

onde

$$0.00625 \leq T_s, T_h \leq 5 \text{ (em passos constantes de 0.0625)}$$

$$10 \leq R, L \leq 200$$

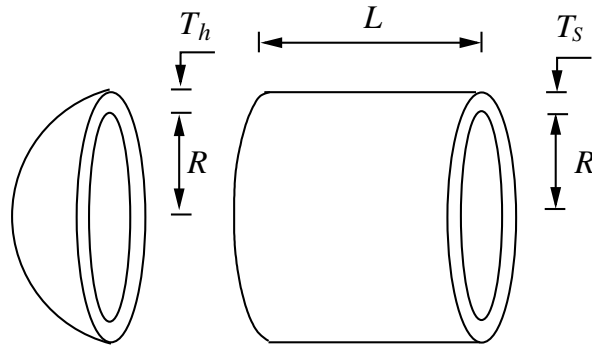


Figura 5.12: Vaso de pressão.

Adotou-se para este problema um valor de 80000 para o número de avaliações da função objetivo. A Tabela 5.5.4 apresenta os resultados encontrados para o vaso de pressão. Observa-se que os 4 métodos analisados obtiveram o mesmo resultado 6059.7143.

Tabela 5.18: Resultados encontrados para o problema do vaso de pressão.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	6059.7143	6090.5263	6474.8760	3.1086e+03	7544.4925	35/35
APM_Med_3	6059.7143	6370.7797	6427.6676	2.6221e+03	7544.4925	35/35
APM_Worst	6059.7143	6318.9481	6359.9781	2.2021e+03	7544.4925	35/35
APM_Spor_Mono	6059.7143	6090.5263	6352.0563	2.5773e+03	7544.4925	35/35

A Tabela 5.19, reproduzida de [7], mostra os resultados da literatura para o problema do vaso de pressão. Comparando-se os dados, é possível verificar que o resultado encontrado nessa dissertação para o problema se mostrou competitivo.

5.5.5 Viga Engastada e Livre

O problema corresponde em minimizar o volume V de uma viga engastada e livre [188], sujeita a uma carga de $P = 50000\text{N}$. A viga é ilustrada pela Figura 5.13. As variáveis de projeto são dez e correspondem à altura (H_i) e a largura (B_i) da seção transversal retangular de cada uma das cinco partes que compõem a viga. As variáveis B_1 e H_1 são inteiras, B_2, B_3 , assumem valores discretos para serem escolhidos a partir de um conjunto 2.4, 2.6, 2.8, 3.1, H_2 e H_3 são discretas e são escolhidos dentro de um conjunto 45.0,

Tabela 5.19: Resultados estatísticos para o problema do vaso de pressão. Os resultados foram extraídos da referência [7].

Autores	Melhor	Média	Pior	dp
Akhtar <i>et al.</i> [177]	6171.00	6335.05	6453.65	-
Aragon <i>et al.</i> [15]	6390.55	7694.06	6737.06	357
Barbosa and Lemonge [179]	6059.71	6384.05	6447.20	419
Bernardino <i>et al.</i> [14]	6060.13	6845.49	6385.94	-
Bernardino <i>et al.</i> [14]	6059.85	7388.16	6545.12	124
Cagnina <i>et al.</i> [180][33]	6059.71	-	-	-
Coello [181]	6288.75	6293.84	6308.15	7.41
Coello [152]	6177.25	-	-	130.93
Coello and Montes [155]	6059.95	6177.25	6469.32	130.93
Deb [182]	6410.38	-	-	-
Dos Santos Coelho [156]	6059.71	-	-	-
Hadj-Alouane and Bean [183]	6303.50	8065.66	10569.70	821.30
He and Wang [157]	6061.08	6147.13	6363.8	86.45
He <i>et al.</i> [158]	6059.71	6289.93	-	305.78
Homaifar <i>et al.</i> [184]	6295.11	8098.03	9528.07	831.69
Huang <i>et al.</i> [161]	6059.73	6085.23	6371.05	43.01
Joines and Houck [185]	6273.28	8092.87	10382.10	1017.99
Lemonge and Barbosa [13]	6060.188	-	-	-
Li and Chang [186]	7127.30	-	-	-
Michalewicz and Attia [187]	6572.62	8164.56	9580.51	789.65
Montes and Coello [162]	6059.75	6850	7332.88	426
Montes and Ocana [163]	6059.73	6081.78	6150.13	67.24
Runarsson and Yao [17]	6832.58	8012.61	7187.31	267
Silva <i>et al.</i> [18]	6090.76	6261.08	6771.76	-
Gandomi <i>et. al.</i> [7]	6059.71	6179.13	6318.95	137.223
Este estudo	6059.71	6359.97	7544.49	2202.10

50.0, 55.0, 60.0 e, finalmente B_4 , H_4 , B_5 e H_5 são contínuas. As variáveis são dadas em centímetros e o módulo de elasticidade *Young* do material é igual a 200 *GPa*. O volume da viga e as restrições do problema podem ser calculados como:

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

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

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

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

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

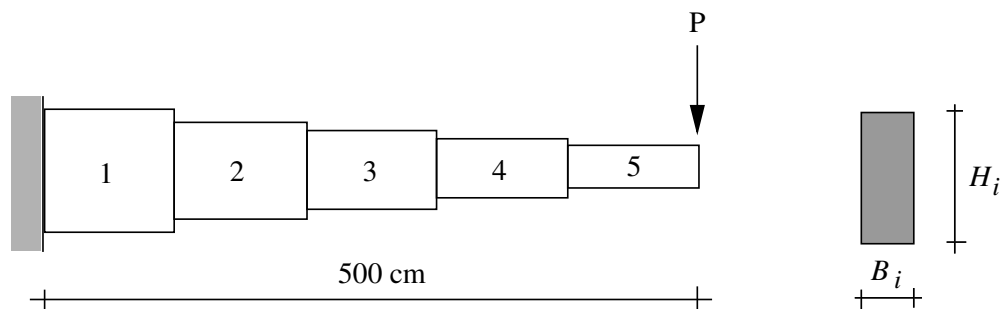


Figura 5.13: Viga engastada e livre.

O número de avaliações da função objetivo utilizada foi de 35000. A Tabela 5.20 apresenta uma comparação entre o APM_Worst, que obteve o melhor resultado 64578.229, o APM e as outras duas variantes utilizadas neste estudo.

Tabela 5.20: Resultados encontrados para o problema da viga engastada e livre.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	64965.071	67943.462	67901.329	1.6162e+04	75143.537	35/35
APM_Med_3	64578.271	68294.702	71817.816	1.0431e+05	173520.325	35/35
APM_Worst	64578.229	67943.452	68240.2182	1.6177e+04	73943.453	35/35
APM_Spor_Mono	64584.1326	68673.2969	70516.5469	4.2187e+04	106637.8329	35/35

A Tabela 5.21, extraída de [8], compara os resultados encontrados das variáveis de projeto e o volume final disponíveis na literatura com o resultado da melhor variante obtida neste estudo.

Tabela 5.21: Problema da viga engastada e livre. Os resultados foram extraídos da Tabela 8 da referência [8]. ^aestratégia de evolução baseada em ranqueamento de nicho, ^bcontínuo/aproximado, ^cdiscreto preciso, ^daproximação discreta linear, ^eaproximação discreta conservativa, ^fmétodo de penalização adaptativa, ^gclareira. Os autores são: Chen e Chen [9], Lamberti e Pappalettere [10], Thanedar e Vanderplaats [11], Erbatur *et. al* [12], Lemonge e Barbosa [13] e Bernardino *et. al* [14].

Referência	Método	na	b1	h1	b2	h2	b3	h3	b4	h4	b5	h5	Volume
[9]	RNES ^a 1	12000	3	60	3.1	55	2.6	50	2.311	43.108	1.822	34.307	64269.59
	RNES 2	12000	3	60	3.1	55	2.6	50	2.267	43.797	1.849	34.282	64322.43
	RNES 3	12000	3	60	3.1	55	2.6	50	2.348	42.804	1.783	34.753	64299.11
	RNES 4	12000	3	60	3.1	55	2.6	50	2.491	41.51	2.113	33.231	65416.90
[11]	C/RU ^b	-	4	62	3.1	60	2.6	55	2.205	44.09	1.751	35.03	73555.00
	PD ^c	-	3	60	3.1	55	2.6	50	2.276	45.528	1.75	34.995	64537.00
	LAD ^d	-	3	60	3.1	55	2.6	50	2.262	45.233	1.75	34.995	64403.00
	CAD ^e	-	3	60	3.1	55	2.6	50	2.279	45.553	1.75	35.004	64403.00
[12]	GA 1	10000	3	60	3.1	55	2.6	50	2.3	45.5	1.80	35	64558.00
	GA 2	10000	3	60	3.1	55	2.6	50	2.27	45.25	1.75	35	64447.00
[13]	GA-APM ^f	35000	3	60	3.1	55	2.6	50	2.289	45.626	1.793	34.593	64698.56
[14]	AIS-GA	35000	3	60	3.1	55	2.6	50	2.235	44.395	2.004	32.879	65559.60
	AIS-GA-C ^g	35000	3	60	3.1	60	2.6	50	2.311	43.186	2.225	31.250	66533.47
	FA	50000	3	60	3.1	55	2.6	50	2.205	44.091	1.750	34.995	63893.52
Este estudo	PSO	35000	4	60	3.1	55	2.6	50	2.204	44.091	1.749	34.995	64578.229

5.5.6 Discussão dos problemas de engenharia

Para o conjunto de experimentos envolvendo os problemas de engenharia, uma análise dos resultados obtidos é feita utilizando o melhor valor e a média da função objetivo. Inicialmente, utilizou-se na análise o melhor valor da função objetivo. A Figura 5.14(a) apresenta o gráfico dos perfis de desempenho no intervalo $\tau \in [1; 1.000005]$, onde verifica-se que o APM obteve o melhor desempenho em uma quantidade maior de problemas em relação às demais variantes. A Figura 5.14(b) que representa o gráfico dos perfis de desempenho onde os menores valores de τ , tal que $\rho(\tau) = 1$ são os métodos de melhor desempenho, apresenta as quatro variantes como as mais robustas. Os valores das áreas sob as curvas dos perfis de desempenho podem ser observados na Tabela 5.22, onde o APM_Med_3 obteve o maior valor de área, seguido do APM_Spor_Mono. Assim, ambas as variantes são consideradas as que possuem o melhor desempenho global, ao passo que elas chegaram em segundo lugar na análise obtida na Figura 5.14(a).

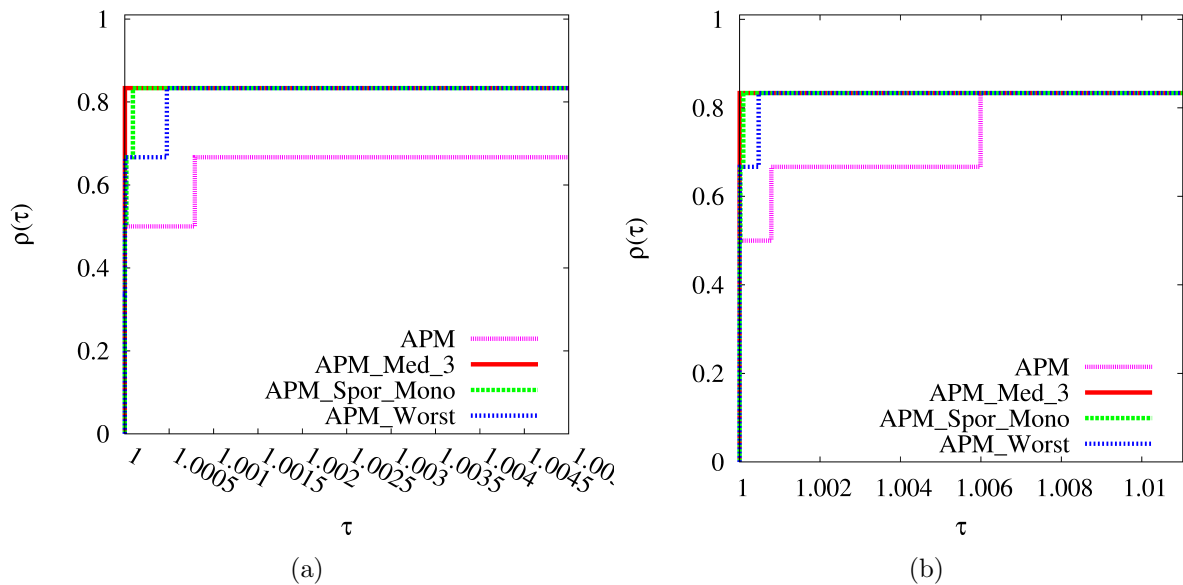


Figura 5.14: Perfis de desempenho utilizando o **melhor** valor.

Tabela 5.22: Área normalizada sob as curvas dos perfis de desempenho utilizando o **melhor** valor.

Método	APM_Med_3	APM_Spor_Mono	APM_Worst	APM
Área	1	0.99641	0.98404	0.77373

A análise envolvendo o valor da média da função objetivo, apresenta o método APM como aquele que obteve o melhor desempenho em um número maior de problemas e, portanto, possui o maior valor de $\rho(1)$, conforme a Figura 5.15(a) no intervalo $\tau \in [1; 1.005]$. Na Figura 5.15(b) novamente todos os métodos obtiveram o menor valor de τ , tal que $\rho(\tau) = 1$. Por fim, a Tabela 5.23 que apresenta os valores das áreas sob as curvas dos perfis de desempenho, apresenta o APM_Worst e o APM como os métodos como melhor desempenho global, respectivamente.

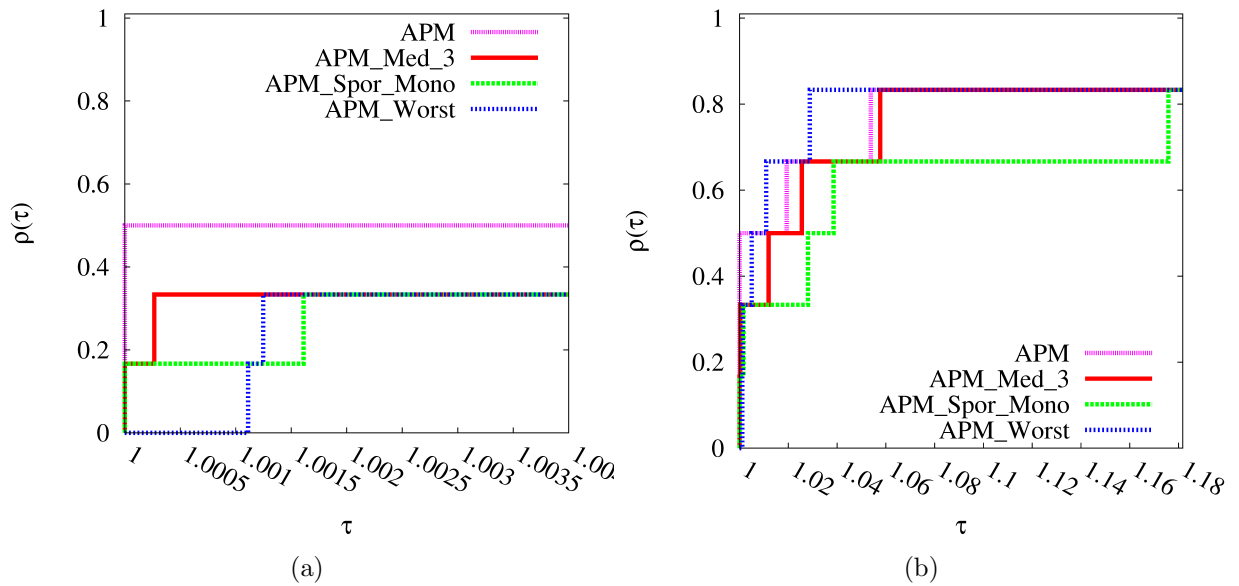


Figura 5.15: Perfis de desempenho utilizando a **média**.

Tabela 5.23: Área normalizada sob as curvas dos perfis de desempenho utilizando a **média**.

Método	APM_Worst	APM	APM_Med_3	APM_Spor_Mono
Área	1	0.96866	0.94193	0.76332

5.6 Problemas de otimização estrutural

5.6.1 Treliça de 10 barras

A Treliça de 10 barras é um problema clássico de otimização em treliças, ilustrada pela Figura 5.16 e amplamente estudada [135, 189, 190]. A função objetivo é bastante simples

e pode ser descrita da seguinte forma: encontrar o conjunto de áreas da seção transversal das barras (A_i , $i = 1, \dots, 10$) que minimize o peso da estrutura, conforme a Equação 5.3.

$$W(a) = \sum_{i=1}^n \rho A_i L_i \quad (5.3)$$

onde L_i é o comprimento do i -ésimo membro da treliça e ρ a densidade do material.

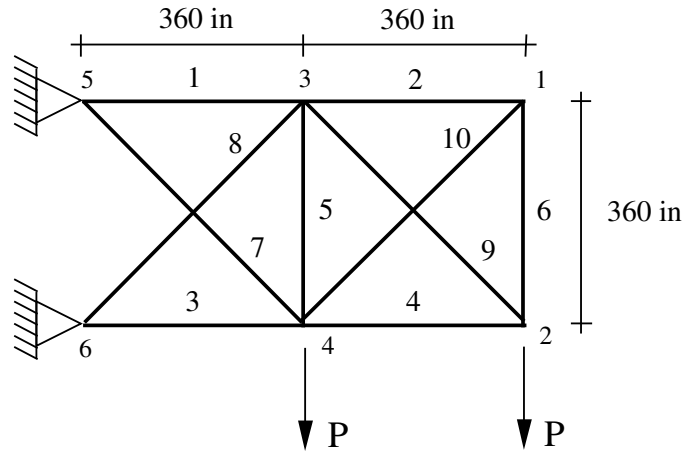


Figura 5.16: Treliça de 10 barras.

As restrições normalizadas do problema envolvem as tensões em cada membro apresentadas na Equação 5.4 e de deslocamentos de cada nó descritas na Equação 5.5.

$$\frac{\sigma_i}{\bar{\sigma}} - 1 \leq 0, \quad i = 1, 2, 3, \dots, m \quad (5.4)$$

onde m é o número de barras.

$$\frac{u_j}{\bar{u}} - 1 \leq 0, \quad j = 1, 2, 3, \dots, n \quad (5.5)$$

onde n é o número de grau de liberdade da estrutura.

A tensão normal máxima $\bar{\sigma}$ é limitada em $\pm 25 \text{ ksi}$ e os deslocamentos \bar{u} são limitados em 2 in nas direções x e y . A densidade do material é de 0.1 lb/in^3 , o módulo de elasticidade é $E = 10^4 \text{ ksi}$ e as cargas verticais descendentes $P = 100 \text{ kips}$ são aplicadas nos nós 2 e 4.

Para esse problema, dois casos de variáveis são analisados: um com variáveis contínuas e outro com variáveis discretas. Para o caso contínuo, as áreas variam de 0.1 in^2 a

33.50 in^2 . E para o caso com variáveis discretas, as áreas das seções transversais são escolhidas dentro de um conjunto que contem as 32 possibilidades que seguem (in^2): 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.93, 3.13, 3.38, 3.47, 3.55, 3.63, 3.88, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.97, 11.50, 13.50, 14.20, 15.50, 16.90, 18.80, 19.90, 22.00, 26.50, 30.00, 33.50.

Em ambos os casos foram feitas 35 execuções independentes e para o caso contínuo, adotou-se um valor de 280000 avaliações da função objetivo. A Tabela 5.24 apresenta os resultados obtidos para o caso contínuo considerando o APM, APM_Med.3, APM_Worst e APM_Spor_Mono. O melhor valor para o peso final foi obtido pelo APM_Worst 5060.9234 lbs .

Tabela 5.24: Resultados encontrados para a treliça de 10 barras (caso contínuo).

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	5060.9770	5076.8523	5072.2409	4.8201e+01	5087.7249	35/35
APM_Med.3	5060.9234	5073.8662	5069.7385	4.8100e+01	5087.1895	35/35
APM_Worst	5061.0453	5074.5300	5070.6926	4.5738e+01	5087.5734	35/35
APM_Spor_Mono	5060.9379	5076.8950	5071.0983	5.4067e+01	5088.3651	35/35

A Tabela 5.25 apresenta os resultados encontrados na literatura para as variáveis de projeto e o peso final da estrutura.

Para o caso onde as variáveis de projeto são discretas, foram consideradas 90000 avaliações da função objetivo. A Tabela 5.26 apresenta os resultados encontrados para o problema. Os quatro métodos analisados obtiveram o mesmo peso final 5509.7173.

A Tabela 5.27 apresenta os resultados encontrados na literatura para as variáveis de projeto e peso final da treliça de 10 barras caso discreto. As soluções apresentadas nas referências [19, 20, 21] são ineficazes, conforme observado em [13].

5.6.2 Treliça de 25 barras

O problema da treliça de 25 barras tem como objetivo minimizar o peso da estrutura e é ilustrada conforme a Figura 5.17. As variáveis de projeto são as áreas das seções transversais a_k . A estrutura é composta por barras de comprimento L_k , sendo k o índice das barras da estrutura e ρ é a massa específica do material.

Tabela 5.25: Comparação entre os resultados da literatura e o resultado encontrado neste estudo para o problema da treliça de 10 barras (caso contínuo), com peso final em lb. Os autores são: Aragon *et. al* [15], Barbosa e Lemonge [13], Bernardino *et. al* [14] Bernardino *et. al* [16], Runarsson e Yao [17] e Silva *et al.* [18].

Variáveis	Ref.[15]	Ref.[13]	Ref.[14]	Ref.[16]	Ref.[17]	Ref. [18]	Este estudo
1	31.23829	29.22568	29.78121	30.52684	30.01400	30.5431	30.6481
2	0.316625	0.10000	0.10031	0.10000	0.10000	0.1000	0.1000
3	23.61073	24.18212	22.55140	22.91574	26.14460	23.1906	23.3151
4	14.50669	14.94714	15.50462	15.48294	15.29260	15.1972	15.2528
5	0.316234	0.10000	0.10002	0.10000	0.10000	0.1000	0.1000
6	0.316464	0.39463	0.52377	0.54620	0.55610	0.5425	0.5453
7	8.135098	7.49579	7.52854	7.47594	7.43980	7.4581	7.4516
8	21.61828	21.92486	21.15708	21.01566	21.00560	21.0317	21.0146
9	21.22159	21.29088	22.21351	21.55362	21.93900	21.5483	21.3700
10	0.31634	0.10000	0.10018	0.10000	0.10000	0.1000	0.1000
Peso	5142.30	5069.086	5064.67	5061.16	5061.71	5060.876	5060.9234

Tabela 5.26: Resultados encontrados para a treliça de 10 barras (caso discreto).

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	5509.7173	5528.0869	5628.8689	1.4468e+03	6593.1205	35/35
APM_Med_3	5509.7173	5626.0477	5665.151541	1.5179e+03	6540.4413	35/35
APM_Worst	5509.7173	5532.1210	5624.9286	1.3148e+03	6540.4413	35/35
APM_Spor_Mono	5509.7173	5540.4065	5700.1346	1.9561e+03	6654.2616	35/35

As barras da treliça são agrupadas e cada grupo possui uma única área A , a fim de manter a simetria da estrutura. Assim, o objetivo é encontrar um conjunto de áreas $a_k = \{A_1, A_2, \dots, A_i\}$ que minimize seu peso.

A treliça está sujeita a restrições de tensão em cada barra, que deve ter seu valor no intervalo de $[-40, 40]$ e deslocamentos máximos nos nós 1 e 2 limitados a 0.35 in . As áreas das seções transversais devem ser escolhidas dentro de um conjunto com 30 opções (in^2): 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4.

A densidade do material das barras é de 0.1 lb/in^3 e o módulo de elasticidade (*Young*) é igual a 10^4 ksi . O agrupamento das barras é feito conforme Tabela 5.28 e o carregamento aplicado sobre a estrutura é mostrado na Tabela 5.29.

A Tabela 5.30 apresenta uma comparação dos resultados encontrados para o problema,

Tabela 5.27: Variáveis de projeto e peso total encontrados para o problema da treliça de 10 barras (caso discreto). Os autores são: Krishnamoorthy e Rajeev [19], Lemonge e Barbosa [13], Galante [20] e Ghasemi *et. al* [21].

Variáveis	Ref.[19]	Ref.[13]	Ref.[20]	Ref.[13]	Ref.[13]	Ref.[21]	Este estudo
1	33.50	26.50	33.50	30.00	33.50	33.50	30.00
2	1.62	1.62	1.62	1.62	1.62	1.62	1.62
3	22.00	26.50	22.00	22.00	22.90	22.00	33.50
4	15.50	16.90	14.20	16.90	14.20	14.20	16.90
5	1.62	1.80	1.62	1.62	1.62	1.62	1.62
6	1.62	1.62	1.62	1.62	1.62	1.62	1.62
7	14.20	13.50	7.97	11.50	7.97	7.97	7.97
8	19.90	22.00	22.90	22.00	22.90	22.90	18.80
9	19.90	19.90	22.00	22.00	22.00	22.00	22.00
10	2.62	1.99	1.62	1.80	1.62	1.62	1.62
Peso	5613.58	5619.662	5458.3	5572.60	5490.738	5493.36	5509.7173

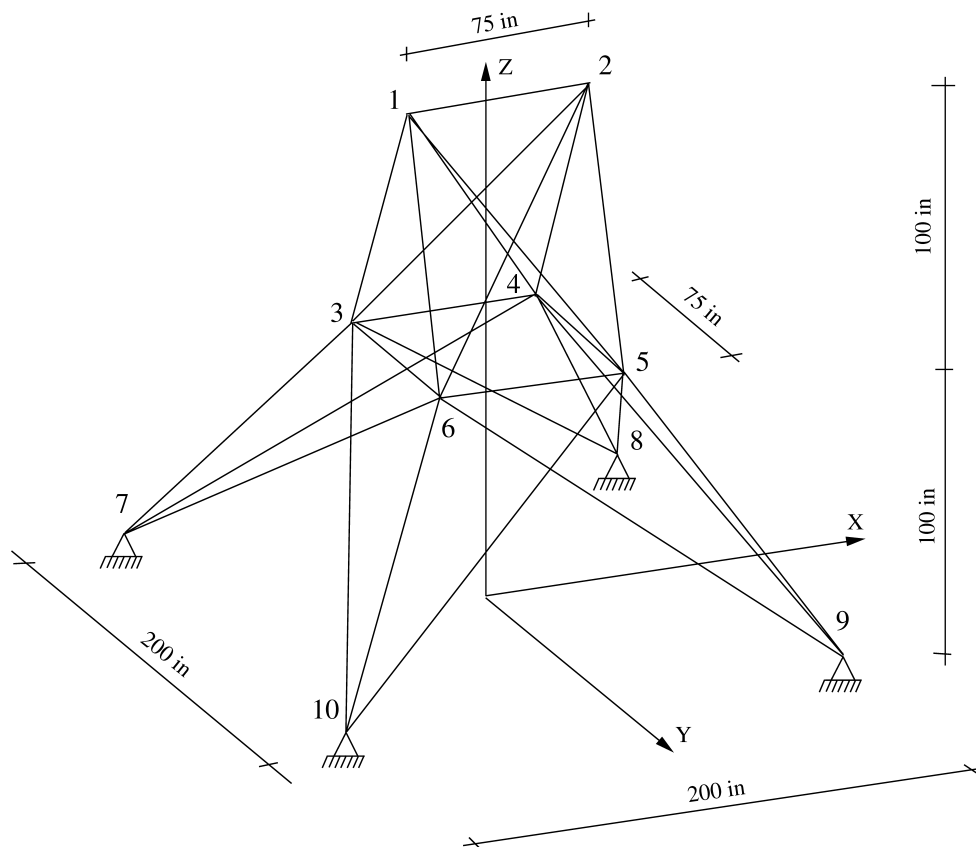


Figura 5.17: Treliça de 25 barras.

onde os quatro métodos analisados obtiveram o mesmo peso final 484.8541 *lbs*. A Tabela 5.31 apresenta uma comparação dos resultados encontrados na literatura das variáveis de

projeto e o peso final da estrutura. Este estudo juntamente com os resultados da Ref. [13] obtiveram o melhor valor dentre as demais referências.

Tabela 5.28: Agrupamento para a Treliça de 25 barras.

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

Tabela 5.29: Carregamento para a treliça de 25 barras (em *kips*).

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

Tabela 5.30: Resultados encontrados para o problema da treliça de 25 barras.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	484.8541	485.0487	485.97338	1.3601e+01	496.2520	35/35
APM_Med_3	484.8541	485.0487	487.22217	4.1581e+01	526.8441	35/35
APM_Worst	484.8541	485.0487	485.4513	6.2653e+00	490.1923	35/35
APM_Spor_Mono	484.8541	485.0487	485.5626	7.2751e+00	490.9124	35/35

5.6.3 Treliça de 52 barras

O objetivo dessa treliça é minimizar o peso das 52 barras conforme ilustrado na Figura 5.18. A densidade do material de composição das barras é de 7.860 kg/m^3 e o módulo de elasticidade é igual a $2.07 \times 10^5 \text{ MPa}$. A treliça está sujeita apenas a restrições de tensão,

Tabela 5.31: Comparação com os resultados da literatura para a treliça de 25 barras, com peso final em lb. Os autores são: Krishnamoorthy e Rajeev [19], Lemonge e Barbosa [13], Zhu [22], Erbatur *et. al* [12], Wu e Chow [23] e Lemonge e Barbosa [13].

Variáveis	Ref.[19]	Ref.[13]	Ref.[22]	Ref.[12]	Ref.[23]	Ref. [13]	Este estudo
A ₁	0.1	0.1	0.1	0.1	0.1	0.1	0.1
A ₂	1.8	0.7	1.9	1.2	0.5	0.3	0.6
A ₃	2.3	3.4	2.6	3.2	3.4	3.4	3.2
A ₄	0.2	0.1	0.1	0.1	0.1	0.1	0.1
A ₅	0.1	1.8	0.1	1.1	1.5	2.1	2.1
A ₆	0.8	1.0	0.8	0.9	0.9	1.0	1.0
A ₇	1.8	0.3	2.1	0.4	0.6	0.5	0.2
A ₈	3.0	3.4	2.6	3.4	3.4	3.4	3.2
Peso	546.01	486.743	562.93	493.80	486.29	484.854	484.854

definidas como 180 *MPa*. As barras estão agrupadas em 12 grupos conforme Tabela 5.32 e o carregamento para a treliça é observado na Tabela 5.33.

Os valores das áreas transversais a serem escolhidas são apresentadas em um conjunto com 64 opções (em mm^2): 71.613, 90.968, 126.451, 161.290, 198.064, 252.258, 285.161, 363.225, 388.386, 494.193, 506.451, 641.289, 645.160, 792.256, 816.773, 940.000, 1008.385, 1045.159, 1161.288, 1283.868, 1374.191, 1535.481, 1690.319, 1858.061, 1890.319, 1993.544, 2019.351, 2180.641, 2238.705, 2290.318, 2341.191, 2477.414, 2496.769, 2503.221, 2696.769, 2722.575, 2896.768, 2961.284, 3096.768, 3206.445, 3303.219, 3703.218, 4658.055, 5141.925, 5503.215, 5999.998, 6999.986, 7419.340, 8709.660, 8967.724, 9161.272, 9999.980, 10322.560, 10903.204, 12129.008, 12838.684, 14193.520, 14774.164, 15806.420, 17096.740, 18064.480, 19354.800 e 21612.860.

Os resultados encontrados para o problema são apresentados na Tabela 5.34. Todos os métodos obtiveram o mesmo peso final 1977.8997 *kg*. Uma comparação com resultados da literatura é apresentada nas Tabelas 5.35 e 5.36.

5.6.4 Treliça de 60 barras

O problema da treliça de 60 barras tem como objetivo a minimização do peso da estrutura. O módulo de elasticidade do material é de 10^4 *ksi* e a massa específica é igual a 0.1 *lb/in*³. Pode-se observar através da Figura 5.19 o raio externo do anel que vale 100 *in* e o raio interno 90 *in*.

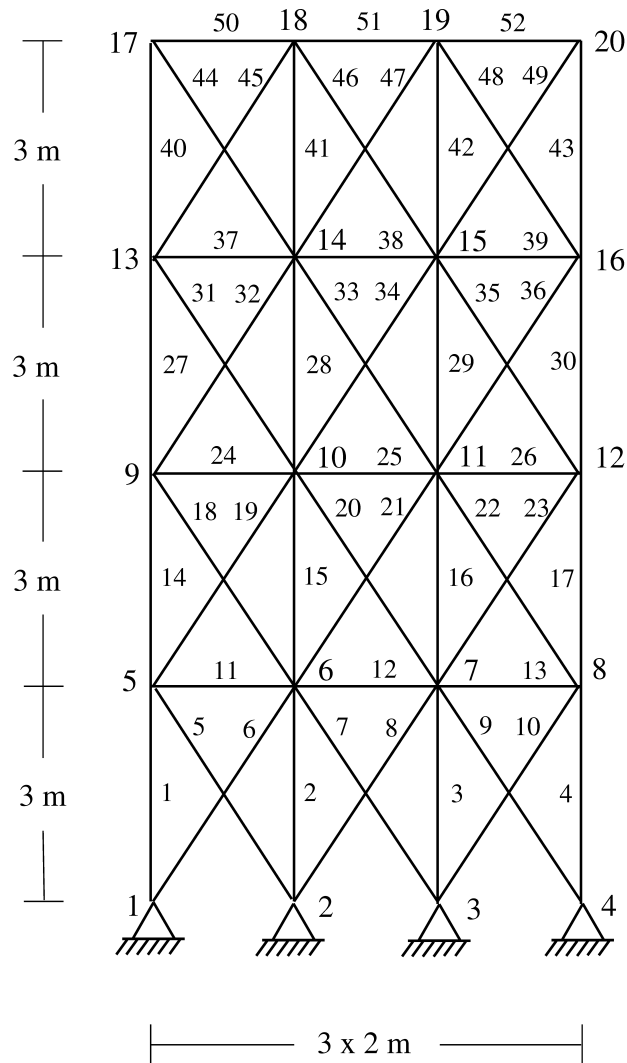


Figura 5.18: Treliça de 52 barras.

A treliça é submetida a três casos de carregamento, observados na Tabela 5.37. O número de restrições de tensão e deslocamento é de 198, onde a tensão máxima é de 10 ksi e os deslocamentos são: 1.75 no nó 4, 2.25 no nó 13 e 2.75 no nó 19. As barras são agrupadas conforme Tabela 5.38 e variam em um espaço de busca contínuo de 0.5 in^2 a 5 in^2 .

Na Tabela 5.39 são apresentados os resultados encontrados para o problema. O método que obteve o melhor desempenho foi o APM, com um peso final de 291.2477 lbs . A Tabela 5.40 apresenta uma comparação dos resultados encontrados na literatura para o problema, onde o valor encontrado para este estudo foi superior aos demais.

Tabela 5.32: Agrupamento para a treliça de 52 barras.

Grupo	Barras
A_1	1, 2, 3 e 4
A_2	5, 6, 7, 8, 9 e 10
A_3	11, 12 e 13
A_4	14, 15, 16 e 17
A_5	18, 19, 20, 21, 22 e 23
A_6	24, 25 e 26
A_7	27, 28, 29 e 30
A_8	31, 32, 33, 34, 35 e 36
A_9	37, 38 e 39
A_{10}	40, 41, 42 e 43
A_{11}	44, 45, 46, 47, 48 e 49
A_{12}	50, 51 e 52

Tabela 5.33: Carregamento para a treliça de 52 barras (em kN).

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

Tabela 5.34: Resultados encontrado para o problema da treliça de 52 barras.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	1977.8997	1984.3213	1994.3507	1.8893e+02	2112.8701	35/35
APM_Med_3	1977.8997	1979.3400	1983.2759	4.3450e+01	2009.2914	35/35
APM_Worst	1977.8997	1981.2606	1984.9845	5.0894e+01	2007.4309	35/35
APM_Spor_Mono	1977.8997	1984.3213	1993.4951	1.6879e+02	2089.6992	35/35

5.6.5 Treliça de 72 barras

A treliça de 72 barras é um problema de otimização que objetiva a minimização do peso da estrutura e é ilustrada pela Figura 5.20. Os valores das variáveis de projeto variam de 0.1 in^2 a 5 in^2 e são organizadas em 16 grupos conforme Tabela 5.41.

As restrições do problema envolvem a tensão máxima permitida de cada barra no intervalo de $[-25,25] \text{ ksi}$ e o deslocamento máximo permitido de 0.25 in nas direções x e

Tabela 5.35: Comparação dos resultados para o problema da treliça de 52 barras, com peso final em Kg. Os autores são: Lee e Geem [24], Li *et. al* [25] e Kaveh e Talatahari [26].

Variáveis	HS [24]	PSO [25]	PSOC [25]	HPSO [25]	DHPSACO [26]
A ₁	4658.055	4658.055	5999.988	4658.055	4658.055
A ₂	1161.288	1374.190	1008.380	1161.288	1161.288
A ₃	506.451	1858.060	2696.380	363.225	494.193
A ₄	3303.219	3206.440	3206.440	3303.219	3303.219
A ₅	940.000	1283.870	1161.290	940.000	1008.385
A ₆	494.193	252.260	729.030	494.193	285.161
A ₇	2290.318	3303.220	2238.710	2238.705	2290.318
A ₈	1008.385	1045.160	1008.380	1008.385	1008.385
A ₉	2290.318	126.450	494.190	388.386	388.386
A ₁₀	1535.481	2341.930	1283.870	1283.868	1283.868
A ₁₁	1045.159	1008.380	1161.290	1161.288	1161.288
A ₁₂	506.451	1045.160	494.190	792.256	506.451
Peso	1906.76	2230.16	2146.63	1905.49	1904.83

Tabela 5.36: Comparação dos resultados para o problema da treliça de 52 barras, com peso final em Kg. As colunas 1 a 4 desta tabela referem-se aos resultados obtidos para as variantes de um algoritmo de colônia de formigas proposto por Capriles *et. al* [27].

Variáveis	RBAS	RBAS+	RBASLU	RBASLU, 2	Este estudo
A ₁	4658.055	4658.055	4658.055	4658.055	2961.284
A ₂	1161.288	1161.288	1161.288	1161.288	792.256
A ₃	363.225	494.193	388.386	506.451	388.386
A ₄	3303.219	3303.219	3303.219	3303.219	2290.318
A ₅	940.000	940.000	940.000	940.000	641.289
A ₆	494.193	494.193	641.289	506.451	161.290
A ₇	2238.705	2238.705	2238.705	2238.705	1690.319
A ₈	1008.385	1008.385	1008.385	1008.385	494.193
A ₉	494.193	363.225	388.386	388.386	71.613
A ₁₀	1283.868	1283.868	1283.868	1283.868	71.613
A ₁₁	1161.288	1161.288	1161.288	1161.288	71.613
A ₁₂	641.289	645.160	645.160	506.451	90.968
Peso	1903.366	1903.549	1906.683	1899.350	1977.8997

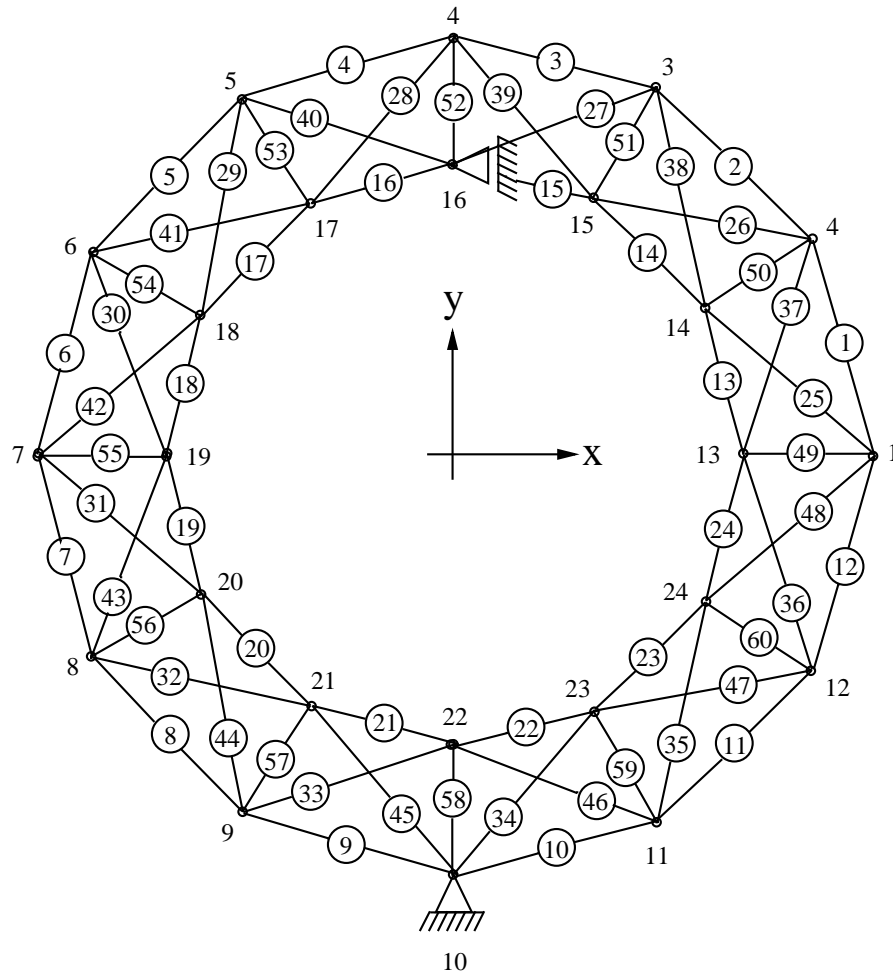


Figura 5.19: Treliça de 60 barras.

Tabela 5.37: Carregamento para a treliça de 60 barras (em *kips*).

Carregamento	Nó	F_x	F_y
1	1	-10.0	0
	7	9.0	0
2	15	-8.0	3.0
	18	-8.0	3.0
3	22	-20.0	10.0

y . A densidade do material é de 0.1 lb/in^3 e o módulo de elasticidade é igual a 10^4 ksi . Para esse problema são definidos dois casos de carregamentos que podem ser observados na Tabela 5.42.

Na Tabela 5.43 observa-se os resultados encontrados para o problema considerando os quatro métodos analisados. O APM_Med_3 obteve o melhor valor para o peso final dentre

Tabela 5.38: Agrupamento para a treliça de 60 barras.

Grupo	Barras	Grupo	Barras
A_1	49 ao 60	A_{14}	25 e 37
A_2	1 e 13	A_{15}	26 e 38
A_3	2 e 14	A_{16}	27 e 39
A_4	3 e 15	A_{17}	28 e 40
A_5	4 e 16	A_{18}	29 e 41
A_6	5 e 17	A_{19}	30 e 42
A_7	6 e 18	A_{20}	31 e 43
A_8	7 e 19	A_{21}	32 e 44
A_9	8 e 20	A_{22}	33 e 45
A_{10}	9 e 21	A_{23}	34 e 46
A_{11}	10 e 22	A_{24}	35 e 47
A_{12}	11 e 23	A_{25}	36 e 48
A_{13}	12 e 24		

Tabela 5.39: Resultados encontrados para o problema da treliça de 60 barras.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	291.2477	302.0956	319.2243	1.9759e+02	436.8457	35/35
APM_Med_3	292.2123	319.4826	319.3956	1.4329e+02	385.6968	35/35
APM_Worst	292.2592	323.54300	325.5208	1.8135e+02	437.9921	35/35
APM_Spor_Mono	291.8345	321.1936	322.1222	1.5402e+02	395.2332	35/35

os demais métodos. Contudo, os outros métodos obtiveram valores bem aproximados.

A Tabela 5.44, apresenta uma comparação para a treliça de 72 barras com resultados encontrados na literatura, onde as soluções marcadas com um asterisco não são rigorosamente factíveis. Este estudo obteve um valor superior para o peso final em relação às demais referências.

Tabela 5.40: Comparação dos resultados para o problema da treliça de 60 barras, com peso final em lb. Os autores são: Barbosa e Lemonge [28], Bernardino [29], Silva *et al.* [30].

Variáveis	Ref.[28]	Ref.[29]	Ref.[30]	Este estudo
A ₁	1.1202	1.1726	1.1628	1.16186
A ₂	2.0219	2.0621	2.0741	2.05396
A ₃	0.5087	0.5100	0.5000	0.55622
A ₄	1.7272	1.7654	1.7550	1.56836
A ₅	1.5205	1.6658	1.6937	1.42066
A ₆	0.5263	0.5720	0.5831	0.56823
A ₇	1.9032	1.9062	1.9866	1.92526
A ₈	2.1275	1.9484	1.8817	1.90424
A ₉	0.9882	1.0510	1.0210	1.11336
A ₁₀	2.0527	1.7466	1.8557	1.49336
A ₁₁	2.0527	1.6457	1.7946	1.49575
A ₁₂	0.7243	0.5153	0.5001	0.80284
A ₁₃	1.9604	2.0797	2.0715	2.06031
A ₁₄	1.2302	1.2616	1.2532	1.30629
A ₁₅	0.9970	1.1398	1.0677	1.00834
A ₁₆	0.6055	0.6888	0.7102	0.54040
A ₁₇	0.7287	0.7888	0.7795	0.50147
A ₁₈	0.0938	1.0401	1.0434	0.94529
A ₁₉	1.1158	1.1605	1.1508	1.14428
A ₂₀	1.1686	1.1499	1.1600	1.19860
A ₂₁	1.0674	1.0097	0.9856	1.15119
A ₂₂	1.0630	1.0680	1.0803	1.08666
A ₂₃	0.5879	0.8199	0.6282	0.50457
A ₂₄	1.0674	1.0693	1.0245	1.02406
A ₂₅	1.2679	1.2743	1.2642	1.26823
Peso	311.875	310.880	309.969	291.2477

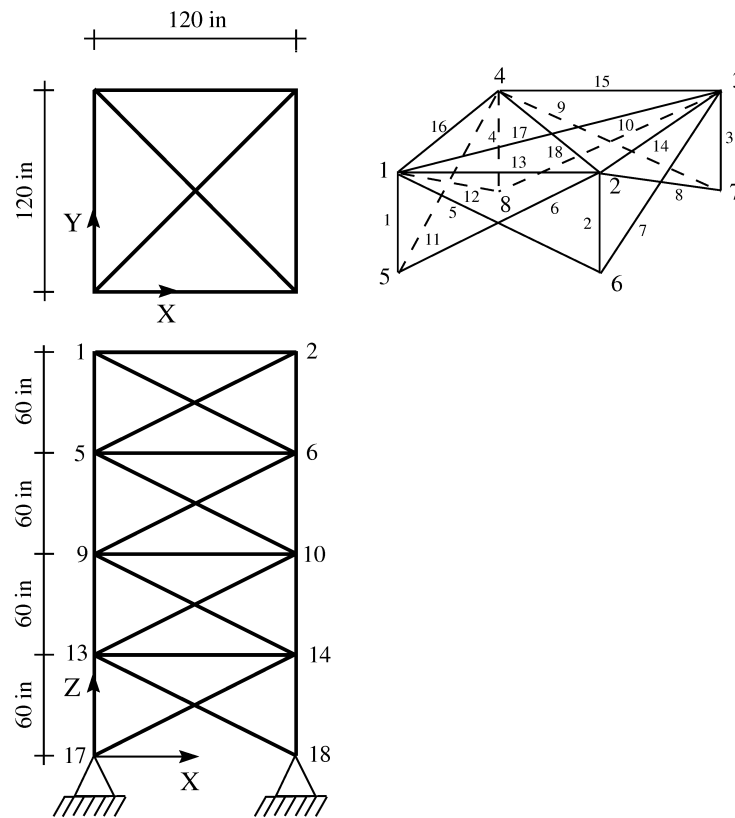


Figura 5.20: Treliça de 72 barras.

Tabela 5.41: Agrupamento para a Treliça de 72 barras.

Grupo	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

Tabela 5.42: Carregamento para a treliça de W é o peso final em lb 72 barras (em *kips*).

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

Tabela 5.43: Resultados encontrados para o problema da treliça de 72 barras.

Método	Melhor	Mediana	Média	dp	Pior	nesf
APM	379.65373	379.74019	383.87949	1.0493e+02	475.87770	35/35
APM_Med_3	379.65356	379.72416	394.40452	2.3324e+02	545.23741	35/35
APM_Worst	379.65388	379.72401	383.86117	1.0496e+02	475.91189	35/35
APM_Spor_Mono	379.65592	379.72484	381.10749	4.7899e+01	428.31708	35/35

5.6.6 *Discussão dos problemas de otimização estrutural*

Para o grupo de problemas de otimização estrutural a análise dos métodos também é feita utilizando o melhor valor e a média da função objetivo. A Figura 5.21(a), no intervalo $\tau \in [1; 1.000005]$, apresenta o gráfico dos perfis de desempenho onde a variante APM_Med_3 apresentou o maior valor de $\rho(1)$ e, portanto, obteve o melhor desempenho em uma quantidade maior de problemas. Na Figura 5.21(b), todas as variante obtiveram o menor valor de τ , tal que $\rho(\tau) = 1$. A área sob as curvas dos perfis de desempenho é apresentada na Tabela 5.45. O APM seguido da variante APM_Spor_Mono apresentaram os maiores valores de área, 1 e 0.90334, respectivamente. O APM ficou em segundo lugar na análise observada na Figura 5.21(a) e, portanto, é considerado o método com o melhor desempenho global.

Tabela 5.44: Comparação dos resultados para o problema da treliça de 72 barras, com peso final em lb. As soluções marcadas com um asterisco não são rigorosamente factíveis. Os autores são: Venkaya [31], Gellatly e Berke [32], Schimit e Farshi [33], Erbatur *et. al* [12] e Lemonge e Barbosa [13].

Variáveis	Ref.[31]	Ref.[32]	Ref.[33]	Ref.[12]	Ref.[12]*	Ref.[13]	Este estudo
A ₁	0.161	0.1492	0.1585	0.155	0.161	0.15500	0.15664
A ₂	0.557	0.7733	0.5936	0.535	0.544	0.54534	0.54419
A ₃	0.377	0.4534	0.3414	0.480	0.379	0.27496	0.40256
A ₄	0.506	0.3417	0.6076	0.520	0.521	0.51853	0.57634
A ₅	0.611	0.5521	0.2643	0.460	0.535	0.60365	0.51796
A ₆	0.532	0.6084	0.5480	0.530	0.535	0.66607	0.51204
A ₇	0.100	0.1000	0.1000	0.120	0.103	0.10159	0.10015
A ₈	0.100	0.1000	0.1509	0.165	0.111	0.13008	0.10062
A ₉	1.246	1.0235	1.1067	1.155	1.310	1.19954	1.26850
A ₁₀	0.524	0.5421	0.5793	0.585	0.498	0.47368	0.51759
A ₁₁	0.100	0.1000	0.1000	0.100	0.110	0.10059	0.10007
A ₁₂	0.100	0.1000	0.1000	0.100	0.103	0.10945	0.10001
A ₁₃	1.818	1.4636	2.0784	1.755	1.910	1.95307	1.91257
A ₁₄	0.524	0.5207	0.5034	0.505	0.525	0.51653	0.50978
A ₁₅	0.100	0.1000	0.1000	0.105	0.122	0.10000	0.10004
A ₁₆	0.100	0.1000	0.1000	0.155	0.103	0.10105	0.10045
Peso	381.2	395.97	388.63	385.76	383.12	387.036	379.65356

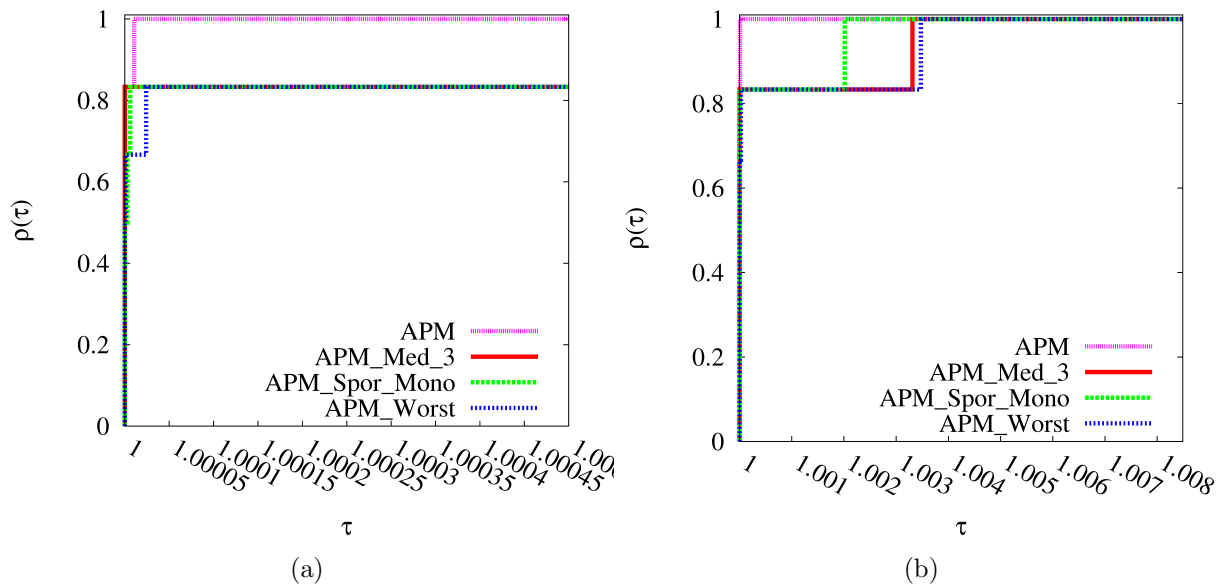


Figura 5.21: Perfis de desempenho utilizando o **melhor** valor.

Tabela 5.45: Área normalizada sob as curvas dos perfis de desempenho utilizando o **melhor** valor.

Método	APM	APM_Spor_Mono	APM_Med_3	APM_Worst
Área	1	0.90334	0.84151	0.83257

Uma última análise é feita utilizando o valor da média da função objetivo. A Figura 5.22(a), no intervalo $\tau \in [1; 1.0005]$, apresenta os métodos APM_Med_3 e APM_Worst como sendo os que obtiveram o melhor desempenho em uma quantidade maior de problemas. Na Figura 5.22(b), todos os métodos obtiveram os menores valores de τ , tal que $\rho(\tau) = 1$. O método que apresentou o maior valor de área, conforme Tabela 5.46, foi o APM seguido do APM_Worst.

Tabela 5.46: Área normalizada sob as curvas dos perfis de desempenho utilizando a **média**.

Método	APM	APM_Worst	APM_Spor_Mono	APM_Med_3
Área	1	0.93372	0.93321	0.83987

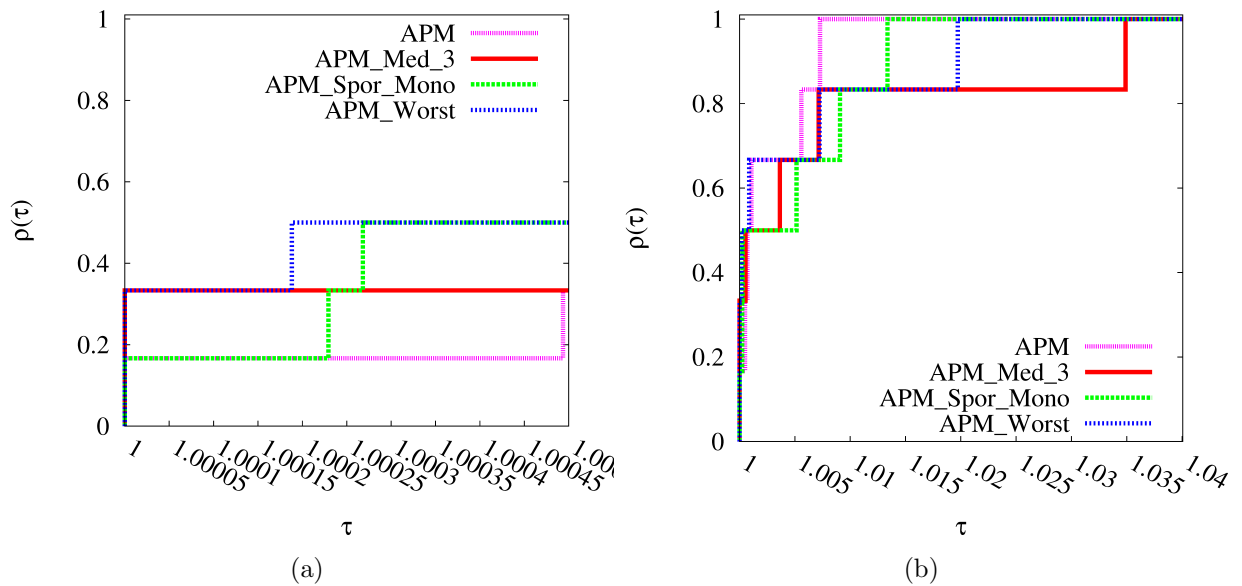


Figura 5.22: Perfis de desempenho utilizando a **média**.

5.7 Análise dos Resultados

A análise dos resultados para os três níveis de avaliações do *suite* de funções utilizando o melhor valor da função objetivo mostra que a variante que obteve o melhor desempenho foi a APM_Med_3, classificada em primeiro lugar em 2 das 3 análises, seguida do APM com uma ocorrência. Em segundo lugar, aparece a variante APM_Spor_Mono que obteve o melhor desempenho em 2 das 3 análises, seguida do APM_Worst com uma ocorrência. Utilizando a média da função objetivo a variante que obteve o melhor desempenho foi novamente a APM_Med_3, com 2 ocorrências em 3 análises, seguida do APM_Worst com uma ocorrência. Em segundo lugar, aparece o APM em 2 das 3 análises, seguido do APM_Spor_Mono. Assim, para o *suite* de funções a variante que obteve o melhor desempenho nas duas métricas foi a APM_Med_3.

Para o conjunto de problemas de engenharia a variante APM_Med_3 obteve o melhor desempenho na análise utilizando o melhor valor da função objetivo, seguida do APM_Spor_Mono. Utilizando o valor da média da função objetivo, o APM_Worst obteve o melhor desempenho, seguido do APM.

No conjunto de problemas de otimização estrutural e, inicialmente, utilizando o melhor valor da função objetivo o APM obteve o melhor desempenho, seguido do APM_Spor_Mono. A análise utilizando o valor da média da função objetivo mostrou o APM novamente com o melhor desempenho dentre as variantes analisadas, seguido do

APM_Worst. Logo, o APM foi o método com o melhor desempenho nesse conjunto de problemas.

Um resumo das análises envolvendo todos os experimentos com o objetivo de determinar a variante com o melhor desempenho global em cada métrica é apresentado a seguir. A Tabela 5.47 apresenta o número de ocorrências em que cada variante alcançou o primeiro, segundo, terceiro e quarto lugar utilizando o melhor valor da função objetivo. Observa-se que a variante APM_Med_3 obteve o melhor desempenho global alcançando o primeiro lugar em 3 das 5 análises. Em segundo lugar, destaca-se o APM_Spor_Mono e o APM.

Tabela 5.47: Análise das variantes envolvendo todos os experimentos utilizando o **melhor** valor da função objetivo.

	APM	APM_Med_3	APM_Worst	APM_Spor_Mono
1º lugar	2	3	0	0
2º lugar	0	0	1	4
3º lugar	2	1	1	1
4º lugar	1	1	3	0

O número de ocorrências que cada variante alcançou o primeiro, segundo, terceiro e quarto lugar utilizando agora o valor da média da função objetivo é apresentado na Tabela 5.48. Observa-se que as variantes APM_Worst e APM apresentaram o melhor desempenho e a variante APM_Spor_Mono apresentou um desempenho inferior em relação às demais variantes.

Tabela 5.48: Análise das variantes envolvendo todos os experimentos utilizando o valor da **média** da função objetivo.

	APM	APM_Med_3	APM_Worst	APM_Spor_Mono
1º lugar	1	2	2	0
2º lugar	3	0	1	1
3º lugar	0	1	2	2
4º lugar	1	2	0	2

Com base nas análises observadas ao longo do capítulo 5, conclui-se que a variante que apresentou um desempenho satisfatório em todos os experimentos alcançando um número maior de vezes o primeiro lugar é a APM_Med_3 e, portanto, pode ser considerada a variante com o melhor desempenho global.

6 Conclusões

Na presente dissertação foi abordado um algoritmo por enxame de partículas, conhecido como PSO, para o tratamento de problemas de otimização com restrições. O PSO vem sendo largamente estudado na literatura e é um tipo de inteligência de enxame inspirado no comportamento de bandos de pássaros (chamados de partículas) que fazem uso da experiência de cada um deles assim como da experiência do próprio bando para encontrar a melhor região do espaço de busca.

O PSO possui rápida convergência e utiliza poucos parâmetros de controle. Em contrapartida, uma deficiência do algoritmo é que o mesmo pode alcançar apenas mínimos locais, que foi tratado aqui através da introdução de um operador de loucura que tem como objetivo manter a diversidade das partículas.

As técnicas de penalização são frequentemente utilizadas para o tratamento de problemas de otimização com restrições por serem de fácil implementação e apresentarem bons resultados. Neste trabalho utilizou-se um método de penalização adaptativa (APM) e suas variantes APM_Med_3, APM_Worst e APM_Spor_Mono acoplados ao algoritmo PSO para o tratamento de problemas de otimização com restrições. O APM é livre de parâmetros a serem definidos pelo usuário, trata restrições de igualdade e desigualdade e não demanda o conhecimento explícito da função objetivo ou restrições.

Inicialmente as variantes do APM foram selecionadas e analisadas a fim de obter aquelas que apresentaram o melhor desempenho. Em seguida, as variantes com o melhor desempenho juntamente com o APM foram utilizadas na apresentação dos resultados e estes foram comparados com outros algoritmos encontrados na literatura.

Destaca-se que o objetivo principal desta dissertação foi explorar a capacidade do APM e suas variantes acoplados ao PSO como ferramenta de otimização na busca de soluções para problemas de otimização com restrições, envolvendo funções matemáticas, problemas de engenharia mecânica e estrutural tradicionalmente encontrados na literatura.

Os resultados apresentados nos experimentos comprovaram a competitividade do APM e suas variantes acoplados ao algoritmo PSO. No conjunto de problemas da matemática conhecido como *suite* de funções, a variante APM_Med_3 obteve o melhor desempenho dentre as outras variantes alcançando resultados satisfatórios. No segundo grupo de

experimentos envolvendo problemas de engenharia, destacam-se as variantes APM_Med.3 e APM_Worst. Os resultados apresentados nesse grupo foram competitivos quando comparados com resultados encontrados na literatura. Nos experimentos em problemas de otimização estrutural, o APM obteve o melhor desempenho. Quando comparado com outros métodos encontrados na literatura, o APM obteve resultados superiores em vários problemas.

As sugestões para trabalhos futuros são:

- Estudo mais detalhado dos parâmetros do PSO, tais como a inércia, fator de aceleração e atualização da velocidade;
- Desenvolvimento de um PSO com uma topologia de vizinhança local a fim de compará-lo com o PSO com topologia global apresentado nesta dissertação;
- Desenvolvimento de um PSO multiobjetivo acoplando, testando e adaptando o APM para este tipo de problema;
- Testar a nova *suíte* de funções disponibilizada na literatura [191], verificando o desempenho das variantes destacadas nas análises conduzidas neste texto.
- Aplicação do APM em problemas de otimização estrutural considerando restrições de frequência de vibração e restrições de cardinalidade para agrupamento ótimo das barras;
- Acoplamento do PSO a plataformas comerciais para análises com maior complexidade de problemas de otimização em engenharia.

REFERÊNCIAS

- [1] GANDOMI, A. H., YANG, X.-S., TALATAHARI, S., ALAVI, A. H., *Metaheuristic applications in structures and infrastructures*. Access Online via Elsevier, 2013.
- [2] RAO, S. S., *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.
- [3] KURIHARA, T., JIN'NO, K., “Analysis of convergence property of PSO and its application to nonlinear blind source separation”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 976–981, 2013.
- [4] BARBOSA, H. J., LEMONGE, A. C., “An adaptive penalty method for genetic algorithms in constrained optimization problems”, *Frontiers in Evolutionary Robotics*, v. 34, 2008.
- [5] GARCIA, R. P., CARVALHO, E. C. R., MONTÁ, B. G., LEMONGE, A. C. C., BERNARDINO, H. S., BARBOSA, H. J. C., “Novas Variantes para o Método de Penalização Adaptativo (APM) para problemas de otimização com restrições”, *Simpósio Brasileiro de Pesquisa Operacional - SBPO*, 2013.
- [6] 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”, *J. of Applied Mechanics*, v. 41, 2006.
- [7] GANDOMI, A., YANG, X.-S., ALAVI, A., TALATAHARI, S., “Bat algorithm for constrained optimization tasks”, *Neural Computing and Applications*, pp. 1–17, 2012.
- [8] GANDOMI, A., YANG, X.-S., ALAVI, A., “Mixed variable structural optimization using Firefly Algorithm”, *Computers & Structures*, v. 89, n. 23, pp. 2325 – 2336, 2011.
- [9] T.Y. CHEN, H. C., “Mixed-discrete structural optimization using a rank-niche evolution strategy”, *Engineering Optimization*, v. 41, n. 1, pp. 39–58, 2009.

- [10] LAMBERTI, L., PAPPALETTERE, C., “Move limits definition in structural optimization with sequential linear programming. Part II: Numerical examples”, *Computers & Structures*, v. 81, pp. 215–238, 2003.
- [11] THANEDAR, P., VANDERPLAATS, G., “Survey of discrete variable optimization for structural design”, *J Struct Eng, ASCE*, v. 2, n. 121, pp. 215–238, 1995.
- [12] ERBATUR, F., HASANÇEBİ, O., TÜTÜNCÜ, I., KILÇ, H., “Optimal design of planar and space structures with genetic algorithms”, *Computers & Structures*, v. 75, pp. 209–224, 2000.
- [13] LEMONGE, A., BARBOSA, H., “An adaptive penalty scheme for genetic algorithms in structural optimization”, *Int. J. Numer. Meth. Engng.*, v. 59, pp. 703–736, 2004.
- [14] BERNARDINO, H., BARBOSA, H., LEMONGE, A., “A hybrid genetic algorithm for constrained optimization problems in mechanical engineering”. In: *2007 IEEE congress on evolutionary computation (CEC 2007)*, pp. 646–653, IEEE Press, New York: Singapore, 2007.
- [15] ARAGÓN, 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.
- [16] 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.
- [17] RUNARSSON, T., YAO, X., “Stochastic ranking for constrained evolutionary optimization”, *IEEE Transactions on Evolutionary Computation*, v. 4, n. 3, pp. 284–294, 2000.
- [18] SILVA, A. F., *Análise de uma Técnica de Penalização Adaptativa Aplicada ao Algoritmo de Otimização por Enxame de Partículas*, Dissertação (Mestrado), Universidade Federal do Rio de Janeiro, 2010.

- [19] KRISHNAMOORTY, C., RAJEEV, S., “Discrete Optimization of Structures Using Genetic Algorithms.” *Journal of Structural Engineering*, v. 118, n. 5, 1992.
- [20] GALANTE, M., “Genetic Algorithms as an Approach to Optimize Real-World Trusses.” *Int. Journal for Numerical Methods in Engineering*, v. 39, pp. 361–382, 1996.
- [21] GHASEMI, M., HINTON, E., WOOD, R., “Optimization of trusses using genetic algorithms for discrete and continuous variables”, *Engineering Computations*, v. 16, pp. 272–301, 1997.
- [22] ZHU, D., “An improved Templeman’s algorithm for optimum design of trusses with discrete member sizes”, *Engineering Optimization*, v. 9, pp. 303–312, 1986.
- [23] WU, S., CHOW, P., “Steady-State Genetic Algorithms for Discrete Optimization of Trusses”, *Computers & Structures*, v. 56, n. 6, pp. 979–991, 1995.
- [24] LEE, K., GEEM, Z., “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, pp. 3902 – 3933, 2005.
- [25] LI, L., HUANG, Z., LIU, F., “A heuristic particle swarm optimization method for truss structures with discrete variables”, *Computers & Structures*, v. 87, n. 7–8, pp. 435–443, 2009.
- [26] KAVEH, A., TALATAHARI, S., “A particle swarm ant colony optimization for truss structures with discrete variables”, *Journal of Constructional Steel Research*, v. 65, pp. 1558–1568, 2009.
- [27] CAPRILES, P., FONSECA, L., BARBOSA, H., LEMONGE, A., “Rank-based ant colony algorithms for truss weight minimization with discrete variables”, *Communications in Numerical Methods in Engineering*, v. 23, pp. 553–575, 2007.
- [28] BARBOSA, H., LEMONGE, A., “A New Adaptive Penalty Scheme for Genetic Algorithms”, *Information Sciences*, v. 156, n. 3-4, pp. 215–251, 2003.

- [29] BERNARDINO, H. S., *Hibridização de Algoritmos Genéticos e Sistemas Imunológicos Artificiais para Problemas de Otimização com Restrições em Engenharia*, Dissertação (Mestrado), Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, 2008.
- [30] SILVA, E., BARBOSA, H., LEMONGE, A., “An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization”, *Optimization and Engineering*, v. 12, pp. 31–54, 2011.
- [31] VENKAYYA, V. B., “Design of Optimum Structures”, *Journal of Computers & Structures*, v. 1, n. 1-2, pp. 265–309, 1971.
- [32] GELLATLY, R., BERKE, L., *Optimal Structural Design*, Tech. Rep. AFFDL-TR-70-165, Air Force Flight Dynamics Lab., AFFDL, 1971.
- [33] SCHIMIT, L., FARSHI, B., “Some Approximation Concepts in Structural Synthesis”, *AIAA Journal*, v. 12, pp. 692–699, 1974.
- [34] 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, 1995.
- [35] MEZURA-MONTES, E., COELLO COELLO, C. A., “Constraint-handling in nature-inspired numerical optimization: past, present and future”, *Swarm and Evolutionary Computation*, v. 1, n. 4, pp. 173–194, 2011.
- [36] COELLO COELLO, C. A., “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art”, *Computer methods in applied mechanics and engineering*, v. 191, n. 11, pp. 1245–1287, 2002.
- [37] BARBOSA, H. J., LEMONGE, A. C., “An Adaptive Penalty Scheme In Genetic Algorithms For Constrained Optimiazation Problems.” In: *GECCO*, v. 2, pp. 287–294, 2002.
- [38] GALLET, C., SALAUN, M., BOUCHET, E., “An example of global structural optimisation with genetic algorithms in the aerospace field”. In: *VIII International Conference on Computational Plasticity*, 2005.

- [39] ROCHA, A., FERNANDES, E., “Self-Adaptive Penalties in the Electromagnetism-like Algorithm for Constrained Global Optimization Problems”. In: *Proceedings of The 8 th World Congress on Structural and Multidisciplinary Optimization*, pp. 1–10, 2009.
- [40] VENTER, G., HAFTKA, R. T., “Constrained particle swarm optimization using a bi-objective formulation”, *Structural and Multidisciplinary Optimization*, v. 40, pp. 65–76, 2010.
- [41] YOUSEFI, M., ENAYATIFAR, R., DARUS, A. N., ABDULLAH, A. H., “A robust learning based evolutionary approach for thermal-economic optimization of compact heat exchangers”, *International Communications in Heat and Mass Transfer*, v. 39, n. 10, pp. 1605 – 1615, 2012.
- [42] KAR, R., MANDAL, D., MONDAL, S., GHOSHAL, S. P., “Craziness based Particle Swarm Optimization algorithm for FIR band stop filter design”, *Swarm and Evolutionary Computation*, 2012.
- [43] DOLAN, E. D., MORE´, J. J., “Benchmarking optimization software with performance profiles”, *Mathematical Programming*, v. 91, pp. 201–213, 2002.
- [44] MEZURA-MONTES, E., COELLO, C. A. C., “Constrained optimization via multiobjective evolutionary algorithms”, In: *Multiobjective problem solving from nature*, pp. 53–75, Springer, 2008.
- [45] MARTINEZ, J. M., SANTOS, S. A., “Métodos computacionais de otimização”, *Colóquio Brasileiro de Matemática, Apostilas*, v. 20, 1995.
- [46] FRIEDLANDER, A., *Elementos de programação não-linear*. Editora da UNICAMP, 1994.
- [47] SOUSA, T. C. A., *Métodos subgradientes em otimização convexa e diferenciável*, Dissertação (Mestrado), Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, 2008.
- [48] ALIMORADI, A., FOLEY, C. M., PEZESHK, S., “Benchmark problems in structural design and performance optimization: past, present and future—part

- " . In: *19th ASCE Conf. Proc., State of the Art and Future Challenges in Structure*. ASCE Publications, 2010.
- [49] LEMONGE, A. C., BARBOSA, H. J., COUTINHO, A. L., BORGES, C. C., "Multiple cardinality constraints and automatic member grouping in the optimal design of steel framed structures", *Engineering Structures*, v. 33, n. 2, pp. 433–444, 2011.
- [50] SHIMODA, M., IWASA, K., TSUKADA, S., "Optimal Shape Design of Shell Structures", *Proceedings of IV European Conference on Computational Mechanics, ECCM 2010*, 2010.
- [51] RONG, J. H., LIANG, Q. Q., "A level set method for topology optimization of continuum structures with bounded design domains", *Computer Methods in Applied Mechanics and Engineering*, v. 197, n. 17, pp. 1447–1465, 2008.
- [52] ARORA, J., "Introduction to optimum design. 1989", .
- [53] HUGHES, T. J., *The finite element method: linear static and dynamic finite element analysis*. DoverPublications.com, 2012.
- [54] KARGUPTA, H., GOLDBERG, D. E., "Black box Optimization: Implications of SEARCH", *University of Illinois, Urbana-Champaign*, 1995.
- [55] BORGWARDT, K. H., *The Simplex Method. A Probabilistic Analysis, Algorithms and Combinatorics 1*. Springer-Verlag, Berlin, 1987.
- [56] HOLLAND, J. H., "Genetic algorithms and the optimal allocation of trials", *SIAM Journal on Computing*, v. 2, n. 2, pp. 88–105, 1973.
- [57] HOLLAND, J. H., "Adaption in natural and artificial systems", 1975.
- [58] JAMIESON, P., GHARIBIAN, F., SHANNON, L., "Supergenes in a genetic algorithm for heterogeneous FPGA placement". In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 253–260, 2013.
- [59] ELSAYED, S. M., SARKER, R. A., ESSAM, D. L., "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization". In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 356–360, 2013.

- [60] COSTA, E. O., FABRIS, F., RODRIGUES LOUREIROS, A., AHONEN, H., VAREJAO, F. M., FERRO, R., “Using GA for the stratified sampling of electricity consumers”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 261–268, 2013.
- [61] KIRKPATRICK, S., JR., D. G., VECCHI, M. P., “Optimization by simulated annealing”, *science*, v. 220, n. 4598, pp. 671–680, 1983.
- [62] BUREERAT, S., LIMTRAGOOL, J., “Structural topology optimization using simulated annealing with multiresolution design variables”, *Finite Elements in Analysis and Design*, v. 44, n. 12, pp. 738–747, 2008.
- [63] LAMBERTI, L., “An efficient simulated annealing algorithm for design optimization of truss structures”, *Computers & Structures*, v. 86, n. 19, pp. 1936–1953, 2008.
- [64] PARK, H. S., WON SUNG, C., “Optimization of steel structures using distributed simulated annealing algorithm on a cluster of personal computers”, *Computers & structures*, v. 80, n. 14, pp. 1305–1316, 2002.
- [65] DE CASTRO, L. N., TIMMIS, J., “An artificial immune network for multimodal function optimization”. In: *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*, v. 1, pp. 699–704, 2002.
- [66] COLORNI, A., DORIGO, M., MANIEZZO, V., OTHERS, “Distributed optimization by ant colonies”. In: *Proceedings of the first European conference on artificial life*, v. 142, pp. 134–142, 1991.
- [67] DORIGO, M., *Optimization, learning and natural algorithms*, Tese (Doutorado), Politecnico di Milano, Italy, 1992.
- [68] LUH, G.-C., LIN, C.-Y., “Structural topology optimization using ant colony optimization algorithm”, *Applied Soft Computing*, v. 9, n. 4, pp. 1343–1353, 2009.
- [69] ZHANG, Z. Q., LI, H. N., “Two-Level Optimization Method of Transmission Tower Structure Based on Ant Colony Algorithm”, *Advanced Materials Research*, v. 243, pp. 5849–5853, 2011.

- [70] MAJUMDAR, A., MAITI, D. K., MAITY, D., “Damage assessment of truss structures from changes in natural frequencies using ant colony optimization”, *Applied Mathematics and Computation*, v. 218, n. 19, pp. 9759–9772, 2012.
- [71] KARABOGA, D., “An idea based on honey bee swarm for numerical optimization”, *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.
- [72] AKAY, B., KARABOGA, D., “A modified artificial bee colony algorithm for real-parameter optimization”, *Information Sciences*, v. 192, pp. 120–142, 2012.
- [73] GREENWOOD, G. W., CHOPRA, S., “A modified artificial bee colony algorithm for solving large graph theory problems”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 713–717, 2013.
- [74] NARASIMHA, K. V., KIVELEVITCH, E., SHARMA, B., KUMAR, M., “An ant colony optimization technique for solving minâmax Multi-Depot Vehicle Routing Problem”, *Swarm and Evolutionary Computation*, 2013.
- [75] WANG, H., WU, Z., ZHOU, X., RAHNAMAYAN, S., “Accelerating artificial bee colony algorithm by using an external archive”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 517–521, 2013.
- [76] EL-ABD, M., “Local best Artificial Bee Colony algorithm with dynamic sub-populations”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 522–528, 2013.
- [77] YANG, X.-S., *Nature-inspired metaheuristic algorithms*. Luniver Press, 2008.
- [78] GANDOMI, A., YANG, X.-S., TALATAHARI, S., ALAVI, A., “Firefly algorithm with chaos”, *Communications in Nonlinear Science and Numerical Simulation*, v. 18, n. 1, pp. 89–98, 2013.
- [79] YANG, X.-S., “Firefly algorithms for multimodal optimization”, In: *Stochastic algorithms: foundations and applications*, pp. 169–178, Springer, 2009.
- [80] CHAI-EAD, N., AUNGKULANON, P., LUANGPAIBOON, P., “Bees and Firefly algorithms for noisy non-linear optimization problems”. In: *Proceedings of the*

International Multi Conference of Engineering and Computer Scientists, v. 2, 2011.

- [81] YANG, X.-S., DEB, S., “Cuckoo search via Lévy flights”. In: *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pp. 210–214, 2009.
- [82] ZHOU, Y., ZHENG, H., “A Novel Complex Valued Cuckoo Search Algorithm”, *The Scientific World Journal*, pp. 6, 2013.
- [83] YANG, X.-S., DEB, S., “Engineering optimization by cuckoo search”, *International Journal of Mathematical Modelling and Numerical Optimization*, v. 1, n. 4, pp. 330–343, 2010.
- [84] SHI, Y., EBERHART, R. C., “A modified particle swarm optimizer”. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pp. 69–73, 1998.
- [85] SHI, Y., EBERHART, R. C., “Parameter selection in particle swarm optimization”. In: *Evolutionary Programming VII*, pp. 591–600, 1998.
- [86] EBERHART, R. C., SHI, Y., “Comparing inertia weights and constriction factors in particle swarm optimization”. In: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, v. 1, pp. 84–88, 2000.
- [87] RATNAWEERA, A., HALGAMUGE, S. K., WATSON, H. C., “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients”, *Evolutionary Computation, IEEE Transactions on*, v. 8, n. 3, pp. 240–255, 2004.
- [88] SHI, Y., EBERHART, R. C., “Empirical study of particle swarm optimization”. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, v. 3, 1999.
- [89] CHATTERJEE, A., SIARRY, P., “Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization”, *Computers & Operations Research*, v. 33, n. 3, pp. 859–871, 2006.

- [90] KENNEDY, J., “The behavior of particles”. In: *Evolutionary programming VII*, pp. 579–589, 1998.
- [91] CLERC, M., KENNEDY, J., “The particle swarm-explosion, stability, and convergence in a multidimensional complex space”, *Evolutionary Computation, IEEE Transactions on*, v. 6, n. 1, pp. 58–73, 2002.
- [92] MARTINS, A., OLUYINKA, A., “An Adaptive Velocity Particle Swarm Optimization for high-dimensional function optimization”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 2352–2359, 2013.
- [93] PLUHACEK, M., SENKERIK, R., ZELINKA, I., “Investigation on the Performance of a New Multiple Choice Strategy for PSO Algorithm in the task of Large Scale Optimization Problems”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013.
- [94] BASTOS-FILHO, C. J., CARACIOLO, M. P., MIRANDA, P. B., CARVALHO, D. F., “Multi-ring particle swarm optimization”. In: *Neural Networks, 2008. SBRN’08. 10th Brazilian Symposium on*, pp. 111–116, 2008.
- [95] ROSENDO, M., *Um algoritmo de otimização por nuvem de partículas para resolução de problemas combinatórios*, Dissertação (Mestrado), Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná, 2010.
- [96] KENNEDY, J., “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance”. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, v. 3, 1999.
- [97] GENG, N., GONG, D., ZHANG, Y., “Robot path planning in an environment with many terrains based on interval multi-objective PSO”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 813–820, 2013.
- [98] ALJARAH, I., LUDWIG, S. A., “MapReduce Intrusion Detection System based on a Particle Swarm Optimization Clustering Algorithm”, .
- [99] YU, S., WEI, Y.-M., WANG, K., “A PSO–GA optimal model to estimate primary energy demand of China”, *Energy Policy*, v. 42, pp. 329–340, 2012.

- [100] HAMTA, N., FATEMI GHOMI, S., JOLAI, F., AKBARPOUR SHIRAZI, M., “A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect”, *International Journal of Production Economics*, v. 141, n. 1, pp. 99–111, 2013.
- [101] RODRIGUEZ-GARCIA, L., PEREZ-LONDONO, S., MORA-FLÓREZ, J., “Particle swarm optimization applied in power system measurement-based load modeling”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 2368–2375, 2013.
- [102] CAMP, C. V., MEYER, B. J., PALAZOLO, P. J., “Particle Swarm Optimization for The Design of Trusses”. In: *004@sBuilding on the Past, Securing the Future*, pp. 1–10, 2004.
- [103] GOMES, H. M., “Truss optimization with dynamic constraints using a particle swarm algorithm”, *Expert Systems with Applications*, v. 38, n. 1, pp. 957–968, 2011.
- [104] KAVEH, A., TALATAHARI, S., “Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization”, In: *Harmony Search Algorithms for Structural Design Optimization*, pp. 159–198, Springer, 2009.
- [105] LI, L., LIU, F., “Harmony particle swarm algorithm for structural design optimization”, In: *Harmony Search Algorithms for Structural Design Optimization*, pp. 121–157, Springer, 2009.
- [106] LI, L., HUANG, Z., LIU, F., “A heuristic particle swarm optimization method for truss structures with discrete variables”, *Computers & Structures*, v. 87, n. 7, pp. 435–443, 2009.
- [107] PEREZ, R., BEHDINAN, K., “Particle swarm approach for structural design optimization”, *Computers & Structures*, v. 85, n. 19, pp. 1579–1588, 2007.
- [108] CORREIA, A., MATIAS, J., SERÔDIO, C., “Métodos de penalidade exacta para resolução de problemas de optimização não linear”, *Investigação Operacional*, v. 28, n. 1, pp. 17–30, 2008.

- [109] COURANT, R., “Variational methods for the solution of problems of equilibrium and vibrations”, *Bull. Amer. Math. Soc.*, v. 49, n. 1, pp. 23, 1943.
- [110] CARROLL, C. W., “The created response surface technique for optimizing nonlinear, restrained systems”, *Operations Research*, v. 9, n. 2, pp. 169–184, 1961.
- [111] FIACCO, A. V., MCCORMICK, G. P., “Extensions of SUMT for nonlinear programming: equality constraints and extrapolation”, *Management Science*, v. 12, n. 11, pp. 816–828, 1966.
- [112] RUNARSSON, T. P., YAO, X., “Stochastic ranking for constrained evolutionary optimization”, *Evolutionary Computation, IEEE Transactions on*, v. 4, n. 3, pp. 284–294, 2000.
- [113] HOMAIFAR, A., QI, C. X., LAI, S. H., “Constrained optimization via genetic algorithms”, *Simulation*, v. 62, n. 4, pp. 242–253, 1994.
- [114] HOFFMEISTER, F., SPRAVE, J., “Problem-independent handling of constraints by use of metric penalty functions”. Citeseer, 1996.
- [115] JOINES, J. A., HOUCK, C. R., “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, 1994.
- [116] BEAN, J. C., HADJ-ALOUANE, A., “A dual genetic algorithm for bounded integer programs”, *Ann Arbor*, v. 1001, pp. 48109–2117, 1992.
- [117] SMITH, A. E., TATE, D. M., “Genetic optimization using a penalty function”. In: *Proceedings of the 5th international conference on genetic algorithms*, pp. 499–505, 1993.
- [118] COIT, D. W., SMITH, A. E., “Penalty guided genetic search for reliability design optimization”, *Computers & industrial engineering*, v. 30, n. 4, pp. 895–904, 1996.

- [119] 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.
- [120] HAMIDA, S. B., SCHOENAUER, M., “An adaptive algorithm for constrained optimization problems”. In: *Parallel Problem Solving from Nature PPSN VI*, pp. 529–538, 2000.
- [121] LIN, C.-Y., WU, W.-H., “Self-organizing adaptive penalty strategy in constrained genetic search”, *Structural and Multidisciplinary Optimization*, v. 26, n. 6, pp. 417–428, 2004.
- [122] TESSEMA, B., YEN, G. G., “A self adaptive penalty function based algorithm for constrained optimization”. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 246–253, 2006.
- [123] 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*, 2012.
- [124] MEZURA-MONTES, E., *Constraint-handling in evolutionary optimization*. v. 198. Springer, 2009.
- [125] AGUIRRE, A. H., ZAVALA, A. M., DIHARCE, E. V., RIONDA, S. B., “COPSO: Constrained Optimization via PSO algorithm”, *Center for Research in Mathematics (CIMAT). Technical report No. I-07-04/22-02-2007*, 2007.
- [126] RAY, T., LIEW, K., “A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimisation problems”. In: *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, v. 1, pp. 75–80, 2001.
- [127] HU, X., EBERHART, R. C., SHI, Y., “Engineering optimization with particle swarm”. In: *Swarm Intelligence Symposium, 2003. SIS’03. Proceedings of the 2003 IEEE*, pp. 53–57, 2003.
- [128] MUÑOZ ZAVALA, A. E., AGUIRRE, A. H., VILLA DIHARCE, E. R., “Constrained optimization via particle evolutionary swarm optimization

- algorithm (PESO)". In: *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 209–216, 2005.
- [129] 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, pp. 340–349, 2007.
- [130] HE, S., WU, Q., WEN, J., SAUNDERS, J., PATON, R., "A particle swarm optimizer with passive congregation", *Biosystems*, v. 78, n. 1, pp. 135–147, 2004.
- [131] GEEM, Z. W., KIM, J. H., LOGANATHAN, G., "A new heuristic optimization algorithm: harmony search", *Simulation*, v. 76, n. 2, pp. 60–68, 2001.
- [132] DEB, K., "An efficient constraint handling method for genetic algorithms", *Computer methods in applied mechanics and engineering*, v. 186, n. 2, pp. 311–338, 2000.
- [133] CAGNINA, L. C., ESQUIVEL, S. C., COELLO, C. A. C., "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer", *Informatica (Slovenia)*, v. 32, n. 3, pp. 319–326, 2008.
- [134] BARBOSA, H. J., LEMONGE, A. C., "A new adaptive penalty scheme for genetic algorithms", *Information Sciences*, v. 156, n. 3, pp. 215–251, 2003.
- [135] LEMONGE, A. C., BARBOSA, H. J., "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.
- [136] YOUNG, C.-T., ZHENG, Y., YEH, C.-W., JANG, S.-S., "Information-Guided Genetic Algorithm Approach to the Solution of MINLP Problems", *Industrial & Engineering Chemistry Research*, v. 46, n. 5, pp. 1527–1537, 2007.
- [137] GALLET, C., SALAUN, M., BOUCHET, E., "An example of global structural optimisation with genetic algorithms in the aerospace field". In: *Proceedings of The VIII International Conference on Computational Plasticity COMPLAS VIII, Barcelona*, pp. 1–4, 2005.

- [138] BARBOSA, H. J., LEMONGE, A. C., FONSECA, L. G., BERNARDINO, H. S., “Comparing two constraint handling techniques in a binary-coded genetic algorithm for optimization problems”, *Lecture Notes in Computer Science*, v. 6457, pp. 125–134, 2010.
- [139] LUCENA, R., COUTINHO, D., LIMA, B., BAIOCO, J., ALBRECHT, C., JACOB, B., “A comparative study of constraint-handling methodologies applied to genetic algorithm for the optimization of submarine pipeline routes”. In: *Proceedings of 3rd International Conference on Engineering Optimization - EngOpt 2012*, pp. 1–10, Rio de Janeiro, Brazil, 01 - 05 July 2012., 2012.
- [140] OLIVEIRA, L., AFONSO, S., HOROWITZ, B., LEMONGE, A., “Constraints Handling for Hybrid Algorithms in Waterflooding Optimization Problem”. In: *Proceedings of 3rd International Conference on Engineering Optimization - EngOpt 2012*, pp. 1–8, Rio de Janeiro, Brazil, 01 - 05 July 2012., 2012.
- [141] BARBOSA, H., LEMONGE, A., BORGES, C., “A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization.” *Engineering Structures*, v. 30, pp. 3708–3723, 2008.
- [142] LIU, X., CHENG, G., WANG, B., LIN, S., “Optimum design of pile foundation by automatic grouping genetic algorithms”, *International Scholarly Research Network ISRN Civil Engineering*, v. 2012, pp. 1–16, 2012, Article ID 678329.
- [143] YOUSEFI, M., ENAYATIFAR, R., DARUS, A. N., ABDULLAH, A. H., “Optimization of plate-fin heat exchangers by an improved harmony search algorithm”, *Applied Thermal Engineering*, v. 50, n. 1, pp. 877 – 885, 2013.
- [144] CARVALHO, E. C. R., MONTÁ, B. G., GARCIA, R. P., LEMONGE, A. C. C., BERNARDINO, H. S., BARBOSA, H. J. C., “Variants of the Adaptive Penalty Methods (APM) for Constrained Optimization Problems Applied to Engineering Problems”, *XXXIV Iberian Latin-American Congress on Computational Methods in Engineering - CILAMCE*, 2013.
- [145] SILVA, F. B. S., *Algoritmos genéticos para otimização de estruturas reticuladas baseadas em modelos adaptativos e lagrangeano aumentado*, Dissertação

(Mestrado), Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, 2011.

- [146] BARBOSA, H. J., BERNARDINO, H. S., BARRETO, A. D. M. S., “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, 2010.
- [147] BARRETO, A., 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*, pp. 751–758, 2010.
- [148] BERNARDINO, H. S., BARBOSA, H. J., FONSECA, L. G., “Surrogate-assisted clonal selection algorithms for expensive optimization problems”, *Evolutionary Intelligence*, v. 4, n. 2, pp. 81–97, 2011.
- [149] MEZURA-MONTES, E., COELLO, C. A. C., LANDA-BECERRA, R., “Engineering Optimization Using a Simple Evolutionary Algorithm”, , pp. 149–1562003.
- [150] BARBOSA, H. J. C., LEMONGE, A. C. C., “An Adaptive Penalty Scheme In Genetic Algorithms For Constrained Optimiazation Problems”. In: *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 287–294, Morgan Kaufmann Publishers: New York, 9-13 July 2002.
- [151] BELEGUNDU, A., *A Study of Mathematical Programming Methods fos Structural Optimization*, Tech. rep., Dept. of Civil and Environmental Engineering, University of Iowa, 1982.
- [152] COELLO, C., “Self-adaptive penalties for GA based optimization”. In: *Proceedings of Congress on Evolutionary Computation*, v. 1, pp. 573–580, 1999.
- [153] COELLO, C., “Constraint-handling using an evolutionary multiobjective optimization technique”, *Civil Engineering and Environmental Systems*, v. 17, n. 4, pp. 319–346, 2000.

- [154] COELLO, C., BECERRA, R., “Efficient evolutionary through the use of a cultural algorithm”, *Engineering Optimization*, v. 36, pp. 219–236, 2004.
- [155] COELLO, C., MONTES, E., “Use of dominance-based tournament selection to handle constraints in genetic algorithms”. In: *Intelligent Engineering Systems through Artificial Neural Networks (ANNIE2001)*, v. 11, St. Louis, Missouri.
- [156] COELHO, L. S., “Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems”, *Expert Syst Appl*, v. 37, n. 2, pp. 1676–1683, 2010.
- [157] HE, Q., WANG, L., “An effective co-evolutionary particle swarm optimization for engineering optimization problems”, *Appl Artif Intell*, v. 20, pp. 89–99, 2006.
- [158] HE, S., PREMPAIN, E., WU, Q., “An improved particle swarm optimizer for mechanical design optimization problems”, *Engineering Optimization*, v. 36, n. 5.
- [159] HEDAR, A., FUKUSHIMA, M., “Derivative-free filter simulated annealing method for constrained continuous global optimization”, *Journal of Global Optimization*, , n. 32.
- [160] HU, X., EBERHART, R., SHI, Y., “Engineering optimization with particle swarm”. In: *Proceedings of 2003 IEEE swarm intelligence symposium*, pp. 53–57, 2003.
- [161] HUANG, F., WANG, L., LE, Q., “An effective co-evolutionary differential evolution for constrained optimization”, *Appl Math Comput*, v. 1, n. 186, pp. 340–356, 2007.
- [162] MONTES, E., COELLO, C., “An empirical study about the usefulness of evolution strategies to solve constrained optimization problems”, *Int J Gen Syst*, v. 37, n. 4, pp. 443–473, 2008.
- [163] MONTES, E., OCAÑÁ, B., “Bacterial foraging for engineering design problems: preliminary results”. In: *Proceedings of the 4th Mexican Congress on Evolutionary Computation (COMCEV 2008)*, pp. 33–38, CIMAT, México, 2008.

- [164] PARSOPOULOS, K., VRAHATIS, M., “Unified particle swarm optimization for solving constrained engineering optimization problems”. In: *Lecture notes in computer science (LNCS)*, v. 3612, pp. 582–591, 2008.
- [165] RAY, T., LIEW, K., “Society and civilization: An optimization algorithm based on the simulation of social behavior”, *Evolutionary Computation, IEEE Transactions on*, v. 7, n. 4, pp. 386–396, 2003.
- [166] RAY, T., SAINI, P., “Engineering design optimization using a swarm with an intelligent information sharing among individuals”, *Engineering Optimization*, v. 33, n. 3, pp. 735–748, 2001.
- [167] 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.
- [168] MEZURA-MONTES, E., COELLO, C. A. C., LANDA-BECERRA, R., “Engineering Optimization Using a Simple Evolutionary Algorithm”. pp. 149–156, 15th Intl. Conf. on Tools with Art. Intelligence, ICTAI, CA, USA, 2003.
- [169] LEITE, J., TOPPING, B., “Improved genetic operators for structural engineering optimization”, *Advances in Engeneering Software*, v. 29, n. 7-9, pp. 529–562, 1998.
- [170] 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, June 2000.
- [171] DEB, K., “Optimal design of a welded beam via genetic algorithms”, *AIAA Journal.*, v. 29, n. 11, pp. 2012–2015, 1991.
- [172] ATIQULLAH, M., RAO, S., “Simulated annealing and parallel processing: an implementation for constrained global design optimization”, *Engineering Optimization*, v. 32, n. 5.
- [173] LIU, J.-L., “Orthogonal simulated annealing with fractional factorial analysis to solve global optimization problems”, *Engineering Optimizatiom*, , n. 37.

- [174] HWANG, S.-F., HE, R.-S., “A hybrid real-parameter genetic algorithm for function optimization”. N. 20, pp. 7–21, 2006.
- [175] “An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization”, *Applied Mathematics and Computation*, v. 211, n. 2, pp. 392–416, 2009.
- [176] SIDALL, J., *Analytical decision-making in engineering design*. Prentice-Hall, Englewood Cliffs, 1972.
- [177] AKHTAR, S., TAI, K., RAY, T., “A socio-behavioral simulation model for engineering design optimization”, *Engineering Optimization*, v. 34, n. 4, pp. 341–354, 2002.
- [178] RAGSDELL, K., PHILLIPS, D., “Optimal Design of a Class of Welded Structures Using Geometric Programming”, *Journal of Engineering for Industry*, v. 98, n. 3, pp. 1021–1025, 1976.
- [179] BARBOSA, H., LEMONGE, A., “An Adaptive Penalty Method for Genetic Algorithms in Constrained Optimization Problems”, In: IBA, H. (ed), *Frontiers in Evolutionary Algorithms*, pp. 9–34, I-Tech, 2008.
- [180] CAGNINA, L., ESQUIVEL, S., COELLO, C., “Solving engineering optimization problems with the simple constrained particle swarm optimizer”, *Informatica*, v. 32, pp. 319–326, 2008.
- [181] COELLO, C., “Use of a self-adaptive penalty approach for engineering optimization problems”, *Computers in Industry*, v. 41, n. 2, pp. 113–127, 2000.
- [182] DEB, K., “GeneAS: A Robust Optimal Design Technique for Mechanical Component Design”. In: *Evolutionary Algorithms in Engineering Applications*, pp. 497–514, Springer-Verlag, Berlin, 1997.
- [183] BEAN, J., ALOUANE, A., “A genetic algorithm for the multiple-choice integer program”, *Oper Res*, v. 45, pp. 92–101, 1999.
- [184] HOMAIFAR, A., LAI, S., QI, X., “Constrained optimization via genetic algorithms”, *Simulation*, v. 62, n. 4, pp. 242–254, 1994.

- [185] JOINES, J., HOUCK, C. R., “On the use of non-stationary penalty methods to solve nonlinear constrained optimization problems with GAs”. In: *Proc. of 1994 IEEE Conf. on Evolutionary Computation*, pp. 579–585, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 1994.
- [186] LI, H., CHANG, C., “An approximate approach of global optimization for polynomial programming problems”, *Eur J Oper Res*, v. 107, n. 3, 1998.
- [187] MICHALEWICZ, Z., ATTIA, N., “Evolutionary optimization of constrained problems”. In: *Proceedings of the 3rd annual conf on evolutionary programming*, pp. 98–108, 1994.
- [188] ERBATUR, F., HASANÇEBİ, O., TÜTÜNCÜ, I., KILIÇ, H., “Optimal design of planar and space structures with genetic algorithms”, *Computers & Structures*, v. 75, n. 2, pp. 209–224, 2000.
- [189] BERNARDINO, H. S., BARBOSA, H. J., LEMONGE, A. C., FONSECA, L., “A new hybrid AIS-GA for constrained optimization problems in mechanical engineering”. In: *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 1455–1462, 2008.
- [190] ARAGÓN, V. S., ESQUIVEL, S. C., COELLO, C. A. C., “A modified version of a T-Cell Algorithm for constrained optimization problems”, *International Journal for Numerical Methods in Engineering*, v. 84, n. 3, pp. 351–378, 2010.
- [191] LIANG, J., QU, B., SUGANTHAN, P., *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*, Tech. rep., Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.

APÊNDICE A - *Resultados G-Suite*

Resultados encontrados para as funções *G-Suite* considerando os três níveis de avaliação da função objetivo, 5000, 50000 e 500000.

Tabela A.1: Resultados encontrados para a função G1. Melhor valor conhecido: -15.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-14.999916	-14.985747	-13.513761	1.106983e+01	-7.002509	35/35	5000
APM_Med_3	-14.999918	-14.999760	-14.728093	3.831314e+00	-12.997032	35/35	5000
APM_Worst	-14.999957	-14.998778	-13.460143	1.183127e+01	-9.001392	35/35	5000
APM_Spor_Mono	-14.999944	-14.999681	-14.057300	7.858376e+00	-11.999721	35/35	5000
APM	-14.999995	-14.999983	-14.599983	4.732876e+00	-12.999971	35/35	50000
APM_Med_3	-14.999997	-14.999984	-14.885699	2.746421e+00	-12.999987	35/35	50000
APM_Worst	-14.999992	-14.999979	-14.755784	4.049645e+00	-12.453054	35/35	50000
APM_Spor_Mono	-14.999994	-14.999985	-14.771411	3.764531e+00	-12.999958	35/35	50000
APM	-14.999998	-14.999992	-14.771422	3.764498e+00	-12.999991	35/35	500000
APM_Med_3	-14.999997	-14.999993	-14.771422	3.764501e+00	-12.999989	35/35	500000
APM_Worst	-14.999998	-14.999993	-14.599993	4.732872e+00	-12.999987	35/35	500000
APM_Spor_Mono	-14.999998	-14.999992	-14.623653	4.518436e+00	-12.999986	35/35	500000

Tabela A.2: Resultados encontrados para a função G2. Melhor valor conhecido: -0.803619.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-0.623216	-0.429518	-0.427235	5.248323e-01	-0.265298	35/35	5000
APM_Med.3	-0.628424	-0.484628	-0.466756	4.936998e-01	-0.310252	35/35	5000
APM_Worst	-0.675435	-0.472529	-0.461551	5.564616e-01	-0.299121	35/35	5000
APM_Spor_Mono	-0.635249	-0.479895	-0.466707	4.600990e-01	-0.279513	35/35	5000
APM	-0.630207	-0.476232	-0.476353	4.286699e-01	-0.318167	35/35	50000
APM_Med.3	-0.731990	-0.525771	-0.516027	6.102338e-01	-0.343078	35/35	50000
APM_Worst	-0.710287	-0.499941	-0.501189	6.043827e-01	-0.314523	35/35	50000
APM_Spor_Mono	-0.777343	-0.492360	-0.497701	6.623740e-01	-0.287536	35/35	50000
APM	-0.771002	-0.550075	-0.531664	6.841186e-01	-0.326777	35/35	500000
APM_Med.3	-0.786159	-0.502737	-0.525108	7.109076e-01	-0.307339	35/35	500000
APM_Worst	-0.770400	-0.549819	-0.542841	6.025362e-01	-0.322065	35/35	500000
APM_Spor_Mono	-0.751509	-0.469866	-0.503042	6.334190e-01	-0.332635	35/35	500000

Tabela A.3: Resultados encontrados para a função G3. Melhor valor conhecido: -1.000500.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-0.942513	-0.334711	-0.386968	1.428892e+00	-0.000000	16/35	5000
APM_Med.3	-0.846017	-0.510976	-0.435413	1.182529e+00	-0.000000	14/35	5000
APM_Worst	-0.937964	-0.639657	-0.560414	1.163313e+00	-0.000000	17/35	5000
APM_Spor_Mono	-0.008964	-0.000000	-0.000467	1.048240e-02	-0.000000	35/35	5000
APM	-1.000277	-0.895246	-0.827518	9.870566e-01	-0.361582	30/35	50000
APM_Med.3	-0.994341	-0.887950	-0.826004	1.004246e+00	-0.051974	22/35	50000
APM_Worst	-0.997906	-0.955321	-0.842447	1.043972e+00	-0.127671	25/35	50000
APM_Spor_Mono	-0.066992	-0.001467	-0.010752	1.050963e-01	-0.000000	28/35	50000
APM	-1.000497	-1.000494	-1.000441	1.091512e-03	-0.999387	26/35	500000
APM_Med.3	-1.000499	-1.000496	-1.000487	1.158874e-04	-1.000409	29/35	500000
APM_Worst	-1.000498	-1.000495	-1.000329	3.039083e-03	-0.998029	27/35	500000
APM_Spor_Mono	-0.999494	-0.758967	-0.750031	9.462013e-01	-0.307686	26/35	500000

Tabela A.4: Resultados encontrados para a função G4. Melhor valor conhecido: -30665.538671.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-30665.521200	-30665.454312	-30665.287155	3.214469e+00	-30662.540500	35/35	5000
APM_Med.3	-30665.533112	-30665.470744	-30665.212453	5.677576e+00	-30659.950413	35/35	5000
APM_Worst	-30665.511135	-30665.449830	-30665.417291	5.808929e-01	-30665.099098	35/35	5000
APM_Spor_Mono	-30665.518363	-30665.450396	-30665.069148	9.252902e+00	-30656.172493	35/35	5000
APM	-30665.538580	-30665.537904	-30665.537920	2.379537e-03	-30665.537032	35/35	50000
APM_Med.3	-30665.538614	-30665.538025	-30665.537993	2.196542e-03	-30665.537038	35/35	50000
APM_Worst	-30665.538541	-30665.537828	-30665.537826	2.059609e-03	-30665.536943	35/35	50000
APM_Spor_Mono	-30665.538579	-30665.537977	-30665.537891	2.188643e-03	-30665.537066	35/35	50000
APM	-30665.538645	-30665.538531	-30665.538507	6.022388e-04	-30665.538259	35/35	500000
APM_Med.3	-30665.538659	-30665.538498	-30665.538477	5.686566e-04	-30665.538246	35/35	500000
APM_Worst	-30665.538649	-30665.538513	-30665.538501	5.420882e-04	-30665.538312	35/35	500000
APM_Spor_Mono	-30665.538662	-30665.538507	-30665.538504	5.268366e-04	-30665.538276	35/35	500000

Tabela A.5: Resultados encontrados para a função G5. Melhor valor conhecido: 5126.496714.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-	-	-	-	-	0/35	500000
APM_Med_3	5147.087297	5147.087297	5147.087297	0.000000e+00	5147.087297	1/35	500000
APM_Worst	-	-	-	-	-	0/35	500000
APM_Spor_Mono	-	-	-	-	-	0/35	500000

Tabela A.6: Resultados encontrados para a função G6. Melhor valor conhecido: -6961.813875.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-6961.366893	-6956.131479	-6952.265417	6.403016e+01	-6905.333790	33/35	5000
APM_Med_3	-6960.928189	-6951.117622	-6948.159811	5.910853e+01	-6907.940370	32/35	5000
APM_Worst	-6960.860055	-6955.418531	-6952.765660	4.210972e+01	-6932.548301	32/35	5000
APM_Spor_Mono	-6959.891297	-6917.567858	-6893.373598	4.560081e+02	-6500.224434	25/35	5000
APM	-6961.809034	-6961.796293	-6961.793856	5.146332e-02	-6961.766308	35/35	50000
APM_Med_3	-6961.807248	-6961.798553	-6961.797424	3.871357e-02	-6961.779491	35/35	50000
APM_Worst	-6961.810489	-6961.797790	-6961.795362	5.758852e-02	-6961.767216	35/35	50000
APM_Spor_Mono	-6961.810809	-6961.782189	-6961.782789	1.031371e-01	-6961.736478	35/35	50000
APM	-6961.813569	-6961.810918	-6961.810778	8.152511e-03	-6961.807545	35/35	500000
APM_Med_3	-6961.813499	-6961.811485	-6961.811324	7.412243e-03	-6961.807991	35/35	500000
APM_Worst	-6961.813391	-6961.811821	-6961.811310	8.371268e-03	-6961.808208	35/35	500000
APM_Spor_Mono	-6961.813071	-6961.810381	-6961.810551	9.103603e-03	-6961.807236	35/35	500000

Tabela A.7: Resultados encontrados para a função G7. Melhor valor conhecido: 24.306209.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	25.295408	28.781330	33.289840	1.579839e+02	187.660383	35/35	5000
APM_Med_3	25.349708	29.469173	41.746449	2.351985e+02	188.287568	35/35	5000
APM_Worst	25.235925	29.025477	34.349418	1.021189e+02	128.062935	35/35	5000
APM_Spor_Mono	24.739301	29.170985	41.790638	2.074276e+02	184.286184	35/35	5000
APM	24.525137	26.565734	27.202860	1.031970e+01	31.139377	35/35	50000
APM_Med_3	24.593349	26.469898	27.160157	1.291342e+01	32.030171	35/35	50000
APM_Worst	24.481880	27.664497	28.357040	2.382309e+01	43.914534	35/35	50000
APM_Spor_Mono	24.493838	26.589226	28.125475	2.346705e+01	43.166854	35/35	50000
APM	24.483265	25.501782	26.006466	8.860099e+00	29.925030	35/35	500000
APM_Med_3	24.362391	25.273331	25.795374	8.788768e+00	30.138765	35/35	500000
APM_Worst	24.406910	25.498204	25.681991	6.711048e+00	29.112591	35/35	500000
APM_Spor_Mono	24.367583	25.151848	25.366300	5.807761e+00	29.345105	35/35	500000

Tabela A.8: Resultados encontrados para a função G8. Melhor valor conhecido: -0.095825.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-0.095825	-0.095825	-0.095824	5.291974e-05	-0.095771	35/35	5000
APM_Med_3	-0.095825	-0.095825	-0.095825	6.810869e-08	-0.095825	35/35	5000
APM_Worst	-0.095825	-0.095825	-0.095825	5.774690e-07	-0.095824	35/35	5000
APM_Spor_Mono	-0.095825	-0.095825	-0.095825	2.990344e-10	-0.095825	35/35	5000
APM	-0.095825	-0.095825	-0.095825	7.991429e-12	-0.095825	34/35	50000
APM_Med_3	-0.095825	-0.095825	-0.095825	1.414031e-11	-0.095825	35/35	50000
APM_Worst	-0.095825	-0.095825	-0.095825	1.478570e-11	-0.095825	35/35	50000
APM_Spor_Mono	-0.095825	-0.095825	-0.095825	1.066869e-11	-0.095825	34/35	50000
APM	-0.095825	-0.095825	-0.095825	6.551302e-13	-0.095825	35/35	500000
APM_Med_3	-0.095825	-0.095825	-0.095825	5.595308e-13	-0.095825	35/35	500000
APM_Worst	-0.095825	-0.095825	-0.095825	8.743868e-13	-0.095825	35/35	500000
APM_Spor_Mono	-0.095825	-0.095825	-0.095825	6.728809e-13	-0.095825	35/35	500000

Tabela A.9: Resultados encontrados para a função G9. Melhor valor conhecido: 680.630057.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	680.693331	681.091027	681.219722	2.382933e+00	682.682853	35/35	5000
APM_Med_3	680.787388	681.186754	681.301657	2.479356e+00	682.941604	35/35	5000
APM_Worst	680.806389	681.238103	681.374953	2.643107e+00	682.293300	35/35	5000
APM_Spor_Mono	680.836751	681.155798	681.350692	3.334917e+00	683.713122	35/35	5000
APM	680.640700	680.708624	680.719023	3.385538e-01	680.870747	35/35	50000
APM_Med_3	680.636867	680.686727	680.714915	4.750741e-01	680.989708	35/35	50000
APM_Worst	680.637227	680.683680	680.699094	3.057902e-01	680.860921	35/35	50000
APM_Spor_Mono	680.645015	680.693379	680.709292	3.248599e-01	680.883123	35/35	50000
APM	680.633046	680.639659	680.641056	3.633214e-02	680.655539	35/35	500000
APM_Med_3	680.632943	680.640659	680.642579	5.460579e-02	680.684606	35/35	500000
APM_Worst	680.631461	680.638087	680.640618	4.822911e-02	680.666669	35/35	500000
APM_Spor_Mono	680.631642	680.640082	680.641241	4.155849e-02	680.657040	35/35	500000

Tabela A.10: Resultados encontrados para a função G10. Melhor valor conhecido: 7049.248020.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	8015.289175	8063.693422	8063.693422	6.845394e+01	8112.097669	2/35	5000
APM_Med_3	8622.361094	8622.361094	8622.361094	0.000000e+00	8622.361094	1/35	5000
APM_Worst	7975.462148	7978.069471	8109.321751	3.246997e+02	8374.433633	3/35	5000
APM_Spor_Mono	-	-	-	-	-	0/35	5000
APM	7875.252341	7875.252341	7875.252341	0.000000e+00	7875.252341	1/35	50000
APM_Med_3	8154.983760	8154.983760	8154.983760	0.000000e+00	8154.983760	1/35	50000
APM_Worst	7413.856671	7633.136789	7633.136789	3.101089e+02	7852.416906	2/35	50000
APM_Spor_Mono	7491.831468	7930.302987	7930.302987	6.200924e+02	8368.774506	2/35	50000
APM	7469.048455	7469.064398	7474.935221	1.440005e+01	7486.692809	3/35	500000
APM_Med_3	7501.937211	7501.937211	7501.937211	0.000000e+00	7501.937211	1/35	500000
APM_Worst	7469.048248	7469.051578	7485.031809	3.914749e+01	7516.995602	3/35	500000
APM_Spor_Mono	7270.261287	7285.512810	7285.512810	2.156891e+01	7300.764333	2/35	500000

Tabela A.11: Resultados encontrados para a função G11. Melhor valor conhecido: 0.7499.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	0.787213	0.992521	0.961153	3.478622e-01	0.997122	35/35	5000
APM_Med_3	0.750104	0.993233	0.977031	3.341606e-01	0.998061	35/35	5000
APM_Worst	0.823039	0.991374	0.966417	2.851458e-01	0.997555	35/35	5000
APM_Spor_Mono	0.751750	0.992267	0.970510	3.189786e-01	0.997949	35/35	5000
APM	0.750425	0.905711	0.896677	2.219742e-01	0.942969	35/35	50000
APM_Med_3	0.749904	0.902225	0.882162	2.829798e-01	0.925663	35/35	50000
APM_Worst	0.749917	0.890017	0.870791	3.014903e-01	0.941586	35/35	50000
APM_Spor_Mono	0.752217	0.887136	0.879446	2.976230e-01	0.942922	35/35	50000
APM	0.749900	0.749900	0.751129	4.237829e-02	0.792898	35/35	500000
APM_Med_3	0.749900	0.749900	0.749904	1.013221e-04	0.750002	35/35	500000
APM_Worst	0.749900	0.749900	0.749903	5.450554e-05	0.749954	35/35	500000
APM_Spor_Mono	0.749900	0.749900	0.749901	1.425145e-05	0.749910	35/35	500000

Tabela A.12: Resultados encontrados para a função G12. Melhor valor conhecido: -1.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Med.3	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Worst	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Spor_Mono	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Med.3	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Worst	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Spor_Mono	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Med.3	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Worst	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000
APM_Spor_Mono	-1.000000	-1.000000	-1.000000	0.000000	-1.000000	35/35	5000

Tabela A.13: Resultados encontrados para a função G13. Melhor valor conhecido: 0.053941.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	0.788830	0.996656	0.961834	1.897056e-01	0.999819	6/35	5000
APM_Med.3	0.548691	0.998819	0.922871	4.086333e-01	1.000000	7/35	5000
APM_Worst	0.778845	0.998382	0.950907	2.143295e-01	0.999959	8/35	5000
APM_Spor_Mono	0.572335	0.996636	0.896335	5.196877e-01	0.999998	12/35	5000
APM	0.371459	0.971689	1.118360	6.225404e+00	7.198036	31/35	50000
APM_Med.3	0.119425	0.978563	0.953736	4.284284e+00	4.970328	33/35	50000
APM_Worst	0.440732	0.889414	0.805604	1.131975e+00	1.080052	30/35	50000
APM_Spor_Mono	0.075849	0.989516	0.913742	1.569192e+00	1.903880	35/35	50000
APM	0.119859	0.867777	0.789389	1.556636e+00	1.161256	35/35	50000
APM_Med.3	0.055630	0.919704	0.818831	1.292381e+00	1.063849	35/35	50000
APM_Worst	0.103651	0.919718	0.907038	3.229511e+00	3.605466	35/35	50000
APM_Spor_Mono	0.068170	0.985557	1.428848	1.673405e+01	17.769561	35/35	50000

Tabela A.14: Resultados encontrados para a função G14. Melhor valor conhecido: -47.764885.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-47.576899	-45.670398	-45.268715	8.525478e+00	-41.670434	17/35	5000
APM_Med_3	-47.457790	-44.093522	-44.246895	9.648916e+00	-39.430548	17/35	5000
APM_Worst	-47.577338	-45.378192	-44.540612	1.307667e+01	-38.272517	17/35	5000
APM_Spor_Mono	-47.442823	-46.767421	-45.886868	4.826108e+00	-43.242357	8/35	5000
APM	-47.563331	-46.074483	-45.230706	1.313763e+01	-39.571577	35/35	50000
APM_Med_3	-47.515958	-46.127983	-45.205467	1.392030e+01	-38.738744	35/35	50000
APM_Worst	-47.657789	-46.869361	-45.475974	1.414596e+01	-40.656486	35/35	50000
APM_Spor_Mono	-47.655553	-47.432597	-47.175523	2.667478e+00	-45.224320	17/35	50000
APM	-47.672282	-45.490896	-45.196024	1.139824e+01	-40.391310	35/35	500000
APM_Med_3	-47.596602	-45.416246	-45.295818	1.189608e+01	-39.674610	35/35	500000
APM_Worst	-47.600764	-45.634822	-44.611713	1.802523e+01	-35.472343	35/35	500000
APM_Spor_Mono	-47.744885	-47.671146	-46.905324	4.143589e+00	-43.801601	10/35	500000

Tabela A.15: Resultados encontrados para a função G15. Melhor valor conhecido: 961.715022.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	961.715097	964.202033	964.853775	1.477181e+01	972.312765	22/35	5000
APM_Med_3	961.715164	963.089208	964.795759	1.708173e+01	972.315329	26/35	5000
APM_Worst	961.758725	963.868764	965.159677	1.651512e+01	972.313083	22/35	5000
APM_Spor_Mono	961.722864	965.386387	965.736223	1.756738e+01	972.316255	23/35	5000
APM	961.718998	962.152508	964.241129	1.937750e+01	972.311046	35/35	50000
APM_Med_3	961.724070	963.003792	964.973819	2.074334e+01	972.308470	35/35	50000
APM_Worst	961.725527	962.713094	964.312433	1.911385e+01	972.310448	35/35	50000
APM_Spor_Mono	961.719832	962.189481	963.849834	1.790124e+01	972.307061	34/35	50000
APM	961.715186	963.140098	964.214274	1.574122e+01	971.012139	35/35	500000
APM_Med_3	961.716576	966.421902	966.668530	2.486614e+01	972.297068	35/35	500000
APM_Worst	961.715052	962.355489	964.562517	2.171690e+01	972.298357	35/35	500000
APM_Spor_Mono	961.715089	962.278845	964.113188	1.812814e+01	972.286563	35/35	500000

Tabela A.16: Resultados encontrados para a função G16. Melhor valor conhecido: -1.905155.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-1.904932	-1.901606	-1.858155	7.556367e-01	-1.414984	34/35	5000
APM_Med.3	-1.904959	-1.901209	-1.853879	7.797867e-01	-1.430645	34/35	5000
APM_Worst	-1.904961	-1.901479	-1.900748	3.487925e-02	-1.870205	34/35	5000
APM_Spor_Mono	-1.904920	-1.901646	-1.862117	7.601920e-01	-1.430649	35/35	5000
APM	-1.905154	-1.905152	-1.890623	4.667562e-01	-1.430653	35/35	50000
APM_Med.3	-1.905155	-1.905153	-1.877299	6.505595e-01	-1.430653	35/35	50000
APM_Worst	-1.905154	-1.905153	-1.890712	4.668415e-01	-1.430653	35/35	50000
APM_Spor_Mono	-1.905155	-1.905153	-1.890490	4.666286e-01	-1.430651	35/35	50000
APM	-1.905155	-1.905155	-1.904376	7.886580e-03	-1.901827	35/35	500000
APM_Med.3	-1.905155	-1.905154	-1.890438	4.654991e-01	-1.431718	35/35	500000
APM_Worst	-1.905155	-1.905155	-1.904489	7.408038e-03	-1.901823	35/35	500000
APM_Spor_Mono	-1.905155	-1.905155	-1.890497	4.666423e-01	-1.430653	35/35	500000

Tabela A.17: Resultados encontrados para a função G17. Melhor valor conhecido: 8853.539674.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-	-	-	-	-	0/35	5000
APM_Med_3	-	-	-	-	-	0/35	5000
APM_Worst	-	-	-	-	-	0/35	5000
APM_Spor_Mono	9011.912512	9011.912791	9011.956660	1.522880e-01	9012.088545	4/35	5000
APM	9012.491007	9012.491007	9012.491007	0.000000e+00	9012.491007	1/35	50000
APM_Med_3	9012.766571	9012.766571	9012.766571	0.000000e+00	9012.766571	1/35	50000
APM_Worst	9011.912305	9011.912305	9011.912305	0.000000e+00	9011.912305	1/35	50000
APM_Spor_Mono	9011.912583	9011.913108	9011.913108	7.426090e-04	9011.913633	2/35	50000
APM	8886.271385	9010.700826	9031.682308	5.367118e+02	9260.471970	24/35	500000
APM_Med_3	8930.856545	9012.881187	9101.043464	6.452841e+02	9304.244351	21/35	500000
APM_Worst	8861.092190	9011.159278	9026.649371	5.090741e+02	9281.578682	21/35	500000
APM_Spor_Mono	8924.080417	8989.392339	9025.071962	4.278606e+02	9281.485714	14/35	500000

Tabela A.18: Resultados encontrados para a função G18. Melhor valor conhecido: -0.866025.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-0.808874	-0.807758	-0.782255	6.383931e-02	-0.730135	3 /35	5000
APM_Med_3	-	-	-	-	-	0/35	5000
APM_Worst	-0.801925	-0.791673	-0.791673	1.449881e-02	-0.781421	2/35	5000
APM_Spor_Mono	-0.850696	-0.796112	-0.777081	2.955463e-01	-0.613340	21/35	5000
APM	-0.865950	-0.863995	-0.861525	2.716774e-02	-0.850337	35 /35	50000
APM_Med_3	-0.865771	-0.852429	-0.837278	1.047773e-01	-0.743662	6/35	50000
APM_Worst	-0.865900	-0.863748	-0.861977	2.820198e-02	-0.841293	35 /35	50000
APM_Spor_Mono	-0.865938	-0.862629	-0.835783	4.941783e-01	-0.499778	29/35	50000
APM	-0.865974	-0.861501	-0.819687	6.803727e-01	-0.499964	35 /35	500000
APM_Med_3	-0.865964	-0.863863	-0.849143	3.565243e-01	-0.499975	29/35	500000
APM_Worst	-0.865946	-0.862966	-0.840105	4.967131e-01	-0.499973	35 /35	500000
APM_Spor_Mono	-0.865956	-0.863491	-0.846154	3.615676e-01	-0.499976	29/35	500000

Tabela A.19: Resultados encontrados para a função G19. Melhor valor conhecido: 32.655592.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	41.795256	73.266757	115.609399	7.129728e+02	662.877241	35/35	5000
APM_Med_3	39.038034	82.014177	92.240476	3.650674e+02	387.067493	35/35	5000
APM_Worst	42.744810	81.895039	106.561910	6.943432e+02	656.855956	35/35	5000
APM_Spor_Mono	41.323695	68.131967	80.521406	2.477829e+02	262.952869	35/35	5000
APM	34.771183	49.734143	54.642489	9.390163e+01	105.465477	35/35	50000
APM_Med_3	36.325963	51.873621	58.502530	1.212532e+02	114.729288	35/35	50000
APM_Worst	35.744760	52.016033	53.842968	6.966717e+01	82.801023	35/35	50000
APM_Spor_Mono	35.991414	49.621793	58.824625	1.368188e+02	136.941592	35/35	50000
APM	33.161173	50.311549	53.590551	9.177767e+01	85.913573	35/35	500000
APM_Med_3	35.225365	50.913799	54.466448	8.968245e+01	92.615380	35/35	500000
APM_Worst	34.567271	43.193239	50.915809	8.210930e+01	78.208336	35/35	500000
APM_Spor_Mono	34.196632	44.998012	53.117473	1.310396e+02	143.552040	35/35	500000

Tabela A.20: Resultados encontrados para a função G23. Melhor valor conhecido: -400.055099.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	899.994375	899.994375	899.994375	0.000000e+00	899.994375	1/35	5000
APM_Med_3	899.984849	899.988760	899.988760	5.531428e-03	899.992671	2/35	5000
APM_Worst	-0.006894	449.988475	449.988475	6.363896e+02	899.983843	2/35	5000
APM_Spor_Mono	899.987042	899.987042	899.987042	0.000000e+00	899.987042	1/35	5000
APM	-0.003706	-0.003706	-0.003706	0.000000e+00	-0.003706	1/35	50000
APM_Med_3	899.945147	899.945682	899.945682	7.570638e-04	899.946218	2/35	50000
APM_Worst	-	-	-	-	-	0/35	50000
APM_Spor_Mono	-400.052390	-400.005555	-300.045743	1.309162e+02	-357.185021	7/35	50000
APM	899.915394	899.915394	899.915394	0.000000e+00	899.915394	1/35	500000
APM_Med_3	899.938757	899.938757	899.938757	0.000000e+00	899.938757	1/35	500000
APM_Worst	899.938279	899.938279	899.938279	0.000000e+00	899.938279	1/35	500000
APM_Spor_Mono	-400.052878	-400.046200	-375.046343	8.660976e+01	-300.040093	4/35	500000

Tabela A.21: Resultados encontrados para a função G24. Melhor valor conhecido: -5.508013.

Método	Melhor	Mediana	Média	dp	Pior	nesf	na
APM	-5.508010	-5.507996	-5.507994	5.798124e-05	-5.507969	35/35	5000
APM_Med.3	-5.508008	-5.507993	-5.507989	7.087429e-05	-5.507965	35/35	5000
APM_Worst	-5.508008	-5.507985	-5.507987	7.570944e-05	-5.507966	35/35	5000
APM_Spor_Mono	-5.508008	-5.507991	-5.507990	7.167815e-05	-5.507961	35/35	5000
APM	-5.508013	-5.508010	-5.508010	8.702509e-06	-5.508007	35/35	50000
APM_Med.3	-5.508013	-5.508010	-5.508010	9.810831e-06	-5.508004	35/35	50000
APM_Worst	-5.508013	-5.508011	-5.508011	7.624071e-06	-5.508007	35/35	50000
APM_Spor_Mono	-5.508013	-5.508011	-5.508010	9.031124e-06	-5.508006	35/35	50000
APM	-5.508013	-5.508013	-5.508012	2.621720e-06	-5.508012	35/35	500000
APM_Med.3	-5.508013	-5.508013	-5.508013	2.176201e-06	-5.508012	35/35	500000
APM_Worst	-5.508013	-5.508013	-5.508013	1.698174e-06	-5.508012	35/35	500000
APM_Spor_Mono	-5.508013	-5.508012	-5.508012	2.696939e-06	-5.508011	35/35	500000