

Tiago Gonçalves Goto

Propostas de Heurísticas e Estratégias de Feedback Aplicadas ao Recozimento Simulado

São Paulo

2023

Tiago Gonçalves Goto

Propostas de Heurísticas e Estratégias de Feedback Aplicadas ao Recozimento
Simulado

Versão Corrigida

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

Área de concentração Engenharia de Controle e Automação Mecânica

Orientador: Marcos de Sales Guerra Tsuzuki

São Paulo

2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

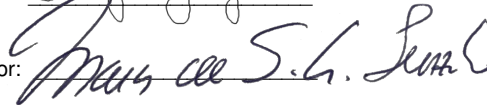
Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 02 de janeiro de 2023

Assinatura do autor:



Assinatura do orientador:



Catálogo-na-publicação

Goto, Tiago Gonçalves

Propostas de Heurísticas e Estratégias de Feedback Aplicadas ao Recozimento Simulado / T. G. Goto -- versão corr. -- São Paulo, 2023.
155 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo.
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Otimização Estocástica 2.Heurística 3.Pareto Otimalidade
I.Universidade de São Paulo. Escola Politécnica. Departamento de
Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

À minha esposa, Pamela Jordana de Menezes, com amor, admiração e gratidão por sua compreensão, carinho, presença e incansável apoio ao longo do período de elaboração deste trabalho.

Agradecimentos

Agradeço ao Professor Marcos de Sales Guerra Tsuzuki pelas orientações durante todo o desenvolvimento deste trabalho. Agradeço aos professores Thiago de Castro Martins e Ahmad Barari pelo apoio e contribuições. Aos Professores Larissa Driemeier e Rafael T. Moura pelas contribuições. Agradeço aos Colegas Hossein e Edson Ueda pela ajuda e contribuições. Agradeço aos meus colegas do Laboratório de Geometria Computacional, que me auxiliaram nesta jornada. Finalmente, agradeço à minha esposa, Pamela Jordana de Menezes pelo companheirismo e paciência e à minha família pelo apoio.

Resumo

Um projeto de engenharia deve contemplar uma série de requisitos técnicos, inclusive de segurança, para que o mesmo possa atingir o objetivo proposto. A otimização é uma ferramenta importante para o projetista no desenvolvimento de projetos que atenda os objetivos necessários. Dentre diversas técnicas de otimização, o Recozimento Simulado (SA, do inglês *Simulated Annealing*) tem se destacado devido suas características. Neste texto é abordado um estudo sobre heurística e estratégias de feedback para o SA, estas heurísticas foram estendidas para a otimização multiobjetivo com o algoritmo CoAnnealing, este baseado em SA. Também abordamos a aplicação do SA em Otimização Topológica (OT) mono-objetivo e multiobjetivo. O objetivo deste trabalho é realizar um estudo do desempenho do algoritmo CoAnnealing com a heurística de cristalização e estratégia de feedback por meio de funções de Benchmark encontradas na literatura. O desempenho foi analisado com o tempo de execução e a qualidade da frente de Pareto obtida. Estes resultados foram comparados com outro algoritmo baseado em SA conhecido como AMOSA. Além disso, aplicamos CoAnnealing com a heurística de cristalização em problemas mecânicos de otimização, como o dimensionamento de eixos de transmissão e dimensionamento de placa com orifício. Outro objetivo deste trabalho foi a aplicação do algoritmo SA em OT mono-objetivo e multiobjetivo, sendo necessário propor novas heurísticas e estratégias para solucionar a OT, tanto em mono-objetivo quanto multiobjetivo. Os resultados demonstraram a eficiência e bom desempenho do CoAnnealing com a heurística de cristalização em resolver diversos problemas de otimização. Nos problemas de OT, os resultados apresentaram distribuição de massa sem a presença de gradiente de cinza e a energia de deformação equivalente ao obtido na literatura.

Palavras-chave: Otimização Estocástica, Recozimento Simulado, Otimização multiobjetivo, Otimização Topológica.

Abstract

In the engineering design process, it is necessary to meet technical requirements, such as safety, to achieve goals. An optimization is a mathematical tool that assists in the development of a project to achieve goals. There are several optimization techniques, and the Simulated Annealing (SA) has stood out due to its characteristics. In this work, we study heuristic and feedback strategies for SA. The proposed heuristics are extended to SA multi-objective algorithm called CoAnnealing and it is based on the SA. This work aims to validate the CoAnnealing ability to solve several optimization problems. Performance is measured by execution time and Pareto front quality, and we compare the results with the AMOSA algorithm and results from the literature. In addition, we applied CoAnnealing with crystallization heuristics to optimize the mechanical transmission design. The second aim of this work was to investigate topology optimization using mono-objective and multi-objective simulated annealing algorithms. The results show that the CoAnnealing algorithm can solve different types of optimization problems and find optimized solutions efficiently. In topology optimization, the results show a mass distribution in the domain without a gray gradient, and the results were comparable to the results in the literature.

Keywords: Stochastic Optimization, Topology Optimization, Simulated Annealing, Multi-objective Optimization.

Lista de ilustrações

Figura 1 – Exemplo de melhor resultado para a função objetivo $f(x)$	24
Figura 2 – Diagrama esquemático do problema do caixeiro viajante, cada nó representa uma cidade, as arestas representam as rotas entre as cidades	25
Figura 3 – Fronteira de Pareto para problema de maximização e minimização . . .	33
Figura 4 – Conceito de dominância entre soluções	33
Figura 5 – Características para análises da frente de Pareto.	34
Figura 6 – O indicador <i>Hypervolume</i> em duas dimensões	36
Figura 7 – Tipos de otimização estrutural	36
Figura 8 – Otimização topológica em meio contínuo e discreta	37
Figura 9 – Padrão conhecido como tabuleiro de xadrez e o gradiente de cinza entre a transição entre o material sólido e vazio	40
Figura 10 – Tratamento térmico do recozimento	48
Figura 11 – A busca por solução e a escalada aceitando solução pior para encontrar a solução ótima global.	48
Figura 12 – Fase de exploração e refinamento durante o processo de otimização. . .	52
Figura 13 – Fluxograma do processo de feedback do algoritmo de otimização. Quando a nova solução gerada é aceita, o fator de cristalização da variável de decisão recebe o feedback positivo. Quando é rejeitada, a variável de decisão recebe o feedback negativo	53
Figura 14 – Comportamento do Fator de cristalização para diferentes estratégias de feedback	57
Figura 15 – A quantidade total de dominância entre duas soluções A e B, consiste na região do retângulo.	60
Figura 16 – Estado de Energia e intensidade de dominância utilizado no CoAnnealing	66
Figura 17 – Frente de Pareto da função ZDT1	75
Figura 18 – Variação do fator de cristalização em relação à temperatura, função ZDT1	76
Figura 19 – Frente de Pareto da função ZDT2	77
Figura 20 – Frente de Pareto da função ZDT3	78
Figura 21 – Frente de Pareto da função ZDT6	79
Figura 22 – Frente de Pareto da função DTLZ1	80
Figura 23 – Frente de Pareto da função DTLZ2	81
Figura 24 – Frente de Pareto da função DTLZ5	82
Figura 25 – Frente de Pareto da função DTLZ7	83
Figura 26 – Frente de Pareto da Função Binh and Korn de Binh e Korn (1997) . .	83
Figura 27 – Frente de Pareto da Função Constr-Ex	84

Figura 28 – Frente de Pareto da Função Chankong e Haimes	84
Figura 29 – Frente de Pareto da Função TNK com restrição	84
Figura 30 – Desenho esquemático da placa com orifício para maximização do fator de segurança	88
Figura 31 – Frente de Pareto obtida por meio da otimização com o CoAnnealing do problema de dimensionamento do furo	90
Figura 32 – Variação do fator de cristalização no processo de otimização da dimensão do furo	91
Figura 33 – Diagrama esquemático do sistema de transmissão, para a otimização do dimensionamento do eixo	92
Figura 34 – Desenho esquemático do eixo e os componentes de transmissão	92
Figura 35 – A frente de Pareto ótima obtida na otimização do eixo 1	95
Figura 36 – Comportamento do fator de cristalização do problema de otimização do eixo 1	95
Figura 37 – Frente de Pareto obtida com processo de otimização do eixo 2	98
Figura 38 – Variação do fator de cristalização no processo de otimização do eixo 2 .	99
Figura 39 – Diagrama esquemático de escolha de nova solução na vizinhança	101
Figura 40 – Problemas clássicos de vigas encontrados na literatura para teste de OT	104
Figura 41 – Resultados de OT do problema <i>Cantilever Beam</i>	105
Figura 42 – Resultados da OT pelo método SIMP do código Andreassen et al. (2011) para o problema <i>Cantilever Beam</i>	106
Figura 43 – Variação do fator de cristalização em relação à temperatura para a OT mono-objetivo de viga engastada	107
Figura 44 – Resultados de OT do problema <i>Half-MBB Beam</i>	107
Figura 45 – Resultados da OT pelo método SIMP do código Andreassen et al. (2011) para problema <i>Half-MBB beam</i>	108
Figura 46 – Resultados da distribuição de massa com a variação do r para o problema <i>Cantilever Beam</i>	109
Figura 47 – Viga engastada do tipo <i>Cantilever Beam</i> com um orifício circular . . .	109
Figura 48 – Resultados do processo de OT na viga <i>Cantilever Beam</i> com um orifício	110
Figura 49 – Resultados obtidos com o código top88.m de Andreassen et al. (2011) para a viga engastada com furo.	111
Figura 50 – Problema de OT em uma viga <i>Cantilever Beam</i> com aplicação da força no centro da ponta	112
Figura 51 – Resultados da distribuição de massa da otimização TO do algoritmo proposto em uma viga engastada com a aplicação no centro da ponta .	112
Figura 52 – Resultados obtidos com o código top88.m de Andreassen et al. (2011) para a viga engastada com furo.	113
Figura 53 – Problema de otimização de viga bi-apoiada com força no centro da barra	114

Figura 54 – Resultados da distribuição de massa do processo de OT com o algoritmo proposto para a viga bi-apoiada	115
Figura 55 – Resultados da distribuição de massa pelo algoritmo top88 de Andreassen et al. (2011) para o problema de viga bi-apoiada	116
Figura 56 – Otimização TO da viga bi-apoiada, com os parâmetros de $FV = 0,4$, $N = 2000$, $T_{\max} = 200$, $T_{\min} = 1,0 \times 10^{-7}$ e $\alpha = 0,85$	117
Figura 57 – Variação da região de seleção das variáveis de decisões na OT com SA .	119
Figura 58 – Resultados do problema <i>Cantilever beam</i> com algoritmo com variação do raio	121
Figura 59 – Resultados do problema <i>Half MBB beam</i> com algoritmo com variação do raio	122
Figura 60 – Os valores do fator R_k no final do processo de otimização	123
Figura 61 – Distribuição de massa obtida no processo de OT da viga engastada com orifício	124
Figura 62 – A distribuição de massa obtida no processo de OT para a viga engastada com a força aplicada no centro da extremidade da viga.	125
Figura 63 – Resultados da distribuição de massa no domínio do processo de OT . .	126
Figura 64 – Otimização TO anterior com a mesma malha, e os parâmetros de $FV = 0,5$, $N = 5000$, $T_{\max} = 200$, $T_{\min} = 1,0 \times 10^{-9}$ e $\alpha = 0,85$	128
Figura 65 – Frente de Pareto resultante da OT com algumas das soluções da frente de Pareto com seus respectivos valores de Fração volumétrica e a energia de deformação	134
Figura 66 – Soluções otimizadas obtidas no processo de OT com filtro	135
Figura 67 – Frente de Pareto obtida no processo de OT com seus respectivos valores de Fração volumétrica e a energia de deformação	136
Figura 68 – Solução antes do pós-processamento e após o pós-processamento. . . .	136
Figura 69 – O esquema de troca de densidades dos elementos utilizados na OT multiobjetivo	137
Figura 70 – Diagrama esquemática de Viga engastada com carregamento de duas forças	139
Figura 71 – Frente de Pareto obtida no processo de OT multiobjetivo	140
Figura 72 – Soluções da frente de Pareto do processo de OT multiobjetivo	140
Figura 73 – Problema de OT em viga engastada com três carregamentos	141
Figura 74 – Frente de Pareto do processo de OT da viga engastada com carregamento de 3 forças	142
Figura 75 – Soluções otimizadas na frente de Pareto da OT da viga engastada. . . .	142

Lista de tabelas

Tabela 1 – Tabela com os principais algoritmos de otimização de acordo com a categoria	26
Tabela 2 – Estratégias da heurística de cristalização.	56
Tabela 3 – As funções de <i>benchmark</i> utilizadas para validação do desempenho dos algoritmos.	73
Tabela 4 – As funções de benchmark com restrição utilizadas para validação do desempenho dos algoritmos	74
Tabela 5 – Tabela com os parâmetros de otimização utilizados	74
Tabela 6 – Resultados dos indicadores de desempenho do AMOSA e CoAnnealing com as funções benchmark.	85
Tabela 7 – Tempo de execução do algoritmo AMOSA e CoAnnealing com as funções de benchmark.	86
Tabela 8 – Soluções ótimas obtidas no processo de otimização do problema da placa de orifício circular	90
Tabela 9 – Tabela de dados do problema de otimização do eixo 1	94
Tabela 10 – Variáveis e valores mínimos e máximos do problema de otimização eixo 1	94
Tabela 11 – Soluções ótimas presentes na frente de Pareto do problema de otimização do eixo 1	94
Tabela 12 – Tabela de dados do problema de otimização do eixo 2	96
Tabela 13 – Variáveis e valores mínimos e máximos do problema de otimização do eixo 2	98
Tabela 14 – Soluções ótimas presentes na frente de Pareto do problema de otimização do eixo 2	98
Tabela 15 – Energia de deformação obtida na otimização topológica pelo algoritmo proposto	108
Tabela 16 – Energia de deformação obtida no processo de OT da viga engastada com orifício	111
Tabela 17 – Valores da energia de deformação resultante do processo de OT da viga engastada com força aplicada no centro da extremidade da viga, os valores de menores são melhores	114
Tabela 18 – Valores da energia da deformação do processo de OT da viga bi-apoiada	115
Tabela 19 – Resultados dos valores da energia de deformação das vigas.	121
Tabela 20 – Valores da energia de deformação obtida no processo de OT da viga engastada com orifício.	123
Tabela 21 – Valores de energia de deformação obtidos no processo TO para a viga engastada e carregamento no centro da ponta	125

Tabela 22 – Energia de deformação obtida no processo de OT para o problema de viga bi-apoiada	127
---	-----

Lista de abreviaturas e siglas

ACO	Ant Colony Optimization
ADD	Approximate Density Distribution
AESO	Additive Evolutionary Structural Optimization
AMOSAS	Archived Multi-objective Simulated Annealing
ASME	The American Society of Mechanical Engineers
BBTO	Black-box Topology Optimization
BESO	Bi-directional Evolutionary Structural Optimization
CONLIN	CONvex LINearization
ESO	Evolutionary Structural Optimization
FE	Finite Element
FEA	Finite Element Analysis
FVM	finite volume method
GA	Genetic Algorithm
GD	General Distance metrics
GRASP	Greedy Randomized Adaptive Search Procedure
GTO	Gradient-based Topology Optimization
HBTO	Homogenization Based Topology Optimization
HV	Hypervolume
MBB	Messerschmitt–Bolkow–Blohm
MMA	Method of Moving Asymptotes
MOEA	Multiobjective Evolutionary Algorithm
MOSA	Multi-objective Simulated Annealing
NISE	Noninferior Set Estimation

NSGA-II	Non-dominated Sorted Genetic Algorithm II
OCM	Optimality Criteria Methods
PAES	Pareto Archived Evolution Strategy
PSA	Pareto Simulated Annealing
PSO	Particle Swarm Optimization
OT	Otimização Topológica
SA	Simulated Annealing
SIMP	Solid Isotropic Material with Penalization method
SLP	Sequential Linear Programming
SPEA	Strength Pareto Evolutionary Algorithm
SQP	Sequential Quadratic Programming
TB	Tabu Search

Lista de símbolos

L	Função lagrangiana
x	Váriável de decisão
λ	multiplicador de lagrange
μ	Escalar
$f(x)$	Função objetivo em função de x
$g(x)$	Função de restrição em função de x
$C(x)$	Função objetivo em função de x ou Energia de deformação em otimização topológica
$V(x)$	Fração volumétrica em função de x
m	Limite móvel positivo no método de Critério de Otimilidade (OCM)
B_e	Fator escala no método de Critério de Otimilidade (OCM)
η	Fator de amortecimento método de Critério de Otimilidade (OCM)
U_j	Parâmetros do Método das Assíntotas Móveis (MMA)
L_j	Parâmetros do Método das Assíntotas Móveis (MMA)
X	Variável de decisão na otimização
NPF	Quantidade de soluções no conjunto de soluções ótimas
R	Conjunto de soluções ótimas
R^*	Conjunto de soluções ótimas de referência
d	Distância euclidiana
S	Índice <i>Spacing</i>
\bar{d}	Valor médio da distância euclidiana
M	Número de funções objetivo no problema de otimização
HV	Índice hipervolume
x_e	Densidade fictícia do elemento

E	Módulo de elasticidade
ke	Matriz rigidez do elemento
p	Fator de penalidade na otimização topológica
U	Vetor deslocamento
F	Vetor de força na análise de elementos finitos
U_e	Vetor deslocamento do elemento
σ_e	Tensão de Mises no método BESO
RR_i	Razão de rejeição no método ESO
ER	Razão de evolução
IR	Razão de adição no método BESO
P	Fator de probabilidade de Boltzman
ΔE	Variação de estado de energia no Recozimento Simulado
T	Temperatura no Recozimento simulado
k_b	Constante de Boltzman
Δr	Passo fixo no Recozimento Simulado
u	Vetor direção no Recozimneto Simulado
N	Número de iterações no algoritmo SA
α	Fator de decaimento de temperatura
C_k	Fator de cristalização do variável de decisão k
e_k	Vetor direção no eixo k no Recozimento Simulado
C_{\max}	Fator de cristalização máximo
HL	Limite de soluções <i>hard limit</i>
SL	Limite de soluções <i>Soft Limit</i>
$\Delta dom_{a,b}$	Intensidade de dominância entre a solução a e b
R_i	Diferença entre o maior e menor valor da função objetivo i pertecente ao <i>Archive</i>

$temp$	Temperatura no algoritmo AMOSA
$Archive$	Vetor que armazena as soluções não-dominadas
max_r	Quantidade máxima aceitável de remoção da solução atual do $Archive$ no processo de clusterização
r	Contador de quantidade de vezes que a solução corrente é removido do $Archive$ no processo de clusterização
σ	Tensões normal
A	Área de uma seção transversal
K	Fator de concentração de tensões
FS	Fator de Segurança
d	Diâmetro do furo
S_y	Resistência do escoamento
ρ	Densidade [$\frac{kg}{m^3}$]
y, y_{max}	Deflexão e deflexão máxima
y_{xz}, y_{yz}	Deflexão no plano xz e deflexão no plano yz
α	Fator de contato em engrenagem de dente reto
L	Comprimento da seção de um eixo
d	Diâmetro
N_f	Coeficiente de Segurança de Fadiga
K_f, K_{fm}	Fatores de concentração de tensão de fadiga por flexão para componentes media e alternantes
K_{fs}, K_{fsm}	Fatores de concentração torcional de fadiga para componentes media e alternantes
M_m, M_a	Componente media e alternante do momento fletor
T_m, T_a	Componente media e alternante do torque
S_f	Resistência à fadiga
S_{ut}	Resistência ultima a tração

ω_n	Frequência natural de um eixo
ω_t	Frequência de trabalho de um eixo
r	Raio da região circular na otimização topológica
FV	Fração Volumétrica
$nelx$	Número de elementos na direção x na malha
$nely$	Número de elementos na direção y na malha
r_{\min}	Raio utilizado no filtro de sensibilidade ou filtro de densidade
$penal$	Fator de penalidade
ft	Parâmetro para indicar a utilização do filtro de sensibilidade ou filtro de densidade
J	Energia de deformação na otimização topológica
R_k	Fator de raio
R_{\max}	Valor máximo do fator Raio R_k
Ω	Domínio de busca na otimização
ρ	Pseudo densidade na otimização topológica
N_{viz}	Número de elementos na vizinhança do elemento

Sumário

1	INTRODUÇÃO	20
1.1	Heurísticas e Estratégias	21
1.2	Objetivo	22
1.3	Estrutura da Tese	23
2	REVISÃO BIBLIOGRÁFICA	24
2.1	Algoritmos de Otimização	24
2.1.1	Algoritmos Determinísticos	26
2.1.2	Algoritmos Probabilísticos	29
2.2	Otimização Multiobjetivo	31
2.2.1	Conjunto de Pareto - Frente de Pareto	32
2.2.2	Índice de performance da Frente de Pareto	33
2.3	Otimização Topológica	35
2.3.1	O Método SIMP	38
2.3.2	Método ESO	41
2.3.3	Método BESO	42
2.3.4	Otimização Topológica e Algoritmos Não Gradiente	44
2.4	Considerações Finais	46
3	RECOZIMENTO SIMULADO	47
3.1	Funcionamento do SA	47
3.2	Fase de Exploração e de Refinamento	51
3.3	Principais Parâmetros	52
3.4	Heurística de Cristalização	52
3.5	Estratégias de Feedback	54
3.6	Considerações Finais	57
4	RECOZIMENTO SIMULADO MULTIOBJETIVO	58
4.1	Introdução a SA Multiobjetivo	58
4.2	AMOSAS	59
4.3	CoAnnealing	65
4.4	CoAnnealing com Heurística de Cristalização	69
4.5	CoAnnealing com Estratégia de Feedback	70
4.6	Solucionando as Funções de Benchmark	72
4.7	Considerações Finais	87

5	OTIMIZAÇÃO MULTIOBJETIVO EM PROBLEMAS MECÂNICOS	88
5.1	Placa com Orifício Circular	88
5.2	Problema de Otimização de Eixo 1	91
5.3	Problema de Otimização de Eixo 2	96
5.4	Considerações Finais	99
6	OTIMIZAÇÃO TOPOLÓGICA COM RECOZIMENTO SIMULADO	100
6.1	OT com SA e Heurística de Cristalização	100
6.2	Resultados do SA com a Heurística de Cristalização	104
6.3	OT Mono-Objetivo com Heurística de Raio Variável	118
6.4	Resultados do OT Mono-Objetivo com Heurística de Raio Variável	119
6.5	OT Multiobjetivo	129
6.6	OT Multiobjetivo com Heurística de Raio Variável	130
6.7	Resultados do OT Multiobjetivo Aplicado a Vigas	132
6.8	OT Multiobjetivo com Heurística de Troca de Densidade	137
6.9	OT Multiobjetivo com o Índice de Continuidade e a Heurística de Troca de Densidade	137
6.10	Considerações Finais	143
7	CONCLUSÕES	145
7.1	Trabalhos Futuros	146
	REFERÊNCIAS	147

1 Introdução

A otimização é uma ferramenta matemática para determinar a melhor combinação de argumentos para maximizar ou minimizar um objetivo ou mais objetivos, sendo este um assunto relevante na engenharia. Ao longo dos anos, diversos métodos e algoritmos de otimização foram propostos, não existindo um método que se adeque a todos os tipos de problemas, cada método possui vantagem em algum tipo específico de problema conforme o teorema *No Free Lunch Theorem for Search* (WOLPERT; MACREADY et al., 1995).

Os algoritmos de otimização podem ser classificados em determinísticos e não determinísticos (ou probabilísticos). Os algoritmos clássicos em sua maioria são determinísticos. Nestes algoritmos a função objetivo deve ser diferenciável, ou seja, o algoritmo é dependente das informações do gradiente que em alguns problemas são complexo de ser obtidas. Já os algoritmos probabilísticos não dependem de informações do gradiente, sendo baseados apenas na avaliação da função objetivo (HOLTZ, 2005). Alguns exemplos de algoritmos não determinísticos são: algoritmos genéticos, colônia de formigas, recozimento simulado (SA, do inglês *Simulated Annealing*), dentre outros. O SA foi selecionado por apresentar várias características favoráveis.

O SA é conhecido por obter bons resultados, mas como citado por Suman e Kumar (2006) não necessariamente o resultado será a solução ótima. Porém, o SA é o único algoritmo meta-heurístico que converge para o ótimo global em problemas combinatórios (GRANVILLE; KRIVANEK; RASSON, 1994). É possível obter a solução ótima global utilizando outras técnicas, porém, o custo de processamento é pode ser inviável. E o SA também é conhecido por resolver com sucesso os problemas contínuos (GASPARCUNHA; TAKAHASHI; ANTUNES, 2012). Sendo assim, o SA possui a característica de conseguir resolver problemas com variáveis contínua, discreta e inteira. Outra característica desse algoritmo é a capacidade de evitar solução ótima local, aceitando soluções piores (RUTENBAR, 1989). Esta característica está relacionada ao processo de otimização que possui duas fases: exploratória e refinamento.

Além disso, como descrito nos trabalhos de Suman e Kumar (2006), Amine (2019) o SA é um algoritmo que possui adaptabilidade, conseqüentemente surgiu diversas adaptações para melhorar o desempenho do SA nos últimos anos. Essa flexibilidade do SA também permitiu surgir propostas de algoritmos de otimização multiobjetivo. Dentre os principais algoritmos multiobjetivo baseado em SA, o *Archived Multiobjective Simulated Annealing* (AMOS) apresenta um ótimo desempenho em relação a outros algoritmos evolutivos, porém é um algoritmo complexo e possui algumas desvantagens que serão discutidas no Capítulo 4. Baseado no AMOS, foi proposto por Martins e Tsuzuki (2015) o *CoAnnealing*

que apresenta critérios de decisões mais simples. Nesta tese incluiremos heurísticas e estratégias de feedback para o *CoAnnealing*, em seguida serão apresentados estudos de desempenho com funções de *benchmark*.

Utilizando um sistema de busca de periódico, o Science Direct, verificamos que há aproximadamente 98 mil publicações relacionadas ao SA e 2,1 mil publicações relacionadas a SA multiobjetivo no período de 2000 a 2022. Esses dados demonstram que é um assunto relevante. Muitas das publicações estão relacionadas aos algoritmos híbridos entre o SA e outros algoritmos evolutivos como o algoritmo de busca de tabu. Em relação à otimização multiobjetivo, muitos trabalhos são baseados em técnicas de escalarização (COHON; CHURCH; SHEER, 1979; HAIMES YV; LASDON; WISMER DA, 1971; ZADEH, 1963). Portanto, o estudo do desempenho do algoritmo multiobjetivo e o desenvolvimento de melhorias no algoritmo por meio de heurísticas e estratégias são contribuições relevante ao tema.

Ainda, ao observar em sistema de busca de periódicos, é possível observar que nos últimos anos tem crescido a quantidade de publicações relacionadas à otimização topológica (OT). Principalmente à OT multiobjetivo e a OT baseada em algoritmo meta-heurísticos, demonstrando a relevância do assunto. As OT tradicionais utilizam algoritmos determinísticos que dependem de informações do gradiente, estes são conhecidos como *Gradient-based Topology Optimization* (GTO). Como descrito por Bureerat e Limtragool (2008), em alguns problemas de OT o gradiente pode ser difícil de obter ou não estar disponível. Além disso, problemas de OT multiobjetivo só podem ser resolvidos por meio de técnicas de escalarização, quando utilizado os algoritmos GTO.

Em contrapartida, os algoritmos não determinísticos não precisam do gradiente, porém são mais lentos (GARCIA-LOPEZ et al., 2011; BUREERAT; LIMTRAGOOL, 2008). Além disso, geralmente estes algoritmos necessitam de heurísticas ou estratégias para a resolução de problemas de OT mono-objetivo. Existem algumas propostas de OT multiobjetivo, porém existem poucos trabalhos a respeito desta (GUIRGUIS et al., 2020). Quando se restringe ao SA, são encontrados poucos trabalhos de OT mono-objetivo e multiobjetivo.

Apesar de ser um assunto relevante, observamos que a otimização multiobjetivo ou a OT com SA é pouco explorada. Ainda, devido à flexibilidade do SA, há possibilidade de inserir sensibilidade por meio de heurística ou estratégias. Para cada tipo de problema deve ser proposta uma heurística diferente, podendo ser complexo.

1.1 Heurísticas e Estratégias

A heurística é definida por Gaspar-Cunha, Takahashi e Antunes (2012) como “um procedimento de busca de “boa solução” para o problema que não é dotado de garantias

quanto à qualidade da solução encontrada em comparação com o ótimo exato”. Como descrevemos, o algoritmo SA é um algoritmo flexível, permitindo adaptações conforme o problema de otimização.

O algoritmo SA é conhecido por ter duas fases no processo de otimização, a fase de exploração e a de refinamento. Estas fases tem como objetivo permitir uma maior exploração para evitar mínimos locais e a fase de refinamento para a busca do mínimo global. Devido a sua flexibilidade é possível propor heurísticas e estratégias de controle das fases de exploração e refinamento. Uma das propostas é inserção de estratégias de feedback para controlar a exploração ou refinamento dependendo da função objetivo. Essas características não são encontradas em outros algoritmos de otimização probabilística.

1.2 Objetivo

Esta pesquisa tem como objetivo geral estudar a otimização multiobjetivo baseada no SA, utilizando os algoritmos AMOSA e CoAnnealing, e propor estratégias de feedback no algoritmo de otimização multiobjetivo CoAnnealing, e comparar o desempenho do mesmos. Além disso, temos como objetivo estudar a capacidade dos algoritmos multiobjetivo em aplicações mecânicas. Uma das aplicações a serem implementadas é a otimização de problemas mecânicos e a OT estrutural. Sendo necessário realizar estudos e propor estratégias e heurísticas para o SA multiobjetivo resolver problemas de OT.

Para atingir tal objetivo, foi necessário desdobrá-lo em objetivos específicos para melhor desenvolver o estudo aqui proposto, conforme exposto a seguir:

- realizar um estudo dos algoritmos AMOSA e CoAnnealing com relação à literatura específica;
- realizar um estudo sobre OT, inclusive multiobjetivo, com relação à literatura específica;
- realizar um levantamento sobre heurísticas e estratégias para algoritmo SA;
- implementar a otimização multiobjetivo, usando os algoritmos AMOSA e CoAnnealing em funções de *benchmark* da literatura;
- realizar um estudo comparativo do desempenho dos algoritmos AMOSA e CoAnnealing;
- aplicar o algoritmo CoAnnealing em aplicações de otimização mecânica;
- implementar o algoritmo de SA multiobjetivo em OT;
- implementar OT aplicada em casos conhecidos na literatura como o problema *Cantilever beam* e *Half MBB-Beam*.

1.3 Estrutura da Tese

A estrutura da Tese está organizada em 7 capítulos, incluído este capítulo introdutório e a conclusão do trabalho. No Capítulo 2, é apresentada uma revisão bibliográfica relacionada a algoritmos de otimização, otimização multiobjetivo e OT.

O capítulo 3, é dedicado ao algoritmo de SA e serão discutidas as principais propostas de heurísticas para melhoria do desempenho. No Capítulo 4, apresentamos os dois principais algoritmos de otimização multiobjetivo baseados em SA. Além disso, apresentamos os resultados da otimização multiobjetivo aplicado em funções de *benchmark*.

No Capítulo 5, apresenta-se a aplicação do CoAnnealing em problemas de otimização mecânica. Os casos são de dimensionamento de furo e eixo de sistema de transmissão. No Capítulo 6, apresentamos a OT mono-objetivo e a multiobjetivo baseada no SA com a heurística proposta. Finalmente, no Capítulo 7 é discutida a conclusão do trabalho e sugestões para trabalhos futuros.

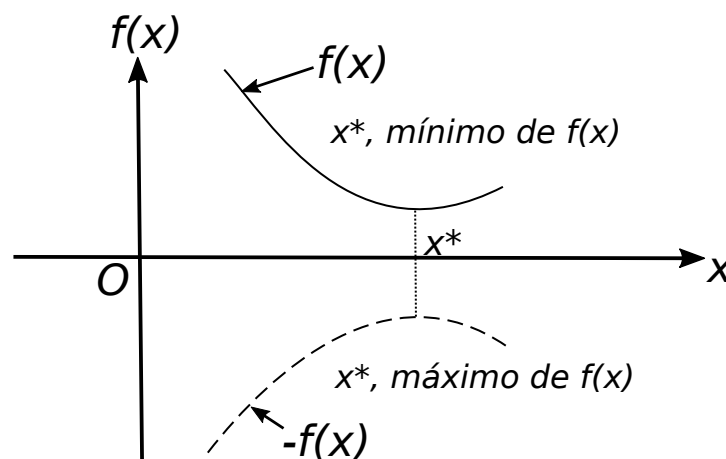
2 Revisão Bibliográfica

Neste capítulo será apresentada uma revisão bibliográfica sobre técnicas de otimização, otimização multiobjetivo e otimização topológica. Em relação aos algoritmos de otimização, será tratado tanto os algoritmos determinísticos como os algoritmos probabilísticos, que é o foco deste trabalho. Os algoritmos determinísticos, descritos na Subseção 2.1.1, são amplamente utilizados na OT. Enquanto o algoritmo utilizado neste trabalho faz parte dos algoritmos probabilísticos descritos na Subseção 2.1.2. Na Seção 2.3 será abordada uma revisão bibliográfica sobre a OT e as principais técnicas encontradas na literatura.

2.1 Algoritmos de Otimização

A otimização é uma técnica matemática para encontrar um valor máximo e mínimo de uma função (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). Autores como Rao (2009), definiram a otimização como a arte de obter o melhor resultado de uma determinada circunstância. Um exemplo de otimização mono-objetivo de uma função objetivo $f(x)$, onde x é um vetor de variável de decisão de n dimensões, é apresentado na Fig. 1.

Figura 1 – Exemplo de melhor resultado para a função objetivo $f(x)$

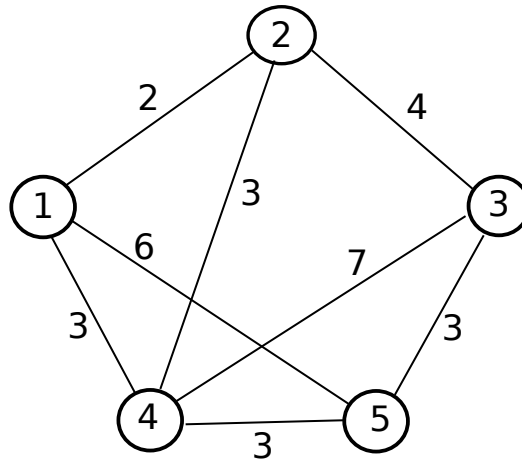


Fonte: Adaptado de (RAO, 2009).

A otimização possui uma vasta aplicação em diversas áreas como economia, física, química e biologia e, obviamente, é utilizada em larga escala em problemas de engenharia. Um problema de otimização clássico é o caixeiro viajante, este problema consiste na procura de um caminho para o vendedor passar por todas as cidades e regressar a cidade

de partida percorrendo a menor distância. O problema do caixeiro viajante é um problema de otimização combinatória, pode ser representado na Fig. 2.

Figura 2 – Diagrama esquemático do problema do caixeiro viajante, cada nó representa uma cidade, as arestas representam as rotas entre as cidades



Fonte: Adaptado de Goyal (2010).

Existem diversos métodos de otimização, podendo ser classificados em programação matemática, métodos estatísticos, métodos de processos estocásticos e métodos modernos. Na Tabela 1, são apresentados alguns métodos de otimização de acordo com sua classificação, esta tabela foi adaptada de Rao (2009). Não existe um método ou algoritmo de otimização que consiga resolver todos os problemas, portanto são necessários diferentes algoritmos e métodos de otimização.

A otimização também pode ser classificada em otimização mono-objetivo ou multiobjetivo. Na primeira, somente uma função objetivo é minimizada ou maximizada, e encontramos uma solução ótima global. Em otimização multiobjetivo, são minimizadas ou maximizadas duas, ou mais funções. Neste processo, diferente do primeiro, não encontramos apenas uma solução ótima e sim um conjunto de soluções ótimas.

Na categoria de otimização por técnica de programação matemática que são métodos determinísticos, destacamos os métodos Critério de Otimalidade (OCM, do inglês *Optimality Criteria Methods*) e Método das Assíntotas Móveis (MMA, do inglês *Method of Moving Asymptotes*). Estes métodos são amplamente utilizados em Otimização Topológica (OT) sendo conhecidos como métodos baseados em gradiente. Portanto, abordaremos o método OC e o método MMA em Subseção 2.1.1.

Dentre os métodos modernos ou métodos probabilísticos, muitos são bio-inspirados, ou seja, os algoritmos foram inspirados em fenômenos da natureza para a solução de otimização. Como exemplo podemos citar os algoritmos genéticos e a colônia de abelhas. Também temos os não bio-inspirados como o SA e a busca de tabu discutido na

Tabela 1 – Tabela com os principais algoritmos de otimização de acordo com a categoria

Técnicas de Programação Matemática ou técnicas de otimização	Técnicas de Processo Estocásticos
Métodos de Cálculo	Teoria de Decisão Estatísticas
Programação Não Linear	Processo de Markov
Calculo da Variação	Teoria da Filas
Programação Geométrica	Teoria da Renovação
Programação Quadrática	Método Simulado
Programação Linear	Teoria da Confiabilidade
Programação Dinâmica	
Programação Inteiro	
Programação Estocástica	
Programação por Separação	
Programação Multiobjetivo	
Teoria dos Jogos	
Técnicas modernas de Otimização ou não Tradicionais	Técnicas Estatísticos
Algoritmos Genéticos	Análises de Regressão
Recozimento Simulado	Análise de Cluster e Reconhecimento de Padrão
Colônia de Formigas	
Otimização por Partícula de Swarm	
Colônia de Abelha	
Rede Neurais	
Otimização por Lógica Fuzzy	

Fonte: Adaptado de Rao (2009)

Subseção 2.1.2, o SA é o algoritmo estudado neste trabalho e será discutido no Capítulo 3.

2.1.1 Algoritmos Determinísticos

A maioria dos métodos clássicos de otimização fazem parte dos algoritmos determinísticos, ou programação matemática como é conhecido. Esses métodos utilizam as informações da derivada das funções objetivo em relação às variáveis de projetos, essa derivação é o gradiente. O algoritmo utiliza essas informações para a busca da solução ótima, não necessariamente a solução ótima global.

Para os algoritmos determinísticos, as funções objetivo devem ser representadas por funções matemáticas e devem ser contínuas e diferenciáveis (HOLTZ, 2005). A principal vantagem dos algoritmos determinísticos é a necessidade de baixa quantidade de avaliações da função objetivo para convergir. Entretanto, o algoritmo possui a desvantagem de ser intrinsecamente mono-objetivo (CAVAZZUTI, 2013).

Dentre os algoritmos de otimização determinísticos, destacamos o método de

Critério de Otimalidade (OCM) e o Método das Assíntotas Móveis (MMA). Esses métodos são amplamente utilizados em algoritmos de OT tradicionais. Portanto, apresentaremos de forma resumida esses algoritmos nesta seção.

A técnica de otimização Critério de Otimalidade é baseada na derivação de um critério para uma condição específica de projeto e o desenvolvimento iterativo para alcançar a solução ótima (RAO, 2009). Segundo SALVADOR (2018) a otimização está relacionada a obtenção do resultado satisfazendo as condições de Karush-Kuhn-Tucker descrito em

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) + \nabla g(x)\lambda + \nabla h(x)\mu = 0, \quad (2.1)$$

onde x é variável de decisão, L é a função lagrangiana, $f(x)$ é a função objetivo em função da variável de decisão x , $g(x)$ é a função de restrição em função de x , λ é o multiplicador de Lagrange e μ é um escalar.

Inicialmente, a técnica foi apresentada para a solução de estruturas elásticas com tensões com restrições de deslocamento e tensões. O algoritmo foi originalmente proposto por Prager (1968 apud RAO, 2009) e melhorado por outros autores.

Esta técnica é simples e usada com frequência em problemas de OT. Além disso, é uma técnica baseada na função Lagrangiana. Segundo Simonetti et al. (2021), para a otimização topológica, a função Lagrangiana L é dada por

$$L = C(x) + \lambda(V(x) - V^*) + \lambda_1^T(K_u - F) + \sum_{i=1}^n \lambda_2^i(x_i^{\min} - x_i) + \sum_{i=1}^n \lambda_3^i(x_i - x_i^{\max}), \quad (2.2)$$

onde λ são os multiplicadores lagrangianos, $V(x)$ é a fração volumétrica (função restrição do problema) em função de x , V^* é o volume prescrito e $C(x)$ é a função objetivo.

A solução ótima é obtida quando a derivada do Lagrangiano L em relação às variáveis de decisão são iguais a zero, conforme

$$\frac{\partial L}{\partial x_i} = 0. \quad (2.3)$$

A partir das condições ótimas obtemos o fator de escala B conforme

$$B_k^e = \frac{-\frac{\partial C}{\partial x^e}}{\lambda \frac{\partial V}{\partial x^e}}, \quad (2.4)$$

onde o numerador é a derivada parcial da da função objetivo em relação as variáveis de decisão e o denominador é a derivada parcial da função restrição em relação ás variáveis de decisão.

E a nova solução é encontrada com

$$x_e^{k+1} = B_k^e x_e^k. \quad (2.5)$$

Com objetivo de evitar uma modificação grande em uma única iteração, é estabelecido um limite móvel positivo m conforme descrito por Simonetti et al. (2021). O esquema de atualização da OC é determinado por

$$x_e^{k+1} = \begin{cases} \max(x_{\min}, (x_e^k - m)) & \text{se } x_e^k (B_k^e)^\eta \leq \max(x_{\min}, x_e^k - m) \\ x_e^k (B_k^e)^\eta & \text{se } \max(x_{\min}, x_e^k - m) \leq x_e^k (B_k^e)^\eta < \min(1, x_e^k + m), \\ \min(1, x_e^k + m) & \text{se } x_e^k (B_k^e)^\eta \geq \min(1, x_e^k + m) \end{cases}, \quad (2.6)$$

onde o η é um fator de amortecimento e é geralmente adotado o valor de $1/2$. Segundo Simonetti et al. (2021), o valor do multiplicador de Lagrange é atualizado iterativamente por algoritmo de bissecção. O critério de parada deste algoritmo ocorre por número máximo de iterações.

O MMA também é um algoritmo de programação matemática, e também é utilizado amplamente em problemas de OT, é o método adequado para problemas com várias restrições e variáveis de decisões. Este método foi desenvolvido por Svanberg (1987) e este algoritmo pode ser considerado uma generalização do método por aproximação linear convexa *CONvex LINearization* (CONLIN) (SALVADOR, 2018).

Segundo SALVADOR (2018), o diferencial do algoritmo MMA são os intervalos adotados para as funções que podem ser modificadas ou movidas a cada iteração. Esse intervalo também é conhecido como assíntotas móveis, e este comportamento possui como objetivo acelerar a convergência e a estabilidade (SALVADOR, 2018). No MMA, cada função de aproximação $f_i^k(x)$ é obtida com a linearização em variáveis do tipo $1/(U_j - x_j)$ ou $1/(x_j - L_j)$. Onde L_j e U_j são parâmetros que satisfaça a condição $L_j < x_j^{(k)} < U_j$ e a função de aproximação é dada por

$$f_i^k(x) = r_i^k + \sum_{j=1}^n \left(\frac{p_{ij}^k}{U_j^{(k)} - x_j} + \frac{q_{ij}^k}{x_j - L_j^{(k)}} \right), \quad (2.7)$$

onde os parâmetros p , q e r são determinados por

$$p_{ij}^k = \begin{cases} (U_j^k - x_j^k)^2 \frac{\partial f_i}{\partial x_j} & \text{se } \frac{\partial f_i}{\partial x_j} > 0 \\ 0 & \text{se } \frac{\partial f_i}{\partial x_j} \leq 0 \end{cases}, \quad (2.8)$$

$$q_{ij}^k = \begin{cases} 0, & \text{se } \frac{\partial f_i}{\partial x_j} \geq 0 \\ -(x_j^k - L_j^k)^2 \frac{\partial f_i}{\partial x_j}, & \text{se } \frac{\partial f_i}{\partial x_j} < 0 \end{cases}, \quad (2.9)$$

$$r_i^k = f_i(x^k) - \sum_{j=1}^n \left(\frac{p_{ij}^k}{U_j^k - x_j} + \frac{q_{ij}^k}{x_j - L_j^k} \right). \quad (2.10)$$

Os valores dos parâmetros L_j e U_j são atualizados a cada nova iteração. A regra padrão de atualização dos valores varia de acordo com o valor de k . Para às duas primeiras iterações onde $k = 1$ e $k = 2$, temos

$$L_j^k = x_j^k - C_0(x_j - x_j), \quad (2.11)$$

$$U_j^k = x_j^k + C_0(x_j - x_j), \quad (2.12)$$

onde C_0 é uma constante, usualmente o valor atribuído é $C_0 = 0,5$. Para iterações $k \geq +3$, se o termo $(x_j^k - x_j^{k-1}) \times (x_j^{k-1} - x_j^{k-2}) \leq 0$, a assíntota é movida próxima entre eles com (SVANBERG, 1993)

$$L_j^k = x_j^{(k)} - C_1(x_j^{(k-1)} - L_j^{(k-1)}), \quad (2.13)$$

$$U_j^k = x_j^{(k)} + C_1(U_j^{(k-1)} - x_j^{(k-1)}), \quad (2.14)$$

onde C_1 é constante e valores usuais são menores que 1. Para o caso de $k \geq 3$ e satisfaça as condições $(x_j^k - x_j^{k-1}) \times (x_j^{k-1} - x_j^{k-2}) > 0$, a assíntota é movida para longe entre eles com

$$L_j^k = x_j^{(k)} - C_2(x_j^{(k-1)} - L_j^{(k-1)}), \quad (2.15)$$

$$U_j^k = x_j^{(k)} + C_2(U_j^{(k-1)} - x_j^{(k-1)}), \quad (2.16)$$

onde C_2 é uma constante com valor ligeiramente maior que 1. O valor de $C_2 = 1,3$ é usual, mas em alguns casos valores menores podem ser melhores (SVANBERG, 1993).

Estas técnicas de otimização, OCM e MMA, são as técnicas mais populares e utilizadas na OT. Mais detalhes sobre a OT serão apresentados na seção Seção 2.3.

2.1.2 Algoritmos Probabilísticos

Os algoritmos probabilísticos, também conhecidos como não determinísticos, avaliam apenas a função objetivo, não necessitando de informações do gradiente (HOLTZ, 2005). A busca desses algoritmos é baseada na aleatoriedade (CAVAZZUTI, 2013). Segundo Holtz (2005) e Cavazzuti (2013), algumas das vantagens dos algoritmos probabilísticos são: não precisam de informação do gradiente (a função não necessita ser diferenciável); a função objetivo não precisa de uma representação matemática, podendo ser um resultado gerado por software (como uma caixa preta com entrada das variáveis de decisão e saída) ou funções incomputáveis, que esta fora do escopo deste trabalho; eles são menos complicados matematicamente e permitem a otimização com mais de uma função objetivo.

Além dessas vantagens, os algoritmos não determinísticos, em sua maioria, conseguem alcançar a solução ótima global e seus resultados são melhores que os algoritmos determinísticos (DEKKERS; AARTS, 1991). Porém, a velocidade para a convergência é mais lenta em relação aos determinísticos (CAVAZZUTI, 2013).

Dentre os métodos probabilísticos, alguns autores classificam em algoritmos bio-inspirados e os não bio-inspirados. Na categoria de bio-inspirados podemos citar como exemplo os algoritmos genéticos (GA do inglês *Genetic Algorithm*), enxame de partículas (PSO do inglês *Particle Swarm Optimization*) e a colônia de formiga (ACO do inglês

Ant Colony Optimization). Quanto aos não bio-inspirados podemos citar a busca de tabu (TB do inglês *Tabu Search*), Busca Gulosa aleatorizada e adaptativa (GRASP do inglês *Greedy Randomized Adaptive Search Procedure*) e o SA estudado neste trabalho. O SA será abordado no Capítulo 3.

Da família de bio-inspirado, o GA é um dos mais populares e seu método é inspirado nos princípios da seleção natural e da genética (HOLLAND, 1975 apud GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). O GA é baseado em população, e emprega uma estratégia em que ocorre uma transformação (evolução) e a cada iteração é gerada uma nova geração. Os elementos da população são conhecidos como indivíduo (ou cromossomos) e a transformação pode ocorrer por meio dos operadores: recombinação ou mutação (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). Os indivíduos podem possuir representação binária ou real, o que permite ser aplicado tanto em problemas combinatórios onde as variáveis são discretas como em problemas contínuos. Os operadores de transformação são dependentes da forma de representação. Ao longo dos anos foram propostos diversos operadores de transformação e estratégias de seleção dos melhores indivíduos (GOLDBARG; GOLDBARG; LUNA, 2017; GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

O PSO também é baseado em população, população é formada por partículas (indivíduos) (LUH; LIN; LIN, 2011). Os indivíduos realizam a busca pela solução ótima no domínio por meio de troca de informações entre os indivíduos, determinando o melhor trajeto de busca. Originalmente o PSO foi proposto para a otimização de problema contínuo por Eberhart e Kennedy (1995) e posteriormente uma versão para problemas binários (discreto) foi proposta por Kennedy e Eberhart (1997). A vantagem do PSO segundo Luh, Lin e Lin (2011) é a estrutura simples, facilidade em implementar, rápido para obter a solução e robustez. Mas possui desvantagens como a rápida convergência sem alcançar a solução ótima global, dificuldade em controle do balanço da exploração e refinamento da busca.

O ACO foi inspirado no comportamento das formigas que iniciam a busca por comida de forma sem rumo e ao encontrar a comida retornam à colônia sinalizando o caminho por meio da deposição de feromônio. Com o tempo ocorre uma evaporação natural do feromônio, porém o caminho mais curto acaba sendo favorecido por ter mais realimentações. Assim, tornando o caminho mais atrativo para as formigas subsequentes (GOSS et al., 1989; DENEUBOURG et al., 1990; GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). O ACO foi proposto para a otimização de problemas discretos, e existem poucos trabalhos relacionados a problemas contínuos (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

A heurística do TB consiste em explorar o domínio de busca movendo da solução atual para a melhor solução vizinha, mesmo que o valor da função objetivo seja pior. O algoritmo possui um mecanismo de memória de curto prazo, no qual é armazenado

em um histórico de movimentação. O TB é popular e muito utilizado com sucesso em problemas combinatórios (GOLDBARG; GOLDBARG; LUNA, 2017; GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). O algoritmo surgiu dos trabalhos independentes de Glover (1977) e Hansen (1986).

Segundo Gaspar-Cunha, Takahashi e Antunes (2012), o GRASP é um algoritmo que atua em duas fases: a construção da solução inicial por um algoritmo guloso aleatório; e a fase de busca local, o qual aplica a busca local repetidamente aperfeiçoando a solução. O GRASP é aplicado em diversos problemas de otimização combinatória e há diversas propostas de melhoria por meio de estratégias na busca e fase de construção da solução inicial.

2.2 Otimização Multiobjetivo

Em vários problemas de otimização é necessário otimizar mais de um objetivo. E usualmente esses objetivos são conflitantes entre si, em Gaspar-Cunha, Takahashi e Antunes (2012) é dado o exemplo de projeto de um motor, em que a otimização auxilia no desenvolvimento visando uma melhor eficiência do motor e redução do custo do mesmo. Um dos objetivos é um motor com melhor desempenho e o outro objetivo é o custo. Obviamente, o motor com maior eficiência não vai possuir o menor custo e o motor de menor custo não possuirá o melhor desempenho, e teremos diversos projetos com desempenhos e custos variados. Portanto, não teremos uma única solução ótima, e sim um conjunto de soluções ótimas. Estas soluções ótimas são conhecidas por diversas nomenclaturas, tais como: soluções eficientes, soluções não dominadas, soluções Pareto-ótimos, soluções não inferiores. Mais detalhes sobre as soluções ótimas são descritos na Subseção 2.2.1.

A otimização de problemas multiobjetivo era inicialmente realizada através de estratégias de reformular o problema multiobjetivo em um problema de mono-objetivo, ou seja, combinar as funções objetivo em uma única função objetivo. Essa estratégia é conhecida como *escalarização* do problema de otimização (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). Existem diversas formas de realizar a escalarização do problema de otimização multiobjetivo, como exemplos podemos citar o método ϵ – *restrito* (HAIMES YV; LASDON; WISMER DA, 1971), o método de pesos (ZADEH, 1963) e o método *Non inferior set Estimation* (NISE) (COHON; CHURCH; SHEER, 1979).

Nos últimos anos aumentou a procura por técnica para a solução dos problemas de otimização multiobjetivo, alguns métodos foram propostos. Entre esses métodos podemos citar os baseados em evolução computacional, *multiobjective evolutionary algorithm* (MOEA), mais sobre esses métodos podem ser consultado em (COELLO et al., 2007). Algumas dessas técnicas são:

- *Nondominated Sorting Genetic Algorithm* (NSGA-II) (DEB et al., 2002a);
- *Pareto Archived Evolution Strategy* (PAES) (KNOWLES; CORNE, 2000);
- *Strength Pareto Evolutionary Algorithm* (SPEA) (ZITZLER; THIELE, 1998).

Os algoritmos acima citados são os principais mencionados na literatura, mas existem muitos outros algoritmos evolutivos. Porém, os algoritmos evolutivos não são a única técnica para otimização multiobjetivo, temos a busca de Tabu, SA, PSO, colônia de formiga, dentre outras técnicas. Estas técnicas também apresentam desempenho comparável aos algoritmos evolutivos.

O SA para otimização multiobjetivo, o algoritmo estudado neste trabalho, é descrito com mais detalhes na Capítulo 4. Os outros métodos citados possuem uma breve revisão bibliográfica em Coello et al. (2007) e não serão discutidos com mais detalhe, pois está fora do escopo deste trabalho.

2.2.1 Conjunto de Pareto - Frente de Pareto

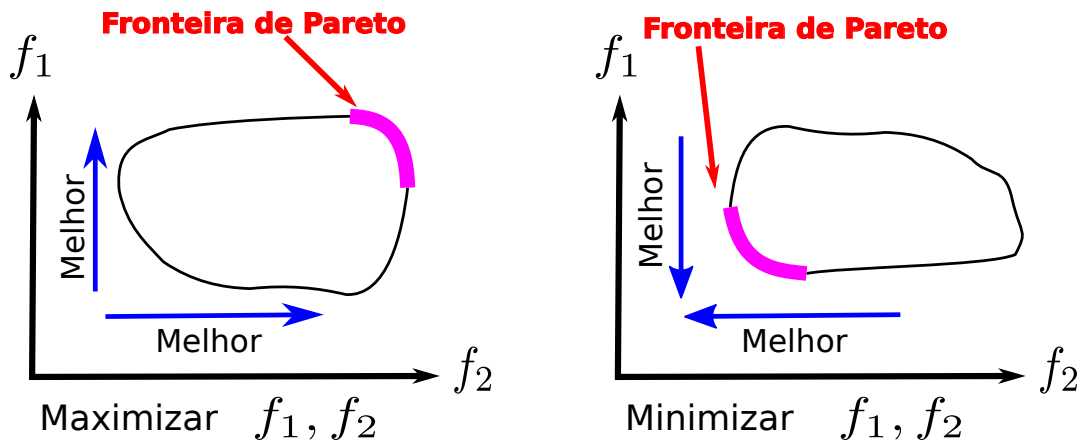
Em otimização multiobjetivo uma solução pode ser ótima em um critério, porém pode ser ruim no segundo critério. O desafio é encontrar um conjunto de soluções ótimas para todos os critérios. Portanto, na otimização multiobjetivo teremos um conjunto de soluções ótimas. A solução ótima ou solução eficiente, como é conhecida, é definida segundo Gaspar-Cunha, Takahashi e Antunes (2012) como um vetor de variáveis de decisão X factível e que não seja dominado por outra solução factível.

A solução ótima também é definida como um conjunto de soluções ótimas ou Pareto-ótimo que é representado pelo símbolo matemático \mathcal{P} . A definição de Gaspar-Cunha, Takahashi e Antunes (2012) diz que o conjunto \mathcal{P} contém todas as soluções eficientes de um problema de otimização multiobjetivo. Os valores dos objetivos correspondentes ao conjunto de soluções ótimas é conhecido como a frente de Pareto ou fronteira de Pareto. Nas Figs. 3a e 3b são apresentadas as fronteiras de Pareto para problema de otimização de maximização e minimização, respectivamente.

Em relação à dominância, podemos definir da seguinte forma: dois vetores de variáveis de decisão, X_a e X_b , dizemos que a solução eficiente X_a domina a solução X_b somente quando $f_i(X_a) < f_i(X_b)$ e $f_i(X_a) \neq f_i(X_b)$ em todas as funções do problema. Esta definição pode ser observada na Fig. 4, assim, a região onde estão os pontos verdes é denominada região de busca. Os pontos verdes são as soluções dominadas, por exemplo, a solução X_b é dominada pela solução X_a , pois $f_1(X_a) < f_1(X_b)$ e $f_2(X_a) < f_2(X_b)$. As soluções X_a e X_c são soluções não dominadas, pois, $f_1(X_a) > f_1(X_c)$ porém $f_2(X_a) < f_2(X_c)$. Os pontos em preto são as soluções não dominadas ou o conjunto de Pareto-ótimos e a linha vermelha representa a fronteira de Pareto.

Figura 3 – Fronteira de Pareto para problema de maximização e minimização

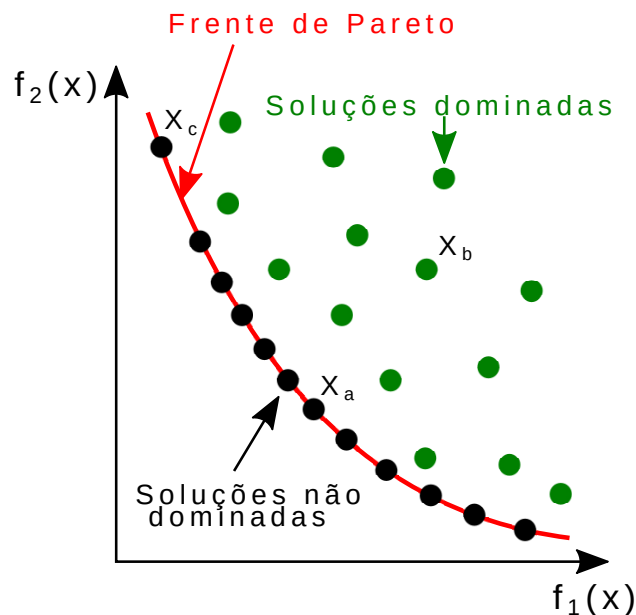
- (a) Fronteira de Pareto para um problema de maximização (b) Fronteira de Pareto para um problema de minimização



Fonte: Adaptado de Ngatchou, Zarei e El-Sharkawi (2005).

Em algoritmo de otimização multiobjetivo espera-se que o resultado forme um conjunto de soluções ótimas (Conjunto-Pareto-ótimo) semelhante ao da Fig. 4, ou seja, deve existir várias soluções distintas, mas elas devem possuir diversidade.

Figura 4 – Conceito de dominância entre soluções

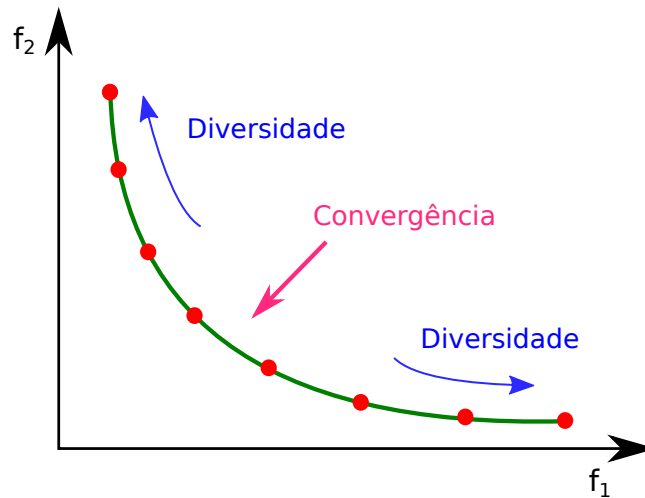


Fonte: autor.

2.2.2 Índice de performance da Frente de Pareto

A otimização multiobjetivo resulta em um conjunto de soluções ótimas. Portanto, os valores da função objetivo não permite avaliar o desempenho real do algoritmo. A forma

Figura 5 – Características para análises da frente de Pareto.



Fonte: O Autor.

correta de mensurar o desempenho do algoritmo é por meio da análise da qualidade da fronteira de Pareto. As características que podem ser avaliadas são:

- convergência, a distância da frente de Pareto obtida em relação à frente de Pareto ótima;
- a distribuição das soluções ao longo da frente de Pareto.

Essas duas características são apresentadas na Fig. 5. Como podemos observar na figura, quanto mais próximas as soluções estiverem em relação à fronteira (Frente de Pareto), melhor será a convergência. E soluções com espaçamento equidistantes e com maior diversidade, melhor será a distribuição da frente de Pareto.

Dentre as diversas métricas encontradas na literatura, temos o índice *General Distance metrics* (GD) apresentado por Veldhuizen e Lamont (1998). Este índice analisa a convergência da frente de Pareto e quanto menor o seu valor, melhor a frente de Pareto. O índice avalia a distância da frente de Pareto obtida em relação à frente de Pareto ótima. O índice GD pode ser determinado por

$$GD = \frac{1}{NPF} \left(\sum_{i=1}^{NPF} d_i^2 \right)^2 \quad i = 1, 2, \dots, NPF, \quad (2.17)$$

onde NPF é a quantidade de soluções no conjunto de soluções R , pertencente a frente de Pareto obtida. E R_i é a solução i do conjunto de soluções R , R^* é o conjunto de soluções pertencente a frente de Pareto ótima obtida de forma analítica, ou obtido na literatura ou gerado pelo algoritmo de otimização de referência. d_i é a distância euclidiana entre a solução i da frente de Pareto obtida com a frente de Pareto ótima.

Para avaliação da distribuição das soluções ao longo da frente de Pareto pode ser utilizada a métrica *Spacing metrics* (S), proposta por Schott (1995). Essa métrica avalia o desvio padrão da distância entre cada solução e sua solução adjacente mais próxima. Quanto melhor a distribuição, menor será o valor de S . O valor da métrica

$$S = \sqrt{\frac{1}{NPF - 1} \sum_{i=1}^{NPF} (d_i - \bar{d})^2}, \quad (2.18)$$

onde d_i é definido por

$$d_i = \min_j \left(\sum_{m=1}^M |f_m^i(x) - f_m^j(x)| \right), \quad (2.19)$$

e \bar{d} é a média de todos valores de d_i .

A variável M é o número de funções objetivo do problema, i e j são soluções pertencentes à frente de Pareto obtida.

Outra métrica para análise da qualidade da frente de Pareto é o *hypervolume* (HV), e foi originalmente proposta por Zitzler e Thiele (1998) e nomeado como *size of dominated space* (tamanho da área dominada) e *size of the space covered* (tamanho da área coberta) (EMMERICH; BEUME; NAUJOKS, 2005). O HV mensura convergência e a distribuição da frente de Pareto.

O HV é definido como a união das áreas (bi-dimensional) ou volumes (Tri-dimensional) formado entre um ponto de referência R e as soluções pertencentes ao conjunto S . Essa definição é apresentada na Fig. 6 e em

$$HV(S, R) = \bigcup_{i=1}^{|S|} v_i. \quad (2.20)$$

O ponto de referência deve ser um ponto dominado por todas as soluções do conjunto S , e define os limites dos hipercubos.

Quando comparamos resultados de dois algoritmos de otimização multiobjetivo distintos, o algoritmo com melhor desempenho apresentará valores de HV maior.

Há algumas bibliotecas com funções para calcular o hypervolume da frente de pareto para a linguagem de programação Python e são disponibilizadas gratuitamente, entre elas podemos citar PyMoo ¹ (Blank; Deb, 2020) e PyGMO ².

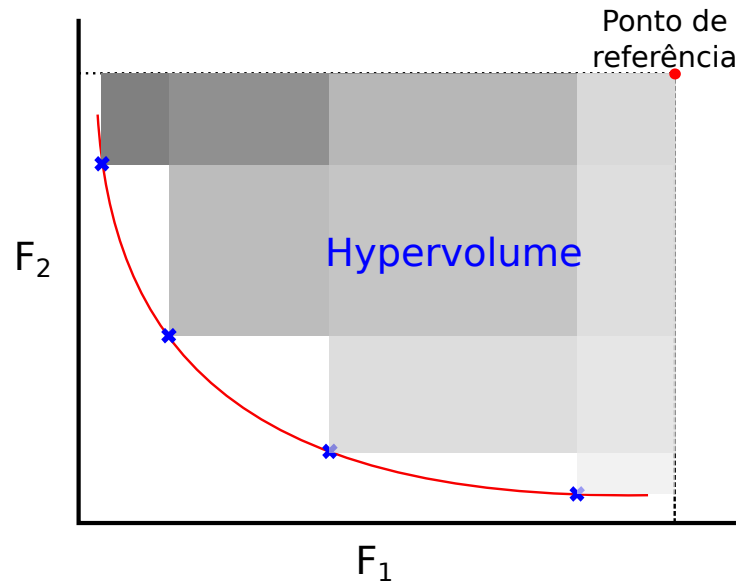
2.3 Otimização Topológica

A OT é uma ferramenta que auxilia o projetista no desenvolvimento de componentes, máquinas e estruturas, sendo definida como um método matemático para otimização do

¹ <https://pymoo.org/>

² <https://esa.github.io/pygmo/>

Figura 6 – O indicador *Hypervolume* em duas dimensões

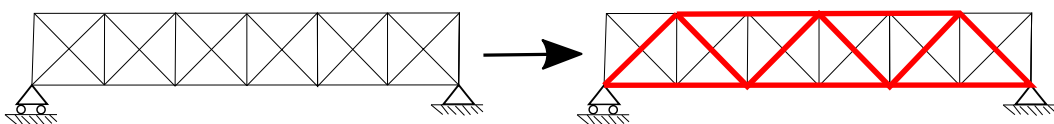


Fonte: O Autor

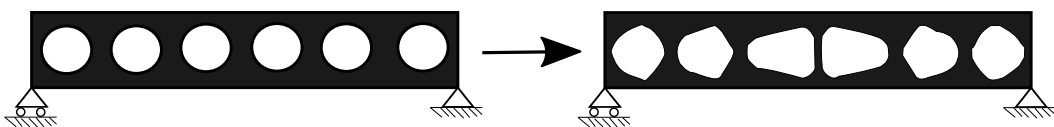
leiaute de um material, ou seja, o preenchimento de material no espaço de forma a satisfazer os requisitos de projetos e restrições, maximizando o desempenho e reduzindo os recursos empregados. A OT é diferente da otimização de forma, ou da otimização dimensional, como pode ser observado nas Figs 7a, 7b e 7c.

Figura 7 – Tipos de otimização estrutural

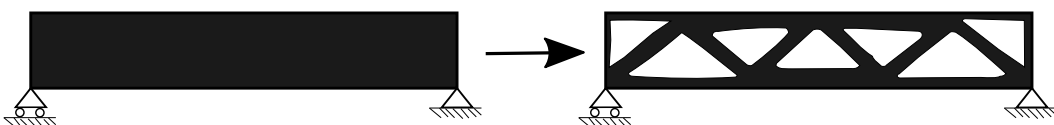
(a) Otimização dimensional, otimização de estrutura de treliça. No lado direito é apresentado o resultado da otimização



(b) otimização de forma, com o lado direito o resultado da otimização



(c) OT com o lado direito o resultado da otimização



Fonte: adaptado de Bendsøe e Sigmund (2004).

Podemos definir os tipos de otimizações da seguinte forma:

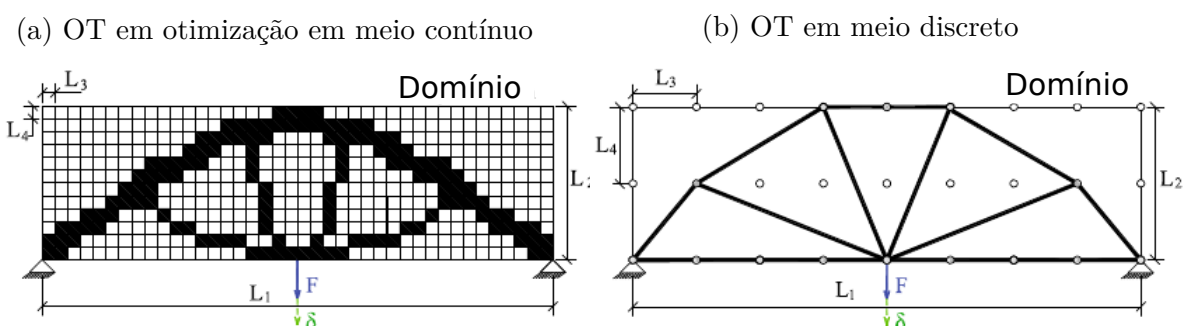
- **otimização dimensional:** neste tipo de otimização os parâmetros a otimizar são as dimensões ou razão de dimensão entre as peças da estrutura, como exemplo a espessura ou área transversal;
- **otimização de forma:** nesta otimização o objetivo é encontrar o melhor formato da estrutura, como o formato do contorno (VITORINO, 2019);
- **otimização topológica (ou de leiaute):** essa modalidade de otimização consiste em distribuir a massa do material de forma a minimizar as deformações do componente e, ainda, maximizar a rigidez (BENDSØE; SIGMUND, 2004).

A otimização de forma diferencia-se da OT na questão de que ela muda o contorno do corpo, porém não permite alterar a topologia, ou seja, não permite inserção de novos furos no domínio (BAHIA, 2005).

A OT é utilizada na otimização de estrutura e pode ser dividida em OT em meio contínuo e OT em meio discreto, de forma simples podemos diferenciar da seguinte forma:

- **OT em meio contínuo:** a OT em meio contínuo discretiza o domínio em diversos elementos retangulares, a qual deve conter o material ou não (KICINGER; ARCISZEWSKI; JONG, 2005), este conceito é apresentado na Fig. 8a;
- **OT em meio discreto:** nesta otimização são considerados vários pontos distribuídos dentro do domínio, e a otimização consiste em determinar as combinações ótimas de interligação entre os pontos por meio de barras, resultando na quantidade de barras, pontos interligados e as posições (KICINGER; ARCISZEWSKI; JONG, 2005; BAHIA, 2005; ZAVALA et al., 2014). Esta técnica é aplicada na otimização de treliças e estruturas, esta otimização é descrita na Fig. 8b.

Figura 8 – Otimização topológica em meio contínuo e discreta



Fonte: Adaptado de Zavala et al. (2014).

Os primeiros estudos relacionados à OT são atribuídos ao inventor Michell (1904 apud BENDSØE; DÍAZ; KIKUCHI, 1993a), em seu trabalho desenvolveu a otimização de

estruturas armadas para minimizar o peso. Após várias décadas, surgiram novos trabalhos utilizando os princípios teóricos desenvolvidos por Michell (1904), no entanto, aplicados em outros sistemas, como o sistema de viga desenvolvido pelo grupo de pesquisa de Rozvany (1972). Baseando-se nestas aplicações, Rozvany e Zhou (1993) apresentaram a teoria geral de otimização de leiaute (*Optimal Layout Theory*). Este conceito foi aplicado em otimização em análise exata, também foi aplicado em análise numérica (ROZVANY, 2009).

Devido à complexidade da OT, atualmente é utilizada a OT baseada na técnica de análise de elementos finitos (FE). A utilização de análise FE iniciou na década de 80, tendo como principal referência o trabalho de Bendsøe (1989), o qual apresentou o método de OT da homogeneização ou *Homogeneization Based Topology Optimization* (HBTO) como é conhecido. Na década de 90 surgiram novos métodos como o *Solid isotropic material with penalization* (SIMP) (BENDSØE; DÍAZ; KIKUCHI, 1993a), *additive evolutionary structural optimization* (AESO) (QUERIN; STEVEN; XIE, 2000), *Evolutionary structural optimization* (ESO) (XIE; STEVEN, 1993) e o *bidirectional Evolutionary Structural Optimisation* (BESO) (QUERIN; STEVEN; XIE, 1998).

Outros métodos foram desenvolvidos no decorrer dos anos, como a OT com o SA (SHIM; MANOOCHEHRI, 1997), procedimento com algoritmos genéticos (Chakraborty; Chakraborty, 1997), e as otimizações multiobjetivo aplicadas à otimização de estrutura como apresentado por Coello e Christiansen (2000).

Os principais métodos de otimização e OT serão discutidos com mais detalhes nas próximas subseções. No Capítulo 4 será abordada a otimização multiobjetivo por SA e em seguida, no Seção 6.5, será abordada a OT multiobjetivo.

Vale ressaltar que os métodos de OT não se restringem a projetos de componentes estruturais, também existem trabalhos que empregam os métodos de OT em problemas de transferência de calor, entre outras aplicações (DEATON; GRANDHI, 2014; MARCK et al., 2012).

2.3.1 O Método SIMP

O método *Solid isotropic material with penalization*, conhecido como SIMP ou *power law method*, é o mais popular e foi proposto por Bendsøe, Díaz e Kikuchi (1993a). Este também é considerado o método mais simples baseado no método numérico de FE (ROZVANY, 2009).

O método consiste em distribuir a quantidade de massa de material em cada FE de forma que minimize a energia de deformação. O método utiliza a técnica de criar uma densidade fictícia x_e , o valor desta densidade pode ser qualquer valor entre 0 e 1. Quando a densidade $x_e = 0$, significa que naquele elemento não há material, e quando $x_e = 1$, significa material sólido (PARDO, 2009).

A densidade do elemento e , determina o módulo Young de cada elemento e é determinado por

$$E_e(x_e) = E_{\min} + (E_0 - E_{\min})x_e^p, \quad (0 \leq x_e \leq 1), \quad (2.21)$$

onde E_e é o módulo de elasticidade do elemento, E_0 é a rigidez do material, E_{\min} é a rigidez mínima ligeiramente maior que zero para evitar uma matriz singular e p é o fator de penalização, o valor típico é $p = 3$. A escolha do fator de penalização p com valor muito alto ou baixo, ocasiona escala de cinza ou a convergência para o mínimo local. Segundo Sigmund e Maute (2013), este valor mágico $p = 3$ assegura a convergência da densidade para 0 ou 1.

Lembrando que esta abordagem é o SIMP modificado apresentado por Sigmund (2007), na abordagem original procurava-se evitar a densidade do elemento com o valor zero, impondo limite ligeiramente maior que zero, geralmente $x_e = 10^{-3}$, para evitar uma matriz de rigidez singular. Nesta nova abordagem este limite é garantido com o termo E_{\min} . Segundo Sigmund (2007), esta proposta apresentou diversas vantagens em relação ao original, permitindo a utilização de filtros adicionais.

Para um problema de OT baseado no método SIMP, que deseja minimizar a energia de deformação C , a formulação matemática pode ser escrita por

$$\begin{aligned} \min : \quad & C(x) = u^T K U = \sum_{e=1}^N E_e(x_e) u_e^T K_0 u_e \\ \text{sujeito a :} \quad & \frac{V(x)}{V_0} = f \\ & K U = F \\ & 0 \leq x \leq 1, \end{aligned} \quad (2.22)$$

onde U e F são os vetores deslocamento e a força, respectivamente. K é a matriz rigidez global, u_e é o deslocamento do elemento, $V(x)$ e V_0 são o volume do material e o volume do espaço do domínio, respectivamente. x é o vetor de variável decisão, e f é a fração volumétrica prescrita.

A otimização do método SIMP pode ser realizada por meio do algoritmo critério de Optimalidade (*Optimality criteria method*), esta formulação é descrita matematicamente em

$$x_e^{novo} = \begin{cases} \max(0, x_e - m) & \text{Se } x_e B_e^\eta \leq \max(0, x_e - m) \\ \min(0, x_e + m) & \text{Se } x_e B_e^\eta \geq \min(0, x_e - m) \\ x_e B_e^\eta & \text{Se não} \end{cases}, \quad (2.23)$$

onde m é o limite de movimentação positiva, $\eta = 1/2$ é um coeficiente de amortecimento, e B_e é obtido na condição ótima em

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}}, \quad (2.24)$$

onde a variável λ é o multiplicador de Lagrange e deve ser escolhida de forma que satisfaça a restrição de volume, o valor apropriado pode ser obtido por meio do algoritmo da bisseção (ANDREASSEN et al., 2011). A sensibilidade do objetivo da função, $c(x)$, e o volume em relação a densidade x_e são determinados em

$$\frac{\partial c}{\partial x_e} = -px_e^{p-1}(E_0 - E_{\min})u_e^T k_0 u \quad (2.25)$$

$$\frac{\partial V}{\partial x_e} = 1. \quad (2.26)$$

Em outros trabalhos, o método de otimização de critério ótimo é substituído por outros métodos, como o *Method of Moving Asymptotes* (MMA) ou *sequential quadratic programming* (SQP), um trabalho comparativo entre vários métodos foi publicado por Fanni, Shabara e Alkalla (2013).

Sigmund (2001) demonstrou que para o correto funcionamento da OT é necessária a combinação do método SIMP com restrições no perímetro, restrição de gradiente ou uma técnica de filtro. O método SIMP permite aplicar estes métodos em problemas com múltiplas restrições e múltiplos materiais. Estas técnicas são necessárias para evitar problemas conhecidos como tabuleiro de xadrez ou a formação de gradiente de cinza entre a transição do material sólido e o espaço vazio. Este padrão conhecido como tabuleiro de xadrez e a transição de gradiente cinza pode ser observado na Fig. 9.

Mais informações sobre a instabilidade, gradiente e tabuleiro de xadrez e, ainda, soluções para estes problemas são descritos em Sigmund e Petersson (1998) e Sigmund (2007). Um estudo comparativo entre diversos filtros é apresentado por Sigmund (2007) em diversos exemplos de testes de OT.

Figura 9 – Padrão conhecido como tabuleiro de xadrez e o gradiente de cinza entre a transição entre o material sólido e vazio



Fonte: Díaz e Sigmund (1995).

A aplicação do método SIMP em casos de testes, escrito em Matlab com 99 linhas de códigos, com objetivo didático, foi disponibilizada por Sigmund (2001). Uma versão aprimorada deste código, com redução para 88 linhas de código foi apresentada

por Andreassen et al. (2011), ainda foi disponibilizada uma implementação em Python. Mais informações sobre o método SIMP podem ser encontradas em uma extensa revisão bibliográfica publicada por Rozvany (2009), Deaton e Grandhi (2014).

2.3.2 Método ESO

Este método começou a ser desenvolvido em 1992 por Xie e Steven (1993), este foi o primeiro método de OT evolucionário e se baseia na remoção de material. O algoritmo basicamente funciona da seguinte forma: no início o espaço de projeto inicia-se cheio e o algoritmo remove a massa de material desnecessário ou que não contribui na resistência da peça. O material é removido gradualmente até que a estrutura evolua para uma distribuição de material ótima, ou seja, até obter uma estrutura que suporta o carregamento com a menor massa possível de material.

Este método possui um algoritmo simples e utiliza a discretização de FE para a solução do sistema. A OT de problema de minimização da energia de deformação, C , é representada matematicamente em

$$\begin{aligned} \min : \quad & C = U^T K U \\ \text{sujeito a} \quad & \frac{V}{V_0} \leq V_f \\ & K U = F \\ & x = [0, 1]. \end{aligned} \quad (2.27)$$

O método ESO se baseia na ideia de que uma estrutura ótima é aquela em que todos os elementos estão sofrendo o mesmo nível de tensão. O material é removido de acordo com o critério de

$$\frac{\sigma_e^{vm}}{\sigma_{\max}^{vm}} = RR_i. \quad (2.28)$$

A qual verifica se a tensão de Von mises para o elemento e , σ_e^{vm} em relação à tensão de Von mises de toda a estrutura σ_{\max}^{vm} é menor do que a razão de rejeição, RR_i . Caso o critério seja satisfeito, o material é removido daquele elemento. Ao fim de cada análise de FE, todos os elementos que satisfazem a (2.28) são removidos (XIE; STEVEN, 1997).

Este processo é repetido com o valor de RR_i constante até que se alcance um estado estável. Este estado é alcançado quando não há mais material sendo removido dos elementos naquela iteração. Neste ponto é introduzida a *razão de evolução* (ER) que é adicionada a razão de rejeição que pode ser descrito como

$$RR_{i+1} = RR_i + ER \quad i = 0, 1, 2, 3 \dots \quad (2.29)$$

Após aumentar o valor da razão de rejeição, é novamente realizada a análise de FE e remoção de material desnecessário, repetindo o ciclo até que alcance o estado estável novamente.

Portanto, podemos resumir os procedimentos do método ESO conforme descrito em Pardo (2009), da seguinte forma:

1. escolher o modelo da distribuição de tensão utilizando FE;
2. usar um critério de rejeição, ER ;
3. estabelecer uma razão de rejeição, RR_i , originalmente o critério da (2.28) em caso de satisfazer, remover o elemento;
4. eliminar todos elementos considerados ineficientes, ou seja, todos que satisfizerem o critério do passo anterior;
5. repetir os passos 2 a 4 até que se alcance o *estado estável*;
6. adicionar ao critério de rejeição o valor da *razão de evolução*;
7. repetir os processos do 1 ao 6 até obter a topologia ótima.

Idealmente, a estrutura ótima ocorre quando todos os elementos estão com tensões no mesmo nível. Em relação ao método baseado na densidade, a vantagem do método ESO é não ocorrer o problema de gradiente de cinza entre material sólido e vazio. Isso devido à variável de decisão apenas possuir o valor de $x_e = 0$ ou $x_e = 1$. Este método foi melhorado posteriormente e proposto um novo algoritmo conhecido como *Bi-directional Evolutionary Structural Optimization* - BESO que será discutido na Subseção 2.3.3.

2.3.3 Método BESO

Este método foi proposto por Querin, Steven e Xie (1998) e é uma evolução do método ESO. Diferente do método anterior este é bidirecional, ou seja, ele permite remover ou adicionar material. O processo de adição é baseado no método AESO e o processo de remoção do material é baseado no método ESO, porém nos dois casos há algumas modificações para que o algoritmo BESO funcione adequadamente.

As regras do algoritmo para o correto funcionamento segundo Querin, Steven e Xie (1998) são:

1. deve ser especificado o domínio máximo que a estrutura pode ocupar;
2. o domínio físico tem que ser dividido em malha regular. Os elementos devem ser quadrado 2D ou elementos em cubo para 3D;
3. definir todos os contornos cinemáticos, restrições e propriedades do material;

4. no método ESO todo domínio é utilizado, enquanto que no AESO foi estabelecido um mínimo entre as cargas e suportes. No BESO é estabelecido inicialmente uma quantidade mínima de elementos entre as cargas e suporte;
5. é atribuído o valor zero a todos os elementos não inclusos no domínio inicial;
6. especificar o critério de otimização, Querin, Steven e Xie (1998) citam o critério de tensão de Von mises;
7. o critério de condução da otimização deve ser a maximização do critério escolhido na etapa anterior (6);
8. conduzir a análise linear de FE na estrutura;
9. remover ou adicionar elemento em regiões de baixa ou alta tensão, respectivamente. Em região de baixa tensão, o elemento é removido de acordo com o critério

$$\sigma_e \leq RR \cdot \sigma_{\max}. \quad (2.30)$$

Para região de alta tensão é adicionado elemento de acordo com o critério descrito por

$$\sigma_e \geq IR \cdot \sigma_{\max}, \quad (2.31)$$

onde σ_e é a tensão do critério de Von mises ou outro critério selecionado, σ_{\max} tensão máxima do critério de Von mises máximo da estrutura, RR é a razão de rejeição, IR é a razão de adição.

10. se alcançar um estado em que nenhum elemento do domínio pode satisfazer os critérios (2.30) e (2.31), considera-se que alcançou o estado estável e para continuar o processo de otimização, o número de estado estável SS deve ser incrementado em 1 e voltar a etapa 9;
11. se um grupo de elementos é removido e depois adicionado de volta após iterações, significa que alcançou um estado oscilatório. O número oscilatório ON é incrementado em 1 e repete-se a partir da etapa 9;
12. as etapas 8 - 11 são repetidas até que o índice de performance (PI) alcance o valor mínimo, ou até alcançar o limite de tensão do critério.

A função objetivo a ser minimizada e as restrições da OT são representadas por

$$\begin{aligned}
 \min : \quad & C = f(x) = PI = \frac{\sum_{e=1}^N \sigma_{VM_e} V_e}{FL} \\
 \text{sujeito a:} \quad & \sum_{e=1}^N [[K]^e \{u\}^e - \{F_0\}^e - \{F_b\}^e - \{F_s\}^e] - \{P\} = 0 \\
 & x_e (x_e \sigma_{VM_e} - RR \sigma_{VM_{\max}}) \geq 0 \\
 & \eta_e (IR \sigma_{\max} - \eta_e \sigma_e) \geq 0 \\
 & x_e \in X_e = \{0, 1\}, \quad e \in E \\
 & \eta_e \in H_e = \{1, t\},
 \end{aligned} \tag{2.32}$$

onde E é um conjunto de elementos discretos, η_e é um multiplicador de adição de elementos com intervalo de $0 \leq \eta_e \leq 1$, e H_e é o conjunto de valores de η_e .

Segundo Querin, Steven e Xie (1998), o método apresentou resultados de OT consistente com o esperado e com tempo de processamento menor em relação ao método ESO. Outros trabalhos relacionados ao método BESO foram publicados, como o trabalho de Zuo, Xie e Huang (2009) que combinou o algoritmo genético com este método. Ainda temos a pesquisa de Sun et al. (2011) que empregou o método para a otimização em materiais compósitos, ou seja, material anisotrópico, com um refinamento no critério de otimização.

2.3.4 Otimização Topológica e Algoritmos Não Gradiente

A maioria dos trabalhos sobre OT encontrados na literatura utilizam algoritmos baseados em gradiente, entre eles os algoritmos OCM, MMA ou SLP. Esses algoritmos tradicionais são mais simples de serem implementados e convergem de forma rápida. Porém, é necessário determinar o vetor gradiente, para determinar a sensibilidade da função objetivo descrita na Seção 2.3.1. O vetor gradiente é a derivada da função objetivo, e este pode ser complicado em se obter dependendo do domínio. Alguns trabalhos publicados substituíram os algoritmos baseados no gradiente por algoritmos baseados em não gradientes, como os genéticos ou SA. Estes estudos de OT baseado em algoritmo não gradiente foram realizados por Kane e Schoenauer (1996), Garcia-Lopez et al. (2011), Bureerat e Limtragool (2006) e Bureerat e Limtragool (2008).

Bureerat e Limtragool (2006) e Kane e Schoenauer (1996) descrevem que os algoritmos evolutivos são simples e mantêm suas vantagens, como encontrar o mínimo global. Bureerat e Limtragool (2006) destaca ainda que o desempenho do SA é melhor em relação ao algoritmo genético na OT, isso devido à modificação da solução no SA ser melhor em relação ao cruzamento utilizado no algoritmo GA em problemas de OT. Bureerat e Kunakote (2006) também observou que o desempenho do operador de mutação apresentou resultados melhores em relação ao cruzamento no GA.

Porém, os algoritmos evolutivos como o genético e SA são mais lentos em relação aos algoritmos tradicionais, baseado em gradiente, devido à grande quantidade de variáveis do problema de OT. Para contornar este problema, alguns autores propuseram algumas estratégias. Bureerat e Limtragool (2006), realizaram alguns estudos da busca local para os algoritmos genéticos e recozimento simulado e propôs uma técnica numérica conhecida como *Approximate Density Distribution* (ADD). Esta técnica reduz a quantidade de variáveis sem alterar a quantidade de elementos finitos. Os resultados demonstraram que os SA obtiveram resultados melhores em relação aos algoritmos genéticos, devido a a forma de gerar a nova solução.

Bureerat e Limtragool (2008) propôs uma nova técnica denominada *Multiresolution Design Variable* (MRDV). Nesta nova técnica, o algoritmo aplica a técnica ADD com baixa resolução no início e aumenta gradualmente a resolução da malha ao longo do processo de otimização, dessa forma melhorando o desempenho e os resultados obtidos. Mas mesmo com a utilização da técnica proposta, os algoritmos tradicionais baseados em gradiente convergem mais rápido.

Garcia-Lopez et al. (2011) propôs um algoritmo híbrido de OT com o SA e o método SIMP. Neste novo algoritmo, são utilizadas estratégias de seleção de elementos e estratégias de mutação da densidade desses elementos, removendo as densidades intermediárias (valores entre 0 e 1). Os resultados obtidos por Garcia-Lopez et al. (2011) apresentaram valores de energia de deformação comparáveis aos obtidos pelos métodos tradicionais e apresentaram menor área com gradiente de cinza.

Em seu trabalho Sigmund (2011) detalha as desvantagens dos métodos não-gradientes, e uma das desvantagens é a solução de problemas com malhas com elevado número de elementos. Nesses casos são necessárias milhões de avaliações da função objetivo, exigindo uma grande capacidade computacional. Alguns autores, como Guirguis, Melek e Aly (2018), descrevem que essas limitações podem ser contornadas.

Uma ampla revisão da literatura foi realizada por Guirguis et al. (2020), e em seu trabalho os autores classificaram os algoritmos não gradiente como *Black-box topology optimization* (BBTO). No mesmo trabalho os autores classificam os algoritmos de acordo com o domínio: domínio da função objetivo (*phenotype*) e o domínio das variáveis de decisão (*genotype*).

Guirguis et al. (2020) salienta que os problemas de OT apresentam diversas inovações e possuem inúmeras propostas de algoritmos diferentes para uma otimização eficiente. Ainda destaca que os BBTO podem ser um complemento, e preencher a lacuna de aplicabilidade dos algoritmos baseados em gradiente.

2.4 Considerações Finais

Neste capítulo discutimos os conceitos fundamentais e principais métodos de otimização, otimização multiobjetivo e a OT. No próximo capítulo será apresentado de forma mais aprofundada a otimização por meio do recozimento simulado. Além disso, apresentaremos algumas heurísticas utilizadas neste trabalho. No Capítulo 4 apresentaremos a otimização multiobjetivo baseada no SA e os resultados obtidos pela proposta deste trabalho. No capítulo seguinte apresentamos a aplicação do algoritmo multiobjetivo em aplicações mecânicas. Em seguida apresentamos aplicação do recozimento simulado em OT mono-objetivo e multiobjetivo.

3 Recozimento Simulado

Este capítulo trata dos conceitos do SA, será detalhado os princípios de funcionamento do algoritmo, assim como suas desvantagens e vantagens. Além disso, serão apresentadas as heurísticas e estratégias para o recozimento simulado e seus principais parâmetros. O intuito deste capítulo é apresentar os conceitos do recozimento simulado e as heurísticas utilizadas no recozimento multiobjetivo.

3.1 Funcionamento do SA

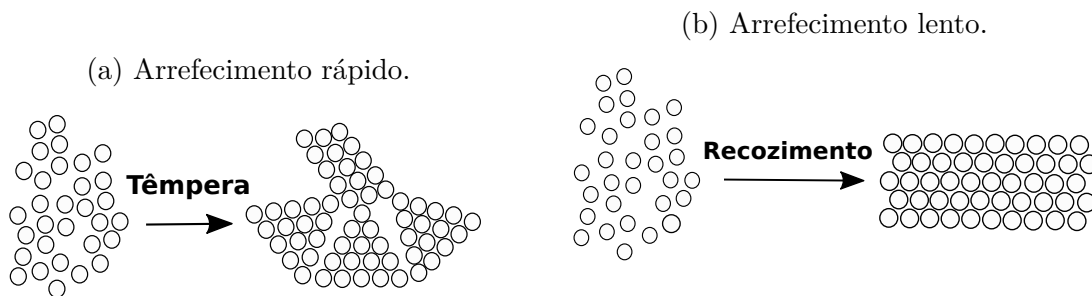
O algoritmo de SA foi introduzido por Kirkpatrick, Gelatt e Vecchi (1983) e Černý (1985) na década de 80 baseado nos algoritmos Metropolis, o qual foi proposto por Metropolis et al. (1953). A maioria dos trabalhos teóricos e aplicações do recozimento simulado está relacionado a otimização de problemas discretos (HENDERSON; JACOBSON; JOHNSON, 2003). Entretanto, o algoritmo tem sido utilizado amplamente em problemas de otimização com domínio contínuo. Há um grande interesse no SA devido sua capacidade de resolver problemas contendo muitos mínimos locais (HENDERSON; JACOBSON; JOHNSON, 2003).

O SA faz uma analogia ao processo de recozimento de metais. O processo conhecido como recozimento se refere a um tratamento térmico em materiais metálicos, o qual consiste em um aquecimento a temperatura elevada por um longo período e em seguida um resfriamento lento até a temperatura do ambiente (JUNIOR, 2006), o tratamento térmico é ilustrado na Fig. 10. Em altas temperaturas os átomos podem se mover livremente, e durante o resfriamento ocorre a reordenação dos átomos (processo de cristalização) minimizando a energia livre do sólido.

O processo de resfriamento é de grande importância para o tratamento térmico. Se ocorrer um resfriamento muito rápido ou a temperatura de aquecimento for baixa, não é possível obter a cristalização e sim um estado amorfo com alta energia, como apresentado na Fig. 10a. Na otimização isto significa obter o mínimo local, com um estado de alta energia (COELLO et al., 2007). Quando o processo é resfriado lentamente, os átomos se arranja formando uma estrutura cristalina, procurando uma configuração para minimizar o estado de energia, conforme representado na Fig. 10b que foi adaptada de (GOLDBARG; GOLDBARG; LUNA, 2017).

O pseudo código do SA é apresentado no Algoritmo 1. O algoritmo minimizará o valor de uma função f por meio de pequenas mudanças aleatórias na *variável de controle* X , em sua vizinhança, este processo é apresentado na Fig. 11. Isto considerando a variação

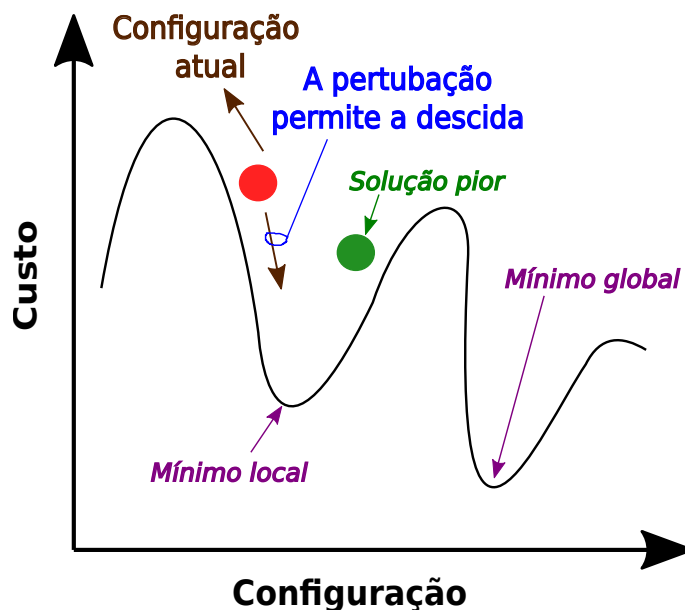
Figura 10 – Tratamento térmico do recozimento



Fonte: Adaptado de Goldberg, Goldberg e Luna (2017).

da função f , se o valor da função $f(X_n)$ diminuir, a nova solução X_n é aceita. Se o valor da função $f(X_n)$ aumentar, a nova solução poderá ser aceita ou não conforme o critério, este processo de aceitar uma solução pior, permite fugir do mínimo local para alcançar o mínimo global como apresentado na Fig. 11. Este critério é conhecido como critério de Metropolis, sendo esta a principal contribuição do algoritmo Metropolis para o SA.

Figura 11 – A busca por solução e a escalada aceitando solução pior para encontrar a solução ótima global.



Fonte: Adaptado de (RUTENBAR, 1989).

O critério para aceitar ou não a solução é a decisão mais importante deste algoritmo. Como descrito anteriormente, se a função objetiva, $f(X_n)$ é menor que $f(X_a)$, a solução será aceita. Se o valor de $f(X_n)$ for maior que $f(X_a)$ ¹ a solução é aceita ou não de acordo com o fator de probabilidade de Boltzmann. Este fator de probabilidade é determinado por

¹ X_a é a solução corrente

$$P = e^{-\frac{\Delta E}{k_b T}}. \quad (3.1)$$

Sendo baseado na distribuição de probabilidade de Maxwell-Boltzman. Onde a variável T é a temperatura, k_b é a constante de Boltzman e ΔE é a variação do estado de energia, que pode ser determinado por

$$\Delta E = f(X_n) - f(X_a), \quad (3.2)$$

onde $f(X_a)$ é o valor da função objetivo da solução corrente e $f(X_n)$ é o valor da função objetivo da solução nova.

Observe que, quando a temperatura T é alta, a probabilidade de aceitar uma nova solução aumenta (SUPPAPITNARM et al., 2000), independente do sinal de ΔE . Caso T seja pequeno, apenas valores positivos e pequenos de ΔE serão aceitos. De forma mais simples, quando a temperatura T é alta, o algoritmo aceita a busca por novas soluções mais distantes, ou seja, aceita valores mais aleatórios (GOLDBARG; GOLDBARG; LUNA, 2017). Enquanto, em temperaturas T menores, o algoritmo aceita apenas buscas nas regiões próximas à solução atual.

Observe que no Algoritmo 1, existe um laço de repetição, caracterizado como o comprimento de temperatura, ou de forma mais simples é a quantidade de vezes realizada a procura por uma nova solução na vizinhança em cada temperatura (PIRLOT, 1996).

Em relação à busca local, inicialmente o SA era aplicado em diversos problemas combinatório, desde problemas como o caixeiro viajante até projeto de circuitos integrados (KIRKPATRICK; GELATT; VECCHI, 1983; RUTENBAR, 1989). Nos problemas de otimização combinatório a variável de decisão é do tipo discreta, e a nova solução X_n é obtida por meio de permutação ou outra operação discreta. Para problemas contínuos, (BOHACHEVSKY; JOHNSON; STEIN, 1986) propuseram a geração da nova solução por

$$X_n = X_a + \Delta r \cdot u, \quad (3.3)$$

onde, u é um vetor direção, e Δr é passo fixo. Este passo fixo deve ser selecionado conforme a função objetivo, sendo um parâmetro crítico.

O SA é apresentado no Algoritmo 1. O algoritmo funciona da seguinte forma:

1. o primeiro passo é a definição das variáveis de entrada T_0 e α , temperatura e o fator de decaimento da temperatura;
2. o algoritmo inicia-se gerando a solução atual X_a , de forma aleatória conforme a linha 2;
3. o algoritmo gera uma nova solução X_n conforme a linha 5, a partir da perturbação da solução atual X_a , é modificado uma variável de decisão por vez;

4. em seguida é calculada a diferença do estado de energia ΔE , com a (3.2), na linha 6;
5. é gerado um valor i , aleatoriamente entre 0 e 1, se este valor for menor que o fator de probabilidade de Boltzman $e^{-\frac{\Delta E}{k_b T}}$, a solução será aceita e atribuída à solução atual X_a . Se não for aceito será mantida a solução atual;
6. as etapas 3 a 5 são repetidas N vezes, sendo esta a quantidade de busca na vizinhança estabelecida;
7. a temperatura decresce de acordo com a estratégia de resfriamento adotada, geralmente utiliza a redução geométrica proposta por Kirkpatrick, Gelatt e Vecchi (1983) conforme a linha 17.
8. as etapas 3 a 7 (linha 4 a 17) são repetidas enquanto a temperatura for maior que a temperatura mínima.

Algoritmo 1: Recozimento simulado

Entrada: T_0 : Temperatura inicial; α : fator de decaimento da temperatura

```

1  $T_j \leftarrow T_0$ 
2  $X_a \leftarrow$  Solução inicial aleatória
3 enquanto  $T_j >$  Temperatura de equilíbrio faça
4   enquanto Critério de parada local não satisfeita faça
5      $X_n \leftarrow$  pertubação da solução  $X_a$ 
6      $\Delta E = F(X_n) - F(X_a)$ 
7     se  $\Delta E \leq 0$  então
8        $X_a \leftarrow X_n$ 
9     senão
10      se  $random(0, 1) < e^{-\frac{\Delta E}{k_b T_j}}$  então
11         $X_a \leftarrow X_n$ 
12      senão
13         $X_n$  é rejeitado
14      fim
15    fim
16  fim
17   $T_{j+1} \leftarrow T_j \times \alpha$ 
18 fim

```

Em relação ao critério de aceitação a nova solução do Algoritmo 1, se o estado da energia ΔE for menor ou igual a zero, a solução nova será aceita, pois o valor do fator de probabilidade de Boltzman será 1. Em caso do valor ser maior que zero, a aceitação dependerá da probabilidade calculada em (3.1). Este comportamento de aceitar ou não a solução é descrito por

$$P(X_n) = \begin{cases} 1 & \text{se } f(X_n) \leq f(X_a) \\ \exp\left(\frac{\Delta E}{k_b T}\right) & \text{se } f(X_n) > f(X_a). \end{cases} \quad (3.4)$$

Lembrando que a probabilidade também depende da temperatura, quando a temperatura é alta, a chance de aceitar a nova solução é alta. Quando a temperatura for pequena, a probabilidade de aceitar esta nova solução diminui. Portanto, esse critério de aceitação da nova solução permite a procura pela solução em pontos mais distantes e aceitar soluções piores que a solução atual para fugir do mínimo global (ULUNGU et al., 1999). E quando a temperatura é pequena, ou seja, está chegando perto do equilíbrio, o critério aceita a procura por novas soluções em regiões próximas e não permite aceitar uma solução pior.

A temperatura decai ao longo do processo de otimização, este decaimento da temperatura ou resfriamento ocorre conforme a estratégia escolhida. Inicialmente, era utilizada a estratégia de resfriamento logarítmica, porém o mais comum é a estratégia de resfriamento geométrica proposta por Kirkpatrick, Gelatt e Vecchi (1983), apresentado em

$$T_{j+1} = \alpha T_j, \quad (3.5)$$

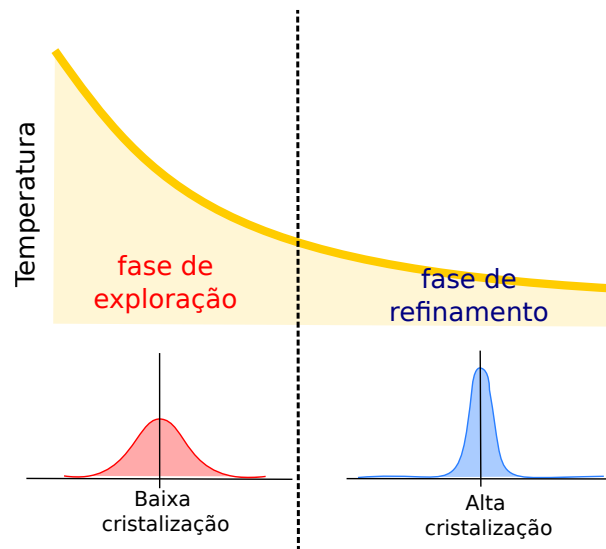
onde α é o decaimento de temperatura, geralmente o valor adotado é entre 0,8 a 0,99. E a otimização finaliza quando a temperatura é igual ou inferior a zero. E como descrito anteriormente, existem outras propostas de resfriamento que podem ser consultados em Goldbarg, Goldbarg e Luna (2017). Em relação à estratégia de temperatura, o mais importante é que o processo de resfriamento seja lento o suficiente e a estratégia geométrica apresenta desempenho satisfatório (PIRLOT, 1996). Entretanto, quando se deseja um desempenho melhor, a estratégia de resfriamento pode se tornar crucial (PIRLOT, 1996).

3.2 Fase de Exploração e de Refinamento

No SA, durante o processo de otimização temos duas fases: exploração e refinamento. Estas fases são descritas na Fig. 12, onde apresenta a fase de exploração em altas temperaturas e a fase de refinamento em temperaturas baixas. Inicialmente o algoritmo inicia com temperatura alta, conseqüentemente o desvio padrão é mais alto, permitindo saltos maiores (RUTENBAR, 1989; SATO et al., 2014).

Devido essa capacidade de saltos mais altos, o algoritmo consegue realizar buscas mais longe da solução atual, ou seja, explorando mais. Enquanto em baixa temperatura, a probabilidade é maior para saltos menores (SATO et al., 2014). Conseqüentemente, a busca pela nova solução é próxima da solução atual. Mas em baixa temperatura, ainda é possível o algoritmo realizar grandes saltos.

Figura 12 – Fase de exploração e refinamento durante o processo de otimização.



Fonte: Adaptado de Sato et al. (2014)

A exploração ou refinamento é controlado por meio da aceitação da solução já mencionada anteriormente. O SA aceita uma nova solução na condição que $\Delta E < 0$ ou se um número aleatório for menor que o fator de probabilidade de Boltzman dado em (3.1). Os valores de temperatura alta tende a gerar valores próximos de 1, conforme apresentado em (3.4).

3.3 Principais Parâmetros

No SA temos alguns parâmetros importantes, dentre eles temos a temperatura, o fator de decaimento da temperatura α . A escolha da temperatura inicial e a temperatura mínima influenciam no balanço entre as fases de exploração e o refinamento. O α influencia na velocidade do resfriamento, lembrando que o resfriamento deve ser lento o suficiente para encontrar a solução ótima global.

Outro parâmetro importante para o SA, é a quantidade de iteração N . Este parâmetro determina a busca local, a cada nova iteração o algoritmo realiza uma busca local por uma nova solução. O valor típico depende do tipo de função objetivo e valores muito baixos pode não ser suficiente para convergir a solução ótima.

3.4 Heurística de Cristalização

Com a característica do SA de possuir fases de exploração e refinamento, Martins e Tsuzuki (2010), Tavares, Martins e Tsuzuki (2011) propuseram um novo SA com uma heurística de cristalização. Este novo algoritmo foi denominado recozimento simulado de

vizinhança adaptativa. Foi inserida a heurística de cristalização com objetivo de aumentar o índice de aceitação da nova solução.

Na heurística de cristalização, cada variável de decisão possui o seu fator de cristalização C_k . E na geração da nova solução é selecionada apenas uma variável de decisão de forma aleatória, que será modificada. Para as variáveis contínuas foi proposta a modificação da variável de decisão por

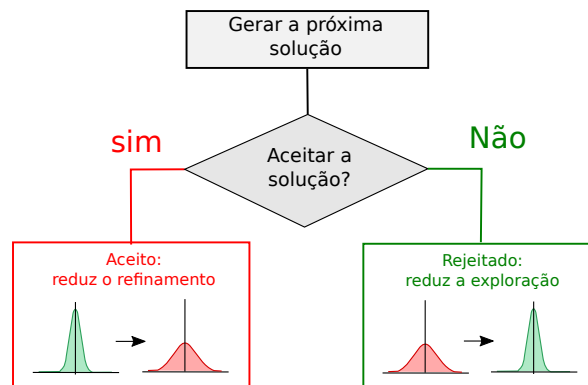
$$x_{i+1} = x_{i,k} + \frac{1}{C_k} \sum_1^{C_k} \text{random}(-1, 1) \cdot \Delta r_k \cdot e_k, \quad (3.6)$$

onde C_k é o fator de cristalização k , e $\text{random}(-1, 1)$ é um número gerado aleatoriamente entre -1 e 1. E e_k é o vetor direção no eixo k e Δr_k é o tamanho do passo, este é um valor constante, e geralmente utiliza o valor de 1/4 do intervalo de busca e é definido como

$$\Delta r_k = \frac{x_{k,\max} - x_{k,\min}}{4}. \quad (3.7)$$

O processo de geração da nova solução utiliza a distribuição de probabilidade de Bates como descrito em (3.6), e o fator de cristalização ajusta a busca local. Quando uma nova solução é rejeitada, a otimização assume que está na fase de exploração e em reposta o fator de cristalização é incrementado. Este aumento do fator de cristalização reduz a probabilidade de saltos maiores. Quando a nova solução é aceita, é assumida a fase de refinamento e o fator de cristalização é reduzido conforme a estratégia de cristalização. Com redução do fator de cristalização aumenta a probabilidade de saltos maiores (GOTO et al., 2021b; DURAN et al., 2021). Como descrito, quando a nova solução é aceita ou rejeitada, o algoritmo ajusta o fator de cristalização por meio de feedback positivo ou negativo, este processo é apresentado no fluxograma da Fig. 13.

Figura 13 – Fluxograma do processo de feedback do algoritmo de otimização. Quando a nova solução gerada é aceita, o fator de cristalização da variável de decisão recebe o feedback positivo. Quando é rejeitada, a variável de decisão recebe o feedback negativo



Fonte: Adaptado de Sato et al. (2014).

Para o problema contínuo, o feedback positivo ou negativo do fator de cristalização utilizado por Tavares, Martins e Tsuzuki (2011) é dado em

$$C_k = C_k - 1 \quad \langle \text{Realimentação Positiva} \rangle \quad (3.8)$$

$$C_k = C_k + 1 \quad \langle \text{Realimentação Negativa} \rangle . \quad (3.9)$$

O fator de cristalização inicia-se com valor igual a um para todas as variáveis de decisão. E o pseudo código do recozimento simulado de vizinhança adaptativa é apresentado no Algoritmo 2.

O SA permite a solução de problemas com variáveis discretas, contínuo e inteiro mesmo que no mesmo problema de otimização, como podemos observar no Algoritmo 2. O SA de vizinhança adaptativa foi aplicado em diversos problemas de otimização como podemos constatar em Duran et al. (2021), Tavares, Martins e Tsuzuki (2011) e Martins e Tsuzuki (2010).

3.5 Estratégias de Feedback

Apresentamos na seção anterior o Algoritmo 2, onde podemos observar que a heurística de cristalização possui um feedback (resposta) de acordo com aceitação ou rejeição da solução. Este feedback incrementa ou decresce o valor do fator de cristalização e a taxa de incremento ou decaimento pode variar conforme a necessidade da função objetivo.

Em Sato et al. (2014) e Duran et al. (2021) são apresentadas diversas estratégias de resposta para a fase de exploração e fase de refinamento, essas estratégias foram compiladas na Tabela 2. A Estratégia I é indicada para funções objetivo que necessitam uma maior exploração ou funções objetivos com muitos mínimos locais, nesta estratégia o fator de cristalização reinicia para o valor 1. Enquanto a Estratégia III é utilizada para casos que seja necessário um maior refinamento da função objetivo (MARTINS; TSUZUKI, 2010). A Estratégia II é uma solução intermediária entre a Estratégia I e III, já a Estratégia IV é uma estratégia híbrida (DURAN et al., 2021).

Duran et al. (2021) apresentou uma evolução na heurística de cristalização, combinando a distribuição de Bates e Gauss. Na SA de vizinhança adaptativa, o fator de cristalização era limitado a um valor C_{max} . Pois, inúmeras somas de números aleatórios entre -1 e 1, o valor cresce substancialmente e a adição de um novo valor entre -1 e 1, não surtirá nenhum efeito. Na nova proposta, o fator de cristalização não possui um limite máximo C_{max} e o problema mencionado é solucionado utilizando a distribuição de Gaussiana. A média da distribuição gaussiana é zero e o desvio padrão é definido por

$$\exp(C_{max} - C_k - 2), \quad (3.10)$$

Algoritmo 2: RECOZIMENTO SIMULADO COM VIZINHANÇA ADAPTATIVA.

Entrada:
 T_0 : Temperatura inicial
 α : Fator de decaimento da temperatura
 C_{\max} : Fator de cristalização máximo

- 1 $T_j \leftarrow T_0$
- 2 $X_a \leftarrow$ Solução inicial aleatório
- 3 $i \leftarrow 0$
- 4 **enquanto** \langle Critério de parada global não satisfeito \rangle **faça**
- 5 $T_{j+1} \leftarrow T_j \times \alpha$
- 6 **enquanto** *Critério de parada local não satisfeita* **faça**
- 7 $k \leftarrow$ Seleção aleatória da variável a ser modificada \rangle
- 8 **se** *a variável de decisão selecionada é discreta* **então**
- 9 \langle Modifica a variável discreta por meio do operador escolhido \rangle
- 10 **senão**
- 11 $X_n \leftarrow$ \langle perturbação da solução para variável contínuo (3.6) \rangle
- 12 **fim**
- 13 $\Delta E = F(X_n) - F(X_a)$
- 14 **se** $\Delta E \leq 0$ **então**
- 15 $X_a \leftarrow X_n$
- 16 **se** $C_k > 1$ **então**
- 17 $C_k \leftarrow$ \langle Realimentação positiva \rangle
- 18 **fim**
- 19 **senão**
- 20 **se** $random(0, 1) < e^{-\frac{\Delta E}{k_b T}}$ **então**
- 21 $X_a \leftarrow X_n$
- 22 **se** $C_k > 1$ **então**
- 23 $C_k \leftarrow$ \langle Realimentação positiva \rangle
- 24 **fim**
- 25 **senão**
- 26 X_n é rejeitada
- 27 **se** $C_k < C_{\max}$ **então**
- 28 $C_k \leftarrow$ \langle Realimentação negativa \rangle
- 29 **fim**
- 30 **fim**
- 31 **fim**
- 32 $i \leftarrow i + 1$
- 33 **fim**
- 34 **fim**

Tabela 2 – Estratégias da heurística de cristalização.

Estratégia	Resposta Positiva		Resposta Negativa
	Fase de Exploração	Fase de Refinamento	
I	$C_k \leftarrow 1$	$C_k \leftarrow 1$	$C_k \leftarrow C_k + 1$
II	$C_k \leftarrow C_k/2$	$C_k \leftarrow C_k/2$	$C_k \leftarrow C_k + 1$
III	$C_k \leftarrow C_k - 1$	$C_k \leftarrow C_k - 1$	$C_k \leftarrow C_k + 1$
IV	$C_k \leftarrow 1$	$C_k \leftarrow C_k - 3$	$C_k \leftarrow C_k + 1$

Fonte: Adaptado de Duran et al. (2021).

onde C_{\max} é o limite máximo do fator de cristalização C_k . No novo algoritmo, a rotina para gerar a nova solução dependerá do valor de C_k . Quando o fator C_k for menor que C_{\max} , o algoritmo utilizará a distribuição de Bates, análogo ao Algoritmo 2. Se o C_k for maior que C_{\max} , será utilizada a distribuição gaussiana, onde, o valor do expoente da (3.10) é negativo. Apesar dos valores da distribuição gaussiana possuírem limites entre $-\infty$ e ∞ , a probabilidade de serem valores pequenos será maior devido aos valores de desvio padrão pequenos (DURAN et al., 2021). O algoritmo de geração da nova solução é apresentado no Algoritmo 3.

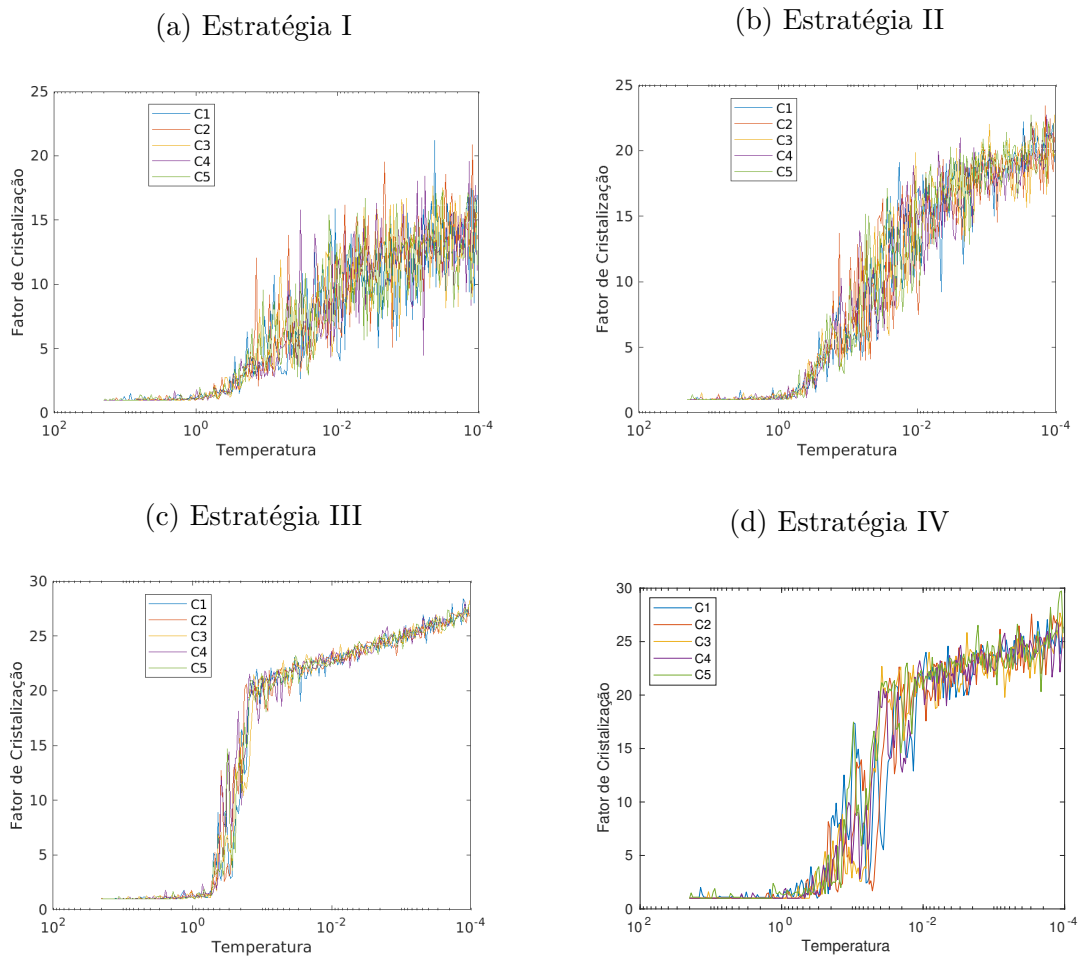
Algoritmo 3: GERAÇÃO DA NOVA SOLUÇÃO DA PROPOSTA DE Duran et al. (2021)

```

1 enquanto As restrições não seja satisfeita faça
2   repita
3      $x_n \leftarrow x_a$ 
4      $k \leftarrow$  seleciona aleatoriamente a variável de decisão a ser modificado
5     se  $C_k \leq C_{\max}$  então
6        $x_{n,k} \leftarrow x_{a,k} + \frac{1}{C_k} \sum_1^{C_k} \text{random}(-1, 1) \cdot \Delta r_k \cdot e_k$ 
7     senão
8        $x_{n,k} \leftarrow x_{a,k} + \text{GaussianDist}(\exp(C_{\max} - C_k - 2)) \cdot \Delta r_k \cdot e_k$ 
9     fim
10  até  $x \in [x_{\min}, x_{\max}]$ ;
11 fim
```

A título de exemplo apresentamos o comportamento do fator de cristalização ao longo do processo de otimização na Fig. 14. Na figura é apresentada a média do valor de cristalização de cada variável em relação à temperatura. Nas figuras podemos ver as diferenças descritas no comportamento do fator de cristalização e a mudança da fase de exploração para o refinamento.

Figura 14 – Comportamento do Fator de cristalização para diferentes estratégias de feedback



Fonte: o Autor.

3.6 Considerações Finais

O SA é conhecido por ser flexível, e ao longo dos anos diversas propostas de heurísticas e estratégias foram apresentadas. A maioria das propostas são algoritmos híbridos as quais combinam as características do SA com outros algoritmos. Como exemplo podemos citar o algoritmo híbrido de SA com busca local (PEDAMALLU; OZDAMAR, 2008), algoritmo híbrido entre a busca de tabu com SA (KATSIKIANNIS; GEORGILAKIS; KARAPIDAKIS, 2012) e algoritmo híbrido entre SA e algoritmo imunológicos (YILDIZ, 2009). Segundo Alnowibet et al. (2022), a maioria das propostas de heurísticas são relacionadas à geração da nova solução, mecanismo de resfriamento, critério de aceitação, critério de parada, loop interno e loop externo. Porém, não encontramos na literatura heurísticas com feedback como a do SA com a heurística de cristalização que possui estratégias de feedback.

4 Recozimento Simulado Multiobjetivo

O SA mono-objetivo é um algoritmo robusto e aplicado em diversos problemas com sucesso. O SA foi estendido à otimização de múltiplos objetivos, e neste capítulo será apresentado o SA multiobjetivo e heurísticas de cristalização para a utilização em multiobjetivo. Inicialmente será apresentada uma revisão bibliográfica e nas seções subsequentes trataremos dos algoritmos SA multiobjetivo AMOSA e CoAnnealing. Ao final do capítulo apresentaremos uma análise de desempenho dos algoritmos.

4.1 Introdução a SA Multiobjetivo

A maioria dos problemas de otimização reais são problemas multiobjetivos, porém a solução desses problemas são difíceis e muitas vezes as funções objetivas são conflitantes. Ao longo dos anos foram propostos diferentes métodos de otimização multiobjetivo, muitos baseados em algoritmos evolucionários. Inicialmente, alguns autores utilizavam algumas estratégias para a otimização de problemas multiobjetivo com algoritmo baseado em SA. Dentre estas estratégias utilizadas, destacamos as seguintes estratégias:

- (a) função de peso ou combinação das funções em uma única função;
- (b) conversão de uma ou mais função objetivo em restrição;
- (c) ou aplicação do algoritmo multiobjetivo (MOSA do inglês *Multi Objective Simulated Annealing*) como o AMOSA e CoAnnealing.

Nas duas primeiras estratégias é utilizado o algoritmo SA mono-objetivo para resolver os problemas multiobjetivo. Portanto, transformamos um problema de otimização multiobjetivo em um problema mono-objetivo. Este procedimento apresenta algumas ineficiências em aplicações práticas e Czyżżak e Jaskiewicz (1998) foram os primeiros a apresentar um algoritmo de SA para problemas de otimização multiobjetivo, o algoritmo foi nomeado por eles como *Pareto Simulated Annealing* (PSA). Esse algoritmo aproveitou diversas ideias do SA, como o conceito de vizinhança, o critério de aceitação, a dependência ao parâmetro temperatura e o esquema de resfriamento. O PSA ainda introduziu o conceito de amostra ou população, que interagem em cada iteração. Este conceito de população é o mesmo utilizado no algoritmo genético de Goldberg (1988).

No algoritmo proposto por Czyżżak e Jaskiewicz (1998), as soluções não-dominadas encontradas são armazenadas e utilizadas no critério de aceite da nova solução. O algoritmo também utiliza a estratégia de função peso, mas com um sistema de controle, aumentando

e diminuindo a probabilidade de ser aceita a nova solução. A otimização de problemas multiobjetivo utilizando a estratégia da função peso é apresentada em Ulungu et al. (1999). Neste mesmo trabalho, Ulungu et al. (1999) apresentou uma proposta MOSA, com um novo critério de aceite das novas soluções baseada na função peso.

Um algoritmo SA multiobjetivo baseado nos algoritmos anteriores foi proposto por Bandyopadhyay et al. (2008) e denominado *Archived Multiobjective Simulated Annealing (AMOSA)*, incorporando o conceito de *Archive*. Esse conceito é definido como um arquivo, onde são armazenadas as soluções não dominadas, ou seja, as que pertencem à frente de Pareto de acordo com o critério do algoritmo (BANDYOPADHYAY et al., 2008). Esse algoritmo apresentou um desempenho muito superior aos anteriores e aos algoritmos genéticos consolidados na literatura como NSGA-II proposto por Deb et al. (2002a), e PAES proposto por Knowles e Corne (2000) (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). Mais sobre este algoritmo será apresentado na Seção 4.2.

Outro algoritmo MOSA baseado no algoritmo AMOSA foi proposto por Martins e Tsuzuki (2015), esse novo algoritmo foi denominado de *CoAnnealing*. Este algoritmo aproveita a ideia do conceito de dominância, porém traz um critério de aceitação de nova solução mais simples. O algoritmo CoAnnealing foi inicialmente proposto para a utilização em regularização de imagem em tomografia computadorizada. Neste trabalho foi implementado o algoritmo CoAnnealing e mais detalhes será descrito na Seção 4.3.

4.2 AMOSA

Como já introduzido, o algoritmo AMOSA foi baseado nos algoritmos MOSA encontrados na literatura. Este algoritmo possui uma análise complexa para aceitação da nova solução, apresentando diversos casos para aceitar ou não a nova solução. Apesar dessa complexidade, este algoritmo permite obter um conjunto de soluções ótimas de forma satisfatória e apresentou um bom desempenho comparado aos algoritmos multiobjetivos evolucionários. Nesta seção são descritos com mais detalhes os procedimentos deste algoritmo.

O conceito de *Archive* utilizado neste algoritmo consiste em armazenar as soluções não-dominantes, ou seja, as soluções que pertencem à frente de Pareto. No AMOSA, ao passo que o algoritmo obtém uma nova solução que pertence à frente de Pareto, ela é armazenada no *Archive*. Portanto, a quantidade de soluções no *Archive* é crescente e para evitar o aumento excessivo de soluções, o algoritmo AMOSA estabelece dois limites: um limite máximo de soluções após a clusterização, chamado HL (do inglês *hard limit*) e um segundo limite máximo SL (do inglês *Soft limit*) e ao atingir este limite é realizado a clusterização até alcançar o limite HL. A quantidade de soluções é reduzida por meio do algoritmo de clusterização, aglutinando soluções próximas no domínio das variáveis

de decisão. O algoritmo original de Bandyopadhyay et al. (2008) utiliza o método de clusterização *Single-linkage algorithm* para reduzir a quantidade de soluções para o limite inferior. Uma revisão bibliográfica sobre o método de clusterização pode ser encontrada em Murtagh (1983).

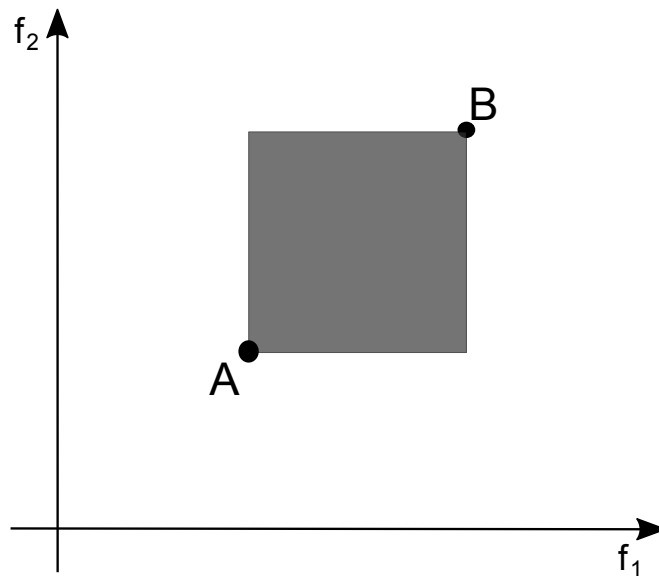
Bandyopadhyay et al. (2008) introduziu um novo conceito conhecido como intensidade de dominância (*amount of dominance*) no algoritmo AMOSA, e é utilizado para computar a probabilidade de aceite da nova solução. Este conceito é dado por

$$\Delta dom_{A,B} = \prod_{i=1, f_i(A) \neq f_i(B)}^M \frac{|f_i(A) - f_i(B)|}{R_i}, \quad (4.1)$$

em que M é o número de funções objetivas e R_i é a diferença do intervalo. Em relação ao R_i , Bandyopadhyay et al. (2008) não especifica com muitos detalhes em seu trabalho, e assumimos como a diferença entre o maior e o menor valor da função, $f_i(x)$. Logo R_i , em princípio, é atualizado a cada iteração i . No código na linguagem C, disponibilizado por Bandyopadhyay et al. (2008) em seu sítio eletrônico ¹ é inserido pelo usuário um valor para cada função objetivo.

A intensidade de dominância é definida como a área formada entre as soluções A e B , como mostrado na Fig. 15. Na figura é apresentada a solução A que domina a solução B , e a área formada entre estas duas soluções representa a intensidade de dominância entre as soluções A e B .

Figura 15 – A quantidade total de dominância entre duas soluções A e B , consiste na região do retângulo.



Fonte: Adaptado de (BANDYOPADHYAY et al., 2008)

¹ O código do AMOSA em C pode ser acessado em <https://www.isical.ac.in/~sriparna_r/>

O critério de aceitação do algoritmo AMOSA é um processo complexo e dividido em três casos diferentes, a saber:

- **Caso 1:** a solução corrente domina a solução nova;
- **Caso 2:** a solução corrente e a nova são não dominantes;
- **Caso 3:** a solução corrente é dominada pela nova solução.

Estes três casos de aceitação da nova solução serão descritos abaixo. E se a solução nova não satisfizer nenhum dos casos descritos, serão descartadas e uma nova solução é gerada pela perturbação.

- **Caso 1:** A nova solução é dominada pela solução atual, ou seja, a solução nova é pior que a solução atual. Neste caso a solução atual domina a solução nova, mas a solução nova ainda poderá ser aceita se um número aleatório, p , entre 0 e 1 for menor que a probabilidade determinada por

$$prob = \frac{1}{1 + \exp(\Delta dom_{avg} temp)}, \quad (4.2)$$

onde $temp$ é a temperatura de resfriamento do algoritmo de recozimento simulado e Δdom_{avg} é a intensidade de dominância média, calculada por

$$\Delta dom_{avg} = \frac{\sum_{i=1}^k \Delta dom_{i, nova\ solução} + (\Delta dom_{solução\ corrente, nova\ solução})}{(k + 1)}. \quad (4.3)$$

A intensidade de dominância média, calculada pela (4.3) é a média entre a intensidade de dominância, de k soluções do *Archive* e da nova solução, com a intensidade de dominância, da solução atual com a nova solução. O valor k é a quantidade de soluções dentro do *Archive*.

Neste caso 1, se a solução nova é não dominante com as soluções do *Archive*, exceto a solução atual se ela pertencer ao *archive*, ou é dominada apenas por algumas soluções do *archive* a probabilidade de ser aceita é maior.

- **Caso 2:** Se a solução atual e a solução nova são não-dominantes, a nova solução será adicionada no *Archive*, nas seguintes condições:
 - a) no caso onde a solução nova é dominada por soluções que pertencem ao *Archive*, esta nova solução será somente aceita de acordo com a probabilidade da (4.2);
 - b) se a solução nova e as soluções do *Archive* são não dominantes, a nova solução será aceita e adicionada ao *Archive*. Caso o tamanho de *Archive* seja maior que o limite superior, ocorre a clusterização do mesmo. O processo de clusterização

realiza a aglutinação de soluções muito próximas no domínio da variável de decisão por meio do algoritmo *single-linkage*;

- c) no caso da solução nova dominar *Archive*, em $k > 1$ soluções, a solução nova se torna a solução corrente e é adicionada ao *Archive* e as soluções dominadas são removidas.
- **Caso 3:** Quando a solução nova domina a solução corrente, também devemos verificar se a solução nova domina as soluções do *Archive*. Portanto, teremos mais três condições em relação à nova solução:
 - a) se a solução nova domina $k > 1$ soluções do *Archive*, a solução nova é atribuída e adicionada no *Archive* de acordo com o critério da probabilidade, (4.2);
 - b) no caso da nova solução e as soluções do *Archive* serem não-dominantes, a solução nova é atribuída à solução atual e adicionada ao *Archive*. Se o tamanho de *Archive* for maior que o limite superior, é realizada a clusterização;
 - c) se alguma solução do *archive* for dominada pela solução nova, a solução nova é atribuída a solução corrente e todas as soluções do *archive* dominadas são removidas.

A nova solução no algoritmo AMOSA é gerada a partir da solução atual por meio de um perturbação. Esta perturbação é gerada em apenas uma das variáveis escolhida aleatoriamente utilizando a distribuição de Laplace. A equação para a busca na vizinhança, dada por

$$x_{j,k} = x_{i,k} + \text{sin}al(rand) \times b \times \log(1 - 2|rand|), \quad (4.4)$$

onde no termo $\text{sin}al(rand)$ é atribuído o valor de 1 ou -1 , dependendo se o valor aleatório $rand$ for positivo ou negativo respectivamente. O valor de $rand$ é gerado aleatoriamente entre $1/2$ até $-1/2$. O método é uma forma simples de busca aleatória na vizinhança, e funciona de forma satisfatória.

O pseudocódigo AMOSA é apresentado na Algoritmo 4, retirada de Bandyopadhyay et al. (2008). O corpo principal do algoritmo funciona da seguinte forma:

1. o algoritmo inicializa atribuindo os limites superior e inferior do *Archive*, a temperatura máxima e mínima, o fator decaimento, α e número de iteração, N ;
2. é inicializado o *Archive*, preenchendo com soluções aleatórias. Durante o processo de inicialização é realizada a técnica *Hill-climbing*, que consiste em aceitar uma nova solução apenas se esta dominar a solução anterior;
3. uma solução aleatória pertencente ao *Archive* é selecionada e atribuída à solução atual, *current-pt*;

4. uma nova solução, *new-pt*, é gerada pela perturbação da solução atual, *current-pt*;
5. verificação do estado de dominância entre a solução nova e a solução atual, baseado no estado de dominância, é determinado em qual caso o algoritmo seguirá;
6. o processo 3-4 é repetido *Numiter* vezes, após a repetição é realizado o decaimento da temperatura e é novamente repetido o processo até a temperatura atingir a T_{\min} .

Como podemos observar, no Algoritmo 4 o mecanismo de decaimento da temperatura utilizado pelo AMOSA original é o decaimento geométrico, o mesmo utilizado no algoritmo recozimento original mono-objetivo. Bandyopadhyay et al. (2008) citam a utilização de outros mecanismos de decaimento da temperatura, porém não apresentam melhorias significativas no desempenho do algoritmo, apesar da necessidade de estudos mais profundos relacionados ao assunto.

O algoritmo AMOSA apresentou desempenho satisfatório em comparação aos métodos evolucionários PAES e NSGA-II. Os testes foram realizados nas principais funções multiobjetivo de *Benchmark*, como ZDT1, ZDT2, ZDT6, entre outras, mais detalhes dos resultados podem ser verificado em Bandyopadhyay et al. (2008).

Algoritmo 4: Pseudocódigo AMOSA

Entrada: T_{\max} : Temperatura máxima, T_{\min} : Temperatura mínima, HL : Limite Hard limit, SL : Soft limit , /* limite para iniciar o processo de clusterização */ α : coeficiente de decaimento da temperatura, ;

1 ;

2 $iter$: número de iterações3 $temp = T_{\max}$

4 Inicializar Archive

5 $current - pt \leftarrow random(Archive)$ /* Seleciona aleatoriamente do Archive */6 **enquanto** ($temp > T_{\min}$) **faça**7 **para** ($i = 0; i < iter; i ++$) **faça**8 $new - pt \leftarrow$ perturbação($current - pt$)

9 Verifica a dominância entre a solução corrente e a nova solução

10 **se** Solução corrente domina a nova solução **então** /* Caso 1 */

$$11 \quad \Delta dom_{avg} = \frac{\left(\sum_{i=1}^k \Delta dom_{i,new-pt}\right) + \Delta dom_{current-pt,new-pt}}{(k+1)}$$

/* k = quantidade de soluções do Archive que domina a nova solução, $k \geq 0$. */

$$12 \quad prob = \frac{1}{1 + \exp(\Delta dom_{avg} \times temp)}$$

13 **se** $random(0,1) < prob$ **então**14 $current - pt \leftarrow new - pt$ 15 **se** a nova solução e corrente são não-dominantes **então** /* Caso 2 */

16 Verifica a dominância da nova solução e dos pontos contido no Archive.

17 **se** nova solução é dominada por k ($k \geq 1$) soluções do Archive **então**

/* Caso 2 (a) */

$$18 \quad prob = \frac{1}{1 + \exp(\Delta dom_{avg} \times temp)}$$

$$19 \quad \Delta dom_{avg} = \frac{\left(\sum_{i=1}^k \Delta dom_{i,new-pt}\right)}{k}$$

20 Aceita a nova solução com a probabilidade = prob

21 **se** Nova solução é não-dominante com todas as soluções do Archive**então** /* Caso 2 (b) */22 $current - pt \leftarrow new - pt$

23 Adiciona a nova solução ao Archive

24 **se** Tamanho do Archive $> SL$ **então**

25 Clusterize o Archive até HL clusteres

26 **se** Nova solução domina k ($k \geq 1$), soluções do Archive **então** /* Caso 2 (c) */27 $current - pt \leftarrow new - pt$

28 Adicione a nova solução ao Archive

29 Remova todas as k soluções dominadas do Archive

Algoritmo 5: Continuação do Pseudocódigo AMOSA

```

26          /* Continuação do algoritmo AMOSA (Algoritmo 4) */
27
28     se a nova solução domina a solução atual então          /* Caso 3 */
29         Verifica a dominância entre a solução nova e as soluções do Archive.
30     se a nova solução é dominada por  $k$  ( $k \geq 1$ ) soluções do Archive então
31         /* Caso 3 (a) */
32          $\Delta dom_{\min}$  = valor mínimo da diferença de dominância entre a nova
33         solução e  $k$  pontos.
34          $prob = \frac{1}{1+exp(-\Delta dom_{\min})}$ 
35         Aceita a nova solução com a probabilidade = prob
36         Se não o valor da solução corrente é atribuído na nova solução
37     se A nova solução é não dominante com as soluções do Archive então
38         /* Caso 3 (b) */
39          $current - pt \leftarrow new - pt$ 
40         se  $current - pt \in Archive$  então
41             Remove a solução  $current - pt$  do Archive
42         senão se quantidade de soluções do Archive  $> SL$  então
43             Clusterize o Archive até HL cluster
44         fim
45     se nova solução domina  $k$  soluções do Archive então /* Caso 3 (c)
46         */
47          $current - pt \leftarrow new - pt$  e adiciona no Archive
48         Remova todas as  $k$  soluções dominadas do Archive.
49     fim
50      $temp \leftarrow \alpha \times temp$ 
51     fim
52     se quantidade de soluções no Archive  $> SL$  então
53     Clusterize o Archive até HL cluster

```

4.3 CoAnnealing

Uma nova proposta de algoritmo multiobjetivo de SA foi proposto por Martins e Tsuzuki (2015). Este algoritmo foi baseado no algoritmo AMOSA, aproveitando alguns conceitos como a utilização do *Archive*, a clusterização do *Archive* e a intensidade de dominância do AMOSA, descrito na (4.5), com algumas considerações. Este novo algoritmo proposto introduz algumas simplificações na análise de aceitação da nova solução em relação ao algoritmo AMOSA, considerado complexo. A intensidade de dominância proposto por Martins e Tsuzuki (2015) pode ser determinada por

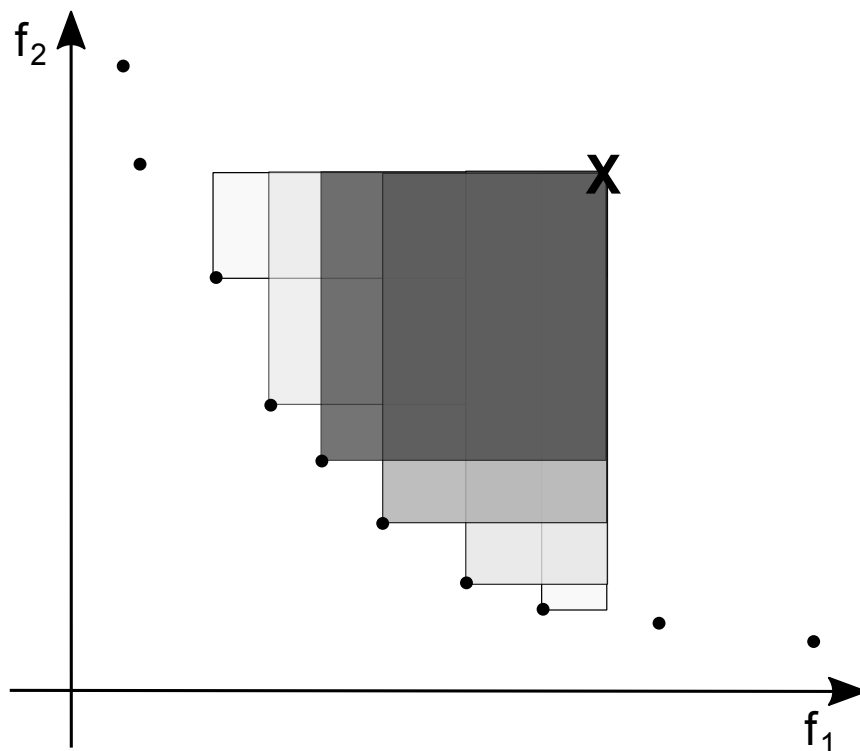
$$\Delta D_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^M (f_i(a) - f_i(b)) R_i \quad (4.5)$$

$$f_i(a) > f_i(b) \forall i.$$

No SA a nova solução é aceita conforme o estado de energia e a distribuição de

Boltzman, como descrito no Capítulo 3. No algoritmo *CoAnnealing*, o estado de energia é determinado pela maior intensidade de dominância entre uma solução em relação a todas as soluções do *Archive*. A intensidade de dominância entre as soluções do *Archive* e uma solução é demonstrada na Fig. 16, área entre as soluções e a solução dominada representa a intensidade de dominância. Quando uma solução pertence à frente de Pareto atual, o estado de energia será igual a zero (MARTINS; TSUZUKI, 2015). No *CoAnnealing*, quando uma nova solução possui o estado de energia menor ou igual a zero, é adicionado ao *Archive*, portanto o mesmo cresce de tamanho de forma arbitrária igual ao AMOSA.

Figura 16 – Estado de Energia e intensidade de dominância utilizado no CoAnnealing



Fonte: Adaptado de Martins e Tsuzuki (2015).

O tamanho do *Archive* é controlado por meio da clusterização do mesmo, da mesma forma que o AMOSA. Porém, o algoritmo de clusterização desse algoritmo proposto por (MARTINS; TSUZUKI, 2015) é mais complexo em relação ao AMOSA. Na clusterização, as soluções são classificadas em 4 categorias, em ordem de prioridade de preservação:

1. as soluções das extremidades da frente de Pareto (maior f_2 e menor f_1);
2. as soluções que pertencem à casca da frente de Pareto;
3. soluções que pertencem a um *cluster* local da casca da frente de Pareto;
4. as soluções remanescentes.

Como podemos observar, as soluções das extremidades sempre serão conservadas, isso para manter as soluções diversificadas na frente de Pareto. O algoritmo de clusterização também é baseado no método *single-linkage*, e segue a hierarquia de preservação de alguns conjuntos de soluções de acordo com a classificação acima. Neste processo de clusterização pode ocorrer a remoção da solução atual, no algoritmo AMOSA não fica claro como lidar. No *CoAnnealing*, quando isso ocorre, ele é re-adicionado no *Archive* e contabilizado na variável r , quando o algoritmo insistir em remover esta solução ela é descartada e é escolhida aleatoriamente outra solução do *Archive* (MARTINS; TSUZUKI, 2015).

O algoritmo CoAnnealing é apresentado no Algoritmo 6, e seu procedimento é descrito abaixo:

1. inicializa as variáveis: temperatura, temperatura máxima e mínima;
2. inicializa o *Archive*, linha 1;
3. uma solução é escolhida aleatoriamente do *Archive*;
4. é atribuída a nova temperatura, conforme o mecanismo de resfriamento;
5. uma nova solução é gerada a partir da perturbação da solução atual;
6. é determinado o estado de energia da nova solução com a solução atual;
7. a nova solução será aceita em caso de satisfazer umas das duas condições, $\Delta E \leq 0$ ou $p < \exp -\frac{\Delta E}{t}$, caso seja aceita o algoritmo segue adiante, caso seja rejeitada, o algoritmo volta para a etapa 5;
8. em caso da nova solução aceita, é utilizado o critério da maior intensidade de dominância da nova solução em relação às soluções do *Archive*, no caso em que, $MaxDom(X_n, archive) \leq 0$, a nova solução pertence à frente de Pareto e será adicionada ao *Archive*;
9. verifica o tamanho do *Archive*, caso seja maior que o limite superior, é realizada a clusterização, após a clusterização é verificado se a solução corrente foi removida do *Archive*;
10. repetem-se as etapas 5 a 9 até atingir a quantidade de passos estabelecidos;
11. enquanto a temperatura for maior que a temperatura mínima estabelecida repetem-se os passos 4 ao 9.

Algoritmo 6: Recozimento simulado multiobjetivo CoAnnealing

Entrada: T_{\max} : Temperatura inicial, ;
 T_{\min} : Temperatura mínima
 N : Número de iterações, HL : Limite do Archive, SL : Limite máximo Archive,
 α : fator de decaimento da temperatura
 \max_r : Quantidade máxima permitida para readicionar solução ao archive

- 1 $Archive \leftarrow$ Solução inicial aleatório
- 2 $X_a \leftarrow$ Solução inicial aleatório do Archive
- 3 $r \leftarrow 0$
- 4 $T_j \leftarrow T_{\max}$
- 5 **enquanto** <Critério de parada global não satisfeito> **faça**
- 6 $T_{j+1} \leftarrow$
 $T_j \times \alpha$ /* próxima temperatura do programa de resfriamento */
- 7 **enquanto** Critério de parada local não satisfeita **faça**
- 8 $X_n \leftarrow$ perturbação da solução (X_a) /* equação (4.6) */
- 9 $\Delta E = MaxDom(X_n, Archive) - MaxDom(X_a, archive)$
- 10 **se** $\Delta E \leq 0 \vee Random(U(0, 1)) < \exp^{-\frac{\Delta E}{t}}$ **então**
- 11 $X_a \leftarrow X_n$
- 12 **se** $MaxDom(X_n, Archive) = 0$ **então**
- 13 X_a adiciona em Archive
- 14 **se** $tamanho(Archive) > Limite$ **então**
- 15 $Archive \leftarrow$ Clusterização(Archive)
- 16 **se** $x_n \notin Archive$ **então**
- 17 **se** $r < \max_r$ **então**
- 18 Adiciona X_n em Archive
- 19 $r \leftarrow r + 1$
- 20 **senão**
- 21 $X_n \leftarrow Random(Archive)$
- 22 **fim**
- 23 **senão**
- 24 $r \leftarrow 0$
- 25 **fim**
- 26 **fim**
- 27 **fim**
- 28 **senão**
- 29 <descarta a nova solução X_n >
- 30 **fim**
- 31 **fim**
- 32 **fim**

4.4 CoAnnealing com Heurística de Cristalização

Neste trabalho, aplicamos no CoAnnealing a heurística de cristalização descrita por Duran et al. (2021) e apresentada na Seção 3.4. Originalmente, a heurística de cristalização foi implementada no SA mono-objetivo e neste trabalho implementamos no CoAnnealing. No algoritmo utilizado neste trabalho, a geração da nova solução é realizada por meio da perturbação da solução atual utilizando os mesmos conceitos do algoritmo SA de vizinhança adaptativa. A nova solução para problemas contínuos é determinada utilizando a equação

$$X_{n,k} = X_{a,k} + \frac{1}{C_k} \sum_1^{C_k} random(-1, 1) \cdot \Delta r_k e_k, \quad (4.6)$$

onde, Δr_k é o tamanho máximo do passo e $random(-1, 1)$ é o valor aleatório entre -1 e 1 , a variável e_k é a direção. E r_k é o tamanho do passo da variável de decisão, no CoAnnealing o valor típico de r_k é 25 % da faixa de variação da variável. Lembrando que essa proposta é para as variáveis contínuas.

Aqui também aplicamos a heurística de Duran et al. (2021), o qual o fator de cristalização não se limita ao valor, C_{\max} aumentando de forma incremental sem limite. Quando o fator de cristalização C_k for menor que o valor de C_{\max} , a nova solução é determinada com a (4.6). Caso o fator de cristalização for maior que C_{\max} , a nova solução é determinada utilizando a (3.10), semelhante ao utilizado no Algoritmo 3.

Os procedimentos do CoAnnealing com heurística de cristalização é semelhante ao CoAnnealing discutido anteriormente, porém com pequenas adaptações para a heurística de cristalização. O procedimento do CoAnnealing com a heurística de cristalização é descrito abaixo:

1. inicializa as variáveis: temperatura, temperatura máxima e mínima;
2. inicializa o *Archive*;
3. inicializa o fator de cristalização com valor igual a 1 para todas as variáveis de decisão;
4. a solução atual é escolhida de forma aleatória do *Archive*;
5. é atribuída a nova temperatura, conforme o mecanismo de resfriamento;
6. é sorteada uma variável de decisão, i , para a perturbação para nova solução X_a ;
7. uma nova solução é gerada a partir da perturbação da variável de decisão selecionada da solução atual, para problemas contínuos é utilizado a (4.6);
8. é determinado o estado de energia da nova solução com a solução atual;

9. a nova solução será aceita em caso de satisfazer um das duas condições, $\Delta E \leq 0$ ou $p < \exp - \frac{\Delta E}{T_j}$;
10. em caso da nova solução ser aceita, o fator de cristalização da variável modificada no processo de geração da nova solução recebe um feedback positivo, e incrementa o contador de passos;
11. e é utilizado o critério da maior intensidade de dominância da nova solução em relação às soluções do *Archive*, no caso em que, $MaxDom(X_n, archive) \leq 0$, a nova solução pertence à frente de Pareto e será adicionada ao *Archive*;
12. caso a nova solução seja rejeitada, o fator de cristalização da variável de decisão modificada recebe um feedback de alimentação negativo e o algoritmo retorna para a etapa 6 e incrementa o contador de passos;
13. verifica o tamanho do *Archive*, caso seja maior que o limite superior, é realizada a clusterização, após a clusterização é verificado se a solução corrente foi removida do *Archive*;
14. repetem-se as etapas 6 a 9 até atingir a quantidade de passos estabelecidos;
15. enquanto a temperatura for maior que a temperatura mínima estabelecida repetem-se os passos 4 ao 9.

O CoAnnealing com a heurística de cristalização é apresentado no Algoritmo 7. Observamos que nesse algoritmo temos o fator de cristalização e a geração da nova solução, considerando o fator de cristalização.

4.5 CoAnnealing com Estratégia de Feedback

Neste algoritmo de CoAnnealing com a heurística de cristalização utilizamos as estratégias de feedback para ajuste da aceitação das soluções. As estratégias utilizadas são as mesmas apresentadas na Tabela 2 da Seção 3.5, que são as propostas de Duran et al. (2021) e Tavares, Martins e Tsuzuki (2011). As estratégias disponíveis são apresentadas na Tabela 2.

As estratégias de feedback ajustam as fases de exploração e refinamento, como já descrito, a estratégia I é utilizada em problemas que necessitam de maior exploração. A estratégia III é indicada para problemas que necessitam de um maior refinamento. A estratégia II é uma estratégia intermediária entre as estratégias I e III. As estratégias usuais são as estratégias I, II e III. A estratégia IV é uma estratégia híbrida e neste trabalho não foi utilizada. Estas estratégias foram utilizadas no CoAnnealing da mesma forma que foram utilizadas no SA mono-objetivo.

Algoritmo 7: Recozimento simulado multiobjetivo CoAnnealing com heurística de cristalização

Entrada: T_{\max} : Temperatura inicial, T_{\min} : Temperatura mínima,
 N : Número de iterações, HL : Limite do Archive, SL : Limite máximo Archive,
 α : fator de decaimento da temperatura,
 \max_r : Quantidade máxima permitida para readicionar solução ao archive

- 1 $Archive \leftarrow$ Solução inicial aleatório
- 2 $X_a \leftarrow$ Solução inicial aleatório
- 3 $r \leftarrow 0$
- 4 $T_j \leftarrow T_{\max}$
- 5 **enquanto** \langle Critério de parada global não satisfeito \rangle **faça**
- 6 $T_{j+1} \leftarrow T_j \times \alpha$ /* próxima temperatura de resfriamento */
- 7 **enquanto** Critério de parada local não satisfeita **faça**
- 8 $i \leftarrow$ Seleção aleatório da variável de decisão \rangle
- 9 **se** a variável de decisão selecionada for discreta **então**
- 10 \langle Modifica a variável discreta por meio do operador \rangle
- 11 **senão**
- 12 $X_n \leftarrow$ perturbação da solução para variável contínuo /* equação (4.6) */
- 13 **fim**
- 14 $\Delta E = MaxDom(X_n, Archive) - MaxDom(X_a, Archive)$
- 15 **se** $\Delta E \leq 0 \parallel Random(U(0,1)) < \exp^{-\frac{\Delta E}{T_j}}$ **então**
- 16 $X_a \leftarrow X_n$
- 17 **se** $C_i > 1$ **então**
- 18 $C_k \leftarrow$ Realimentação Positiva \rangle
- 19 **fim**
- 20 **se** $MaxDom(X_n, Archive) = 0$ **então**
- 21 X_a adiciona em Archive
- 22 **se** $tamanho(Archive) > Limite$ **então**
- 23 $Archive \leftarrow$ Clusterização(Archive)
- 24 **se** $X_n \notin Archive$ **então**
- 25 **se** $r < \max_r$ **então**
- 26 Adiciona X_n em Archive
- 27 $r \leftarrow r + 1$
- 28 **senão**
- 29 $X_n \leftarrow Random(Archive)$
- 30 **fim**
- 31 **senão**
- 32 $r \leftarrow 0$
- 33 **fim**
- 34 **fim**
- 35 **fim**
- 36 **senão**
- 37 $C_k \leftarrow$ Realimentação negativa \rangle
- 38 **fim**
- 39 **fim**
- 40 **fim**

4.6 Solucionando as Funções de Benchmark

Para validação do desempenho do algoritmo, implementamos os algoritmos Co-Annealing com a heurística de cristalização e AMOSA no Matlab[®], em um computador pessoal com o processador AMD[®] Ryzen 5 3600, 16 GB de memória RAM e sistema operacional Linux. Para validar o desempenho dos algoritmos, aplicamos as funções de benchmark encontradas na literatura.

As funções utilizadas são descritas na Tabela 3. As primeiras funções são as funções bi-dimensionais propostas por Zitzler, Deb e Thiele (2000), sendo uma frente de Pareto convexa, uma não-convexa, uma descontínua e uma não uniforme no domínio de busca. Na tabela também apresentamos as funções tri-dimensionais propostas por Deb et al. (2002b), estas funções resultam em frente de Pareto hiperplano, esfera, curva e uma frente de Pareto descontínua. As equações das funções objetivas, número de variáveis e o domínio de busca são apresentadas na Tabela 3.

Para validação também utilizamos algumas funções de benchmark com restrições para o algoritmo AMOSA e o algoritmo CoAnnealing com a heurística de cristalização. As funções escolhidas são apresentadas na Tabela 4. A função Binh and korn (BINH; KORN, 1997) é uma frente de Pareto convexa. A segunda é uma frente de Pareto contínua obtida com a função Chankong and Haimes (CHANKONG; HAIMES, 2008). A outra frente de Pareto é do tipo contínua e descontínua, esta é a função Constr-Ex (DEB, 2011). A última função com restrição é a TNK (TANAKA et al., 1995), a qual apresenta uma frente de Pareto do tipo descontínua. Estas funções foram selecionadas para avaliação da capacidade dos algoritmos solucionar problemas com restrições.

Os parâmetros utilizados para o algoritmo AMOSA foram baseados nos valores utilizados por Bandyopadhyay et al. (2008). Porém, em algumas funções foi necessário incrementar o número de iterações. Os parâmetros utilizados foram: $T_{\min} = 0.00000001$, $T_{\max} = 200$, $HL = 75$, $SL = 100$, $\alpha = 0,95$. Para o algoritmo CoAnnealing com heurística de cristalização, utilizamos os mesmos parâmetros utilizados no algoritmo AMOSA e o fator de cristalização máximo $C_{\max} = 20$. Como descrito, os valores da iteração para o algoritmo CoAnnealing e AMOSA são apresentadas na Tabela 5 e as estratégias da heurística de cristalização do CoAnnealing também são apresentadas na mesma tabela.

Devemos ressaltar que no caso da função DTLZ5 foi necessário aumentar o número de iteração em relação ao AMOSA para convergir de forma satisfatória. Os parâmetros utilizados não são ideais para CoAnnealing, sendo necessário realizar um estudo de sensibilidade dos parâmetros e estratégias. Além disso, nas funções com restrições utilizamos a estratégia I do fator de cristalização. Esta estratégia foi adotada no estudo exploratório e não foi realizado experimento com outras estratégias ou outros parâmetros.

Tabela 3 – As funções de *benchmark* utilizadas para validação do desempenho dos algoritmos.

Função	Equações	Domínio
ZDT1 (ZITZ- LER; THIELE, 2000)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{f_1(x)/g(x)}$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT2 (ZITZ- LER; THIELE, 2000)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - (f_1(x)/g(x))^2$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT3 (ZITZ- LER; THIELE, 2000)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1(x)/g(x))\sin(10\pi f_1)$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
ZDT6 (ZITZ- LER; THIELE, 2000)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - f_1(x)/g(x)^2$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 10$
DTLZ1 (DEB et al., 2002b)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - f_i/g^2$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 12$
DTLZ2 (DEB et al., 2002b)	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - f_i/g^2$ $f_2 = g(x)h(f_1(x), g(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 12$
DTLZ5 (DEB et al., 2002b)	$g(X_M) = \sum_{x_i \in X_M} (x_i - 0.5)^2$ $\theta_i = \frac{\pi}{4(1+g(X_M))} (1 + 2g(X_M)x_i)$ $f_1(x_i) = (1 + g(X_M))\cos(\theta_i\pi/2)\cos(\theta_i)$ $f_2(x_i) = (1 + g)\cos(\pi\theta/2)\sin(\theta_i)$ $f_3(x_i) = (1 + g)\sin(1/2\theta_0\pi)$	$0 \leq x_i \leq 1$ $i = 1, 2, \dots, n$
DTLZ7 (DEB et al., 2002b)	$g(x_M) = 1 + \frac{9}{ X_M } \sum_{x_i \in X_M} x_i$ $h(f_1, f_2, f_3, g) = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right]$ $f_1(x_1) = x_1$ $f_2(x_2) = x_2$ $f_3(x)(1 + g(x))h(f_1, f_2, f_3, g(x))$	$0 \leq x_i \leq 1$ $i = 1, 2, \dots, n$

Tabela 4 – As funções de benchmark com restrição utilizadas para validação do desempenho dos algoritmos

Funções		Equações	Domínio
Binh and Korn	função	$f_1(x, y) = 4x^2 + 4y^2$ $f_2(x, y) = (x - 5)^2 + (y - 5)^2$	$0 \leq x \leq 5$ $0 \leq y \leq 3$
	restrição	$g_1(x, y) = (x - 5)^2 + y^2 \leq 25$ $g_2(x, y) = (x - 8)^2 + (y + 3)^2 \geq 7, 7$	
Chankong and Haimes Functions	função	$f_1(x, y) = 2 + (x - 2)^2 + (y - 1)^2$ $f_2(x, y) = 9x - (y - 1)^2$	$-20 \leq x, y \leq 20$
	restrição	$g_1(x, y) = x^2 + y^2 \leq 225$ $g_2(x, y) = x - 3y + 10 \leq 0$	
Constr-Ex	função	$f_1(x, y) = x$ $f_2(x, y) = \frac{1+y}{x}$	$0.1 \leq x \leq 1$ $0 \leq y \leq 5$
	restrição	$g_1(x, y) = y + 9x \geq 6$ $g_2(x, y) = -y + 9x \geq 1$	
TNK	função	$f_1(x) = x_1$ $f_2(x) = x_2$	$0 \leq x_i \leq \pi$ $i = 1, 2$
	restrição	$g_1(x) = -x_1^2 - x_2^2 + 1 + 0,1\cos(16\arctan(x_1/x_2)) \leq 0$ $g_2(x) = (x_1 - 0,5)^2 + (x_2 - 0,5)^2 \leq 0,5$	

Fonte: O autor.

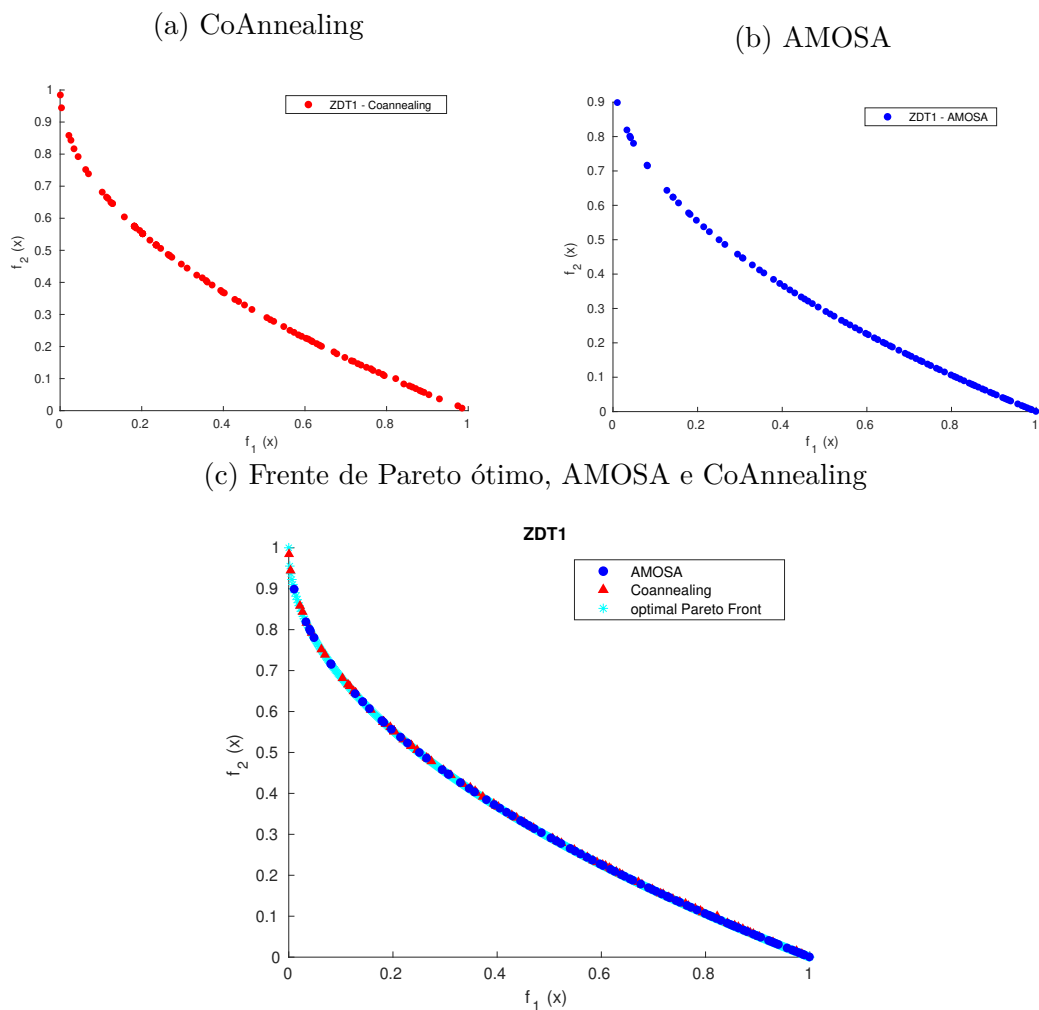
Tabela 5 – Tabela com os parâmetros de otimização utilizados

Problema	AMOSA		CoAnnealing
	N	N	Estratégia de cristalização
ZDT1	500	500	III
ZDT2	500	500	III
ZDT3	500	500	III
ZDT6	1000	1000	III
DTLZ1	5000	30000	III
DTLZ2	1000	1000	I
DTLZ5	1500	2000	III
DTLZ7	9000	9000	III
Constr-Ex	100	50	I
Binh and Korn	50	50	I
Chankong and Haimes	50	50	I
TNK	50	50	I

Fonte: O autor.

A frente de Pareto da função ZDT1 gerada pelos algoritmos CoAnnealing e AMOSA são apresentados na Fig. 17. Nesta figura apresentamos a frente de Pareto obtida pelo CoAnnealing na Fig. 17a, e a frente de Pareto do AMOSA na Fig. 17b. E para efeito de comparação apresentamos na Fig. 17c, a frente de Pareto ótima obtida com os dados de Coello et al. (2007) e a frente de Pareto obtida com AMOSA e CoAnnealing. Os resultados demonstram que tanto o algoritmo AMOSA e o CoAnnealing convergiram de forma satisfatória.

Figura 17 – Frente de Pareto da função ZDT1

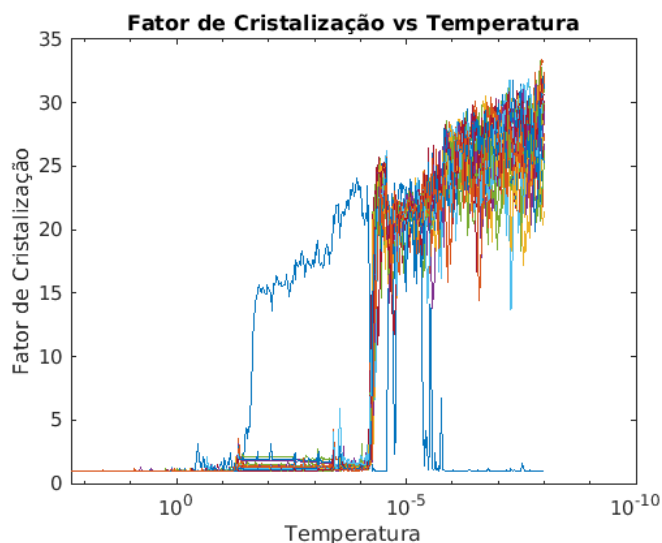


Fonte: O Autor.

Como já descrito anteriormente, neste problema foi utilizada a estratégia III para a heurística de cristalização, o comportamento do fator de cristalização é apresentado na Fig. 18. Na figura, cada variável de decisão possui seu próprio fator de cristalização e são apresentados com linha de cores diferentes. A variação dos fatores de cristalização aumenta com a diminuição da temperatura como esperado. Porém, observamos que há uma linha azul que reduz com a diminuição da temperatura, possivelmente essa linha é o fator de cristalização da variável decisão x_1 . Lembrando que neste problema ZDT1 a

função objetivo é $f_1(x) = x_1$, o que pode explicar o comportamento da linha x_i .

Figura 18 – Variação do fator de cristalização em relação à temperatura, função ZDT1



Fonte: O Autor.

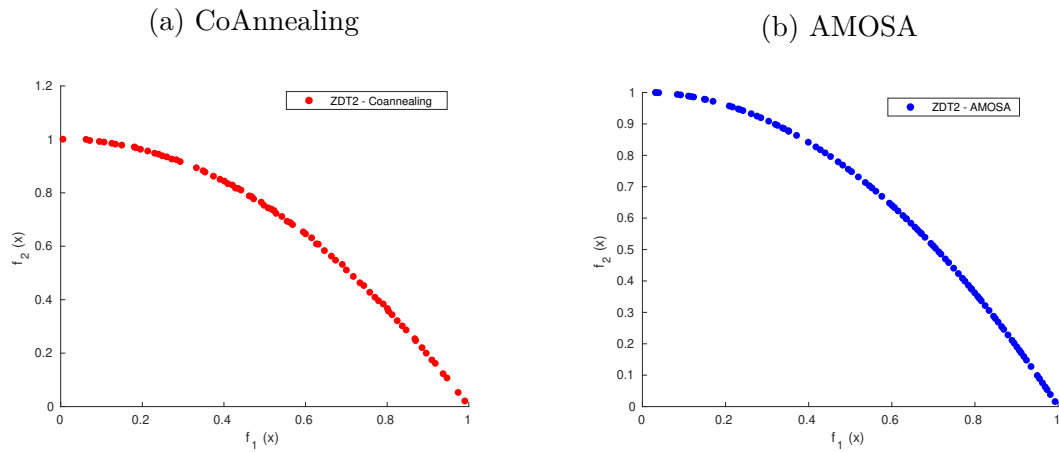
Os bons resultados também foram obtidos com as funções ZDT2 e ZDT3, conforme podemos observar nas Fig. 19 e Fig. 20. Ambos algoritmos obtiveram resultados satisfatórios nestas funções, obtendo uma frente de Pareto bem distribuída e próxima à frente de Pareto ótima.

Em relação à função ZDT6, o algoritmo CoAnnealing apresentou uma frente de Pareto com uma boa distribuição das soluções e convergência, como podemos verificar na Fig. 21c. Este resultado é melhor que o resultado apresentado pelo algoritmo AMOSA. O algoritmo AMOSA não conseguiu convergir, mesmo com o incremento do número de iterações, Custódio et al. (2011) obteve resultados semelhantes com o algoritmo AMOSA na função ZDT6.

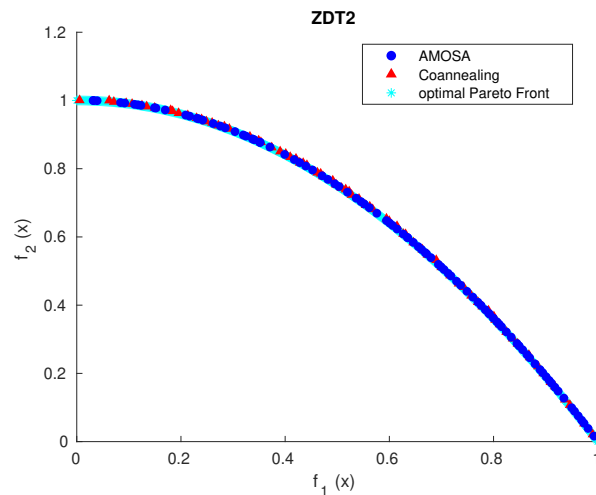
O fator de cristalização para as funções ZDT2 e ZDT3 apresentaram comportamento semelhante ao da Fig. 18. Isto é explicado por possuir a mesma função objetivo $f_1(x) = x_1$ em relação à função ZDT1. A função ZDT6 também obteve um comportamento semelhante, pois a função objetivo $f_1(x)$ é dependente apenas do x_1 .

Nas funções tri-dimensionais, o CoAnnealing e AMOSA obtiveram resultados satisfatórios, com exceção a função DTLZ1. Esta função se trata de uma frente de Pareto no formato hiperplano com $(11^k - 1)$ frente de Pareto locais ótimas. Os resultados do algoritmo AMOSA são apresentados na Fig. 22b e CoAnnealing na Fig. 22a. Mesmo com aumento do número de iteração o algoritmo não conseguiu convergir e os resultados indicam que a estratégia I, II e III não são adequados para esta função, sendo necessário um maior estudo desta função com outras estratégias do fator de cristalização e parâmetros.

Figura 19 – Frente de Pareto da função ZDT2



(c) Frente de Pareto ótimo, AMOSA e CoAnnealing



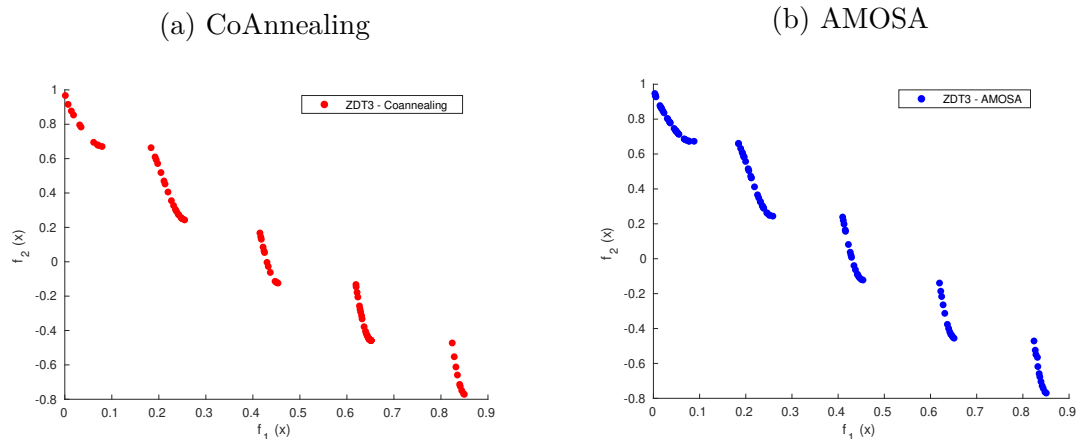
Fonte: O Autor.

Na função DTLZ2, ambos algoritmos apresentaram uma frente de Pareto muito próxima à frente de Pareto ótima, como podemos observar na Fig. 23c. Ambos os algoritmos convergiram e apresentaram uma boa distribuição das soluções na frente de Pareto em formato esférico.

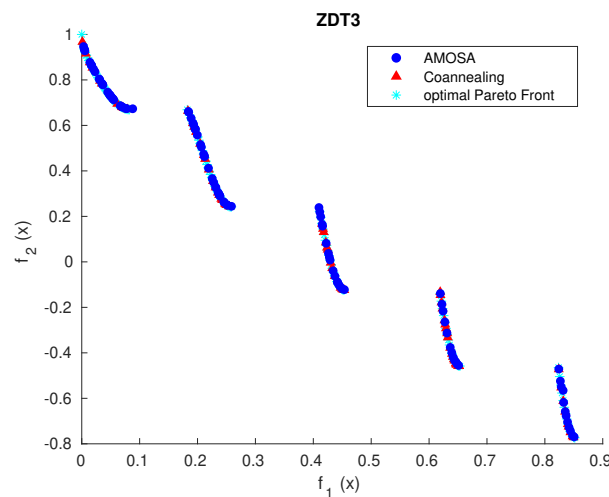
Como descrito anteriormente, a frente de Pareto da função DTLZ5 apresenta um formato de curva, e ambos os algoritmos conseguiram convergir. O algoritmo AMOSA apresenta uma frente de Pareto mais uniforme em relação ao do CoAnnealing. As frentes de Pareto obtidas pelos algoritmos são apresentadas nas Fig. 24a e Fig. 24b. A frente de Pareto ótima da função de DTLZ5 é apresentada na Fig. 24c em conjunto com a frente de Pareto do algoritmo AMOSA e CoAnnealing para fins de comparação.

A frente de Pareto da função DTLZ7 é apresentada na Fig. 25, com a frente de Pareto obtida pelo algoritmo CoAnnealing, AMOSA e a frente de Pareto ótima respectivamente. Com análise visual, ambos algoritmos convergiram para uma frente de Pareto próxima à

Figura 20 – Frente de Pareto da função ZDT3



(c) Frente de Pareto ótimo, CoAnnealing e AMOSA



Fonte: O autor.

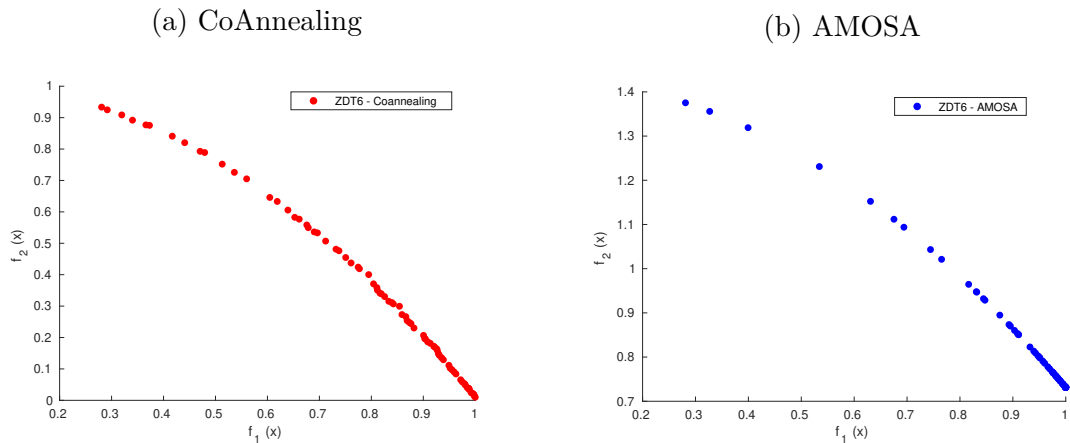
frente de Pareto ótima.

A função com restrição Binh and Korn é apresentada na Fig. 26 com a frente de Pareto obtida pelo CoAnnealing e AMOSA, respectivamente. Nas figuras apresentamos a frente de Pareto em conjunto com a região factível, ou seja a região possível de obter soluções que satisfaçam as restrições do problema. As soluções ótimas obtidas pelo CoAnnealing estão na cor azul, as soluções ótimas obtidas com o AMOSA estão na cor vermelha e as regiões com as soluções factíveis na cor ciano. Os resultados demonstram que ambos algoritmos conseguiram convergir e obter uma frente de Pareto na borda da região factível da função.

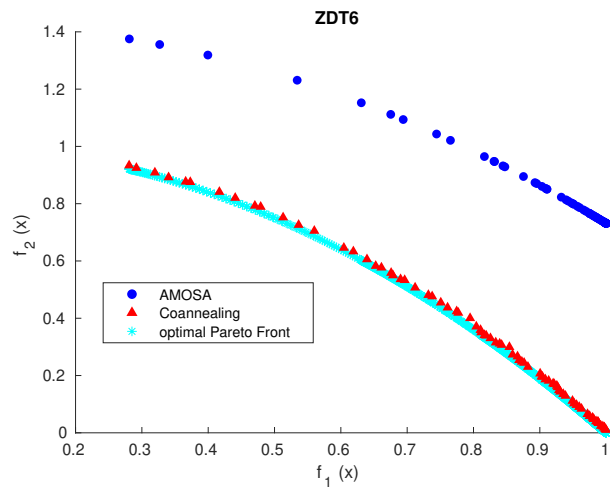
As frentes de Pareto obtidas pelo CoAnnealing e AMOSA da função Constr-Ex é apresentada em conjunto com a região factível na Fig. 27, respectivamente. Nessa função, ambos algoritmos conseguiram convergir conforme podemos observar na Fig. 27.

A função de Chankong e Haimes é apresentada na Fig. 28, com a frente de Pareto do CoAnnealing e AMOSA, respectivamente. E ambos novamente conseguiram convergir

Figura 21 – Frente de Pareto da função ZDT6



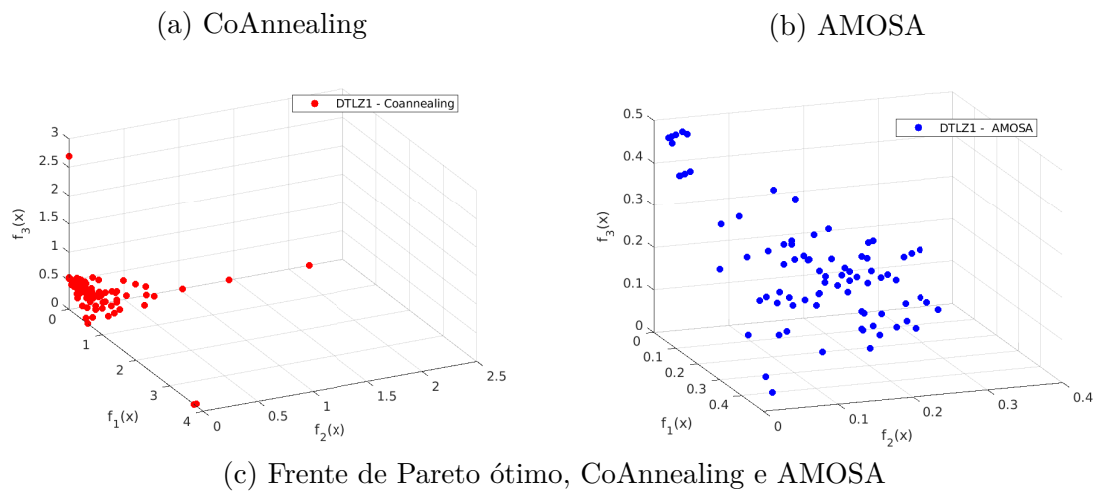
(c) Frente de Pareto ótimo, CoAnnealing e AMOSA



Fonte: O autor.

de forma satisfatória, porém, visualmente o algoritmo CoAnnealing apresentou uma frente de Pareto melhor distribuída na borda da região factível.

Figura 22 – Frente de Pareto da função DTLZ1



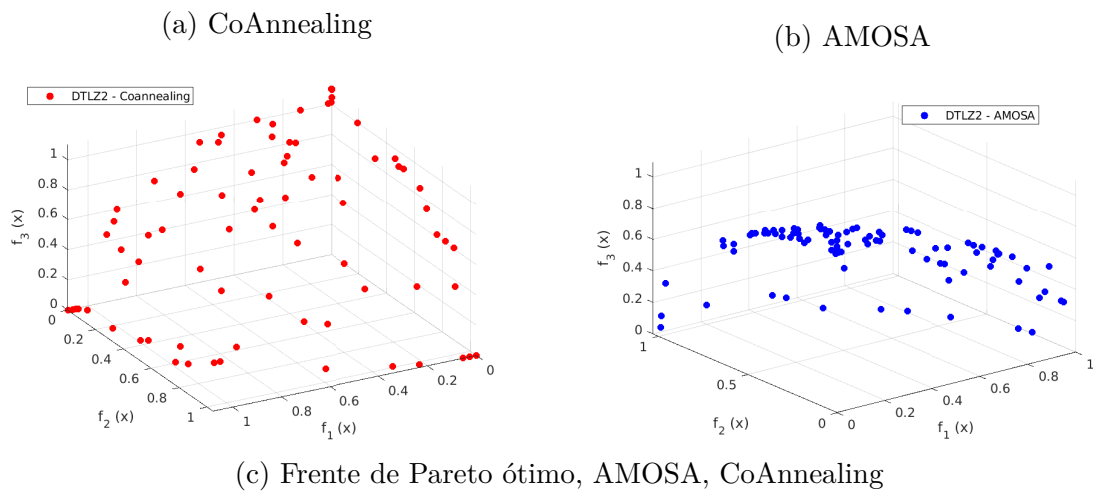
Fonte: O autor.

A frente de Pareto dos algoritmos CoAnnealing e AMOSA para a função TNK é apresentada na Fig. 29. Assim como nos anteriores, ambos algoritmos convergiram e obtiveram uma frente de Pareto satisfatória.

As figuras apresentadas da frente de Pareto obtidas pelos algoritmos permite apenas uma análise visual, ou seja, qualitativa. Para uma análise quantitativa é necessário avaliar os valores dos indicadores descritos na Subseção 2.2.2, os indicadores utilizados neste trabalho foram o *GD*, *Spacing* e *Hypervolume*. Além disso, também foi computado a quantidade de vezes que a função objetivo é avaliada durante o processo de otimização e o tempo de execução. Para mensurar o tempo de execução, foi utilizada a função *CPUTIME* do Matlab[®].

Para análise dos indicadores, o algoritmo foi aplicado nas funções de benchmark por 130 vezes. Os resultados apresentados na Tabela 6 são os valores médio e o desvio padrão dos 100 melhores resultados de cada função. Para as funções com restrição, não foi avaliado nenhum indicador, apenas a avaliação visual da frente de Pareto.

Figura 23 – Frente de Pareto da função DTLZ2



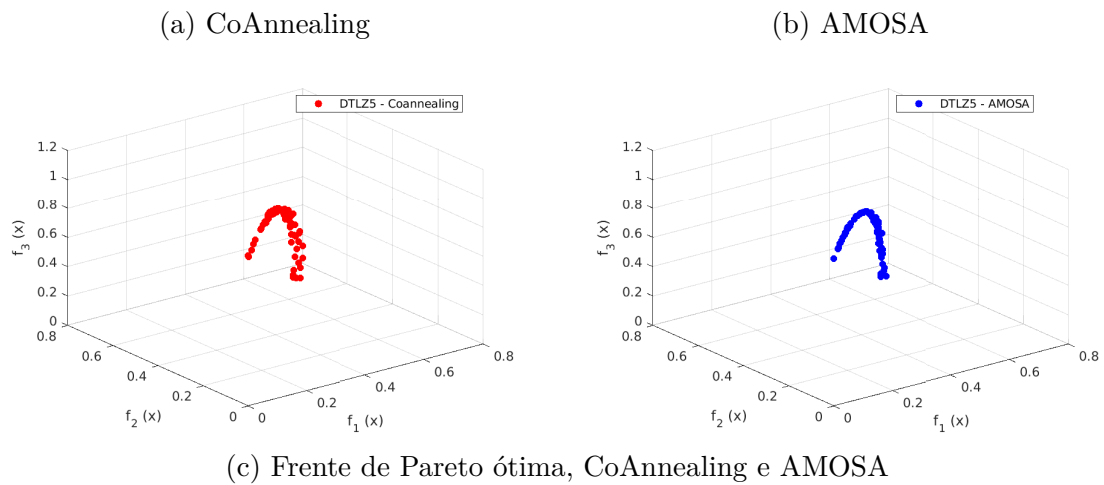
Fonte: O Autor.

Em relação aos valores do indicador GD, nas funções ZDT1 e ZDT2 os valores foram próximos para ambos os algoritmos, com desempenho melhor para o CoAnnealing. Já nas funções ZDT3, ZDT6, DTLZ2 e DTLZ7 o algoritmo CoAnnealing apresentou valores menores, portanto a convergência do CoAnnealing foi melhor que o AMOSA. Nas funções DTLZ1 e DTLZ5 o AMOSA apresentou uma convergência melhor em relação ao CoAnnealing. Lembrando que quanto menor os valores do indicador GD, melhor é a convergência da frente de Pareto. E em relação ao desvio padrão do indicador GD, observamos que, em geral, o CoAnnealing apresentou uma variação menor.

O segundo indicador é o *Spacing*, e como descrito anteriormente, quanto menor o valor do indicador, melhor a distribuição das soluções. Os resultados deste indicador demonstram que o CoAnnealing obteve frente de Pareto com uma melhor distribuição das soluções ao longo da frente de Pareto. Neste requisito, o algoritmo CoAnnealing foi inferior apenas nas funções ZDT3, DTLZ1 e DTLZ5.

Os bons resultados obtidos pelo CoAnnealing neste requisito pode ser explicado

Figura 24 – Frente de Pareto da função DTLZ5



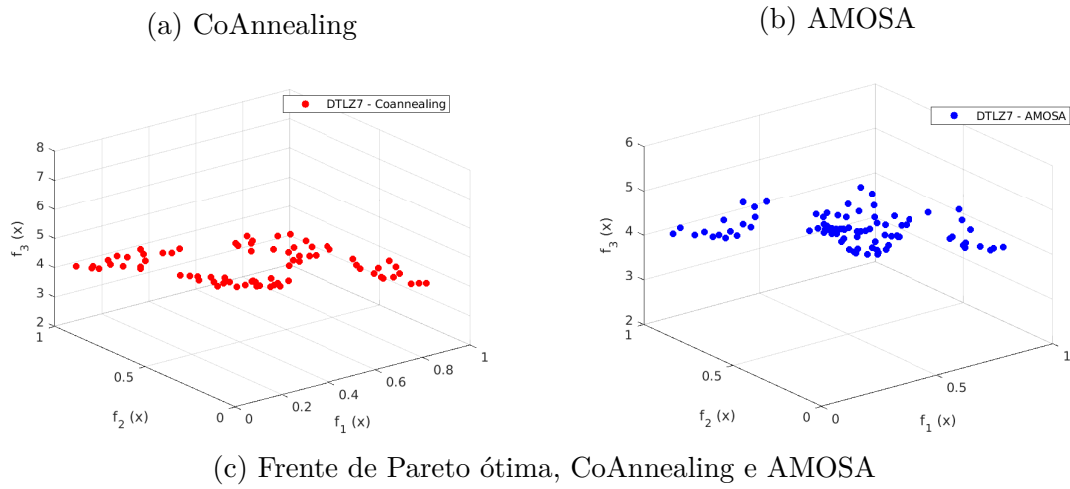
Fonte: O Autor.

pelo processo de controle do tamanho do *Archive*. Enquanto no AMOSA a clusterização aplicada é o algoritmo *Single-linkage*, no CoAnnealing possui uma hierarquia com prioridade de preservação das soluções, conforme detalhado na Seção 4.3.

Outro indicador importante é o Hipervolume, neste indicador é avaliada a convergência como a distribuição das soluções ao longo da frente de Pareto. Neste indicador, quanto maior o valor, melhor é o desempenho do algoritmo. Em relação ao ponto de referência, este deve ser uma solução dominada por todos os pontos pertencentes à frente de Pareto. Nos resultados obtidos, os valores do hipervolume entre os algoritmos AMOSA e CoAnnealing foram muito próximos.

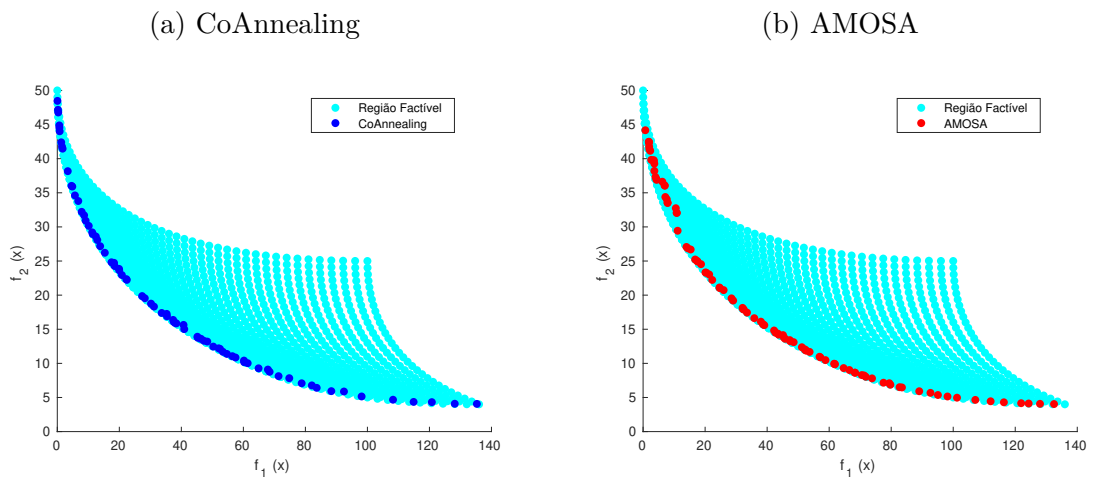
Avaliamos a quantidade de vezes que a função objetivo é avaliada durante o processo de otimização. Em aplicações com avaliação da função complexa ou que demanda grande capacidade de processamento, é vantajoso o algoritmo que converge com menos avaliações. Neste requisito, observamos que o CoAnnealing consegue convergir e obter resultados semelhantes ao AMOSA avaliando menos vezes as funções objetivas.

Figura 25 – Frente de Pareto da função DTLZ7



Fonte: O Autor.

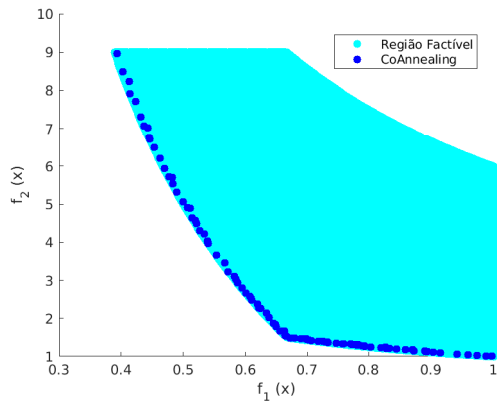
Figura 26 – Frente de Pareto da Função Binh and Korn de Binh e Korn (1997)



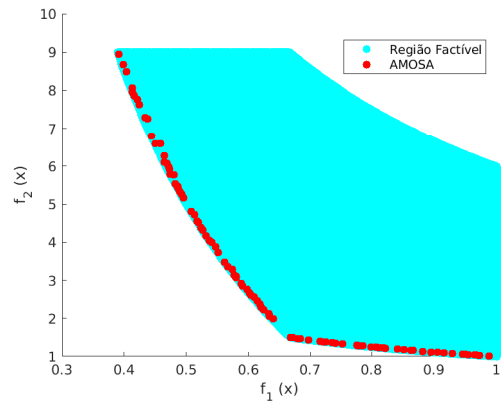
Fonte: O Autor.

Figura 27 – Frente de Pareto da Função Constr-Ex

(a) CoAnnealing



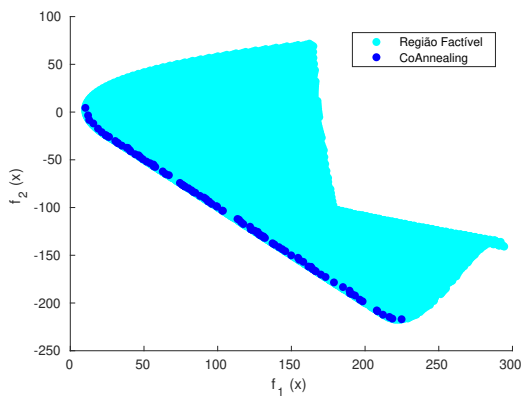
(b) AMOSA



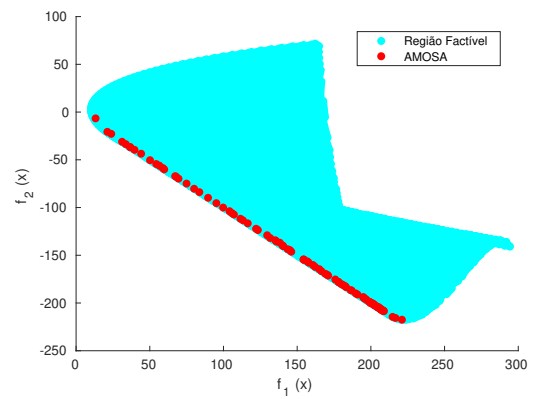
Fonte: O Autor.

Figura 28 – Frente de Pareto da Função Chankong e Haimes

(a) CoAnnealing



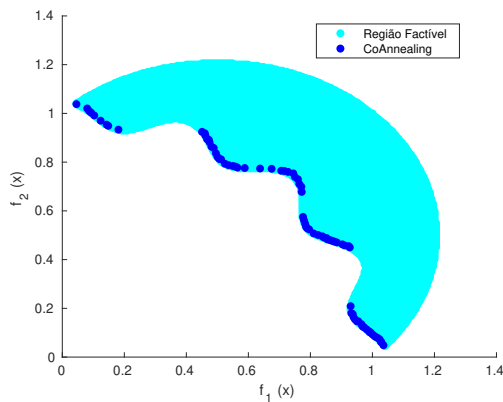
(b) AMOSA



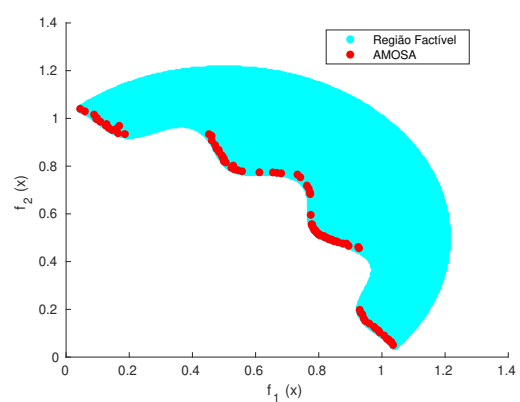
Fonte: O Autor.

Figura 29 – Frente de Pareto da Função TNK com restrição

(a) CoAnnealing



(b) AMOSA



Fonte: O Autor.

Tabela 6 – Resultados dos indicadores de desempenho do AMOSA e CoAnnealing com as funções benchmark.

Função	GD				Spacing				Hipervolume				Avaliação		
	AMOSA		CoAnnealing		AMOSA		CoAnnealing		AMOSA		CoAnnealing		AMOSA	CoAnnealing	
	méd.	D.P	méd.	D.P	méd	D.P	méd	D.P	méd	D.P	méd	D.P	méd.	méd.	D.P
ZDT1	0,0024	0,0033	0,0020	0,0003	0,037	0,032	0,012	0,002	24,5	0,1	24,6	0,0	231963	154208	6813
ZDT2	0,0046	0,0051	0,0046	0,0003	0,014	0,039	0,012	0,002	24,2	0,1	24,2	0,1	231963	146523	7905
ZDT3	0,0049	0,0012	0,0039	0,0023	0,017	0,011	0,019	0,016	28,1	0,0	28,0	0,6	166277	166277	9993
ZDT6	0,4397	0,0331	0,0079	0,0008	0,025	0,189	0,013	0,002	19,5	0,3	23,1	0,0	463463	323801	31072
DTLZ1	0,0069	0,0015	1,6590	1,2037	0,024	0,005	0,439	1,297	124,0	0,9	124,9	0,8	4630463	10787003	608961
DTLZ2	0,2767	0,0245	0,0701	0,0081	0,084	0,014	0,085	0,012	124,0	0,4	124,3	0,1	463463	264174	2164
DTLZ5	0,0072	0,0004	0,0208	0,0211	0,015	0,003	0,020	0,007	119,3	0,5	119,6	0,5	463463	387277	5753
DTLZ7	0,1486	0,1003	0,0222	0,0038	0,549	0,472	0,088	0,010	49,3	0,3	47,8	1,1	4167463	2880632	22062

O tempo de execução dos algoritmos para cada função é apresentado na Tabela 7. Nos problemas bi-dimensionais, o CoAnnealing foi mais rápido que o algoritmo AMOSA. Na função ZDT6 o CoAnnealing é muito mais rápido, conseguindo convergir com a metade do tempo do AMOSA. A quantidade de vezes em que as funções objetivo são avaliadas pode explicar o tempo de execução mais lento do algoritmo AMOSA. A função ZDT3 foi a única onde o AMOSA é mais rápido em relação ao CoAnnealing.

Nas funções tri-dimensionais o AMOSA executa com mais que o dobro do tempo em relação ao CoAnnealing, com exceção as funções ZDT3. Assim como nas funções de duas funções objetivos, o bom desempenho pode ser explicado pela quantidade de vezes necessárias para convergir. Além disso, o mecanismo de aceitação de nova solução mais complexa deve contribuir para a execução mais lenta em relação ao CoAnnealing.

Tabela 7 – Tempo de execução do algoritmo AMOSA e CoAnnealing com as funções de benchmark.

Funções	CPUTIME			
	AMOSA		CoAnnealing	
	med	D.P	med	D.P
ZDT1	25	1,1	14,3	1,3
ZDT2	25,5	2,0	13,4	1,2
ZDT3	6,0	0,2	13,3	0,9
ZDT6	67,3	2,2	31,4	4,0
DTLZ1	1067,9	64,7	265,6	18,2
DTLZ2	77,6	1,3	18,3	1,8
DTLZ5	74,3	1,5	19,7	1,5
DTLZ7	676,7	22,3	288,4	9,0

Os resultados apresentados da frente de Pareto validam a capacidade do CoAnnealing com a heurística de cristalização em resolver problemas de otimização multiobjetivo sem restrições e com restrições. Além disso, os indicadores de avaliação da frente de Pareto demonstram que o CoAnnealing utilizado possui desempenho semelhante ou superior em alguns casos em relação ao AMOSA. Segundo Bandyopadhyay et al. (2008), o algoritmo AMOSA apresentou resultados comparáveis aos algoritmos clássicos da literatura como os algoritmos MOSA, NSGA-II e PAES.

Os resultados indicam o potencial do CoAnnealing com a heurística de cristalização para a solução dos problemas de otimização, indicando que pode ser eficiente em diversos problemas. Pois, em diversos tipos de frente de Pareto testadas neste trabalho, o CoAnnealing obteve bons resultados.

4.7 Considerações Finais

Originalmente a proposta da heurística de cristalização proposta por Duran et al. (2021) foi utilizada no SA mono-objetivo. Neste trabalho foi aplicado este conceito da heurística de cristalização no CoAnnealing para a otimização multiobjetivo. Os resultados demonstraram que o CoAnnealing apresentou desempenho superior ao AMOSA, e em alguns casos desempenho semelhante. Além disso, devemos destacar que o AMOSA é um algoritmo complexo com diversas condições de aceitação da nova solução. Enquanto o CoAnnealing possui critério de aceitação da solução mais simples.

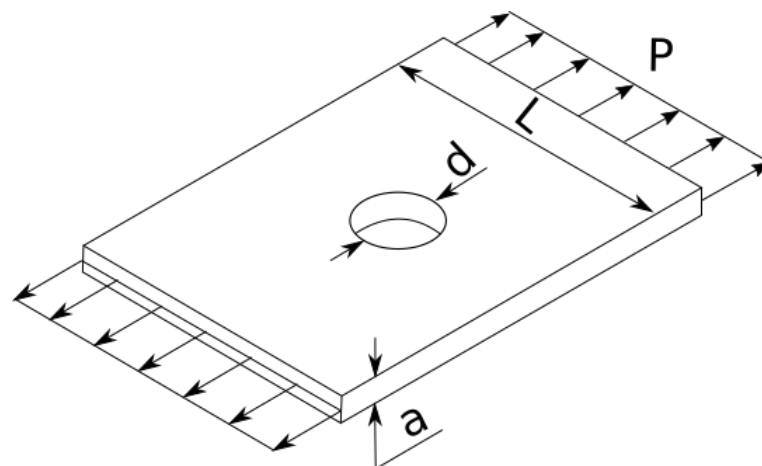
5 Otimização Multiobjetivo em Problemas Mecânicos

O CoAnnealing com a heurística de cristalização (introduzido no Capítulo 4) apresentou bom desempenho em otimização multiobjetivo de funções de benchmark. Com intuito de validação em problemas reais, aplicamos o algoritmo em otimização de problemas de dimensionamento de componentes mecânicos simples. Muitas das aplicações de Engenharia Mecânica necessitam de sistema de transmissão. Serão apresentados casos de dimensionamento de furos (Seção 5.1) em placa e de eixo para sistema de transmissão com polias e engrenagem (Seção 5.2 e Seção 5.3).

5.1 Placa com Orifício Circular

O primeiro caso de aplicação é a otimização de uma placa com formato retangular com carregamento e um orifício, conforme a Fig. 30. O problema de otimização consiste em maximizar o diâmetro do orifício e o fator de segurança do projeto. Este problema apresentado em Pardo (2009) é um problema multiobjetivo com duas funções objetivo contraditórias. A primeira função objetivo visa maximizar o diâmetro do orifício e a segunda maximizar o fator de segurança do projeto. Com o aumento do diâmetro, a área da seção transversal da peça é reduzida, aumentando as tensões normais e conseqüentemente reduzindo o fator de segurança do projeto.

Figura 30 – Desenho esquemático da placa com orifício para maximização do fator de segurança



Fonte: Adaptado de Pardo (2009).

Para a geometria e as condições de carregamento desse problema de otimização, as

tensões normais submetidas à placa são apresentadas em

$$\sigma = K \frac{P}{A}, \quad (5.1)$$

onde P é a carga, A a área transversal e K é o fator de concentração de tensões, que pode ser determinado por

$$K \cong 3,0039 - 3,753 \left(\frac{d}{0,08} \right) + 7,9735 \left(\frac{d}{0,08} \right)^2 - 9,2659 \left(\frac{d}{0,08} \right)^3. \quad (5.2)$$

E o fator de segurança pode ser determinado por

$$FS = \frac{S_y}{\sigma}, \quad (5.3)$$

onde S_y é a resistência de escoamento e o valor utilizado foi de $S_y = 462 \text{ MPa}$ conforme Pardo (2009). E σ é a tensão normal descrita em (5.1).

A função objetivo a ser maximizada é representada matematicamente por

$$f_1(x) = d, \quad (5.4)$$

onde d é o diâmetro do furo como apresentado na Fig. 30.

A segunda função objetivo a ser maximizada é representada por

$$f_2(x) = FS = \frac{S_y}{\sigma} = \frac{462 \text{ MPa}}{K \frac{P}{A}}. \quad (5.5)$$

A primeira função objetivo é o diâmetro do orifício e a segunda função objetivo é o fator de segurança de projeto. O problema de otimização é para maximizar ambas funções objetivo.

A otimização do problema é submetida às restrições

$$g_1(x) = \frac{S_y}{\sigma} - 1 \geq 0 \quad (5.6)$$

$$g_2(x) = 0,65 - \frac{d}{L} \geq 0. \quad (5.7)$$

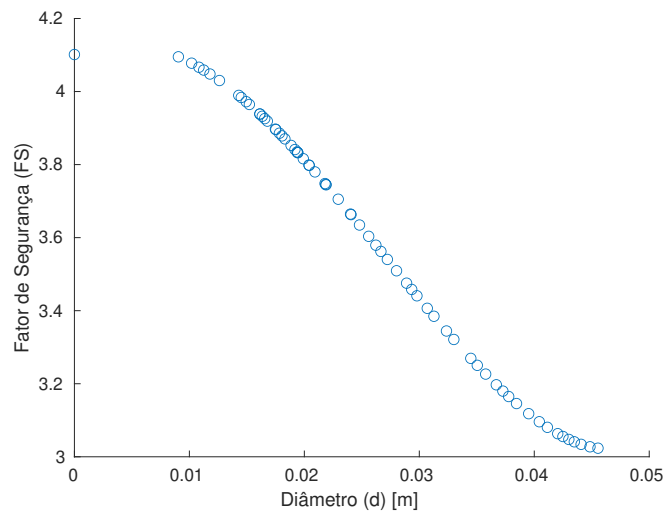
A primeira restrição não permite soluções com fator de segurança do projeto abaixo do valor 1. E a segunda restrição limita a proporção entre o diâmetro e a largura da placa a 0,65. Lembrando que a variável de decisão deste problema é o diâmetro d do furo.

Estas equações e as funções objetivas são as mesmas utilizadas por Pardo (2009). Reproduzimos o problema de otimização apresentado em Pardo (2009) para fim de comparação e validação do algoritmo multiobjetivo CoAnnealing com heurística de cristalização. O domínio do problema de otimização é formado por $0 \leq d \leq 0,08 \text{ m}$. A largura da placa é $L = 80 \text{ mm}$, a espessura $a = 5 \text{ mm}$ e a carga $P = 15 \text{ kN}$. Os parâmetros utilizados no

CoAnnealing foram: $T_{\min} = 0,00001$, $T_{\max} = 200$, $N = 20$, $\alpha = 0,95$, $HL = 50$, $SL = 100$, $C_{\max} = 20$.

A otimização com o CoAnnealing resultou na frente de Pareto apresentada na Fig. 31. E algumas das soluções obtidas que pertencem à frente de Pareto são apresentadas na Tabela 8.

Figura 31 – Frente de Pareto obtida por meio da otimização com o CoAnnealing do problema de dimensionamento do furo



Fonte: O autor.

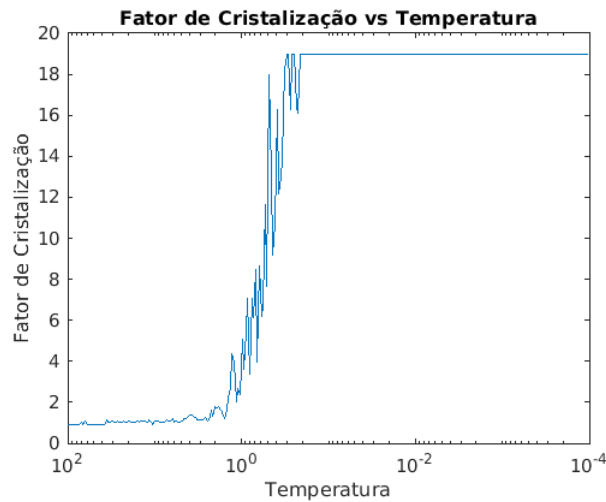
Tabela 8 – Soluções ótimas obtidas no processo de otimização do problema da placa de orifício circular

Diâmetro do orifício [m]	Fator de Segurança
0,0102	4,077
0,0090	4,0951
0,0267	3,5622
0,0324	3,3444

Fonte: O autor.

A frente de Pareto obtida com o CoAnnealing foi satisfatória e semelhante aos resultados obtidos pela otimização pelo algoritmo baseado em quimiotaxia de bactérias de Pardo (2009). A variação do fator de cristalização ao longo do processo de otimização é apresentado na Fig. 32. O comportamento do fator de cristalização foi conforme o esperado, apresentando um aumento do valor conforme a redução da temperatura, indicando o refinamento.

Figura 32 – Variação do fator de cristalização no processo de otimização da dimensão do furo



Fonte: O Autor.

5.2 Problema de Otimização de Eixo 1

Aplicamos a otimização no dimensionamento de um eixo de um sistema de transmissão com uma polia e uma engrenagem de dentes retos, apresentada na Fig. 33. Nesse problema de otimização da geometria do eixo com objetivos de minimizar o peso e a deflexão lateral do eixo, considerando a resistência à fadiga, o torque e o momento fletor aplicado no eixo. O problema foi inspirado no problema de otimização apresentado em Pardo (2009), porém não foi possível reproduzir com os mesmos dados. Pois, algumas informações não foram apresentados.

O sistema de transmissão é composto por dois mancais, ponto R_1 e R_2 , a engrenagem no ponto F_g e a polia no ponto F_s , conforme apresentado na Fig. 33. A engrenagem é do tipo dente reto com ângulo de contato $\alpha = 20^\circ$, e a polia para correia do tipo trapezoidal.

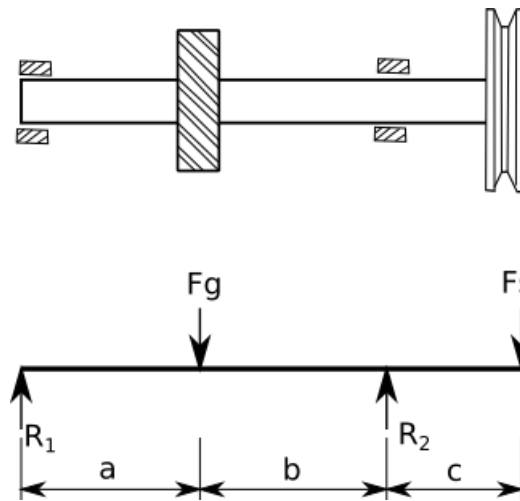
As variáveis de decisão são os diâmetros d_1 , d_2 , d_3 e o comprimento da seção L_1 do eixo. Essas variáveis de decisão são apresentadas na Fig. 34.

O objetivo da otimização é otimizar o eixo para minimizar o peso e, simultaneamente, minimizar a deflexão lateral. A primeira função objetivo é utilizada para determinar o peso do eixo conforme o diâmetro e o comprimento de cada seção do eixo escalonada, sendo representado matematicamente por

$$f_1(x) = \frac{9,81\pi\rho_{aço}}{4}(L_2d_1^2 + L_1d_2^2 + L_2d_3^2). \quad (5.8)$$

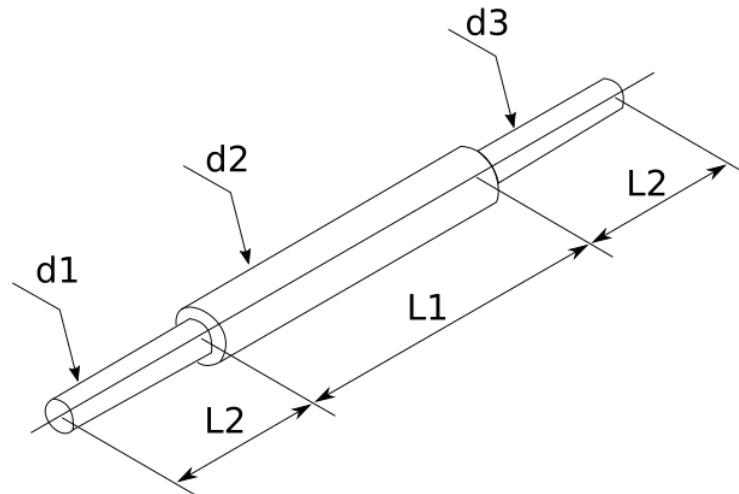
A segunda função objetivo é a deflexão lateral máxima do eixo, sendo este valor máximo obtido da deflexão resultante do plano xz e yz . A deflexão é determinada pela aplicação da integração numérica, a técnica utilizada foi a regra de Simpson. A constante

Figura 33 – Diagrama esquemático do sistema de transmissão, para a otimização do dimensionamento do eixo



Fonte: Adaptado de Pardo (2009).

Figura 34 – Desenho esquemático do eixo e os componentes de transmissão



Fonte: adaptado de Pardo (2009).

de integração C é determinada com aplicação das condições de contorno. A segunda função objetivo é representada matematicamente por

$$f_2(x) = \max(\sqrt{y_{xz}^2 + y_{yz}^2}). \quad (5.9)$$

O problema de otimização é submetido a algumas restrições, e as primeiras restrições é o diâmetro do eixo de cada seção. O diâmetro deve ser submetido ao critério de fadiga e deflexão do eixo, o critério de *The American Society of Mechanical Engineers* (ASME) apresentado em Norton (2013). A primeira restrição é a limitação do diâmetro d_1 que deve satisfazer o dimensionamento do critério mencionado. A restrição é dada por

$$g_1(x) = d_1 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{f1}M_{a1})^2 + \frac{3}{4}(K_{fs1}T_{a1})^2}}{S_f} + \frac{\sqrt{(K_{fm1}M_{m1})^2 + \frac{3}{4}(K_{fsm1}T_{m1})^2}}{S_{ut}} \right] \right\} \geq 0. \quad (5.10)$$

A segunda restrição é a limitação do diâmetro d_2 que deve satisfazer o critério

$$g_2(x) = d_2 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{f2}M_{a2})^2 + \frac{3}{4}(K_{fs2}T_{a2})^2}}{S_f} + \frac{\sqrt{(K_{fm2}M_{m2})^2 + \frac{3}{4}(K_{fsm2}T_{m2})^2}}{S_{ut}} \right] \right\} \geq 0. \quad (5.11)$$

A restrição do diâmetro d_3 é dada por

$$g_3(x) = d_3 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{f3}M_{a3})^2 + \frac{3}{4}(K_{fs3}T_{a3})^2}}{S_f} + \frac{\sqrt{(K_{fm3}M_{m3})^2 + \frac{3}{4}(K_{fsm3}T_{m3})^2}}{S_{ut}} \right] \right\} \geq 0, \quad (5.12)$$

e deve satisfazer o critério de fadiga e deflexão descrito anteriormente. A quarta e última restrição é dada por

$$g_4(x) = 0.000127 - y_{\max} \geq 0, \quad (5.13)$$

a qual limita a deflexão resultante.

Os parâmetros utilizados no problema são apresentados na Tabela 9. Nesta tabela é apresentadas as distâncias dos pontos de apoio e componentes da transmissão. A potência máxima e mínima do sistema também são apresentadas na tabela abaixo. Alguns parâmetros foram semelhantes aos utilizados por Pardo (2009).

As variáveis de decisão do problema são apresentadas na Tabela 10 com seus respectivos intervalos. Estes limites foram adotados semelhantemente aos utilizados por Pardo (2009). Para o fator de segurança do projeto N_f foi adotado o valor de 2,2 de acordo com o valor adotado por Pardo (2009).

Algumas das soluções ótimas presentes na frente de Pareto são apresentadas na Tabela 11. Apresentamos as soluções do extremo da frente de Pareto e algumas outras soluções intermediárias.

A variação do fator de cristalização das variáveis de decisão é apresentada na Fig. 36. O comportamento da variação de fator de cristalização obtido apresentou as características esperadas do processo de otimização. A frente de Pareto como as soluções obtidas foram consistentes, demonstrando a capacidade do CoAnnealing para solução de problemas reais.

Tabela 9 – Tabela de dados do problema de otimização do eixo 1

Parâmetro	Valor ou intervalo
a	0,45 m
b	0,2 m
c	0,15 m
ω	1200 rpm
Material	AISI 1040
S_{ut}	620 MPa
S_y	415 MPa
E	210 GPa
Polia	
Diâmetro	0,3 m
Peso	220 N
Engrenagem	
Diâmetro	0,2 m
Peso	70 N
Ângulo de pressão α :	20°
Motor	
Potência máxima:	2000 watts
Potência mínima:	1100 watts

Fonte: O autor.

Tabela 10 – Variáveis e valores mínimos e máximos do problema de otimização eixo 1

Variável	valor mínimo	valor máximo
d_1	0,005 m	0,025 m
d_2	0,025 m	0,050 m
d_3	0,005 m	0,025 m
L_1	0,200 m	0,500 m

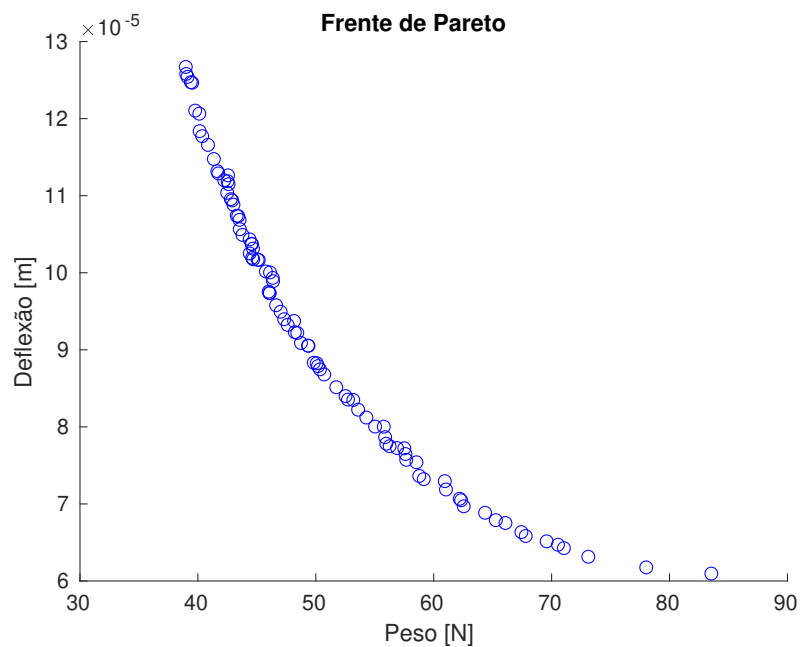
Fonte: O autor.

Tabela 11 – Soluções ótimas presentes na frente de Pareto do problema de otimização do eixo 1

d_1	d_2	d_3	L_1	Peso [N]	deflexão [m]
0,0210	0,0495	0,0247	0,4989	83,54	$6,09 \times 10^{-5}$
0,0247	0,0420	0,0249	0,4975	64,35	$6,88 \times 10^{-5}$
0,0207	0,0347	0,0246	0,4977	45,76	$1,00 \times 10^{-4}$
0,0201	0,0315	0,0245	0,4944	38,97	$1,26 \times 10^{-4}$

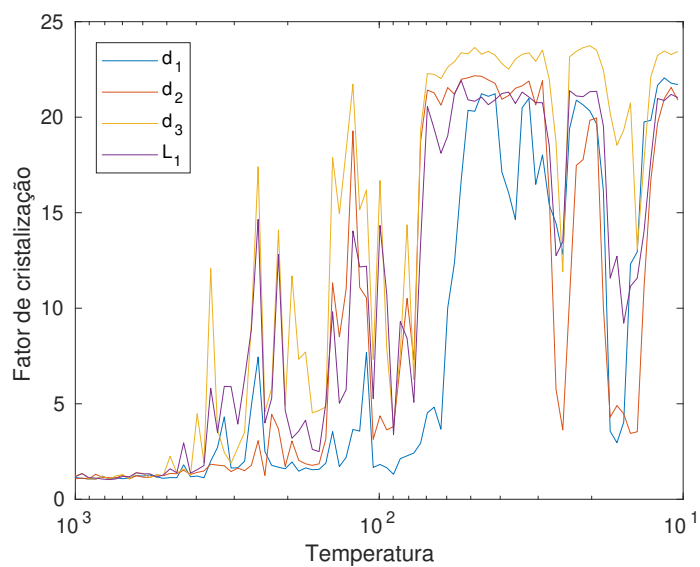
Fonte: O autor.

Figura 35 – A frente de Pareto ótima obtida na otimização do eixo 1



Fonte: O Autor.

Figura 36 – Comportamento do fator de cristalização do problema de otimização do eixo 1



Fonte: O Autor

5.3 Problema de Otimização de Eixo 2

O segundo problema de otimização de transmissão é semelhante ao apresentado na Fig. 34, com os mesmos componentes de transmissão e mancais. O problema consiste em minimizar a deflexão e o peso do eixo com a variação dos diâmetros em cada seção e a distância do mancal c . Neste problema, a potência do motor é constante e submetida às restrições do diâmetro mínimo admissível em cada seção. A otimização também é submetida à restrição de deflexão máxima e a velocidade crítica do eixo que não pode ser inferior a 4 vezes a velocidade de trabalho segundo Pardo (2009). Os parâmetros do problema de transmissão, como potência, rotação de trabalho e distâncias entre os elementos é dada na Tabela 12.

Tabela 12 – Tabela de dados do problema de otimização do eixo 2

Parâmetro	Valor ou intervalo
a	(0,65 - c) m
b	0,2 m
ω	1600 rpm
Material	AISI 1040
S_{ut}	620 MPa
S_y	415 MPa
E	210 GPa
Polia	
Diâmetro	0,3 m
Peso	220 N
Engrenagem	
Diâmetro	0,2 m
Peso	70 N
Ângulo de pressão α :	20°
Motor	
Potência máxima:	7500 watts

Fonte: O autor.

A primeira função objetivo a ser minimizada determina a massa do eixo dada por

$$f_1(x) = 60482,33(d_1^2 L_2 + d_2^2 L_1 + d_3^2 L_2). \quad (5.14)$$

E a segunda função objetivo é minimizar a deflexão lateral máxima determinado por

$$f_2(x) = \max(\sqrt{(y_{xz})^2 + (y_{yz})^2}). \quad (5.15)$$

As variáveis de decisão são as dimensões do eixo, os diâmetros d_1 , d_2 , d_3 e a distância do mancal c , estas dimensões são representados na Fig. 34. Neste problema as dimensões das seções do eixo L_1 e L_2 são fixas com valores de 0,2 m e 0,3 m respectivamente e a posição do mancal c é variável.

A otimização está sujeita ao diâmetro mínimo admissível determinado de acordo com a teoria de falha por fadiga e deflexão. E neste trabalho foi utilizado o critério da ASME apresentado por Norton (2013). O critério é aplicado em cada seção do eixo, portanto a otimização está sujeito à

$$g_1(x) = d_1 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a1}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m1}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (5.16)$$

$$g_2(x) = d_2 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a2}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m2}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (5.17)$$

$$g_3(x) = d_3 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a3}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m3}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0. \quad (5.18)$$

A outra restrição imposta no problema é o limite da deflexão lateral do eixo de 0,000127 metros, conforme utilizado por Pardo (2009) e

$$g_4(x) = 0,000127 - y_{\max} \geq 0. \quad (5.19)$$

A última restrição do problema é a necessidade de a velocidade crítica ser quatro vezes maior que a velocidade de trabalho, conforme a restrição adotada por Pardo (2009). Essa restrição é dada por.

$$g_5(x) = \omega_n - 4\omega_t \geq 0, \quad (5.20)$$

onde ω_t é a velocidade de trabalho e ω_n é a velocidade crítica sendo determinada por meio do método de Rayleigh apresentado em Norton (2013).

Os parâmetros de otimização do CoAnnealing utilizado foram: $T_{\max} = 100$, $T_{\min} = 0,0001$, $N = 1000$, $\alpha = 0,95$, $HL = 50$, $SL = 100$ $C_{\max} = 20$. A estratégia do fator de cristalização utilizada foi a estratégia III descrita na Tabela 2 (na Seção 3.4) e o algoritmo apresentado no Algoritmo 7. Os intervalos das variáveis de decisão são apresentados na Tabela 13, estes valores difere dos valores adotados por Pardo (2009) devido algumas inconsistências nos valores.

Com o processo de otimização obtemos a frente de Pareto apresentada na Fig. 37, e os resultados obtidos foram satisfatórios com o formato e distribuição das soluções de forma distribuída.

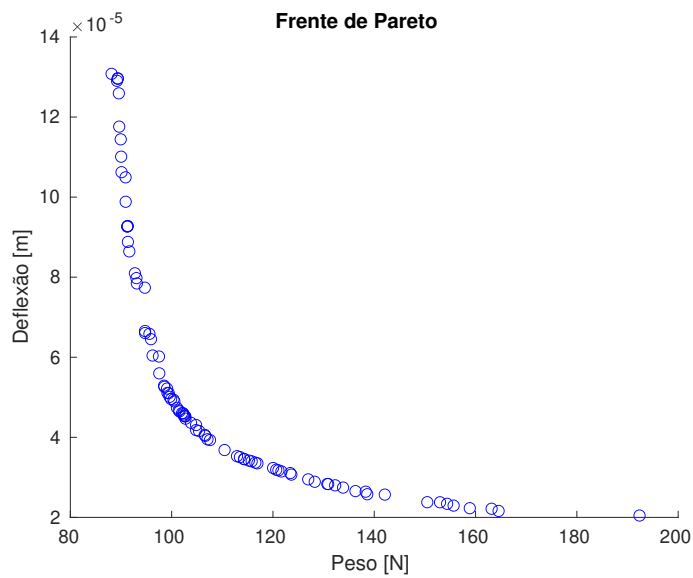
Algumas soluções da frente de Pareto são apresentadas na Tabela 14, apresentamos as soluções da extremidade da frente de Pareto e algumas soluções intermediárias.

Tabela 13 – Variáveis e valores mínimos e máximos do problema de otimização do eixo 2

Variável	valor mínimo	valor máximo
d_1	0,025 m	0,05 m
d_2	0,05 m	0,08 m
d_3	0,025 m	0,05 m
c	0,05 m	0,5 m

Fonte: O autor.

Figura 37 – Frente de Pareto obtida com processo de otimização do eixo 2



Fonte: O Autor.

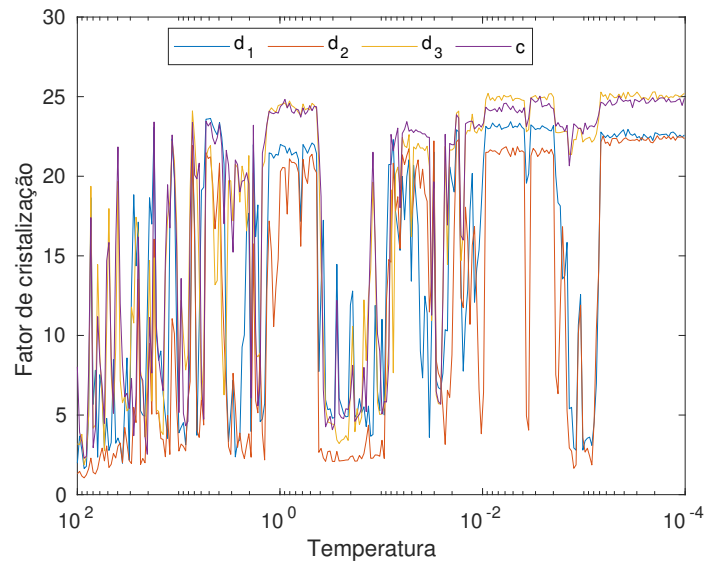
Tabela 14 – Soluções ótimas presentes na frente de Pareto do problema de otimização do eixo 2

d_1	d_2	d_3	c	Peso [N]	deflexão [m]
0,0493	0,07	0,0496	0,0819	192,3703	$2,046 \times 10^{-5}$
0,0498	0,0614	0,0498	0,1054	158,8590	$2,22 \times 10^{-5}$
0,0484	0,0507	0,0493	0,1268	120,028	$3,23 \times 10^{-5}$
0,0259	0,0501	0,0492	0,0545	88,1429	$1,30 \times 10^{-5}$

Fonte: O autor.

O fator de cristalização apresentou o comportamento esperado com o aumento do fator de cristalização com a redução da temperatura. A variação do fator de cristalização é apresentada na Fig. 38.

Figura 38 – Variação do fator de cristalização no processo de otimização do eixo 2



Fonte: O Autor.

Os resultados obtidos foram satisfatórios, obtivemos uma frente de Pareto consistente com o problema proposto. Os valores finais das soluções não foram passíveis de comparação com os valores obtidos por Pardo (2009), pois encontramos algumas inconsistências nos valores dos parâmetros apresentados, como o intervalo das variáveis de decisão.

5.4 Considerações Finais

Neste capítulo aplicamos o CoAnnealing com a heurística de Cristalização em três problemas mecânicos. Os resultados obtidos foram satisfatórios, demonstrando a capacidade do CoAnnealing em solucionar diferentes tipos de problemas de otimização multiobjetivo. Com os resultados concluímos que o CoAnnealing é promissor para solucionar diferentes tipos de problemas.

6 Otimização Topológica com Recozimento Simulado

Na literatura, a maioria dos trabalhos relacionados à otimização topológica estão relacionados aos algoritmos baseado em método gradiente. Porém, apesar da rápida convergência, os métodos possuem algumas limitações como discutido anteriormente na Subseção 2.3.4, principalmente com relação a dificuldade de obtenção do gradiente em várias aplicações de interesse prático.

Na Subseção 2.3.4 comentamos sobre alguns trabalhos de Otimização Topológica (OT) com algoritmos baseados em métodos não-gradientes. Contudo, como descrito anteriormente, problemas de OT envolvem um grande número de variáveis de decisão. Dessa forma, é necessária uma grande quantidade de iterações para convergir à um resultado. Para contornar este problema, Garcia-Lopez et al. (2011) e Burerat e Limtragool (2006) propuseram uma técnica de seleção e modificação das variáveis de decisão.

Neste trabalho foi proposto um algoritmo mono-objetivo baseado no SA e o método de otimização topológica SIMP. Nesta proposta também aplicamos uma técnica de seleção e modificação das variáveis de decisão que será discutida na próxima seção.

6.1 OT com SA e Heurística de Cristalização

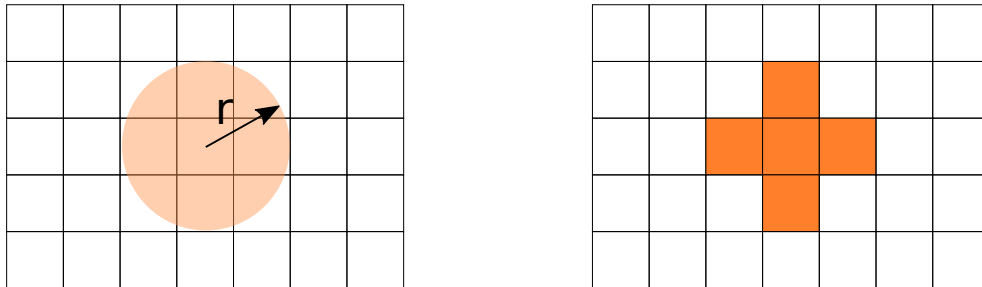
O algoritmo proposto foi uma adaptação do algoritmo de SA com heurística de cristalização utilizado por Duran et al. (2021). Na otimização topológica proposta são utilizados os conceitos do método SIMP, cada elemento da malha possui uma pseudo densidade entre 0 e 1. A densidade de cada elemento é uma variável de decisão. Portanto, quanto maior a quantidade de elementos na malha, maior será a quantidade de variáveis de decisão no problema.

No SA a nova solução é encontrada modificando uma variável por vez, no algoritmo proposto a modificação ocorre em mais de uma variável de decisão. Inicialmente é sorteado um elemento (variável de decisão) de forma aleatória e as modificações são aplicadas no elemento selecionado e nos elementos adjacentes dentro da região circular formada com o raio r . Este parâmetro r é dependente da quantidade de elementos na malha, mas um valor típico é $r = 2$. O valor de r é constante ao longo do processo de Otimização, mas ela pode ser variável e será apresentado em uma nova proposta em Seção 6.3. A representação do elemento selecionado e sua vizinhança é apresentada na Fig. 39. Como podemos observar na figura Fig. 39b, o algoritmo delimita uma região no formato circular e seleciona apenas os elementos que estão completamente dentro da região delimitada. A cada iteração, é

sorteado um novo elemento e delimitada uma nova área.

Figura 39 – Diagrama esquemático de escolha de nova solução na vizinhança

- (a) Região adjacente ao elemento selecionado, com uma distância r (b) Elementos modificados para gerar a nova solução



Fonte: O Autor.

A modificação no elemento e sua vizinhança é realizada conforme o Algoritmo 8. Os elementos modificados são analisados, os que possuem densidades muito próximas de 1 são arredondados para o valor 1 e os próximos de zero são arredondados para a densidade mínima. Este processo de arredondamento do valor da densidade acelera o processo de convergência. Além disso, esse processo reduz o gradiente de cinza ao comparar com o resultado de Goto et al. (2021a) sem a necessidade do pós-processamento utilizado em Najafabadi et al. (2021). A dinâmica de gerar uma nova solução do algoritmo funciona da seguinte forma:

1. a nova solução x_j recebe os valores da solução corrente x_i ;
2. enquanto o critério não é satisfeito;
3. de forma aleatória, é escolhida uma variável de decisão k ;
4. se o fator de cristalização C_k é menor que C_{\max} , o valor de Δ é determinado com a distribuição de Bates;
5. se o fator de cristalização C_k é maior que C_{\max} , o valor de Δ é determinado com a distribuição gaussiana;
6. realiza a modificação na variável de decisão $x_{n,k}$ e as variáveis pertencentes aos elementos vizinhos. Estes elementos adjacentes dentro da região com raio r ;
7. verifica o valor da densidade e realiza o arredondamento para o valor 1 ou a densidade mínima;
8. verifica a fração volumétrica e intervalo, caso não seja satisfeito os critérios retorna ao passo 2.

Algoritmo 8: GERAÇÃO DA NOVA SOLUÇÃO PARA A OT MONO-OBJETIVO

Entrada: x_a : Solução atual,
 C_k : Fator de cristalização,
 C_{\max} : Fator de cristalização máxima,
 r : Raio da região,
 FV : Fração Volumétrica

- 1 **enquanto** $x \in [x_{\min}, x_{\max}]$ e $FV_n \leq FV$ **faça**
- 2 $x_n \leftarrow x_a$
- 3 $k \leftarrow$ seleciona aleatoriamente a variável de decisão a ser modificado
- 4 **se** $C_k \leq C_{\max}$ **então**
- 5 $\Delta \leftarrow \frac{1}{C_k} \sum_1^{C_k} \text{random}(-1, 1) \cdot \Delta r_k \cdot e_k$
- 6 **senão**
- 7 $\Delta \leftarrow \text{GaussianDist}(\exp(C_{\max} - C_k - 2)) \cdot \Delta r_k \cdot e_k$
- 8 **fim**
- 9 **para** $k \in \text{Região selecionada}$ **faça**
- 10 $x_{n,k} \leftarrow x_{n,k} + \Delta$
- 11 $x_{n,k} \leftarrow \text{corrigirosvalores}$
- 12 **fim**
- 13 $FV_n = \text{FraçãoVolumétrica}(x_n)$
- 14 **fim**

Durante o processo de busca da nova solução é realizada uma modificação, Δ , em todos os elementos na região de seleção. Este procedimento evita a formação do padrão tabuleiro de xadrez, já mencionado anteriormente na Fig. 9. Esta técnica de escolha e modificação das variáveis de decisão é acoplada no algoritmo de recozimento simulado de vizinhança adaptativa. As etapas da otimização topológica seguem os seguintes passos:

1. o algoritmo inicia-se com os parâmetros T_{\max} , T_{\min} , N (número de iterações), α e r ;
2. o algoritmo gera aleatoriamente a solução X_a e avalia a função objetiva $f(X_a)$ conforme a linha 1;
3. enquanto a temperatura $T > T_{\min}$;
4. enquanto não seja satisfeito o critério (número de iterações N) ou atingido o equilíbrio térmico;
5. a nova solução X_n é gerada conforme o Algoritmo 8, na linha 6, e avaliada a função objetivo da nova solução $f(X_n)$ na linha 6;
6. é determinada a variação do estado de energia ΔE na linha 7;
7. se $\Delta E \leq 0$ ou $\text{random}(0, 1) < e^{-\frac{\Delta E}{kT}}$;
8. se a nova solução for aceita, a solução atual recebe a nova solução e o fator de cristalização recebe o feedback positivo;

9. se a nova solução é rejeitada, o fator de cristalização recebe o feedback negativo, linha 18;
10. decresce a temperatura de acordo com o esquema de resfriamento, se não atingir a temperatura mínima retorna ao passo 4.

O pseudocódigo do algoritmo de otimização topológica com o SA é apresentado em Algoritmo 9.

Algoritmo 9: Algoritmo de Otimização topológica com recozimento simulado de vizinhança adaptativa em meio contínuo

Entrada: T_0 : Temperatura inicial, α : fator de decaimento da temperatura,
 N : número de iterações, *penalidade*,
nelx : número de elementos na direção x,
nely : número de elementos na direção y,
 r : valor do raio da região selecionada,
 FV : Fração volumétrica

- 1 $X_a \leftarrow$ recebe uma solução aleatória
- 2 $C_k \leftarrow 1$ /* inicia o fator de cristalização das variáveis de decisão atribuindo o valor 1 */
- 3 $F(X_a) \leftarrow$ Energia de deformação
- 4 **enquanto** $T_j > T_{\min}$ **faça**
- 5 **enquanto** *Não satisfaz o critério local ou alcança o equilíbrio térmico* **faça**
- 6 $X_n, k \leftarrow$ GerarNovaSolução(X_a, r, C)
- 7 $F(X_n) \leftarrow$ Análise FEA para obter a energia deformação
- 8 $\Delta E = F(X_n) - F(X_a)$
- 9 **se** $\Delta E \leq 0$ **então**
- 10 $X_a \leftarrow X_n$
- 11 $C_k \leftarrow$ <Realimentação positiva>;
- 12 **senão**
- 13 **se** $random(0, 1) < e^{-\frac{\Delta E}{kT}}$ **então**
- 14 $X_a \leftarrow X_n$
- 15 $C_k \leftarrow$ <Realimentação positiva>;
- 16 **senão**
- 17 X_n é rejeitado
- 18 $C_k \leftarrow$ <Realimentação negativa>;
- 19 **fim**
- 20 **fim**
- 21 **fim**
- 22 $T_{j+1} \leftarrow T_j \times \alpha$
- 23 **fim**

A avaliação da função objetivo da otimização topológica utilizada é a análise de elementos finitos para determinar a energia de deformação. Na avaliação da função objetivo usamos o conceito do método de OT SIMP já descrito na Subseção 2.3.1. Neste método é

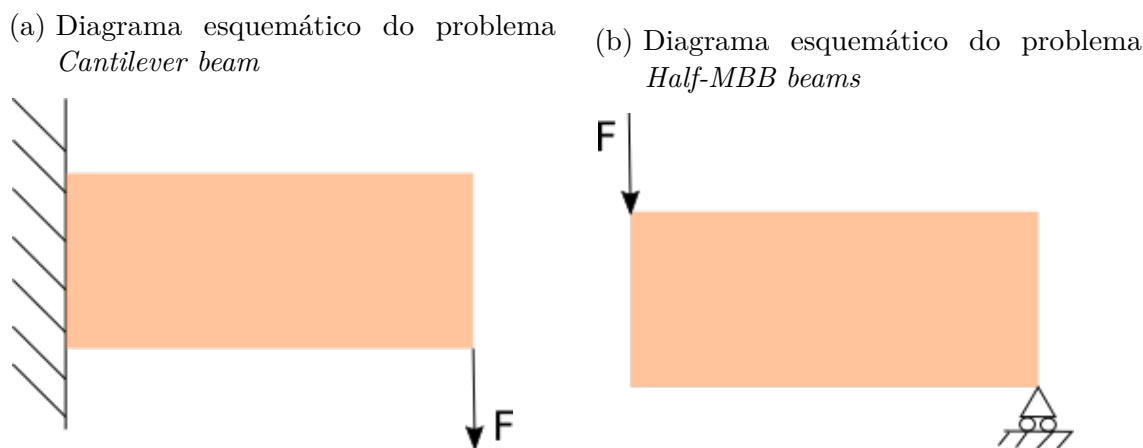
utilizado o conceito de penalidade da densidade e o módulo de elasticidade é determinado pela equação (2.21).

Em relação à análise de elementos finitos, utilizamos o *solver* de elementos finitos do código de otimização topológica disponibilizado por Andreassen et al. (2011). Adaptamos este código para determinar a energia de deformação no algoritmo de OT proposto neste trabalho. O algoritmo de OT proposto foi aplicado em problemas clássicos da literatura, e seus resultados serão discutidos na próxima seção.

6.2 Resultados do SA com a Heurística de Cristalização

Problemas de OT clássicos utilizados na literatura para análise de desempenho dos algoritmos são o *Cantilever beam* e *Messerschmitt–Bolkow–Blohm (Half-MBB beam)*. O primeiro problema trata de uma viga engastada e uma força aplicada na ponta da viga, o diagrama esquemático é apresentado na Fig. 40a. O segundo problema é a metade de uma viga bi-apoiada com uma força aplicada no centro da viga, o diagrama esquemático é apresentado na Fig. 40b. Estes são os principais problemas de OT utilizados na literatura para validação de algoritmos de OT, o que permite utilizar como referência para comparação com o algoritmo proposto.

Figura 40 – Problemas clássicos de vigas encontrados na literatura para teste de OT



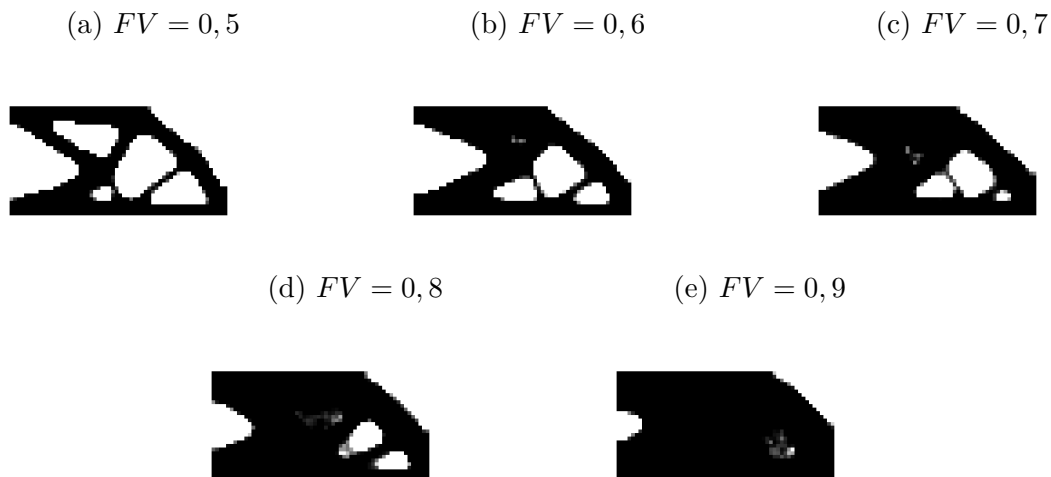
Fonte: O Autor.

O Algoritmo 9 foi implementado no MATLAB® e aplicado nos dois problemas descritos, com os seguintes parâmetros: $nelx = 60$, $nely = 30$, $N = 5000$, $\alpha = 0,85$, $T_{max} = 200$, $T_{min} = 0,0000001$, $penal = 3$ e $r = 1,5$. A fração volumétrica foi colocada como restrição, variando de 0,5 até 0,9. A função objetivo da otimização foi a energia de deformação (*compliance*), e como restrição a fração volumétrica FV. A estratégia da heurística de cristalização utilizada foi a estratégia I.

Para fins de comparação, rodamos o algoritmo de otimização topológica pelo método SIMP com algoritmo gradiente implementado no MATLAB[®]. O código utilizado foi o top88.m¹ disponibilizado por Andreassen et al. (2011), desenvolvido para fins didáticos. Os parâmetros utilizados para este algoritmo foram: $nelx = 60$, $nely = 30$, $penal = 3$, $r_{\min} = 1,5$ e $ft = 1$. O parâmetro $ft = 1$ indica a utilização do filtro de sensibilidade, para utilizar o filtro de densidade o valor deve ser 2.

Os resultados obtidos pela otimização topológica do algoritmo proposto são apresentados na Fig. 41 para o problema *Cantilever Beam*. A distribuição de massa no domínio não apresentou o padrão de tabuleiro de xadrez ou regiões com o gradiente de cinza. Em comparação aos resultados obtidos do método tradicional baseado em GTO, verificamos uma distribuição de massa com alguns pontos com ligações mais finas.

Figura 41 – Resultados de OT do problema *Cantilever Beam*



Fonte: O Autor.

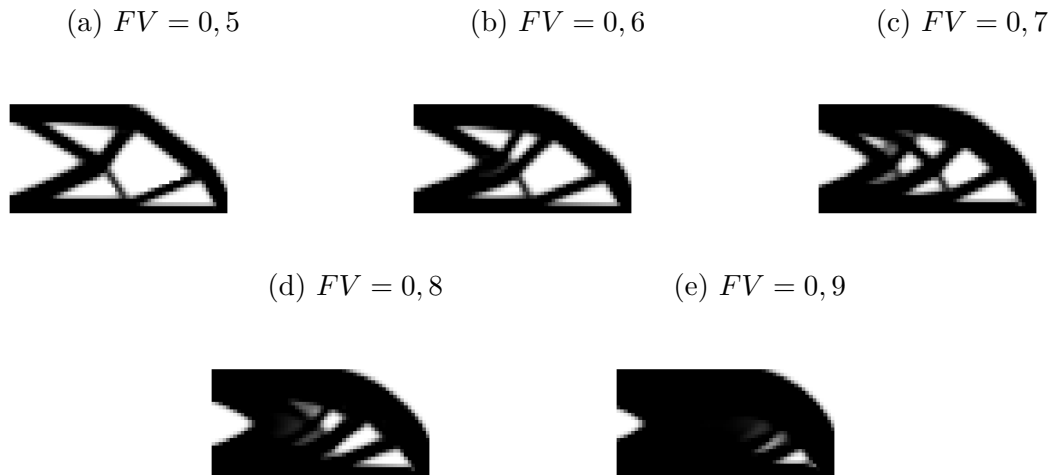
A distribuição de massa para o problema *Cantilever beam* resultante do código de Andreassen et al. (2011) é apresentada na Fig. 42. A distribuição de massa apresentou gradiente de cinza em alguns pontos. Estes resultados ocorreram para todos os casos de fração volumétrica, como podemos observar na Fig. 42.

Na distribuição de massa do algoritmo proposto (Fig. 41), podemos observar que os vazios criados na estrutura possui cantos mais arredondados. Este efeito pode ser explicado pelo formato circular da região de seleção. Além disso, o formato externo também é ligeiramente mais arredondado do que os resultados do algoritmo top88. Os cantos mais arredondados são desejados, para redução da concentração de tensão.

A otimização proposta obteve resultados satisfatórios com o valor de iteração de $N = 5000$, como observamos nas Fig. 41a e Fig. 42a. Em relação ao algoritmo proposto, o

¹ O código é disponibilizado em: <https://www.topopt.mek.dtu.dk/>

Figura 42 – Resultados da OT pelo método SIMP do código Andreassen et al. (2011) para o problema *Cantilever Beam*



Fonte: O Autor.

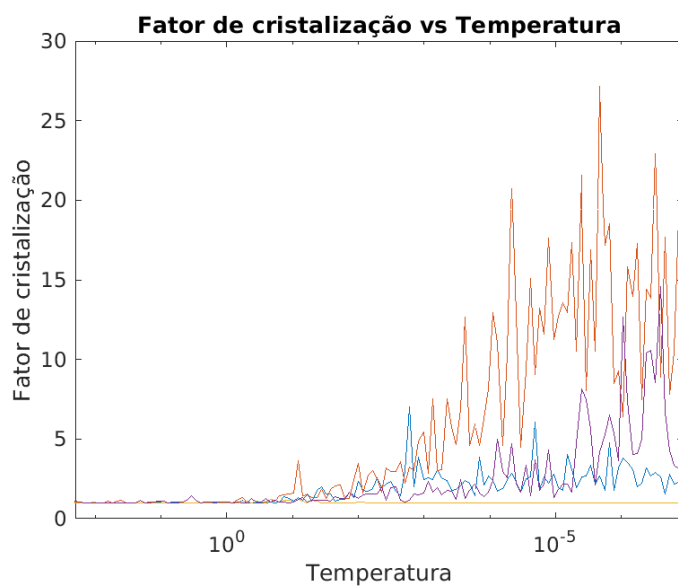
número de iteração, N , maior não influencia expressivamente nos valores da energia de deformação, porém a influência é maior na distribuição de massa. Quando aumentamos o número de iteração, a distribuição de massa resulta em vazios maiores e menor quantidade de furos.

A variação do fator de cristalização em relação à temperatura é apresentada na Fig. 43. Devido à alta quantidade de variáveis, na Fig. 43 apresentamos o fator de cristalização de apenas 4 variáveis escolhidas em pontos distintos da malha, no centro e 3 três pontos perto das bordas escolhidos de forma distribuída. Em alta temperatura, o fator de cristalização apresenta um valor baixo e uma atividade de exploração maior pelo SA. Com o decréscimo da temperatura o valor do fator de cristalização aumenta, o que significa um processo de refinamento do SA. Nos testes de validação realizados neste trabalho, utilizamos a estratégia I da Tabela 2.

A distribuição de massa no domínio para o problema *Half-MBB beam* é apresentada na Fig. 44, e os resultados obtidos pelo método SIMP com otimização baseada em método gradiente são apresentados na Fig. 45. O método proposto não apresentou o padrão tabuleiro de xadrez e apresentou poucas áreas com gradiente de cinza. Em relação à geometria formada pela distribuição de massa, o resultado ficou mais parecido com o resultado dos métodos tradicionais apresentados na Fig. 45. Para este problema *Half-MBB beams*, o comportamento do fator de cristalização é semelhante ao encontrado na Fig. 43.

Os valores da energia de deformação obtidos no processo da otimização são dados na Tabela 15. Nos resultados, verificamos que a energia de deformação resultante do algoritmo proposto apresentou valores semelhantes e, em alguns casos, ligeiramente maiores ou menores do que os métodos tradicionais. Os resultados foram bons para o problema

Figura 43 – Variação do fator de cristalização em relação à temperatura para a OT mono-objetivo de viga engastada



Fonte: O Autor.

Figura 44 – Resultados de OT do problema *Half-MBB Beam*

(a) $FV = 0,5$

(b) $FV = 0,6$

(c) $FV = 0,7$

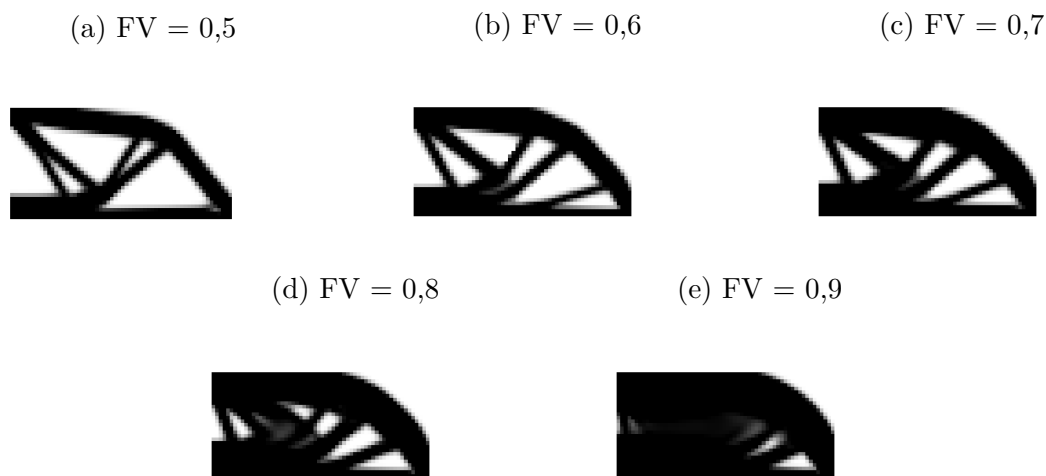


(d) $FV = 0,8$

(e) $FV = 0,9$



Figura 45 – Resultados da OT pelo método SIMP do código Andreassen et al. (2011) para problema *Half-MBB beam*



Fonte: O Autor.

Cantilever beam quanto para o problema *Half-MBB beam*.

Os valores do problema *Cantilever beam* apresentou uma diferença de cerca de até 2 %, dependendo do caso em relação aos resultados do algoritmo tradicional. A diferença dos valores obtidos no algoritmo proposto e da literatura para o problema *Half-Mbb beam* foi cerca de 3% para $FV = 0,5$.

Tabela 15 – Energia de deformação obtida na otimização topológica pelo algoritmo proposto

FV	<i>Half MBB beam</i>			<i>Cantilever beam</i>		
	Alg. Proposto	literatura	diferença [%]	Alg. Proposto	literatura	diferença [%]
0,5	78,34	81,29	-3,77%	71,40	72,97	-2,2%
0,6	68,20	68,26	-0,09%	61,35	62,24	-1,45%
0,7	59,10	59,97	-1,47%	54,09	54,82	-1,35%
0,8	54,72	54,48	0,44%	49,25	49,62	-0,75%
0,9	50,43	50,48	-0,10%	46,16	46,24	-0,17%

Fonte: O autor.

Os resultados obtidos foram com o parâmetro $r = 1,5$, para determinar a influência do raio nos resultados, rodamos o algoritmo para diferentes raios no problema *Cantilever beam*. Dependendo da quantidade de elementos na malha, o desempenho pode melhorar ou piorar com a modificação do parâmetro r da região selecionada. Neste estudo foi utilizada a malha de $nelx = 60$, $nely = 30$, $N = 5000$ e $FV = 0,5$, e obteve o melhor resultado com $r = 1,5$ em relação aos valores da energia de deformação. Os resultados da distribuição com a variação do parâmetro r são apresentados na Fig. 46. Com a mudança dos valores

de r , verificamos que alguns parâmetros, como número de iteração, não são os ideais, sendo necessário um estudo de sensibilidade dos parâmetros.

Em relação à distribuição de massa no domínio, verificamos que com o aumento do raio r , a quantidade de furos diminui e aumenta o tamanho dos furos. Portanto, com o r maior, obtemos geometria com menor complexidade. Porém, os valores da energia de deformação aumenta consideravelmente.

Figura 46 – Resultados da distribuição de massa com a variação do r para o problema *Cantilever Beam*

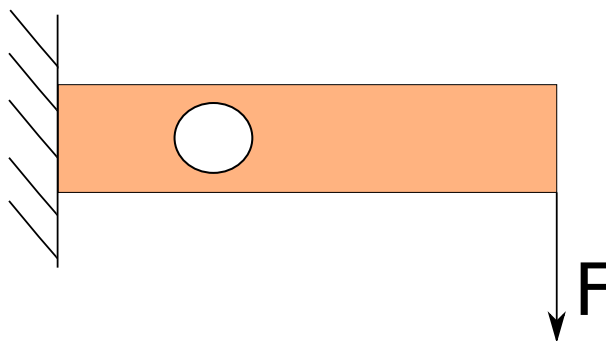
(a) $r = 2$ e $J_i = 71,4545$ (b) $r = 3$ e $J_i = 80,6148$ (c) $r = 4$ e $J_i = 85,4343$



Fonte: O Autor.

Com algoritmo proposto no Algoritmo 9 aplicamos em outros problemas adicionais relacionados a vigas. O primeiro problema é a viga engastada, *Cantilever beam*, com um orifício circular. Este problema é apresentado no diagrama da Fig. 47.

Figura 47 – Viga engastada do tipo *Cantilever Beam* com um orifício circular



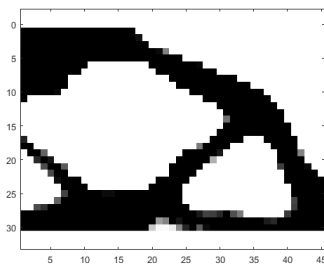
Fonte: O Autor.

Os parâmetros utilizados no processo de otimização foram $nex = 45$, $nely = 30$, $r = 2$, $penal = 3$, $T_{max} = 200$, $T_{min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. Utilizamos a malha com as mesmas dimensões e o mesmo valor do fator de penalidade utilizado por Pardo (2009). Pardo (2009) realizou o processo de otimização com restrição a fração volumétrica de $FV = 0,5$, em nosso caso realizamos a otimização para várias frações volumétricas, variando de $FV = 0,4$ até $FV = 0,7$. Em frações volumétricas acima de $FV = 0,7$ os valores da energia da deformação e a distribuição de massa não tem modificações significativas.

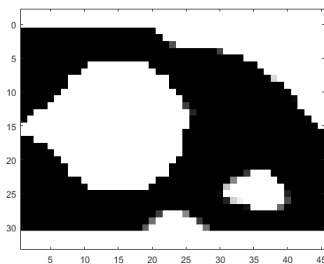
Os resultados da OT com algoritmo proposto são apresentados na Fig. 48, a distribuição em massa no domínio foi semelhante ao resultado obtido por Pardo (2009). A distribuição de massa possui furos com cantos mais arredondados em comparação ao algoritmo baseado em método gradiente. Em geral, os resultados da distribuição de massa foram satisfatórios e comparáveis aos métodos baseado em gradiente e ao método baseado em quimiotaxia de bactérias apresentados por Pardo (2009). Optamos por apresentar os resultados para a FV entre 0,4 a 0,7. Valores de FV acima de 0,7 a viga é totalmente preenchida com por massa, com exceção do furo.

Figura 48 – Resultados do processo de OT na viga *Cantilever Beam* com um orifício

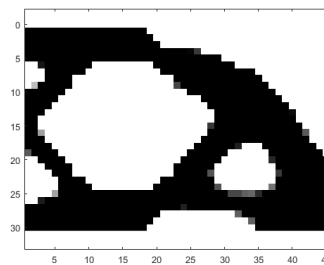
(a) $FV = 0,4$.



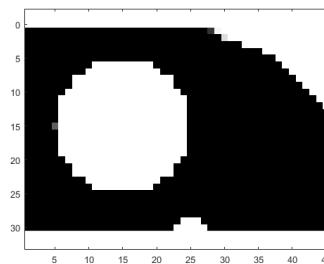
(c) $FV = 0,6$



(b) $FV = 0,5$.



(d) $FV = 0,7$



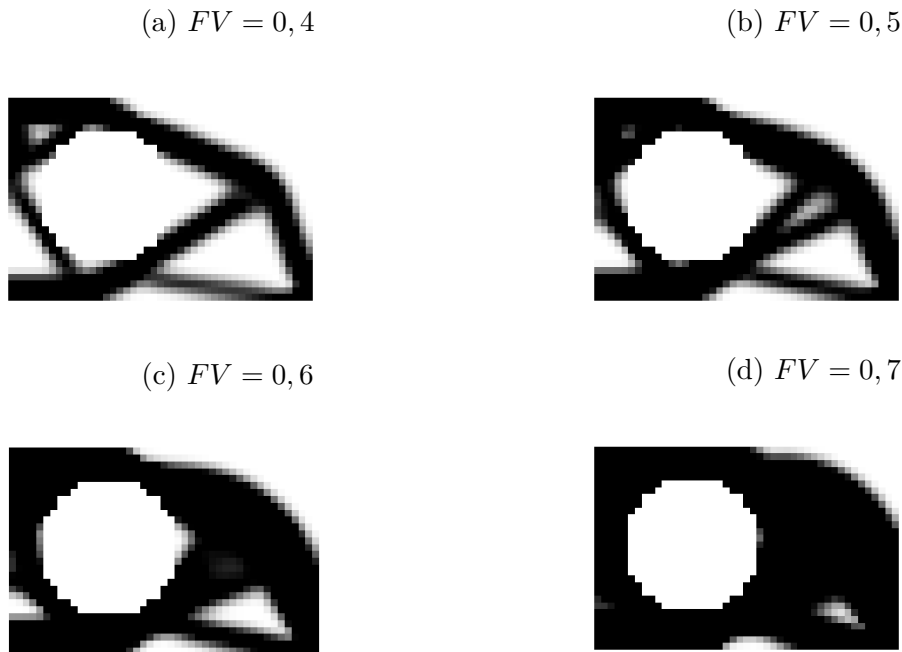
Fonte: O Autor.

A distribuição de massa obtida com o código top88 de Andreassen et al. (2011) é apresentado na Fig. 49 para fins de comparação.

Os valores da energia de deformação obtidos são apresentados na Tabela 16, em conjunto adicionamos os valores obtidos com a otimização utilizando o código de Andreassen et al. (2011) com o método baseado em método gradiente. Não foi possível comparar com a técnica utilizada por Pardo (2009), pois não foi apresentado o valor da energia de deformação.

O código de Andreassen et al. (2011), método gradiente, foi utilizado com os seguintes parâmetros: $nelx = 45$, $nely = 30$, $penal = 3$, $r_{\min} = 2$ e $ft = 1$. Os valores da energia de deformação obtidos com o algoritmo proposto foram menores que os valores obtidos com o código de Andreassen et al. (2011). Os valores do algoritmo proposto foram melhores para todos os casos de fração volumétrica. Portanto, o algoritmo proposto

Figura 49 – Resultados obtidos com o código top88.m de Andreassen et al. (2011) para a viga engastada com furo.



Fonte: O Autor.

Tabela 16 – Energia de deformação obtida no processo de OT da viga engastada com orifício

FV	Algoritmo Proposto	literatura	diferença [%]
0,4	61,82	70,51	-14,06
0,5	52,05	58,21	-11,83
0,6	48,20	53,53	-11,06
0,7	46,65	51,79	-11,02

Fonte: O autor.

apresentou desempenho competitivo e comparável aos métodos tradicionais.

O segundo problema de OT adicional é uma viga engastada com uma força aplicada no centro da ponta conforme apresentado na Fig. 50.

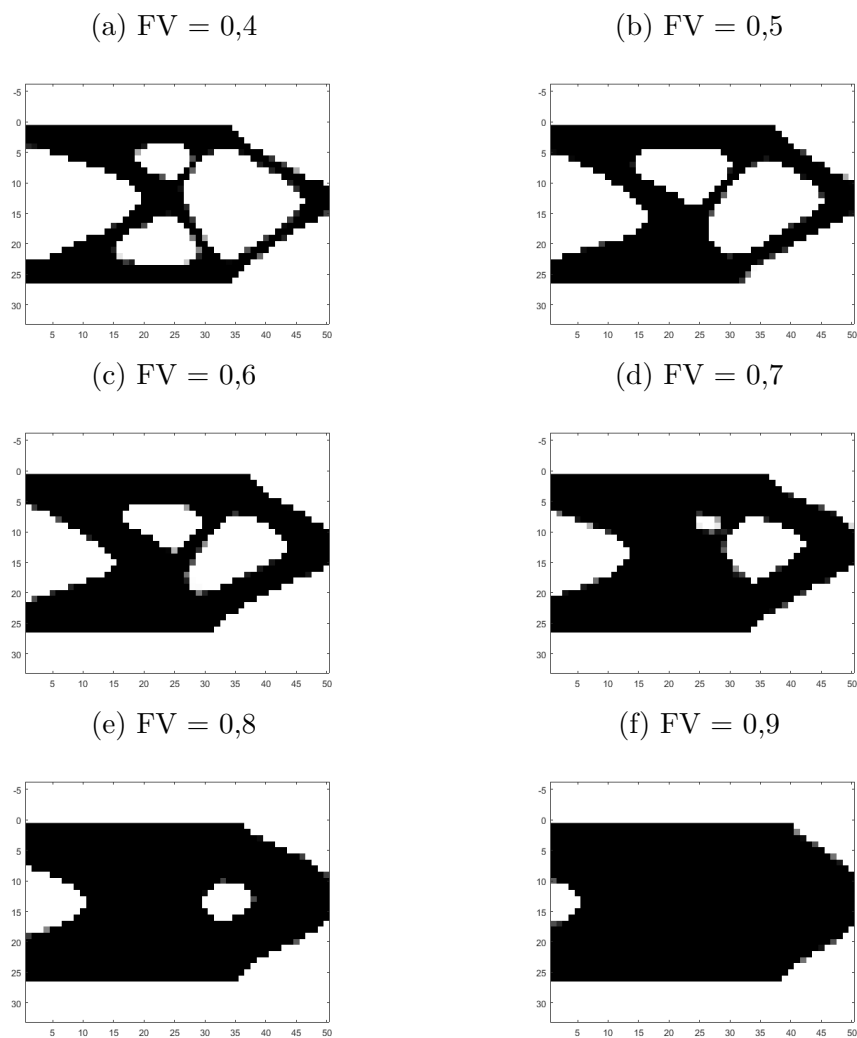
Os parâmetros utilizados no processo de otimização foram $nelx = 50$, $nely = 26$, $r = 2$, $penal = 3$, $T_{max} = 200$, $T_{min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. Utilizamos a malha com as mesmas dimensões e o mesmo valor do fator de penalidade utilizado por Pardo (2009). Nos trabalhos de Pardo (2009) foi realizado o processo de otimização com restrição para a fração volumétrica de $FV = 0,5$, em nosso caso realizamos a otimização para várias frações volumétricas, variando de $FV = 0,4$ à $FV = 0,9$. Os resultados da OT são apresentados na Fig. 51.

Figura 50 – Problema de OT em uma viga *Cantilever Beam* com aplicação da força no centro da ponta



Fonte: O Autor.

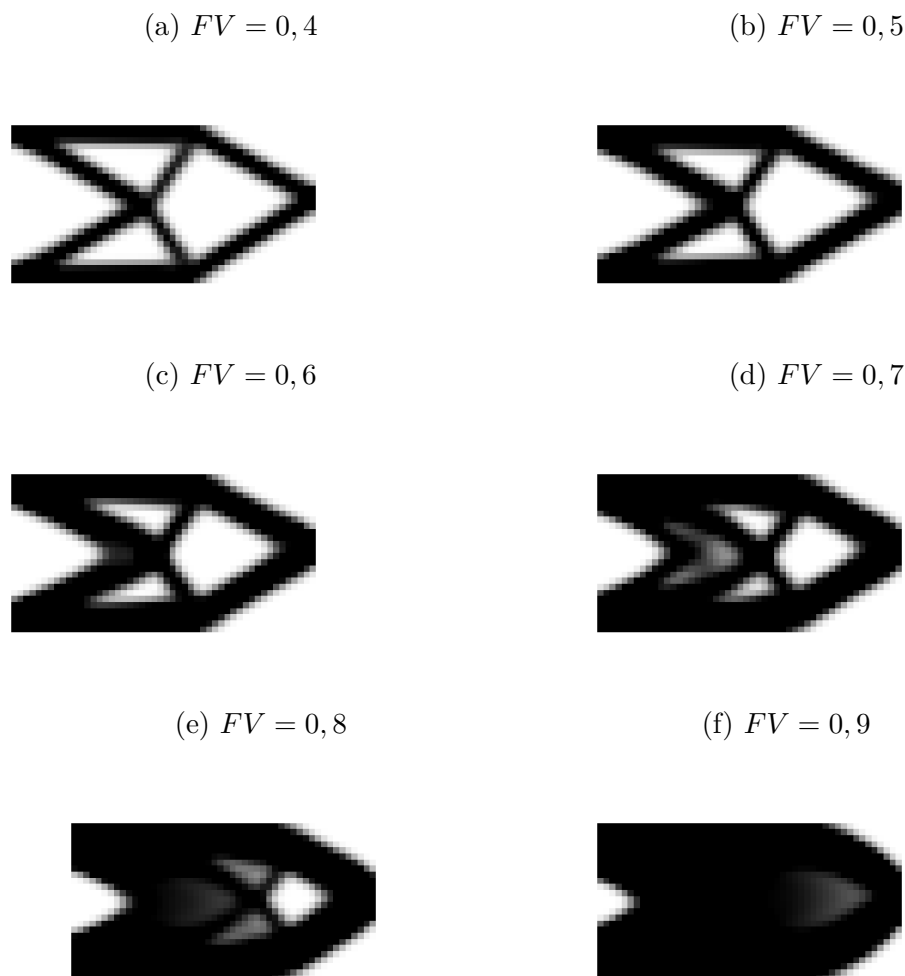
Figura 51 – Resultados da distribuição de massa da otimização TO do algoritmo proposto em uma viga engastada com a aplicação no centro da ponta



Fonte: O Autor.

A distribuição de massa obtida com a OT com o código top88 de Andreassen et al. (2011) para cada fração volumétrica é apresentada na Fig. 52. Com essas figuras é possível realizar uma comparação visual com os resultados obtidos com o algoritmo proposto.

Figura 52 – Resultados obtidos com o código top88.m de Andreassen et al. (2011) para a viga engastada com furo.



Fonte: O Autor.

A distribuição de massa no domínio obtida no processo de OT pelo algoritmo proposto foi satisfatória. Porém, a distribuição de massa possui cantos mais arredondados, menos furos e a distribuição é menos simétrica em comparação aos resultados obtidos pelo algoritmo de Andreassen et al. (2011), que é aprensato em Fig. 52.

Os valores da energia de deformação obtidos são apresentados na Tabela 17, em conjunto adicionamos os valores obtidos com a otimização utilizando o código de Andreassen et al. (2011) com o método baseado em método gradiente. Os parâmetros utilizados foram semelhantes ao problema anterior, diferenciando apenas a quantidade de elementos que foram $nelx = 50$ e $nely = 26$. Não foi possível comparar com a técnica

utilizada por Pardo (2009), pois não foi apresentado o valor da energia de deformação.

Tabela 17 – Valores da energia de deformação resultante do processo de OT da viga engastada com força aplicada no centro da extremidade da viga, os valores de menores são melhores

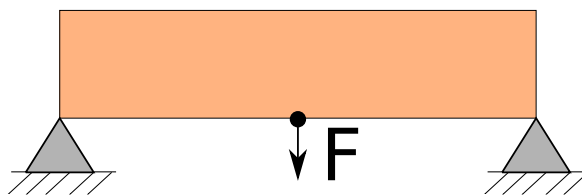
FV	Algoritmo Proposto	literatura	diferença [%]
0,4	77,13	81,80	- 6,05
0,5	59,83	62,88	-5,10
0,6	49,59	52,29	-5,44
0,7	43,65	45,27	-3,71
0,8	39,34	40,19	-2,16
0,9	36,61	36,82	-0,57

Fonte: O autor.

Em relação aos valores da energia de deformação, o algoritmo proposto obteve soluções com menor valor de energia de deformação para a mesma fração volumétrica. Os valores menores foram obtidos em todos os casos de fração volumétrica, conforme observamos na Tabela 17. As diferenças no valor da energia de deformação são expressivas e demonstram que o algoritmo proposto é competitivo em relação aos métodos tradicionais.

Outro problema de otimização topológica é a viga bi-apoiada com aplicação da força no centro da mesma, conforme a Fig. 53. Neste problema também foi realizado a otimização pelo algoritmo proposto e com o código utilizado por Andreassen et al. (2011).

Figura 53 – Problema de otimização de viga bi-apoiada com força no centro da barra

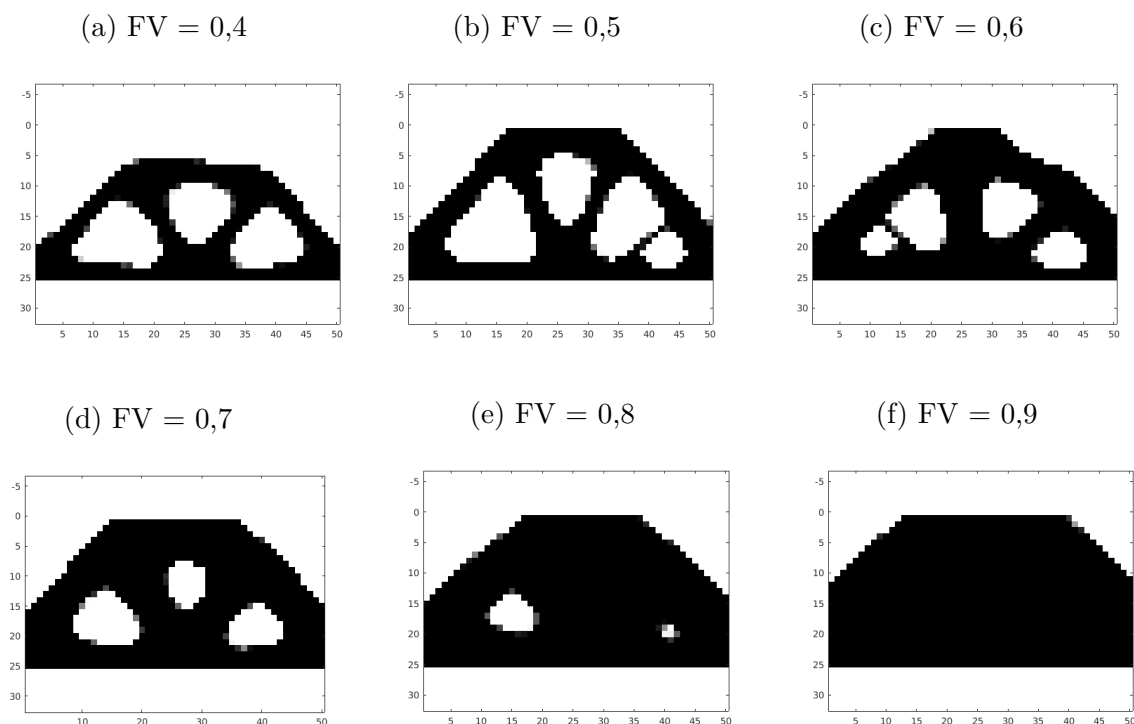


Fonte: O Autor.

Os parâmetros utilizados no processo de otimização no algoritmo proposto foram $nex = 50$, $nely = 25$, $r = 2$, $penal = 3$, $T_{max} = 200$, $T_{min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. Utilizamos a malha com as mesmas dimensões e o mesmo valor do fator de penalidade utilizado por Pardo (2009). Nos trabalhos de Pardo (2009) foi realizado o processo de otimização com restrição a fração volumétrica de $FV = 0,5$, em nosso caso realizamos a otimização para várias frações volumétricas, variando de $FV = 0,4$ até $FV = 0,9$. Os resultados da OT são apresentados na Fig. 54.

Assim como nos problemas anteriores, a distribuição da massa no domínio resultante apresentou menos furos com cantos mais arredondos. A distribuição da massa

Figura 54 – Resultados da distribuição de massa do processo de OT com o algoritmo proposto para a viga bi-apoiada



Fonte: O Autor.

também apresentou uma assimetria, diferente da solução obtida pelo método gradiente de Andreassen et al. (2011).

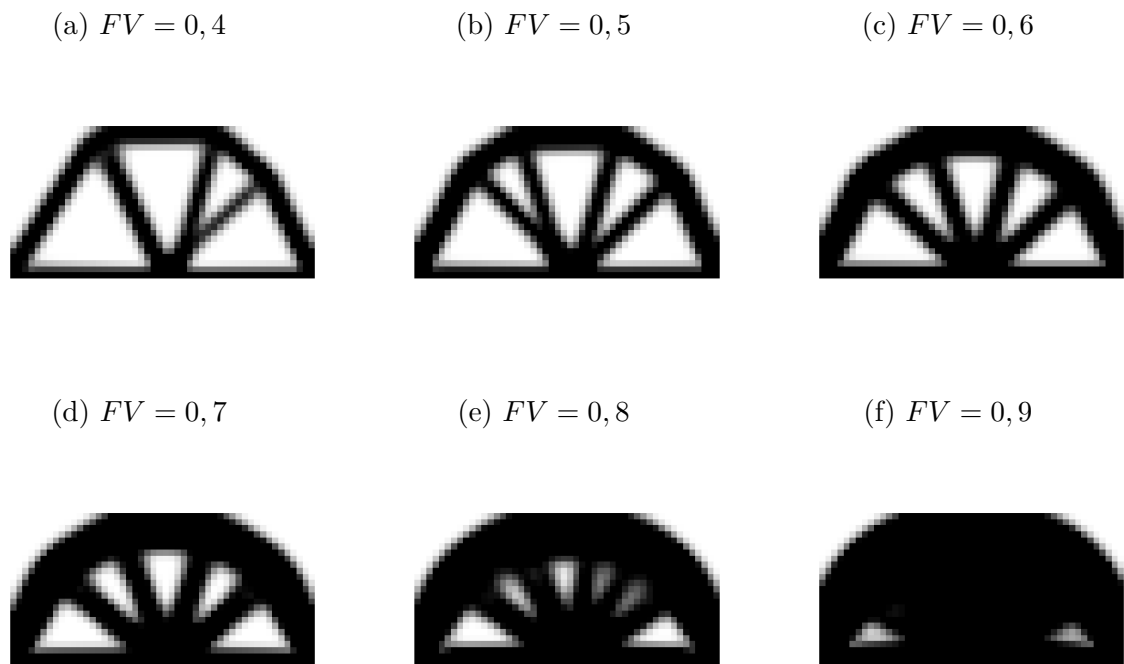
Os valores da energia de deformação obtidos são apresentadas na Tabela 17, em conjunto adicionamos os valores obtidos com a otimização utilizando o código de Andreassen et al. (2011) com o método baseado em método gradiente. Não foi possível comparar com a técnica utilizada por Pardo (2009), pois não foi apresentado o valor da energia de deformação.

Tabela 18 – Valores da energia da deformação do processo de OT da viga bi-apoiada

FV	Algoritmo Proposto	literatura	diferença [%]
0,4	28,42	19,56	31,18
0,5	18,08	14,93	17,42
0,6	13,47	12,37	8,17
0,7	11,21	10,92	2,59
0,8	9,95	9,92	0,3
0,9	9,20	9,20	0

Fonte: O autor.

Figura 55 – Resultados da distribuição de massa pelo algoritmo top88 de Andreassen et al. (2011) para o problema de viga bi-apoiada



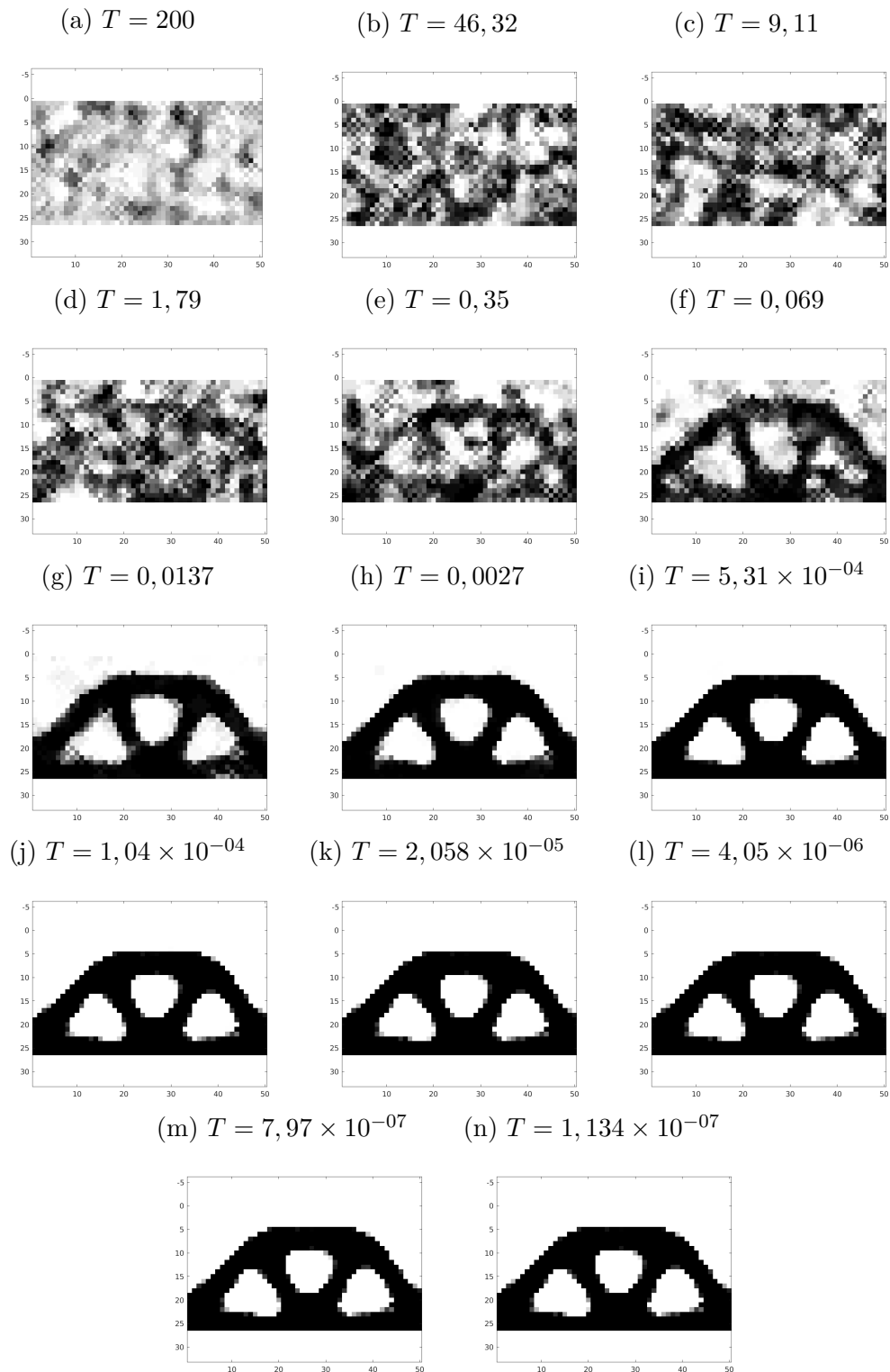
Fonte: O Autor.

Os valores da energia de deformação obtidos no processo de otimização da viga bi-apoiada foram atípicos em relação aos problemas anteriores. O valor obtido para fração volumétrica $FV = 0,4$ é muito maior no algoritmo proposto em relação ao valor da literatura, a diferença foi de 31%, reduzindo a diferença até se igualar na $FV = 0,9$. Na $FV = 0,9$, a distribuição de massa apresentou uma geometria sólida sem furos conforme a Fig. 54f. Enquanto no método gradiente, a distribuição de massa possui uma curvatura mais suave na borda superior e pequenos furos na geometria.

Para análise da convergência, foi realizado OT no problema de viga bi-apoiado com parâmetros semelhantes ao caso anterior. Foi armazenado a distribuição de massa, no domínio ao final de cada loop da temperatura. A convergência ao longo do decaimento da temperatura é apresentada na Fig. 56.

O algoritmo proposto apresentou como resultado desempenho satisfatório em relação à distribuição de massa e valores de energia de deformação. O algoritmo proposto é do tipo não-gradiente e foi necessária uma grande quantidade de iterações. Portanto, o tempo de execução do algoritmo proposto é muito maior em relação ao método gradiente. Quanto maior a quantidade de elementos, maior será a quantidade de iterações necessária para convergir. Ainda assim, o método proposto possui a vantagem de não necessitar da análise de sensibilidade. Essa análise de sensibilidade pode ser complicada de ser resolvida

Figura 56 – Otimização TO da viga bi-apoiada, com os parâmetros de $FV = 0,4$, $N = 2000$, $T_{\max} = 200$, $T_{\min} = 1,0 \times 10^{-7}$ e $\alpha = 0,85$



Fonte: O Autor.

para alguns problemas, dependendo da geometria ou função objetivo.

Embora o tempo de execução do algoritmo proposto seja muito maior que os algoritmos tradicionais, o algoritmo é promissor. O algoritmo pode ser mais flexível em relação a restrições e geometrias, e o tempo de execução pode ser melhorado utilizando técnicas de computação paralela ou outras técnicas.

6.3 OT Mono-Objetivo com Heurística de Raio Variável

Com os resultados do algoritmo anterior, propusemos um novo algoritmo baseado no Algoritmo 9 com uma mudança na dinâmica de seleção da região. Nesta nova proposta, é adicionado um novo fator que representa o raio da região circular variando ao longo do processo de otimização.

Esse fator se comporta semelhantemente ao fator de cristalização, o raio da região circular aumenta ou diminui conforme a aceitação ou rejeição da solução. Todo elemento da malha possui seu próprio fator R_k . E no processo de geração da nova solução, é sorteado aleatoriamente um elemento (variável decisão) e em seguida é selecionada uma região em formato circular conforme o fator R_k daquele elemento. A modificação para gerar a nova solução é semelhante à proposta anterior, e está descrita no Algoritmo 10.

Algoritmo 10: GERAÇÃO DA NOVA SOLUÇÃO COM O RAIOS VARIÁVEL.

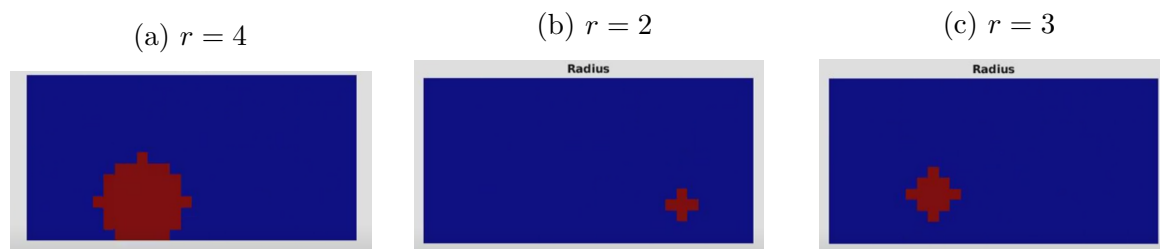
Entrada: x_a : Solução atual,
 C : Fator de cristalização,
 C_{\max} : Fator de cristalização máximo,
 R : Fator de raio,
 FV : Fração volumétrica

- 1 **enquanto** $x \in [x_{\min}, x_{\max}]$ e $FV_n \leq FV$ **faça**
- 2 $x_n \leftarrow x_a$
- 3 $k \leftarrow$ seleciona aleatoriamente a variável de decisão a ser modificado
- 4 **se** $C_k \leq C_{\max}$ **então**
- 5 $\Delta \leftarrow \frac{1}{C_k} \sum_1^{C_k} random(-1, 1) \cdot \Delta r_k \cdot e_k$
- 6 **senão**
- 7 $\Delta \leftarrow GaussianDist(\exp(C_{\max} - C_k - 2)) \cdot \Delta r_k \cdot e_k$
- 8 **fim**
- 9 **para** $k \in$ Região selecionada com raio de distância R_k **faça**
- 10 $x_{n,k} \leftarrow x_{n,k} + \Delta$
- 11 $x_{n,k} \leftarrow$ corrigir os valores
- 12 **fim**
- 13 $FV_n = FV(x_n)$
- 14 **fim**

Após a geração da nova solução, esta é analisada conforme os critérios do SA. Em caso de a nova solução ser aceita, é realizado um feedback positivo e em caso de ser

rejeitada é aplicado um feedback negativo. O feedback positivo é acrescido 1 no valor do R_k , e o feedback negativo reduz o fator R_k por -1. Na próxima iteração, quando um novo elemento é selecionado no processo de gerar a nova solução, o algoritmo utilizará o valor de R_k como raio da região de seleção dos elementos da vizinhança. Essa mudança do raio, aumentando ou diminuindo a região de seleção, é apresentada na Fig. 57. As figuras demonstram a variação da região de seleção ao longo do processo de otimização, a primeira figura apresenta a região já com seu tamanho R_{\max} , enquanto as figuras seguintes com raios menores.

Figura 57 – Variação da região de seleção das variáveis de decisões na OT com SA



O comportamento da geração da nova solução para este algoritmo é muito semelhante ao Algoritmo 8, porém o raio é obtido do fator R_k . Em relação às etapas do algoritmo, também muito semelhante ao algoritmo anterior, com pequenas mudanças com a adição do novo fator e o feedback positivo e negativo deste novo fator. Essas modificações são apresentadas no Algoritmo 11, onde podemos observar a resposta positiva (incremento de 1) quando a nova solução é aceita. Quando a solução é rejeitada, a resposta é negativa (decrece 1), ou seja, diminuímos o tamanho da região na próxima vez que o elemento for escolhido para gerar a nova solução.

6.4 Resultados do OT Mono-Objetivo com Heurística de Raio Variável

Para validar o desempenho do algoritmo foi necessário implementá-lo nos problemas comuns encontrados na literatura, ou seja, os problemas utilizados na Seção 6.2. E os parâmetros utilizados foram semelhantes ao da seção Seção 6.2, $nelx = 60$, $nely = 30$, $\alpha = 0,85$, $N = 5000$, $penal = 3$, $T_{\max} = 200$ e $T_{\min} = 0,0000001$.

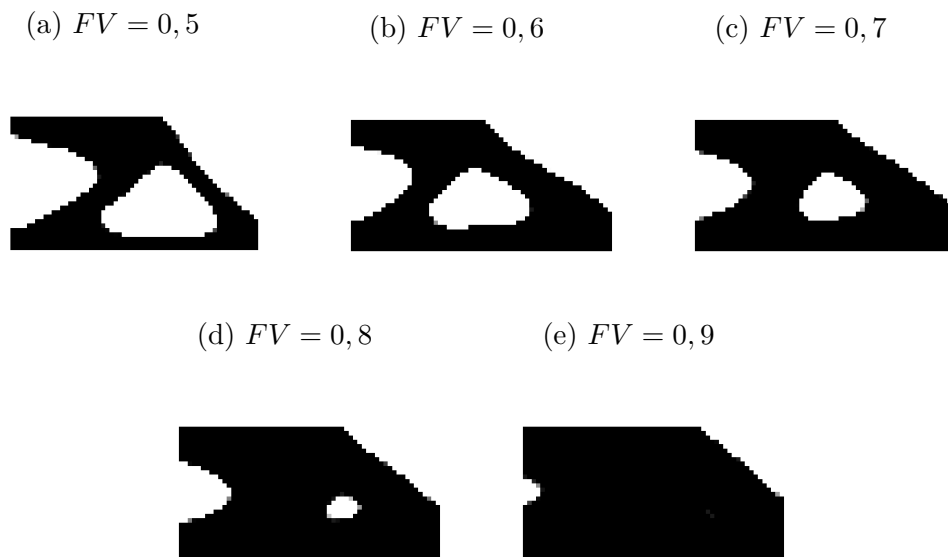
Os resultados com a distribuição de massa no domínio do problema *Cantilever problem* são apresentados na Fig. 58. Os resultados não apresentaram regiões de gradiente de cinza ou padrão tabuleiro de xadrez e a distribuição apresentou menos furos e uma geometria diferente do algoritmo proposto anteriormente ou do algoritmo top88.

Já os resultados para o problema *Half-MBB beam* são apresentados na Fig. 59. Os resultados foram semelhantes ao problema anterior com distribuição de massa concentrada,

Algoritmo 11: Otimização topológica mono-objetivo com raio variável

Entrada: T_0 : Temperatura inicial,
 α : Fator de decaimento de temperatura,
 C_{\max} : Fator de cristalização máxima,
 FV : Fração volumétrica,
 N : número de iterações,

- 1 $T_j \leftarrow T_0$
- 2 $C_k \leftarrow 1$; /* Inicializa o fator de cristalização para todas váriaiveis de decisão com valor 1 */
- 3 $R_k \leftarrow 1$; /* Inicializa o fator de raio para todas váriaiveis de decisão com valor 1 */
- 4 $X_a \leftarrow$ gerar uma nova solução
- 5 $F(X_a) \leftarrow$ Avaliação da função objetiva
- 6 **enquanto** $T_j > T_{\min}$ **faça**
- 7 **enquanto** *Enquanto o critério não for satisfeito faça*
- 8 $X_n \leftarrow$ Nova solução gerada apartir da solução X_a
- 9 $F(X_n) \leftarrow$ Avaliação da função objetiva
- 10 $\Delta E = F(X_n) - F(X_a)$
- 11 **se** $\Delta E \leq 0$ **então**
- 12 $X_a \leftarrow X_n$
- 13 $C_k \leftarrow$ <Realimentação positiva> ; /* De acordo com a estratégia escolhida da Tabela 2 */
- 14 $R_k \leftarrow R_k + 1$; /* Realimentação positiva */
- 15 **senão**
- 16 **se** $\text{random}(0,1) < e^{-\frac{\Delta E}{kT}}$ **então**
- 17 $X_a \leftarrow X_n$
- 18 $C_k \leftarrow$ <Realimentação positiva> ; /* De acordo com a estratégia escolhida da Tabela 2 */
- 19 $R_k \leftarrow R_k + 1$; /* Realimentação positiva */
- 20 **senão**
- 21 X_n é rejeitado
- 22 $C_k \leftarrow$ <Realimentação negativa> ; /* De acordo com a estratégia escolhida da Tabela 2 */
- 23 $R_k \leftarrow R_k - 1$; /* Realimentação negativa */
- 24 **fim**
- 25 **fim**
- 26 **fim**
- 27 $T_{j+1} \leftarrow T_j \times \alpha$
- 28 **fim**

Figura 58 – Resultados do problema *Cantilever beam* com algoritmo com variação do raio

Fonte: O Autor.

com menos furos e furos maiores. Mas neste problema, também não foram apresentadas regiões com o gradiente de cinza.

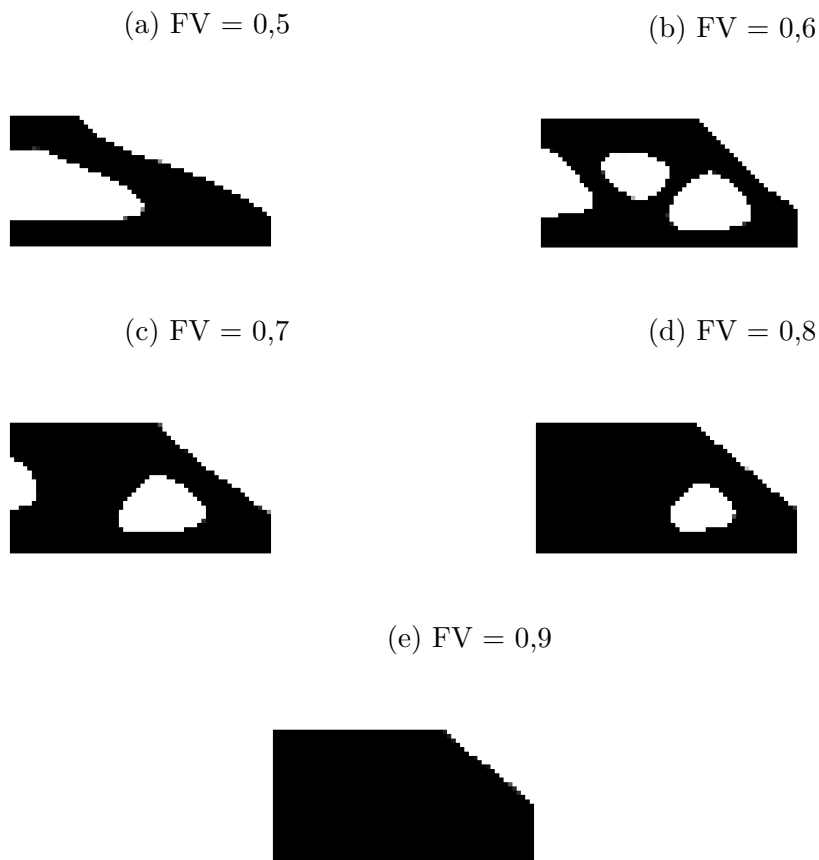
Os valores da energia da deformação obtidos com este algoritmo proposto são apresentados na Tabela 19, são apresentados os resultados do algoritmo proposto de raio variável e os valores do algoritmo de Andreassen et al. (2011) com os mesmos parâmetros mencionados anteriormente. A energia de deformação obtida no problema *Cantilever beam* foi próxima ao método GTO, com a maior diferença em torno de 1,5 %.

Já os valores obtidos para *Half-MBB beam* apresentaram uma diferença maior, cerca de 10 % para o caso $FV = 0,5$. Nos casos com fração volumétrica maior ocorreu uma diferença muito menor. Em média, a diferença da energia de deformação ficou em torno de 1,6%.

Tabela 19 – Resultados dos valores da energia de deformação das vigas.

FV	<i>Half MBB beam</i>			<i>Cantilever beam</i>		
	Algoritmo proposto	literatura	diferença [%]	Algoritmo Proposto	literatura	diferença [%]
0.5	91,16	81,29	10,83	74,12	72,97	1,55
0.6	68,83	68,26	0,1	62,92	62,24	1,08
0.7	60,96	59,97	1,62	54,88	54,82	0,11
0.8	54,67	54,48	0,35	49,61	49,62	-0,02
0.9	50,59	50,48	-7,69	46,36	46,24	0,26

Para o caso $FV = 0,5$ do *Half-MBB beam* a diferença no valor da energia de

Figura 59 – Resultados do problema *Half MBB beam* com algoritmo com variação do raio

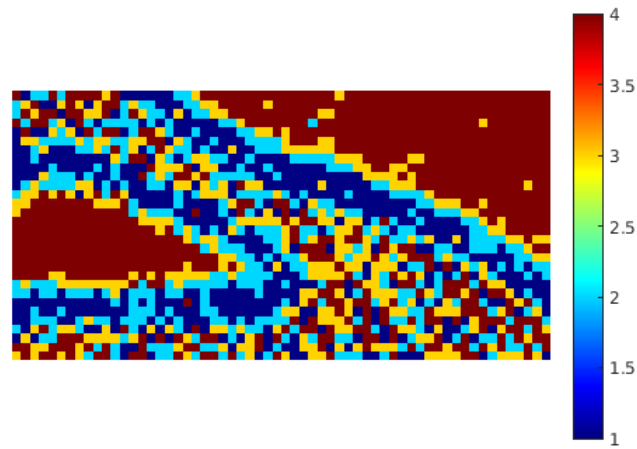
Fonte: O Autor.

deformação em relação ao da literatura foi maior. Portanto, verificamos os valores do fator R_k no final do processo de otimização, apresentado na Fig. 60. Na figura podemos observar que o valor do fator R_k é maior (valor $R_{k_{\max}}$) em regiões onde não apresenta massa. Na borda das concentrações de massas, apresentaram valores menores de R_k , significando uma maior rejeição das soluções nestas regiões.

Os resultados para este algoritmo proposto foram satisfatórios, ainda que os valores da energia de deformação seja maiores que os valores do algoritmo apresentado na Seção 6.1. Mas a distribuição de massa com uma geometria menos complexa e com poucos furos ou caminhos finos, pode ser uma vantagem para algumas aplicações. Em relação ao tempo de execução, foi semelhante ao algoritmo proposto da Seção 6.1 e muito mais lento em relação ao algoritmo de Andreassen et al. (2011).

Aplicamos o algoritmo proposto de raio variável a outros problemas adicionais relacionados a vigas. O primeiro problema é a viga engastada, conhecido como *Cantilever beam*, com um orifício circular. Este problema é apresentado no diagrama da Fig. 47.

Os parâmetros utilizados no processo de otimização foram $n_{elx} = 45$, $n_{ely} = 30$,

Figura 60 – Os valores do fator R_k no final do processo de otimização

Fonte: O Autor.

$penal = 3$, $T_{max} = 200$, $T_{min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. Utilizamos a malha com as mesmas dimensões e o mesmo valor do fator de penalidade utilizado no método anterior.

Os resultados da OT são apresentados na Fig. 61. A distribuição de massa obtida é satisfatória, no entanto, possui menos furos e cantos mais arredondados. As dimensões dos furos são maiores que os observados no método proposto anteriormente e nos resultados do algoritmo de Andreassen et al. (2011).

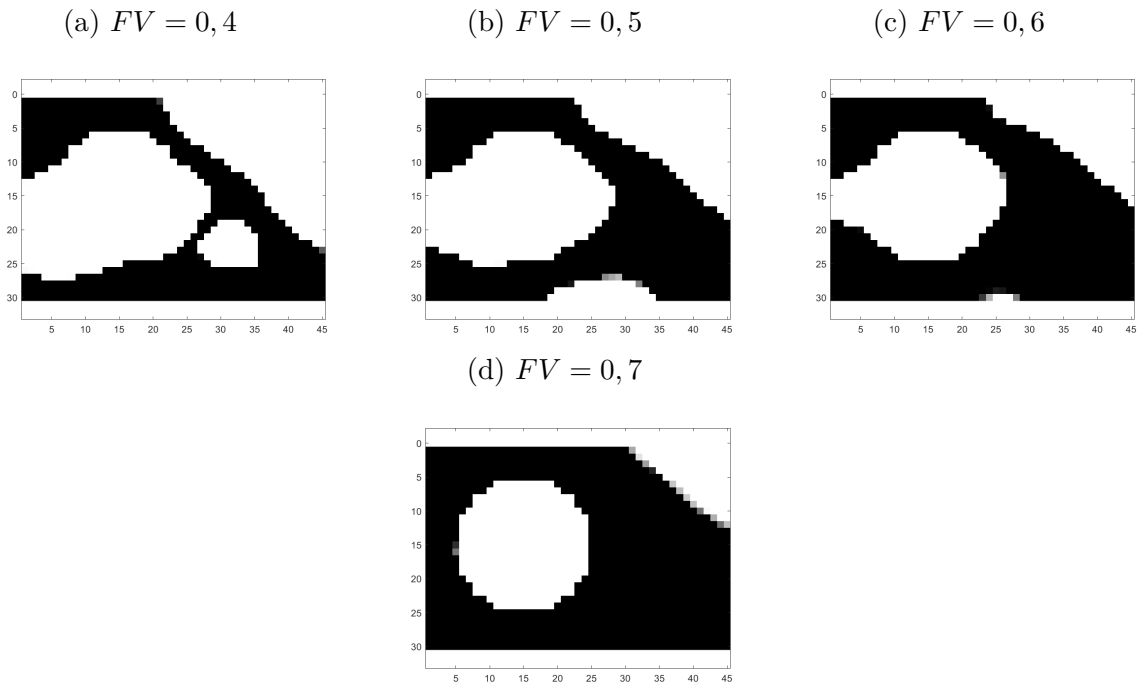
Os valores da energia de deformação otimizada obtida no processo de OT são apresentadas na Tabela 20. Os valores da energia de deformação obtidos por este método foram relativamente próximos aos obtidos com o método da literatura. Os valores obtidos na literatura estão apresentados na Tabela 16.

Tabela 20 – Valores da energia de deformação obtida no processo de OT da viga engastada com orifício.

FV	Energia de deformação	literatura	diferença [%]
0,4	74,91	70,51	5,87
0,5	58,02	58,21	-0,33
0,6	49,80	53,53	-7,49
0,7	46,81	51,79	-10,64
0,8	46,38	51,53	-11,10

Aplicamos o algoritmo no problema de viga engastada com uma força no centro da extremidade, esse problema é descrito na Fig. 50. Os parâmetros utilizados no processo

Figura 61 – Distribuição de massa obtida no processo de OT da viga engastada com orifício



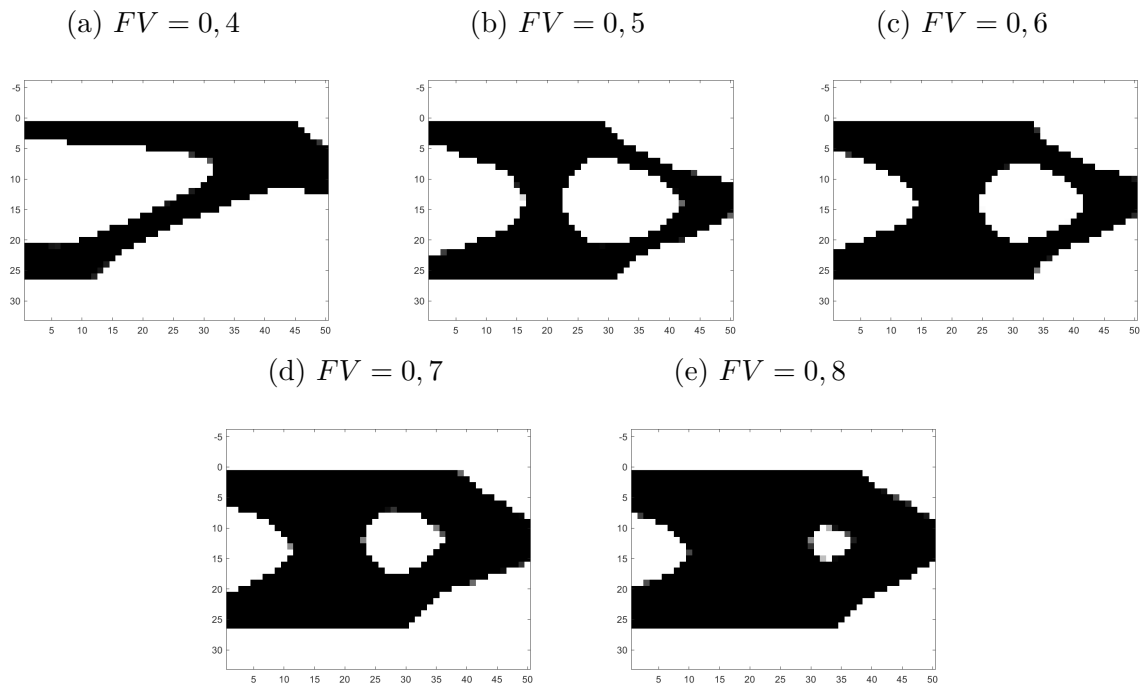
Fonte: O Autor.

de otimização foram $nelx = 45$, $nely = 30$, $penal = 3$, $T_{max} = 200$, $T_{min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. Utilizamos a malha com as mesmas dimensões e o mesmo valor do fator de penalidade utilizado no método anterior.

A distribuição de massa no domínio resultante do processo de OT é apresentada na Fig. 62. A distribuição de massa obtida é satisfatória, porém apresentou uma geometria diferente das obtidas com a proposta anterior e do algoritmo baseado em gradiente de Andreassen et al. (2011). A geometria obtida apresentou menor quantidade de furos e dimensões menores. As curvas e cantos com geometria mais arredondada e geometria não simétrica em comparação aos outros métodos.

Os valores da energia de deformação para todas as frações volumétricas são apresentadas na Tabela 21. Os valores também se apresentaram ligeiramente maior em relação ao método tradicional, exceto no caso de $FV = 0,4$ que o valor foi muito maior, em torno de 20% enquanto nos outros casos a diferença ficou em torno de 2,5%.

Figura 62 – A distribuição de massa obtida no processo de OT para a viga engastada com a força aplicada no centro da extremidade da viga.



Fonte: O Autor.

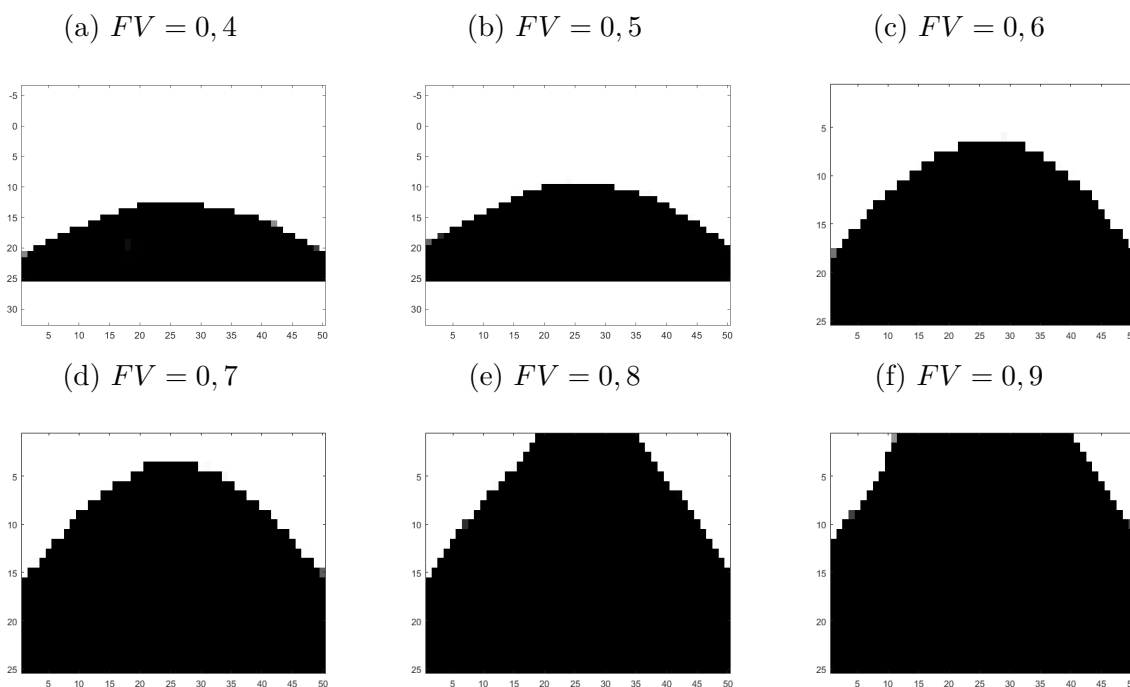
Tabela 21 – Valores de energia de deformação obtidos no processo TO para a viga engastada e carregamento no centro da ponta

FV	Alg. proposto	literatura	diferença [%]
0,4	100,02	81,8	18,22
0,5	64,50	62,88	2,51
0,6	51,77	52,29	-1,0
0,7	45,01	45,27	-0,58
0,8	39,73	40,19	-1,16
0,9	37,1	36,82	0,75

Fonte: O autor.

O último problema aplicado foi a viga bi-apoiada com a força no centro descrito na Fig. 53. Os parâmetros do algoritmo utilizados foram $nelx = 50$, $nely = 25$, $penal = 3$, $T_{\max} = 200$, $T_{\min} = 0,0000001$, $\alpha = 0,85$ e $N = 5000$. A distribuição de massa no domínio resultante do processo de OT é apresentada na Fig. 63. Apresentamos os resultados para cada fração volumétrica testada, $FV = 0,4$ até $FV = 0,9$.

Figura 63 – Resultados da distribuição de massa no domínio do processo de OT



Fonte: O Autor.

Os resultados da distribuição de massa obtidos apresentaram geometria muito diferente em relação aos métodos anteriores, com nenhum furo. A geometria sólida com uma curvatura na borda superior. Este formato geométrico reflete nos valores da energia de deformação obtida, como podemos observar na Tabela 22. Os valores da energia de deformação são muito maiores que os métodos tradicionais, apresentados na Tabela 18. Os valores da energia de deformação no algoritmo proposto foi 31% maior na $FV = 0,4$, e essa diferença decai conforme aumento da FV , chegando a 0,2 % na $FV = 0,8$ e valores iguais na $FV = 0,9$ onde a geometria é sólida em ambos resultados.

O algoritmo proposto nesta seção apresentou soluções otimizadas com valores maiores de energia de deformação comparado a técnica apresentada em Seção 6.1. Mas, em todos os casos, o limite do fator de raio de cada elemento era 4 e observamos que o valor se demonstrou inadequado para o tamanho da malha. Sendo necessário maiores estudos e testes para determinar os parâmetros ideais para o algoritmo.

Para análise da convergência, foi realizada OT no problema de viga bi-apoiada com parâmetros semelhantes ao caso anterior. Foi armazenada a distribuição de massa no

Tabela 22 – Energia de deformação obtida no processo de OT para o problema de viga bi-apoiada

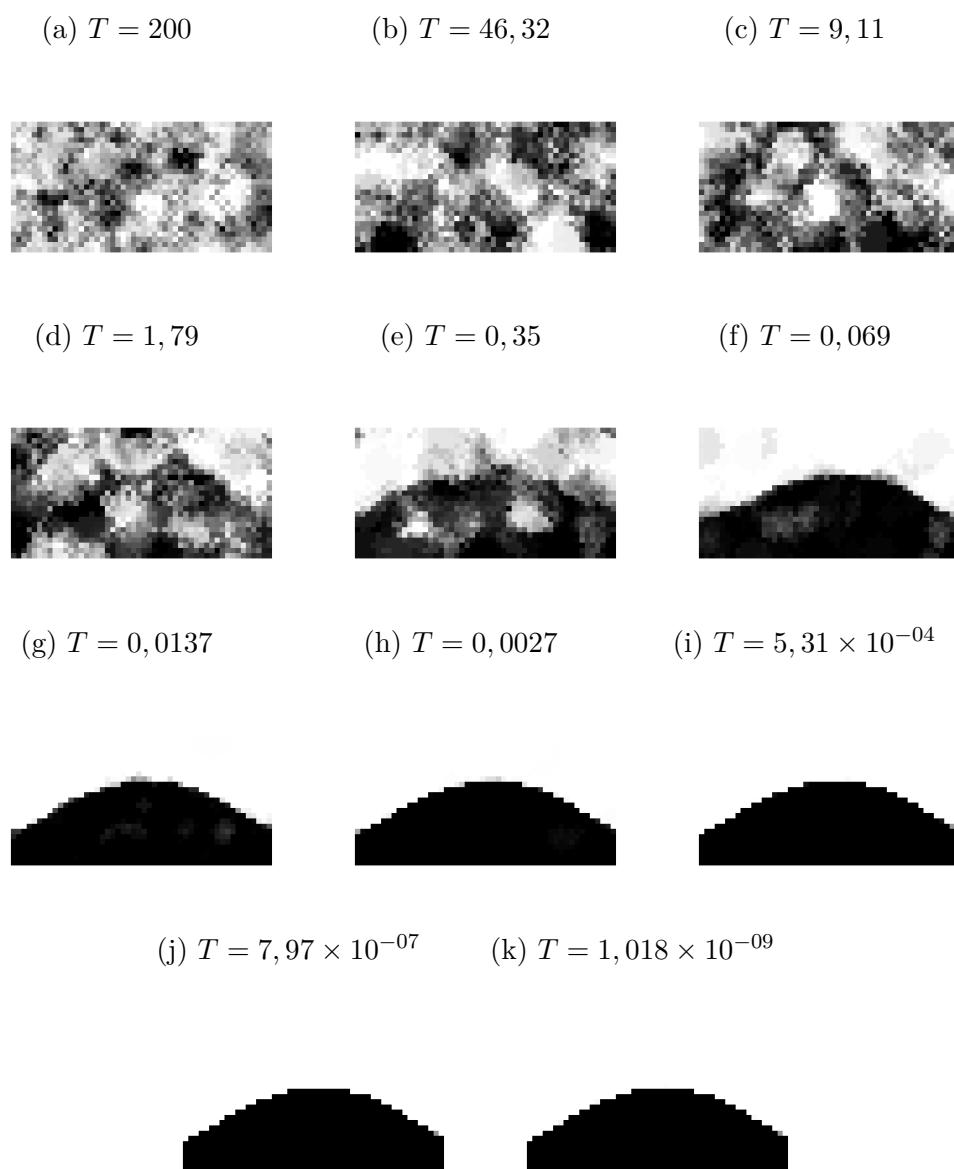
FV	Alg. proposto	literatura	diferença [%]
0,4	28,42	19,55	31,21
0,5	18,08	14,93	17,42
0,6	13,47	12,37	8,17
0,7	11,21	10,91	2,68
0,8	9,94	9,92	0,2
0,9	9,2	9,2	0

Fonte: O Autor.

domínio ao final de cada loop da temperatura. A convergência ao longo do decaimento da temperatura é apresentada na Fig. 64.

Como descrito anteriormente, a geometria das soluções são sólidas com menos furos e essas características podem ser uma vantagem em algumas aplicações e processos de fabricação. Como exemplo podemos citar os processos de fabricação por fundição ou forja, essas restrições de manufatura são descritas em Vatanabe et al. (2016). Outra vantagem, é que não foi observado gradientes de cinza no domínio.

Figura 64 – Otimização TO anterior com a mesma malha, e os parâmetros de $FV = 0,5$, $N = 5000$, $T_{\max} = 200$, $T_{\min} = 1,0 \times 10^{-9}$ e $\alpha = 0,85$



Fonte: O Autor.

6.5 OT Multiobjetivo

Muitos problemas de OT multiobjetivo são transformados em problemas mono-objetivos com funções de restrições. Os recursos computacionais para resolver o problema de OT mono-objetivo com restrições não é muito menor em relação aos problemas OT multiobjetivo, segundo Hamda, Roudenko e Schoenauer (2002). Além disso, na otimização multiobjetivo temos um conjunto de soluções ótimas (uma frente de Pareto), enquanto na mono-objetivo temos apenas uma solução e, dependendo do problema, é interessante possuir mais de uma solução ótima para auxiliar na fase de projeto do produto. Devido a isto, o desenvolvimento de métodos de otimização topológica multiobjetivo desperta o interesse dos pesquisadores.

Uma breve revisão sobre otimização topológica e os principais métodos foram discutidos em Seção 2.3, porém a revisão foi limitada aos métodos de otimização topológica para problemas de otimização mono-objetivo. Mas como discutido no Capítulo 4, em casos reais de otimização é desejada a otimização multiobjetivo, muitas vezes estas funções objetivo são conflitantes entre si. Podemos citar o exemplo dado por Zavala et al. (2014), de um projeto de uma ponte no qual é desejado reduzir os custos e, ao mesmo tempo, maximizar a segurança.

Os primeiros trabalhos relacionados a otimização estrutural com a otimização multiobjetivo são da década de 80, publicados por Goldberg e Samtani (1986 apud ZAVALA et al., 2014) e Kicinger, Arciszewski e Jong (2005). Uma ampla revisão bibliográfica sobre a otimização topológica multiobjetivo é discutida por Zavala et al. (2014), Kicinger, Arciszewski e Jong (2005). Nesses trabalhos são elencados os principais estudos de otimização estrutural multiobjetivo.

Na literatura é possível encontrar trabalhos de otimização topológica multiobjetivo que utilizam o método SIMP, ESO ou BESO com diversos algoritmos de otimização, como NSGA-II, SPEA2, colônia de formiga (ZULIANI et al., 2015), otimização por enxame de partículas (*Particle Swarm optimization* - PSO), recozimento simulado ou por meio da *escalarização*. Para análise das equações diferenciais, é possível encontrar trabalhos que utilizam técnica de elementos finitos (FEA), volumes finitos (FVM) (MARCK et al., 2012) ou ainda o método dos elementos de contorno (*boundary element method* - BEM) (SIMONETTI et al., 2019).

No caso da otimização topológica estrutural em meio contínuo, as funções de minimização podem ser a energia de deformação, o deslocamento ou deformação da estrutura, entre outras funções, matematicamente um problema de otimização topológica multiobjetivo pode ser representado matematicamente por

$$\begin{aligned} \text{Minimizar} \quad & F = [f_1(x), \dots, f_i(x)]^T \in \mathbb{R}^i \\ \text{Sujeito a :} \quad & x \in \Omega, \end{aligned} \tag{6.1}$$

onde Ω é o domínio, x é a variável decisão e $F(x)$ é a função de i , o qual pertence a \mathbb{R}^i .

Como exemplo de trabalho, podemos citar Hamda, Roudenko e Schoenauer (2002) que realizou a otimização topológica, no qual as funções objetivo foram o deslocamento máximo e o peso. O algoritmo de otimização utilizado foi o algoritmo genético NSGA-II e o método de OT foi o método de homogeneização e, ainda, foi utilizado o diagrama de Voronoy para determinar os sítios. O problema de otimização testado foi o *cantilever plate* e a análise numérica do deslocamento foi realizada por análise de elementos finitos. Os resultados das simulações foram comparados em relação a OT mono-objetiva com restrições aplicadas no mesmo problema.

Simonetti et al. (2019) publicaram sua pesquisa de otimização topológica utilizando algoritmo de otimização topológica ESO com a solução de equações diferenciais parciais *Boundary Element Method* para a solução de problemas de otimização topológica multiobjetivo. O método de otimização multiobjetivo utilizado por Simonetti et al. (2019) foi o Método da soma ponderada (*weighted summing methods*).

Neste trabalho propomos a OT com o CoAnnealing, apresentado na Seção 4.3, baseado em SA multiobjetivo. Para a solução do problema OT foi necessário propor uma nova heurística que será descrita nas próximas seções. As propostas apresentado neste trabalhos são OT discreta, ou seja, a pseudo densidade são ρ_{\min} ou valor igual a 1. A OT em meio discreto não é possível por meio dos métodos GTO sem alguma técnica ou pós-processamento.

6.6 OT Multiobjetivo com Heurística de Raio Variável

O CoAnnealing apresentou um bom desempenho na solução de funções de benchmark. Devido a este bom desempenho, aplicamos o algoritmo em problemas de OT. Porém, o CoAnnealing se trata de um algoritmo meta-heurístico dependente da aleatoriedade, exigindo uma grande quantidade de iterações. Problemas com malhas maiores aumentam substancialmente o número de iterações, sendo necessária de alguma técnica de seleção de elementos e busca local.

Nesta proposta utilizamos uma técnica semelhante ao descrito na Seção 6.3 na geração da nova solução em conjunto com o CoAnnealing para a OT multiobjetivo. O algoritmo proposto, a pseudo densidade ρ pode possuir o valor da densidade mínima ou material sólido com densidade $\rho = 1$. A modificação da solução é por meio da seleção aleatória de um elemento no domínio, então é selecionado aleatoriamente entre a densidade ρ e ρ_{\min} . A nova densidade do elemento escolhido é replicado nos elementos da vizinhança na região circular formada pelo raio R , conforme a Fig. 39. O passo a passo da geração da nova solução é descrito em Algoritmo 12.

Na geração da nova solução, inicializa a variável x_n com o valor de solução atual x_a , a fração volumétrica FV_n . Enquanto a fração volumétrica não está no intervalo desejado, é selecionado aleatoriamente uma variável de decisão. Em seguida é sorteado aleatoriamente o valor da nova pseudo densidade dessa variável. Sendo o valor de $\rho = 1$ ou $\rho_{\min} = 0,001$. Em seguida, são selecionadas as variáveis de decisão na vizinhança com distância de raio R_k . Esses elementos selecionados, são aplicados a pseudo densidade do elemento escolhido k . Em seguida é determinado a FV da nova solução.

Algoritmo 12: GERAÇÃO DA NOVA SOLUÇÃO DE RAIOS VARIÁVEL PARA O COANNEALING.

Entrada: x_a, R, FV

```

1  $FV_n \leftarrow$  inicializa
2  $x_n \leftarrow x_a$ 
3 enquanto  $FV_n \notin [FV_{\min}, FV_{\max}]$  faça
4    $k \leftarrow$  seleciona aleatoriamente a variável de decisão a ser modificado
5    $\rho_k \leftarrow \text{rand}(\rho_{\min} \text{ ou } \rho)$ 
6   para  $k \in$  Região selecionada com raio,  $R_k$  faça
7      $x_{n,k} \leftarrow \rho_k$ 
8   fim
9    $FV_n = FV(x_n)$  /* Calcula a nova FV */
10 fim
```

Este raio R varia de acordo com fator raio R_k , cada elemento do domínio possui seu fator R_k . E este fator varia de acordo com a aceitação ou rejeição das soluções. Quando a solução é aceita, o elemento k escolhido no processo de geração da nova solução recebe um incremento no fator $R_k = R_k + 1$, até atingir o valor máximo. Em caso de rejeição da nova solução, o fator R_k do elemento selecionado recebe um decréscimo unitário até atingir o valor 1. O fator de raio varia conforme a aceitação ou rejeição da solução.

O algoritmo proposto utilizando o algoritmo CoAnnealing em conjunto com a técnica de geração de nova solução é apresentado no Algoritmo 13. Este algoritmo proposto para OT multiobjetivo é baseado no algoritmo apresentado em Algoritmo 6, porém sem a heurística de cristalização por se tratar de um problema de OT em que a pseudo densidade pode possuir o valor da ρ_{\min} estipulado e a densidade sólida $\rho = 1$. O processo de otimização descrito no Algoritmo 14 segue os seguintes passos:

1. o algoritmo inicia com a entrada dos parâmetros $SL, HL, N, T_{\max}, T_{\min}, \alpha, \max_r$;
2. o algoritmo inicia o *Archive* e seleciona a solução x_a de forma aleatória do *Archive*;
3. uma nova solução é gerada com a função do Algoritmo 12;
4. o valor das funções objetivo é determinado com o cálculo dos elementos finitos;

5. se a condição $\Delta E \leq 0$ ou número aleatório entre 0 e 1 for menor que $\exp^{-\frac{\Delta E}{T_j}}$ a nova solução será aceita;
6. a solução atual x_a recebe o valor da nova solução x_n ;
7. se o valor de $MaxDom(X_n, Archive) \leq 0$, a solução pertence à frente de Pareto e a solução é adicionada ao *Archive*;
8. se o tamanho de *Archive* é maior que o limite superior SL , é realizada a clusterização até o tamanho do *Archive* atingir o limite inferior HL ;
9. o algoritmo verifica se a solução atual foi removida do *Archive*, caso tenha sido removida, ela é adicionada novamente e o contador r é incrementado até atingir o limite. Caso alcance o limite, a solução atual é descartada e outra solução é escolhida aleatoriamente do *Archive*;
10. enquanto a condição de equilíbrio térmico não é alcançada ou a quantidade de iteração N não é alcançada, repete-se o processo 3 a 9;
11. enquanto a temperatura T_j é maior que a T_{\min} , a temperatura é decrescida conforme o mecanismo utilizado e repete-se o processo do item 3 a 10;
12. se a temperatura T_j é menor que a temperatura T_{\min} , o algoritmo encerra o processo de otimização.

O algoritmo proposto possui o mesmo mecanismo de aceitação ou rejeição do CoAnnealing descrito na Seção 4.3, assim como o processo de clusterização. O algoritmo proposto pode ser aplicado com duas ou três funções objetivo, porém as funções objetivo devem ser selecionadas cuidadosamente.

6.7 Resultados do OT Multiobjetivo Aplicado a Vigas

O algoritmo proposto foi aplicado em problema com a viga engastada *Cantilever Beam* e a viga bi-apoiada *Half MBB Beam* com objetivo de minimizar a fração volumétrica e a energia de deformação. Este problema de OT pode ser descrito matematicamente por

$$\begin{aligned}
 \text{Minimizar :} \quad & f_1(x) = FV(x) \\
 & f_2(x) = C = U^T KU \\
 \text{Sujeito a :} \quad & x \in \Omega, \\
 & 0,20 \leq FV \leq 0,8.
 \end{aligned} \tag{6.2}$$

A avaliação da função objetivo para avaliar a energia de deformação é realizada por meio de elementos finitos. Devemos ressaltar que na análise de elementos finitos utilizamos

Algoritmo 13: Algoritmo de OT multiobjetivo com Coannealing.

```

Entrada:  $\alpha, SL, HL, T_{\max}, T_{\min}, \max_r$ 
/* Entrada dos parâmetros do CoAnnealing */
1  $Archive \leftarrow$  Solução inicial aleatória
2  $x_a \leftarrow$  Solução inicial aleatória
3  $r \leftarrow 0$ 
4  $T_j \leftarrow T_{\max}$ 
5  $R_k \leftarrow 1$  /* Inicializa o fator  $R_k$  com o valor 1 para todas variáveis
de decisão */
6 enquanto <Critério de parada global não é satisfeito> faça
7    $T_{j+1} \leftarrow T_j \cdot \alpha$  próxima temperatura do programa de resfriamento
8   enquanto <Critério de parada local não é satisfeito> faça
9      $x_n \leftarrow$  perturbação da solução  $x_a$ 
10     $\Delta E = MaxDom(x_n, Arch) - MaxDom(x_a, arch)$ 
11    se  $\Delta E \leq 0 \parallel Random(U(0, 1)) < \exp^{-\frac{\Delta E}{T_j}}$  então
12       $x_a \leftarrow x_n$ 
13       $R_k \leftarrow$  < Realimentação positiva >
14      se  $MaxDom(x_n, Arch) = 0$  então
15        < adiciona  $x_a$  em Archive >
16        se  $tamanho(Arch) > Limite$  então
17          Archive  $\leftarrow$  Clusterização(Archive) ; /* O algoritmo de
clusterização reduz a quantidade de soluções até HL
soluções */
18          se  $x_n \notin Arch$  então
19            se  $r < \max_r$  então
20              <Adiciona  $x_n$  em Archive>
21               $r \leftarrow r + 1$ 
22            senão
23               $x_n \leftarrow Random(Archive)$ 
24            fim
25          senão
26             $r \leftarrow 0$ 
27          fim
28        fim
29      fim
30    senão
31       $x_n$  é rejeitada
32       $R_k \leftarrow$  < realimentação negativa >
33    fim
34  fim
35 fim

```

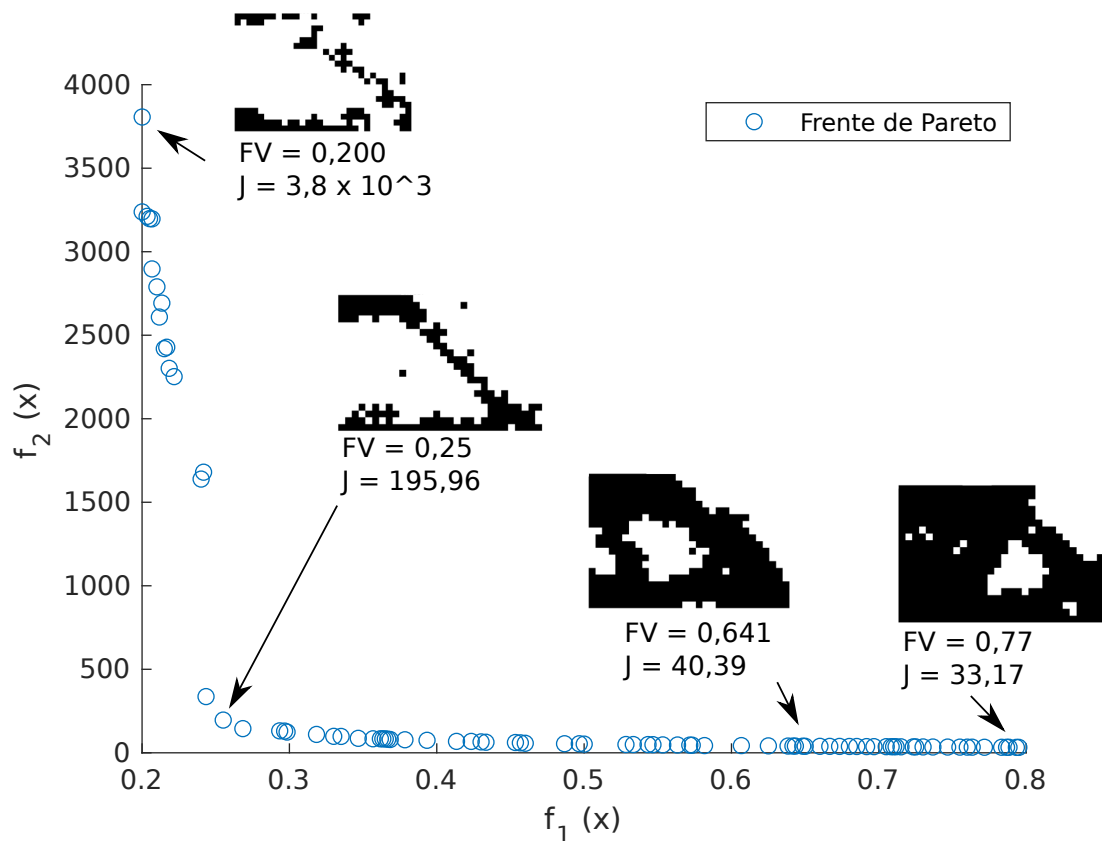
o conceito do método SIMP proposto por Bendsøe, Díaz e Kikuchi (1993b), descrito em Subseção 2.3.1. Nessa técnica, utiliza-se a densidade fictícia para determinar o módulo de elasticidade conforme a (2.21).

O algoritmo proposto foi implementado no software MATLAB[®], dessa forma permitindo a solução de exemplos numéricos. O algoritmo proposto foi escrito em funções e script do MATLAB[®] e a análise de elementos finitos foi escrita baseada no código disponibilizado por Andreassen et al. (2011), Sigmund (2001).

A solução do problema viga bi-apoiada *Half MBB Beam* foi otimizada com os seguintes parâmetros: $T_{\max} = 300$, $T_{\min} = 0.0001$, $N = 150000$, $\alpha = 0,95$, $SL = 100$, $HL = 75$, $nelx = 30$, $nely = 20$, $penal = 3$.

A otimização multiobjetivo com o algoritmo proposto resultou na frente de Pareto apresentada na Fig. 65. Na figura, foi inserida a distribuição de massa no domínio de algumas das soluções da frente de Pareto. As soluções apresentadas foram satisfatórias, porém observamos algumas falhas com pontos com material sólidos isolados, o que pode indicar que o valor de iteração N é insuficiente.

Figura 65 – Frente de Pareto resultante da OT com algumas das soluções da frente de Pareto com seus respectivos valores de Fração volumétrica e a energia de deformação



Fonte: O Autor.

A distribuição de massa pode ser melhorada por meio de filtros de densidade baseado em distância. No entanto, o filtro altera o valor de FV e a energia de deformação. Em Sehlström (2013) foi utilizado filtro de densidade para correção do padrão tabuleiro de xadrez e gradiente de cinza das soluções obtidas na otimização multiobjetivo utilizando a técnica de escalarização em conjunto com algoritmo GTO. Além do filtro de densidade, Sehlström (2013) utilizou um filtro para transformar a distribuição de massa de um domínio contínuo para um domínio discreto.

Neste trabalho aplicamos o filtro de densidade baseado na distância dada em Sigmund (2007), o qual suaviza bordas, aumenta as linhas finas e reduz gradiente de cinza e tabuleiro de xadrez. E aplicamos uma transformação da densidade intermediária para 0 ou 1, conforme utilizado em Najafabadi et al. (2021). A aplicação de filtro de densidade como pós-processamento, melhora com a remoção de massas isoladas no domínio ou em alguns casos reforçando algumas linhas finas no domínio. Na Fig. 66 é apresentado o filtro de densidade como exemplo de melhoria. Porém, devemos ressaltar a modificação da frente de Pareto com alteração das soluções pós-processamento.

Figura 66 – Soluções otimizadas obtidas no processo de OT com filtro

- (a) Solução antes do pós-processamento: $FV = 0,2551$, $J = 195,96$ (b) Solução após o pós-processamento: $FV = 0,27$, $J = 165,62$



Fonte: O Autor.

Outro problema aplicado foi no problema *MBB beam* apresentado na Fig. 53. A solução do problema viga bi-apoiada *MBB Beam* foi otimizada com os seguintes parâmetros: $T_{\max} = 300$, $T_{\min} = 0.0001$, $N = 150000$, $\alpha = 0,95$, $SL = 100$, $HL = 75$, $nelx = 40$, $nely = 15$, $penal = 3$. E a frente de Pareto obtida é apresentada na Fig. 67, com algumas das soluções da frente de Pareto.

As soluções apresentadas foram satisfatórias, porém observamos algumas falhas com pontos com materiais sólidos isolados, o que pode indicar que o valor de iteração N é insuficiente. Neste problema aplicamos o pós-processamento descrito acima e obtemos os resultados da Fig. 68. Na figura observamos que houve melhorias na distribuição de massa, porém com alteração na FV conforme o caso anterior.

Os resultados foram satisfatórios, para a minimização FV e energia de deformação.

Figura 67 – Frente de Pareto obtida no processo de OT com seus respectivos valores de Fração volumétrica e a energia de deformação

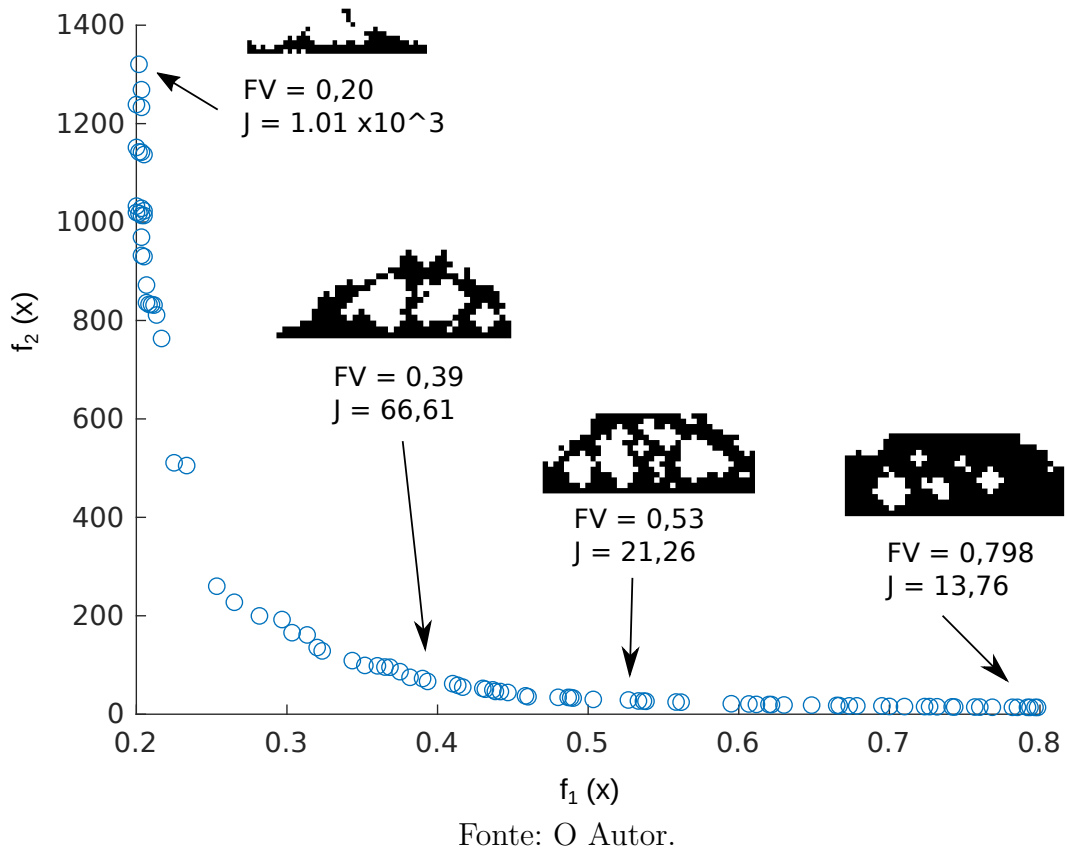


Figura 68 – Solução antes do pós-processamento e após o pós-processamento.

- (a) Resultados antes do pós-processamento: FV = 0,72, J = 15,47
- (b) Resultados após o pós-processamento: FV = 0,75, J = 14,39



Figure: O Autor.

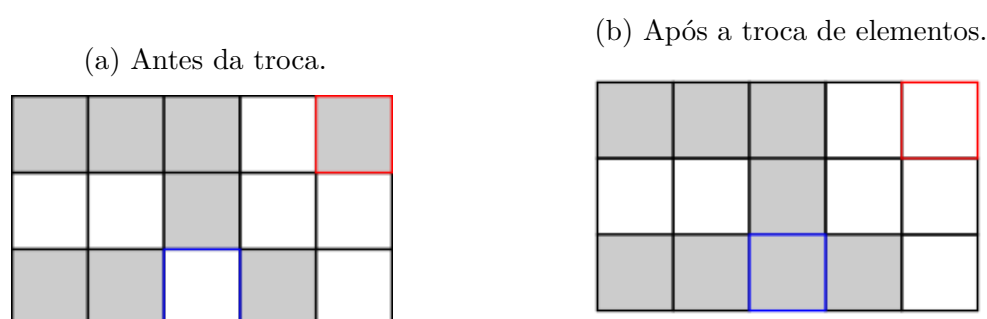
A técnica ainda não foi aplicada em outros tipos de função objetivo. Além disso, é necessário estudo dos parâmetros como a quantidade de iteração N , o fator de raio R em relação ao tamanho da malha.

6.8 OT Multiobjetivo com Heurística de Troca de Densidade

Nesta proposta, aplicamos uma técnica de troca de densidade entre elementos escolhidos aleatoriamente e de densidades opostas para gerar a nova solução. Portanto, nesta técnica a fração volumétrica é mantida constante a cada iteração. Nesta técnica, a pseudo densidade ρ possui o valor $\rho_{\min} = 0.0001$ ou é um elemento sólido $\rho = 1$.

O algoritmo utilizado para gerar a nova solução por meio de troca de densidade fictícia é descrito no Algoritmo 14. O algoritmo seleciona de forma aleatória dois elementos, um com a densidade igual a 1 e outro com a densidade mínima, como apresentado na Fig. 69a. Após a seleção dos elementos ocorre a troca da densidade, o que a densidade é igual a 1 atribui ρ_{\min} e o outro elemento com a $\rho = 1$, como apresentado na Fig. 69b. Na Fig. 69, é possível verificar essa troca da densidade do elemento entre o elemento com as bordas na cor vermelha com o elemento de borda na cor azul.

Figura 69 – O esquema de troca de densidades dos elementos utilizados na OT multiobjetivo



Esse procedimento de troca é descrito no Algoritmo 14, que detalha esse procedimento da troca da densidade entre os dois elementos com densidades opostas.

6.9 OT Multiobjetivo com o Índice de Continuidade e a Heurística de Troca de Densidade

Originalmente o fator de continuidade foi proposto para a penalização da função objetivo na OT com o SA mono-objetivo em domínio discreto. No processo de OT com SA discreto, ou seja, a pseudo densidade com valores de 0.001 ou 1. O fator de continuidade trata-se de um filtro que realiza análise dos elementos da malha e sua vizinhança. Esta técnica analisa e penaliza elementos isolados ou descontinuidades na distribuição de massa.

Algoritmo 14: Técnica de geração da nova solução

Entrada: x_i

- 1 $[i, j] \leftarrow$ sorteia um elemento aleatório
- 2 $\rho_a \leftarrow x_{i,j}$
- 3 **enquanto** <Critério não satisfeito> **faça**
- 4 $[k, l] \leftarrow$ sorteia um elemento aleatório
- 5 $\rho_b \leftarrow x_{k,l}$
- 6 **se** $x_{i,j} \neq x_{k,l}$ **então**
- 7 $x_{i,j} \leftarrow \rho_b$
- 8 $x_{k,l} \leftarrow \rho_a$
- 9 critério \leftarrow satisfeito
- 10 **senão**
- 11 Critério \leftarrow não satisfeito
- 12 **fim**
- 13 **fim**

O objetivo do fator de continuidade é a penalização de soluções com padrão de tabuleiro de xadrez ou massas isoladas no domínio.

Para determinar o fator de continuidade, é calculada a soma da densidade dos elementos vizinhos pela quantidade de elementos da vizinhança de elemento com densidade $\rho = 1$. A soma dessa relação de cada elemento $\rho = 1$ e sua vizinhança é o fator de continuidade. Esse cálculo é realizado por

$$\text{continuidade} = \sum_{e=1}^{N_{el}} \frac{\sum_{i=1}^n \rho_i}{N_{viz}}, \quad (6.3)$$

onde N_{el} é a quantidade de elementos na malha, N_{viz} é a quantidade de elementos na vizinhança do elemento i .

Este processo de determinar o fator de continuidade é apresentado no Algoritmo 15.

Algoritmo 15: Fator de continuidade

Entrada: $\rho = x(i, j)$, $nelx$, $nely$, $n_{el} = 0$, $SomaDen = 0$

- 1 **para** $k = \max(1, j - 1) : \min(j + 1, nely)$ **faça**
- 2 **para** $l = \max(1, i - 1) : \min(i + 1, nelx)$ **faça**
- 3 $n_{el} = n_{el} + 1$
- 4 **se** $x(i, j) == \rho$ **então**
- 5 $SomaDen = SomaDen + 1$
- 6 **fim**
- 7 **fim**
- 8 **fim**
- 9 **retorna** fator de continuidade = $\frac{SomaDen}{n_{el}}$

O processo descrito no Algoritmo 15 pode ser calculado para todos os elementos da

malha, e realizar a soma do fator de continuidade de todos os elementos. A soma do fator de continuidade dos elementos da malha pode ser utilizada como uma função objetivo para ser minimizada. Outra possibilidade é a utilização como restrição na OT.

Esta técnica de penalidade pode ser combinada com a técnica de troca de densidade descrita na Seção 6.8 para a OT. Essa combinação de técnica foi aplicada na OT de uma viga engastada com duas forças aplicadas, F_1 e F_2 . Neste problema descrito na Fig. 70, as funções objetivo $f_1(x)$ e $f_2(x)$ avaliam a energia de deformação para as forças F_1 e F_2 , respectivamente. E a terceira função objetivo é o fator de continuidade, a otimização consiste em minimizar as funções objetivo. Portanto, este problema de minimização pode ser representado matematicamente por

$$\begin{aligned}
 \text{Minimizar} \quad & f_1(x) = J_1(x) = u^T K U_1 \quad \text{Para a Força } F_1 \\
 & f_2(x) = J_2(x) = u^T K U_2 \quad \text{Para a Força } F_2 \\
 & f_3(x) = \text{Continuidade}(x) \\
 \text{Sujeito a :} \quad & x \in \Omega.
 \end{aligned} \tag{6.4}$$

O diagrama da viga engastada e os pontos de aplicação das forças são apresentados na Fig. 70. Os parâmetros utilizados neste problema de OT foram: $nelx = 20$, $nely = 10$, $N = 300000$, $T_{\max} = 300$, $T_{\min} = 0,0001$, $HL = 75$, $SL = 100$, $\alpha = 0,95$ e $FV = 0,6$.

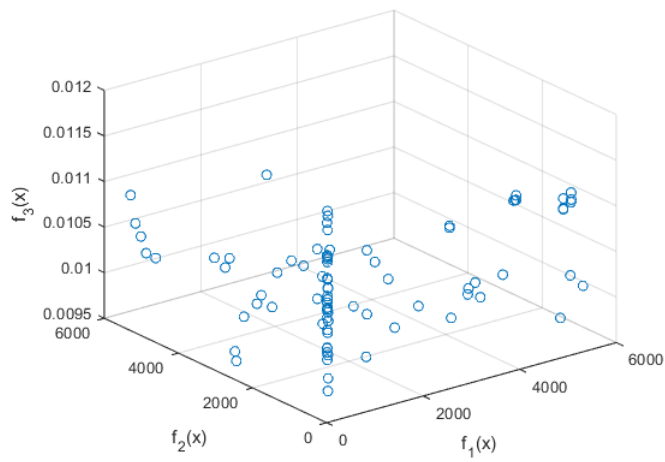
Figura 70 – Diagrama esquemática de Viga engastada com carregamento de duas forças



Fonte: O Autor.

O resultado da OT multiobjetivo resultou na frente de Pareto da Fig. 71. Algumas das soluções pertencentes a frente de Pareto são apresentadas na Fig. 72. Foram selecionadas as soluções otimizadas de pontos diferentes da frente de Pareto.

Figura 71 – Frente de Pareto obtida no processo de OT multiobjetivo



Fonte: O Autor.

Figura 72 – Soluções da frente de Pareto do processo de OT multiobjetivo

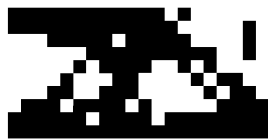
- (a) $J_1 = 70,23$, $J_2 = 76,20$ e $J_3 = 0,0103$ (b) $J_1 = 72,04$, $J_2 = 73,08$ e $J_3 = 0,0105$



- (c) $J_1 = 64,57$, $J_2 = 4,95 \times 10^3$ e $J_3 = 0,0104$ (d) $J_1 = 886,88$, $J_2 = 67,86$ e $J_3 = 0,0105$



- (e) $J_1 = 65,15$, $J_2 = 4,69 \times 10^3$ e $J_3 = 0,0104$ (f) $J_1 = 5,09 \times 10^3$, $J_2 = 65,08$ e $J_3 = 0,0104$



Fonte: O Autor.

Os resultados apresentaram distribuição de massa compatíveis com os encontrados na literatura, porém, a distribuição apresentou massas isoladas e bordas não suaves. Algumas das soluções otimizadas também apresentaram o padrão tabuleiro de xadrez em alguns pontos do domínio. Estas falhas nos resultados estão relacionadas ao tamanho da malha pequena e ao número de iterações insuficiente. Apesar destas falhas, os resultados são satisfatórios e em trabalhos futuros podemos aumentar o número de iteração e implementar filtro de densidade para melhorar a distribuição de massa.

Um segundo problema de OT é a viga engastada com a aplicação de três carregamentos, conforme apresentado na Fig. 73. Este problema de OT foi apresentado por Madeira, Rodrigues e Pina (2006) que utilizaram algoritmos genéticos.

Figura 73 – Problema de OT em viga engastada com três carregamentos



Fonte: O autor.

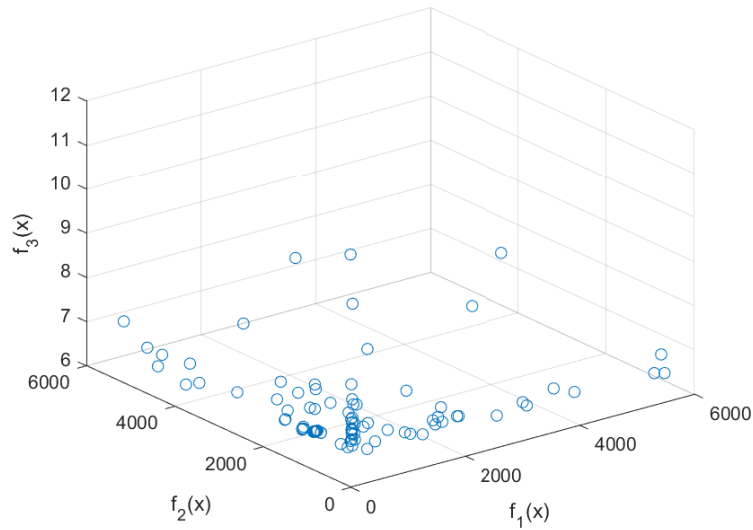
Esse problema de OT possui três funções objetivo, e estas funções objetivo são as energias de deformação para cada força aplicada. Neste caso em específico, o fator de continuidade foi utilizado como um fator de penalidade e não como uma função objetivo, ou seja, o fator de penalidade foi adicionado às funções objetivos. O problema de minimização pode ser descrito matematicamente por

$$\begin{aligned}
 \text{Minimizar } f_1(x) &= J_1(x) + \text{penalidade} = u^T K U_1 + \text{penalidade} && \text{Para a Força } F_1 \\
 f_2(x) &= J_2(x) + \text{penalidade} = u^T K U_2 + \text{penalidade} && \text{Para a Força } F_2 \\
 f_3(x) &= J_3(x) + \text{penalidade} = u^T K U_3 + \text{penalidade} && \text{Para a Força } F_3 \\
 \text{Sujeito a : } &&& x \in \Omega.
 \end{aligned}
 \tag{6.5}$$

Nesse problema de OT, não possui nenhuma restrição e os parâmetros utilizados na OT foram: $nelx = 30$, $nely = 16$, $N = 5000$, $T_{\max} = 100$, $T_{\min} = 0,0001$, $HL = 75$, $SL = 100$, $\alpha = 0,95$ e $FV = 0,5$.

A OT com o CoAnnealing e a heurística de troca resultou na frente de Pareto apresentada na Fig. 74. Algumas das soluções da frente de Pareto, são apresentadas na figura Fig. 75.

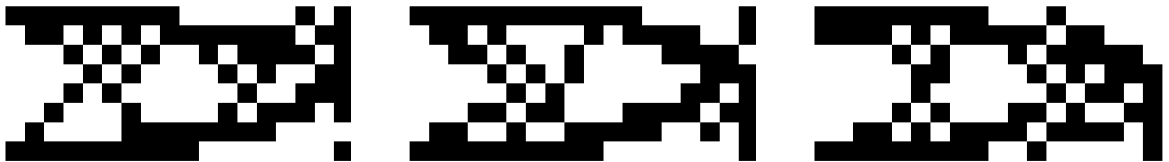
Figura 74 – Frente de Pareto do processo de OT da viga engastada com carregamento de 3 forças



Fonte: O Autor.

Figura 75 – Soluções otimizadas na frente de Pareto da OT da viga engastada.

- (a) $J_1 = 80,3$, $J_2 = 88,88$ e $J_3 = 7,22$ (b) $J_1 = 3 \times 10^3$, $J_2 = 80,07$ e $J_3 = 6,81$ (c) $J_1 = 80,85$, $J_2 = 1,5 \times 10^3$ e $J_3 = 7,0$



- (d) $J_1 = 2,69 \times 10^3$, $J_2 = 74,47$ e $J_3 = 10,32$ (e) $J_1 = 2,17 \times 10^3$, $J_2 = 75,72$ e $J_3 = 9,30$ (f) $J_1 = 96$, $J_2 = 92,99$ e $J_3 = 6,86$



Fonte: O Autor.

As soluções otimizadas encontradas apresentaram o padrão de tabuleiro de xadrez, mesmo realizando a penalização com fator de continuidade. Isso pode ser consequência da malha com poucos elementos, e a quantidade de iterações N muito abaixo do necessário. Os resultados foram satisfatórios e demonstram a capacidade do CoAnnealing de resolver problemas de OT multiobjetivo. Estes resultados não permitem realizar a comparação com os resultados de Madeira, Rodrigues e Pina (2006), devido ao tamanho da malha ser muito diferente.

Os resultados podem ser melhorados por meio de aumento do tamanho da malha e iterações, implementação de filtros de densidade e, ainda, o implemento de melhorias no fator de penalidade.

6.10 Considerações Finais

Neste capítulo foi apresentada a implementação da OT com o SA mono-objetivo e multiobjetivo, com algumas propostas de heurística para a solução de OT. Existem poucos trabalhos na literatura de OT com o SA, principalmente quando comparamos aos algoritmos genéticos.

Em relação a OT mono-objetivo proposto neste trabalho, os resultados foram comparáveis aos métodos tradicionais GTO. A distribuição de massa não apresentou problemas de gradiente de cinza nem o padrão de tabuleiro de xadrez. Além disso, a energia de deformação das soluções otimizadas foram melhores em alguns casos, nos outros casos os valores foram próximos em relação ao do GTO. A desvantagem das técnicas propostas é o tempo para convergência que variaram entre 1 a 2 horas, enquanto os GTO levam alguns minutos.

A OT multiobjetivo foi possível por meio do CoAnnealing e heurística de troca de densidade combinado ao fator de penalidade. Estes apresentaram resultados satisfatórios com algumas soluções otimizadas apresentando massas isoladas. Apesar dos problemas de OT aplicado serem simples, as heurísticas propostas demonstram-se promissoras. A principal desvantagem é a dependência de um alto valor de iteração. A necessidade de um alto valor de iterações é uma característica dos algoritmos não gradiente, que dependem fortemente da aleatoriedade. Quanto maior a quantidade de variáveis de decisão (composição na malha) maior é a quantidade de iterações, já que o espaço de busca é maior. Portanto, a maior desvantagem dos algoritmos probabilísticos são os custos computacionais.

Apesar desta desvantagem, a OT por SA possui outras vantagens como a flexibilidade e a otimização multiobjetivo verdadeira, sem a utilização de técnicas de escalarização. Lembrando que não é possível realizar a otimização multiobjetivo sem técnicas de escalarização com algoritmo GTO. Além da otimização multiobjetivo, Guirguis et al. (2020) cita a limitação dos algoritmos GTO em resolver alguns problemas de OT com restrição

relacionada à manufatura aditiva ou estrutura de multicomponentes, como o apresentado por Guirguis et al. (2015).

Algumas heurísticas podem melhorar o desempenho da OT com a técnica proposta, porém é necessário mais estudos. Uma das possibilidades é a adição de algum tipo de sensibilidade ou utilizar algoritmos híbridos, combinando o SA com outras técnicas.

7 Conclusões

Neste trabalho estudamos as heurísticas e estratégias para SA, e aplicamos estas na otimização multiobjetivo com o CoAnnealing. Avaliamos o desempenho do CoAnnealing com a heurística de cristalização por meio de funções de benchmark Capítulo 4. Os resultados demonstraram que o CoAnnealing com a heurística de cristalização possui desempenho comparável ao AMOSA sendo melhor em várias funções de benchmark. Em outras funções como ZDT3, ZDT6, DTLZ2 e DTLZ7 o CoAnnealing apresentou resultados melhores. Na maioria dos casos estudados o CoAnnealing convergiu de forma mais rápida em relação ao AMOSA. Comprovando a eficiência e o desempenho do CoAnnealing.

O CoAnnealing com a heurística de cristalização também foi aplicado em problema de otimização mecânica como o dimensionamento de chapas com furo e sistema de transmissão (Capítulo 5). Os resultados validaram o desempenho e eficiência em solução de problemas mecânicos.

Com o conhecimento adquirido foi proposto heurísticas e estratégias para utilizar o SA em conjunto com a técnica SIMP para a solução de problemas de OT mono-objetivo ((Seção 6.1 e Seção 6.3)). O SA com a heurística proposta foi aplicado em problemas clássicos de vigas encontrado na literatura e apresentou resultados satisfatórios com características interessantes para algumas aplicações específicas.

As técnicas desenvolvidas para a OT mono-objetivo foram adaptadas para a OT multiobjetivo utilizando o CoAnnealing (Seção 6.6 e Seção 6.8). Aplicamos em problemas de OT de vigas e a técnica proposta permitiu obter uma frente de Pareto, contendo soluções ótimas. Estas técnicas demonstraram um potencial, pois não depende de informações de gradiente e permitem uma maior flexibilidade em relação a restrições. Além de ser uma otimização multiobjetivo sem técnicas de escalarização. Devemos também ressaltar que o algoritmo propos realiza a OT discreta, o que não é possível realizar com GTO sem técnicas adicionais, filtros ou pós-processamento. Consideramos que as vantagens superam as desvantagens como custo computacional e o tempo de execução.

As principais contribuições deste trabalho foram:

- análise do desempenho e a validação do CoAnnealing com heurística de cristalização;
- proposta e implementação de heurísticas e estratégias para OT baseado em SA mono-objetivo;
- a implementação da OT multiobjetivo utilizando o CoAnnealing.

7.1 Trabalhos Futuros

Como sugestão de trabalhos futuros temos os seguintes itens:

- realizar alguns estudos de sensibilidade de parâmetros e estratégias da cristalização heurística no algoritmo multiobjetivo CoAnnealing;
- propor novas estratégias para a heurística de cristalização específica para o CoAnnealing;
- melhoria no desempenho das técnicas propostas para a OT mono-objetivo e multiobjetivo por meio do SA;
- estudar a inserção de avaliação parcial da função objetivo na OT, com objetivo de melhorar o desempenho;
- implementar as técnicas propostas em problemas de multi-física e problemas mais complexos;
- realizar estudos relacionados a utilização de inteligência artificial para a solução de elementos finitos com objetivo de acelerar a OT.

Referências

- ALNOWIBET, K. A. et al. Efficient modified meta-heuristic technique for unconstrained optimization problems. *Axioms*, MDPI AG, v. 11, n. 9, p. 483, set. 2022. 57
- AMINE, K. Multiobjective simulated annealing: Principles and algorithm variants. *Advances in Operations Research*, Hindawi Limited, v. 2019, p. 1–13, maio 2019. 20
- ANDREASSEN, E. et al. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, v. 43, n. 1, p. 1–16, 2011. ISSN 1615147X. 8, 9, 40, 41, 104, 105, 106, 108, 110, 111, 113, 114, 115, 116, 121, 122, 123, 124, 134
- BAHIA, M. T. *Otimização topológica aplicada ao projeto de mecanismos flexíveis*. 193 f. Dissertação (Mestrado em Engenharia Mecânica) — Universidade Federal de Santa Catarina, Florianópolis, SC, 2005. 37
- BANDYOPADHYAY, S. et al. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE transactions on evolutionary computation*, IEEE, v. 12, n. 3, p. 269–283, 2008. 59, 60, 62, 63, 72, 86
- BENDSØE, M. P. Optimal shape design as a material distribution problem. *Structural optimization*, v. 1, n. 4, p. 193–202, Dec 1989. ISSN 1615-1488. 38
- BENDSØE, M. P.; DÍAZ, A.; KIKUCHI, N. Topology and Generalized Layout Optimization of Elastic Structures. In: _____. *Topology Design of Structures*. Dordrecht: Springer Netherlands, 1993. p. 159–205. ISBN 978-94-011-1804-0. 37, 38
- BENDSØE, M. P.; DÍAZ, A.; KIKUCHI, N. Topology and Generalized Layout Optimization of Elastic Structures. In: BENDSØE, M. P.; SOARES, C. A. M. (Ed.). *Topology Design of Structures*. Dordrecht: Springer Netherlands, 1993. p. 159–205. ISBN 978-94-011-1804-0. 134
- BENDSØE, M. P.; SIGMUND, O. *Topology Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. ISBN 978-3-642-07698-5. 36, 37
- BINH, T. T.; KORN, U. Mobes: A multiobjective evolution strategy for constrained optimization problems. In: *The third international conference on genetic algorithms (Mendel 97)*. [S.l.: s.n.], 1997. v. 25, p. 27. 7, 72, 83
- Blank, J.; Deb, K. pymoo: Multi-objective optimization in python. *IEEE Access*, v. 8, p. 89497–89509, 2020. 35
- BOHACHEVSKY, I. O.; JOHNSON, M. E.; STEIN, M. L. Generalized simulated annealing for function optimization. *Technometrics*, Taylor & Francis, v. 28, n. 3, p. 209–217, 1986. 49
- BUREERAT, S.; KUNAKOTE, T. Topological design of structures using population-based optimization methods. *Inverse Problems in Science and Engineering*, v. 14, p. 589–607, 2006. ISSN 17415977. 44

- BUREERAT, S.; LIMTRAGOOL, J. Performance enhancement of evolutionary search for structural topology optimisation. *Finite Elements in Analysis and Design*, v. 42, n. 6, p. 547–566, 2006. 44, 45, 100
- BUREERAT, S.; LIMTRAGOOL, J. Structural topology optimisation using simulated annealing with multiresolution design variables. *Finite Elements in Analysis and Design*, v. 44, n. 12-13, p. 738–747, 2008. 21, 44, 45
- CAVAZZUTI, M. Deterministic optimization. In: _____. *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 77–102. ISBN 978-3-642-31187-1. 26, 29
- ČERNÝ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, v. 45, n. 1, p. 41–51, 1985. 47
- Chakraborty, M.; Chakraborty, U. K. Applying genetic algorithm and simulated annealing to a combinatorial optimization problem. In: *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat. [S.l.: s.n.], 1997. v. 2, p. 929–933 vol.2. 38*
- CHANKONG, V.; HAIMES, Y. Y. *Multiobjective decision making: theory and methodology*. [S.l.]: Courier Dover Publications, 2008. 72
- COELLO, C. A.; CHRISTIANSEN, A. D. Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, v. 75, n. 6, p. 647–660, 2000. ISSN 00457949. 38
- COELLO, C. A. C. et al. *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2007. v. 5. 31, 32, 47, 75
- COHON, J. L.; CHURCH, R. L.; SHEER, D. P. Generating multiobjective trade-offs: An algorithm for bicriterion problems. *Water Resources Research*, v. 15, n. 5, p. 1001–1010, 1979. 21, 31
- CUSTÓDIO, A. L. et al. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, SIAM, v. 21, n. 3, p. 1109–1140, 2011. 76
- CZYŻŻAK, P.; JASZKIEWICZ, A. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, v. 7, n. 1, p. 34–47, 1998. ISSN 10991360. 58
- DEATON, J. D.; GRANDHI, R. V. A survey of structural and multidisciplinary continuum topology optimization: Post 2000. *Structural and Multidisciplinary Optimization*, v. 49, n. 1, p. 1–38, 2014. ISSN 1615147X. 38, 41
- DEB, K. Multi-objective optimisation using evolutionary algorithms: an introduction. In: *Multi-objective evolutionary optimisation for product design and manufacturing*. [S.l.]: Springer, 2011. p. 3–34. 72
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, apr 2002. ISSN 1089778X. 32, 59

- DEB, K. et al. Scalable multi-objective optimization test problems. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. [S.l.: s.n.], 2002. v. 1, p. 825–830 vol.1. 72, 73
- DEKKERS, A.; AARTS, E. Global optimization and simulated annealing. *Mathematical programming*, Springer, v. 50, n. 1, p. 367–393, 1991. 29
- DENEUBOURG, J. L. et al. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, Springer Science and Business Media LLC, v. 3, n. 2, p. 159–168, mar. 1990. Disponível em: <<https://doi.org/10.1007/bf01417909>>. 30
- DÍAZ, A.; SIGMUND, O. Checkerboard patterns in layout optimization. *Structural Optimization*, v. 10, n. 1, p. 40–45, aug 1995. ISSN 0934-4373. 40
- DURAN, G. C. et al. Using feedback strategies in simulated annealing with crystallization heuristic and applications. *Applied Sciences*, v. 11, n. 24, 2021. ISSN 2076-3417. 53, 54, 56, 69, 70, 87, 100
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. [S.l.: s.n.], 1995. p. 39–43. 30
- EMMERICH, M.; BEUME, N.; NAUJOKS, B. An emo algorithm using the hypervolume measure as selection criterion. In: COELLO, C. A. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 62–76. ISBN 978-3-540-31880-4. 35
- FANNI, M.; SHABARA, M.; ALKALLA, M. A comparison between different topology optimization methods. *Mansoura Engineering Journal, MEJ*, Mansoura Engineering Journal, v. 38, n. 1, 2013. 40
- GARCIA-LOPEZ, N. P. et al. A hybrid topology optimization methodology combining simulated annealing and SIMP. *Computers & Structures*, v. 89, n. 15, p. 1512–1522, 2011. 21, 44, 45, 100
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. H. *Manual de computação evolutiva e metaheurística*. [S.l.]: Imprensa da Universidade de Coimbra/Coimbra University Press, 2012. 20, 21, 24, 30, 31, 32, 59
- GLOVER, F. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, v. 8, n. 1, p. 156–166, 1977. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5915.1977.tb01074.x>>. 31
- GOLDBARG, E.; GOLDBARG, M.; LUNA, H. *Otimização combinatória e metaheurísticas: algoritmos e aplicações*. [S.l.]: Elsevier Brasil, 2017. 30, 31, 47, 48, 49, 51
- GOLDBERG, D. Genetic algorithms in search. optimization and machine learning. addism1—wesley. *Reading, MA*, 1988. 58
- GOLDBERG, D. E.; SAMTANI, M. P. Engineering optimization via genetic algorithm. In: ASCE. *Electronic computation*. [S.l.], 1986. p. 471–482. 129
- GOSS, S. et al. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, Springer-Verlag, v. 76, n. 12, p. 579–581, 1989. 30

- GOTO, T. et al. Topological optimization and simulated annealing. *IFAC-PapersOnLine*, v. 54, n. 1, p. 205–210, 2021. ISSN 2405-8963. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021. 101
- GOTO, T. G. et al. Versatility of simulated annealing with crystallization heuristic: Its application to a great assortment of problems. In: *Optimization Problems in Engineering [Working Title]*. [S.l.]: IntechOpen, 2021. 53
- GOYAL, S. A survey on travelling salesman problem. In: *Midwest instruction and computing symposium*. [S.l.: s.n.], 2010. p. 1–9. 25
- GRANVILLE, V.; KRIVANEK, M.; RASSON, J.-P. Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 6, p. 652–656, 1994. 20
- GUIRGUIS, D. et al. Evolutionary black-box topology optimization: Challenges and promises. *IEEE Transactions on Evolutionary Computation*, Institute of Electrical and Electronics Engineers Inc., v. 24, p. 613–633, 8 2020. ISSN 19410026. 21, 45, 143
- GUIRGUIS, D. et al. Multi-objective topology optimization of multi-component continuum structures via a kriging-interpolated level set approach. *Structural and Multidisciplinary Optimization*, Springer Verlag, v. 51, p. 733–748, 3 2015. ISSN 1615-147X. 144
- GUIRGUIS, D.; MELEK, W. W.; ALY, M. F. High-resolution non-gradient topology optimization. *Journal of Computational Physics*, Academic Press Inc., v. 372, p. 107–125, 11 2018. ISSN 10902716. 45
- HAIMES YV, Y.; LASDON, L.; WISMER DA, D. On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, Institute of Electrical and Electronics Engineers Inc., SMC-1, n. 3, p. 296–297, 1 1971. ISSN 0018-9472. 21, 31
- HAMDA, H.; ROUDENKO, O.; SCHOENAUER, M. Multi-objective evolutionary topological optimum design. In: *Adaptive Computing in Design and Manufacture V*. [S.l.]: Springer, 2002. p. 121–132. 129, 130
- HANSEN, P. The steepest ascent mildest descent heuristic for combinatorial programming. In: *Congress on numerical methods in combinatorial optimization, Capri, Italy*. [S.l.: s.n.], 1986. p. 70–145. 31
- HENDERSON, D.; JACOBSON, S. H.; JOHNSON, A. W. The theory and practice of simulated annealing. In: _____. *Handbook of Metaheuristics*. Boston, MA: Springer US, 2003. p. 287–319. ISBN 978-0-306-48056-0. 47
- HOLLAND, J. H. Adaptation in natural and artificial systems. *Ann Arbor*, The University of Michigan Press, 1975. 30
- HOLTZ, G. C. da C. *Traçado automático de envoltórias de esforços em estruturas planas utilizando um algoritmo evolucionário*. 123 f. Dissertação (Mestrado em Engenharia Civil) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 2005. 20, 26, 29
- JUNIOR, W. D. C. Fundamentos da ciência e engenharia de materiais-uma abordagem integrada. *Rio de Janeiro: LTC*, 2006. 47

- KANE, C.; SCHOENAUER, M. Topological optimum design using genetic algorithms. *Control and Cybernetics*, v. 25, n. 5, p. 1059–1087, 1996. 44
- KATSIGIANNIS, Y. A.; GEORGILAKIS, P. S.; KARAPIDAKIS, E. S. Hybrid simulated annealing-tabu search method for optimal sizing of autonomous power systems with renewables. *IEEE Transactions on Sustainable Energy*, IEEE, v. 3, n. 3, p. 330–338, 2012. ISSN 19493029. 57
- KENNEDY, J.; EBERHART, R. A discrete binary version of the particle swarm algorithm. In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. [S.l.: s.n.], 1997. v. 5, p. 4104–4108 vol.5. 30
- KICINGER, R.; ARCISZEWSKI, T.; JONG, K. D. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers and Structures*, v. 83, n. 23-24, p. 1943–1978, 2005. 37, 129
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *science*, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. 47, 49, 50, 51
- KNOWLES, J. D.; CORNE, D. W. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, v. 8, n. 2, p. 149–172, 2000. 32, 59
- LUH, G. C.; LIN, C. Y.; LIN, Y. S. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing Journal*, Elsevier B.V., v. 11, p. 2833–2844, 2011. ISSN 15684946. 30
- MADEIRA, J. A.; RODRIGUES, H. C.; PINA, H. Multiobjective topology optimization of structures using genetic algorithms with chromosome repairing. *Structural and Multidisciplinary Optimization*, v. 32, p. 31–39, 2006. ISSN 1615147X. 141, 143
- MARCK, G. et al. Topology optimization using the SIMP method for multiobjective conductive problems. *Numerical Heat Transfer, Part B: Fundamentals*, v. 61, n. 6, p. 439–470, 2012. ISSN 10407790. 38, 129
- MARTINS, T. C.; TSUZUKI, M. S. Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. *Expert Systems with Applications*, Elsevier Ltd, v. 37, n. 3, p. 1955–1972, 2010. ISSN 09574174. 52, 54
- MARTINS, T. C.; TSUZUKI, M. S. G. EIT image regularization by a new Multi-Objective Simulated Annealing algorithm. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, v. 2015-November, p. 4069–4072, 2015. ISSN 1557170X. 20, 59, 65, 66, 67
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, AIP, v. 21, n. 6, p. 1087–1092, 1953. 47
- MICHELL, A. LVIII. The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 8, n. 47, p. 589–597, nov 1904. ISSN 1941-5982. 37, 38
- MURTAGH, F. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, v. 26, n. 4, p. 354–359, 1983. ISSN 00104620. 60

- NAJAFABADI, H. R. et al. Post-processing of non gradient-based topology optimization with simulated annealing. *IFAC-PapersOnLine*, v. 54, n. 1, p. 755–760, 2021. ISSN 2405-8963. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021. 101, 135
- NGATCHOU, P.; ZAREI, A.; EL-SHARKAWI, A. Pareto Multi Objective Optimization. *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, p. 84–91, 2005. 33
- NORTON, R. L. *Projeto de máquinas*. [S.l.]: bookman editora, 2013. 92, 97
- PARDO, M. A. G. *Técnicas de otimização baseadas em quimiotaxia de bactérias*. Tese (Doutorado) — Universidade de São Paulo, 2009. 38, 42, 88, 89, 90, 91, 92, 93, 96, 97, 99, 109, 110, 111, 114, 115
- PEDAMALLU, C. S.; OZDAMAR, L. Investigating a hybrid simulated annealing and local search algorithm for constrained optimization. *European Journal of Operational Research*, v. 185, n. 3, p. 1230–1245, 2008. ISSN 0377-2217. 57
- PIRLOT, M. General local search methods. *European Journal of Operational Research*, v. 92, n. 3, p. 493–511, 1996. ISSN 03772217. 49, 51
- PRAGER, W. Optimality criteria in structural design. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 61, n. 3, p. 794–796, 1968. 27
- QUERIN, O.; STEVEN, G.; XIE, Y. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations*, v. 15, n. 8, p. 1031–1048, dec 1998. ISSN 0264-4401. 38, 42, 43, 44
- QUERIN, O.; STEVEN, G.; XIE, Y. Evolutionary structural optimisation using an additive algorithm. *Finite Elements in Analysis and Design*, v. 34, n. 3, p. 291 – 308, 2000. ISSN 0168-874X. 38
- RAO, S. S. *Engineering optimization: theory and practice*. 4. ed. [S.l.]: John Wiley & Sons, 2009. 24, 25, 26, 27
- ROZVANY, G. Grillages of maximum strength and maximum stiffness. *International Journal of Mechanical Sciences*, v. 14, n. 10, p. 651–666, oct 1972. ISSN 00207403. 38
- ROZVANY, G. I. N. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, v. 37, n. 3, p. 217–237, jan 2009. ISSN 1615-147X. 38, 41
- ROZVANY, G. I. N.; ZHOU, M. Layout and generalized shape optimization by iterative coc methods. In: _____. *Optimization of Large Structural Systems*. Dordrecht: Springer Netherlands, 1993. p. 103–120. ISBN 978-94-010-9577-8. 38
- RUTENBAR, R. A. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices magazine*, IEEE, v. 5, n. 1, p. 19–26, 1989. 20, 48, 49, 51
- SALVADOR, S. M. B. *Estudo do efeito dos parâmetros algorítmicos nos critérios de resolução OC, SAO e MMA em problemas tridimensionais com o método de otimização SIMP*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2018. 27, 28

- SATO, A. K. et al. Simulated annealing with crystallization heuristic applied to aircraft conceptual design. In: *2014 11th IEEE/IAS International Conference on Industry Applications*. [S.l.: s.n.], 2014. p. 1–6. 51, 52, 53, 54
- SCHOTT, J. R. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Tese (Doutorado) — Massachusetts Institute of Technology, 1995. 35
- SEHLSTRÖM, A. *Multiobjective topology optimization: Tracing of Pareto-optimal structures with respect to volume, compliance and fundamental eigenvalue*. Dissertação (Mestrado) — Chalmers University of Technology, 2013. 135
- SHIM, P. Y.; MANOOCHEHRI, S. Generating optimal configurations in structural design using simulated annealing. *International Journal for Numerical Methods in Engineering*, v. 40, n. 6, p. 1053–1069, 1997. ISSN 00295981. 38
- SIGMUND, O. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, v. 21, n. 2, p. 120–127, 2001. ISSN 1615147X. 40, 134
- SIGMUND, O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, v. 33, n. 4-5, p. 401–424, 2007. ISSN 1615147X. 39, 40, 135
- SIGMUND, O. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, v. 43, p. 589–596, 2011. ISSN 1615147X. 45
- SIGMUND, O.; MAUTE, K. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, v. 48, n. 6, p. 1031–1055, 2013. ISSN 1615147X. 39
- SIGMUND, O.; PETERSSON, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, v. 16, n. 1, p. 68–75, 1998. ISSN 09344373. 40
- SIMONETTI, H. L. et al. Multi-objective topology optimization using the Boundary Element Method. *Structures*, Elsevier, v. 19, n. October 2018, p. 84–95, 2019. ISSN 23520124. 129, 130
- SIMONETTI, H. L. et al. Otimização topológica de estruturas tridimensionais usando mma. *Revista CIATEC-UPF*, v. 13, n. 3, p. 93–110, 2021. 27, 28
- SUMAN, B.; KUMAR, P. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, Informa UK Limited, v. 57, n. 10, p. 1143–1160, out. 2006. 20
- SUN, X. F. et al. Topology optimization of composite structure using Bi-directional Evolutionary Structural Optimization method. *Procedia Engineering*, Elsevier B.V., v. 14, p. 2980–2985, 2011. ISSN 18777058. 44
- SUPPAPITNARM, A. et al. A simulated annealing algorithm for multiobjective optimization. *Engineering optimization*, Taylor & Francis, v. 33, n. 1, p. 59–85, 2000. 49
- SVANBERG, K. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, Wiley Online Library, v. 24, n. 2, p. 359–373, 1987. 28

- SVANBERG, K. The method of moving asymptotes (mma) with some extensions. In: *Optimization of large structural systems*. [S.l.]: Springer, 1993. p. 555–566. 29
- TANAKA, M. et al. Ga-based decision support system for multicriteria optimization. In: *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*. [S.l.: s.n.], 1995. v. 2, p. 1556–1561 vol.2. 72
- TAVARES, R. S.; MARTINS, T. C.; TSUZUKI, M. S. G. Simulated annealing with adaptive neighborhood: A case study in off-line robot path planning. *Expert Systems with Applications*, Elsevier Ltd, v. 38, n. 4, p. 2951–2965, 2011. 52, 54, 70
- ULUNGU, E. L. et al. MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, v. 8, n. 4, p. 221–236, 1999. ISSN 10991360. 51, 59
- VATANABE, S. L. et al. Topology optimization with manufacturing constraints: A unified projection-based approach. *Advances in Engineering Software*, Elsevier Ltd, v. 100, p. 97–112, 10 2016. ISSN 18735339. 127
- VELDHUIZEN, D. A. V.; LAMONT, G. B. *Multiobjective evolutionary algorithm research: A history and analysis*. [S.l.], 1998. 34
- VITORINO, A. *OTIMIZAÇÃO TOPOLÓGICA DE ESTRUTURAS TRIDIMENSIONAIS*. 79 f. Dissertação (Mestrado em Matemática Aplicada) — Universidade de Campinas, 2019. 37
- WOLPERT, D. H.; MACREADY, W. G. et al. *No free lunch theorems for search*. [S.l.], 1995. 20
- XIE, Y.; STEVEN, G. A simple evolutionary procedure for structural optimization. *Computers & Structures*, v. 49, n. 5, p. 885–896, dec 1993. ISSN 00457949. 38, 41
- XIE, Y. M.; STEVEN, G. P. Basic evolutionary structural optimization. In: _____. *Evolutionary Structural Optimization*. London: Springer London, 1997. p. 12–29. ISBN 978-1-4471-0985-3. 41
- YILDIZ, A. R. Hybrid immune-simulated annealing algorithm for optimal design and manufacturing. *International Journal of Materials and Product Technology*, v. 34, p. 217–226, 2009. ISSN 02681900. 57
- ZADEH, L. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, v. 8, n. 1, p. 59–60, jan 1963. ISSN 0018-9286. 21, 31
- ZAVALA, G. R. et al. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, v. 49, n. 4, p. 537–558, apr 2014. 37, 129
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, v. 8, n. 2, p. 173–195, 2000. 72, 73
- ZITZLER, E.; THIELE, L. Multiobjective Optimization Using Evolutionary Algorithm. *Lecture Notes in Computer Science*, v. 1498, p. 292–301, 1998. 32, 35

ZULIANI, J. B. Q. et al. Multi-objective topology optimization with ant colony optimization and genetic algorithms. *Computer-Aided Design and Applications*, v. 12, n. 6, p. 674–682, 2015. ISSN 16864360. 129

ZUO, Z. H.; XIE, Y. M.; HUANG, X. Combining genetic algorithms with BESO for topology optimization. *Structural and Multidisciplinary Optimization*, v. 38, n. 5, p. 511–523, 2009. ISSN 1615147X. 44