



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL

NATALIA CRISTINA THIEM BLEICHVEL

**IMPLEMENTAÇÃO DO ALGORITMO METAHEURÍSTICO DA BALEIA BELUGA
NA OTIMIZAÇÃO DE TRELIÇAS**

FLORIANÓPOLIS
2024

NATALIA CRISTINA THIEM BLEICHVEL

**IMPLEMENTAÇÃO DO ALGORITMO METAHEURÍSTICO DA BALEIA BELUGA
NA OTIMIZAÇÃO DE TRELIÇAS**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Mestre em Engenharia Civil.

Orientador: Prof. Rafael Holdorf Lopez, Dr.

FLORIANÓPOLIS

2024

Bleichvel, Natalia Cristina Thiem
Implementação do algoritmo metaheurístico da baleia
beluga na otimização de treliças / Natalia Cristina Thiem
Bleichvel ; orientador, Rafael Holdorf Lopez, 2024.
76 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia Civil, Florianópolis, 2024.

Inclui referências.

1. Engenharia Civil. 2. Engenharia civil. 3. Otimização.
4. Algoritmos . 5. Métodos heurísticos. I. Lopez, Rafael
Holdorf. II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia Civil. III. Título.

Natalia Cristina Thiem Bleichvel

**IMPLEMENTAÇÃO DO ALGORITMO METAHEURÍSTICO DA BALEIA BELUGA NA
OTIMIZAÇÃO DE TRELIÇAS**

O presente trabalho em nível de Mestrado foi avaliado e aprovado, em 24 de setembro de 2024, pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Tiago Morkis Siqueira
Universidade Federal de Santa Catarina

Prof. Dr. Wanderlei Malaquias Pereira Junior
Universidade Federal de Catalão

Certificamos que esta é a versão original e final do trabalho de conclusão que foi julgado aprovado para obtenção do título de Mestre em Engenharia Civil.

Profa. Dra. Fernanda Fernandes Marchiori
Coordenadora do Curso

Prof. Dr. Rafael Holdorf Lopez
Orientador

Florianópolis, 2024

Dedico este trabalho a Deus, por ser o alicerce das minhas conquistas.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado força e determinação para chegar até aqui.

Ao meu marido, por toda compreensão e apoio, por estar sempre presente e partilhar a vida comigo. Aos meus pais, que me incentivaram e me ensinaram que o estudo pode mudar a vida das pessoas.

Ao professor Rafael Holdorf Lopez, pelos conhecimentos transmitidos nas disciplinas, pela paciência e dedicação na orientação.

A todos os meus amigos, em especial aos que encontrei durante o mestrado, pela companhia e apoio.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

A otimização é a busca pelo resultado ótimo para um problema. Os métodos heurísticos surgiram como uma opção para a resolução de problemas de otimização nos quais os métodos matemáticos não conseguem solucionar, são métodos exploratórios que buscam resolver determinado problema. A busca por novos e cada vez mais robustos métodos heurísticos é crescente, para que seja possível encontrar a solução de problemas de otimização de modo eficaz. Para analisar e comparar os métodos heurísticos, são feitas análises com funções já conhecidas e verificados os dados estatísticos e probabilísticos deles. Este trabalho compreende uma análise para o algoritmo Beluga Whale Optimization (BWO) implementado em MATLAB aplicado em estruturas treliçadas. Para a primeira validação são analisados problemas clássicos de otimização, com e sem restrições, a fim de analisar os resultados estatísticos, como a média e desvio padrão. O procedimento de implementação para as estruturas treliçadas planas, espacial e da torre de transmissão busca apontar os valores estatísticos com os pontos ótimos de massa para cada estrutura em uma determinada simulação. Por fim, são feitas as comparações probabilísticas para verificar a eficiência e possíveis limitações do algoritmo em estudo, apontando que o algoritmo BWO traz resultados de pontos ótimos em três das quatro estruturas treliçadas estudadas.

Palavras-chave: otimização; métodos heurísticos; algoritmos.

ABSTRACT

Optimization is the search for the optimal result for a problem. Heuristic methods emerged as an option for solving optimization problems that mathematical methods cannot solve; they are exploratory methods that seek to solve a given problem. The search for new and increasingly robust heuristic methods is increasing, so that it is possible to find solutions to optimization problems effectively. To analyze and compare heuristic methods, analyzes are carried out with already known functions and their statistical and probabilistic data are verified. This work comprises an analysis for the Beluga Whale Optimization (BWO) algorithm implemented in MATLAB applied to truss structures. For the first validation, classic optimization problems are analyzed, with and without restrictions, in order to analyze statistical results, such as the mean and standard deviation. The implementation procedure for flat, spatial and transmission tower truss structures seeks to identify statistical values with the optimum mass points for each structure in a given simulation. Finally, probabilistic comparisons are made to verify the efficiency and possible limitations of the algorithm under study, indicating that the BWO algorithm provides results of optimal points in three of the four truss structures studied.

Keywords: optimization; heuristic methods; algorithms.

LISTA DE FIGURAS

Figura 1 - Pseudocódigo FA.....	27
Figura 2 - Comportamento de alimentação com rede de bolhas de baleias jubarte..	28
Figura 3 - (a) Vetores de posição 2D e (b) 3D e suas próximas localizações possíveis (X^* é a melhor solução obtida até o momento)	29
Figura 4 - Mecanismo de busca em rede de bolhas implementado no WOA (X^* é a melhor solução obtida até o momento): (a) mecanismo de encolhimento e (b) posição de atualização da espiral.....	31
Figura 5 - Mecanismo de “ <i>exploration</i> ” implementado no WOA (X^* é a baleia escolhida aleatoriamente)	32
Figura 6 - Pseudocódigo do algoritmo WOA.....	33
Figura 7 - Comportamentos das baleias beluga, (a) natação, correspondente à fase de “ <i>exploration</i> ”; (b) caça, correspondente à fase de “ <i>exploitation</i> ”, (c) queda da baleia, para o estágio de queda	37
Figura 8 - Pseudocódigo do algoritmo BWO	38
Figura 9 - Função Goldstein Price	44
Figura 10 - Função Branin.....	45
Figura 11 - Função Shekel em sua forma bidimensional	46
Figura 12 - Problema com restrição <i>Eggholder</i> modificado.....	47
Figura 13 - Mola sob tração/compressão	48
Figura 14 - Viga soldada	48
Figura 15 - Exemplo de benchmark de treliça de onze barras.....	51
Figura 16 - Exemplo de benchmark de treliça de trinta e nove barras.....	51
Figura 17 - Exemplo de referência de vinte e cinco barras (treliça 3D).....	52
Figura 18 - Torre de Cigré adotada para o trabalho.....	55
Figura 19 - Casos de Carga.....	56
Figura 20 - Otimização para a treliça de 11 barras.....	60
Figura 21 - Otimização para a treliça de 39 barras.....	61
Figura 22 - Otimização para a treliça espacial de 25 barras.....	62

LISTA DE QUADROS

Quadro 1 - Valores ótimos das funções com NIR =10.....	56
Quadro 2 - Valores ótimos das funções com NIR =100.....	56
Quadro 3 - Valores ótimos das funções com parâmetros WOA.....	57
Quadro 4 - Comparativo entre ótimos obtidos no algoritmo BWO e valores existentes da literatura para as funções bechmark.....	57
Quadro 5 - Comparativo entre ótimos obtidos no algoritmo BWO e valores existentes da literatura para os problemas da mola e viga soldada.....	58
Quadro 6 - Resultados estatísticos algoritmo BWO - treliça 11 barras.....	59
Quadro 7 - Resultados estatísticos algoritmo WOA - treliça 11 barras.....	60
Quadro 8 - Resultados estatísticos algoritmo FA - treliça 11 barras.....	60
Quadro 9 - Resultados estatísticos algoritmo BWO - treliça 39 barras.....	63
Quadro 10 - Resultados estatísticos algoritmos WOA - treliça 39 barras.....	63
Quadro 11 - Resultados estatísticos algoritmos FA - treliça 39 barras.....	64
Quadro 12 - Resultados estatísticos algoritmo BWO – treliça espacial.....	66
Quadro 13 - Resultados estatísticos algoritmo WOA – treliça espacial.....	67
Quadro 14 - Resultados estatísticos algoritmo FA – treliça espacial.....	67
Quadro 15 - Resultados estatísticos algoritmo BWO – Torre.....	69
Quadro 16 - Resultados estatísticos algoritmo WOA – Torre.....	70
Quadro 17- Resultados estatísticos algoritmo FA – Torre.....	70

LISTA DE TABELAS

Tabela 1 - Parâmetros para o projeto de treliça de onze barras	52
Tabela 2 - Carregamentos.....	53
Tabela 3 - Coordenadas dos nós e agrupamento de membros	53

LISTA DE ABREVIATURAS E SIGLAS

BWO	<i>Beluga Whale Optimization</i>
FA	<i>Firefly Algorithm</i>
MATLAB	<i>Matrix Laboratory</i>
NIR	<i>Number of Independent Runs (Número de execuções independentes)</i>
PSO	<i>Particle Swarm Optimization</i>
WOA	<i>Whale Optimization Algorithm</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVO GERAL.....	17
1.2	OBJETIVOS ESPECÍFICOS.....	17
1.3	TRABALHOS ANTERIORES.....	17
1.4	JUSTIFICATIVA.....	19
2	OTIMIZAÇÃO	20
2.1	OTIMIZAÇÃO COMPUTACIONAL.....	20
2.3	MÉTODOS HEURÍSTICOS.....	21
2.3.1	Busca global	22
2.3.2	Busca local	23
2.4	ALGORITMOS POR ENXAME DE PARTÍCULAS (SWARM).....	23
2.4.1	Estrutura dos algoritmos SWARM	24
2.5	ALGORITMO FA.....	25
2.5.1	Pseudocódigo FA	26
2.6	ALGORITMO WOA.....	26
2.6.1	Fase 1: Operador de presas circundantes	27
2.6.2	Fase 2: Operador de rede de bolhas	29
2.6.3	Fase 3: Operador de busca pela presa	30
2.6.4	Pseudocódigo WOA	31
2.7	ALGORITMO BWO.....	32
2.7.1	Fase de busca global (“exploration”)	34
2.7.2	Fase de busca local (“exploitation”)	35
2.7.3	Queda da baleia	36
2.7.4	Funcionamento do algoritmo BWO	37
2.7.5	Pseudocódigo BWO	37
2.7.6	Diferença entre BWO e WOA	38
2.8	COMPARAÇÕES ESTATÍSTICAS E PROBABILÍSTICAS.....	39
3	METODOLOGIA	41
3.1	VERIFICAÇÃO DO ALGORITMO.....	41
3.2	PROBLEMAS DE OTIMIZAÇÃO GERAIS.....	42
3.2.1	Função Goldstein-Price	42
3.2.2	Função Branin	43

3.2.3	Função Shekel.....	44
3.2.4	Problema com restrição por <i>Eggholder</i> Modificada	45
3.2.5	Problema de engenharia: Mola sob tração/compressão	46
3.2.6	Problema de engenharia: viga soldada.....	47
3.3	TRELIÇAS PROPOSTAS	50
3.3.1	Treliça plana com onze barras.....	50
3.3.2	Treliça plana de trinta e nove barras.....	51
3.3.3	Treliça espacial vinte e cinco barras.....	52
3.4	TORRE DE CIGRÉ	53
4	RESULTADOS E DISCUSSÃO	56
4.1	FUNÇÕES <i>BENCHMARK</i>	56
4.2	TRELIÇAS 58	
4.2.1	Treliça plana de onze barras.....	58
4.2.2	Treliça plana de trinta e nove barras.....	62
4.2.3	Treliça espacial de vinte e cinco barras.....	65
4.3	TORRE DE CIGRÉ	69
	CONCLUSÃO	72
	REFERÊNCIAS	74

1 INTRODUÇÃO

O processo de otimização estrutural é um dos grandes desafios da engenharia. O desenvolvimento da otimização é pautado principalmente na busca pela redução de custos pelos profissionais da área. A utilização de métodos computacionais baseados em algoritmos pode ser considerada uma solução viável, mas é importante ressaltar que pode ocorrer uma variação em relação à eficácia do algoritmo quando se trata de resolver um determinado problema. Este processo é conduzido por um modelo matemático bem definido que possui uma função objetivo, parâmetros (população, iterações máximas, entre outros) e possíveis restrições.

A otimização é fundamentada em encontrar a melhor solução para um dado problema. Conforme Brasil (2022) a utilização de métodos de otimização no dimensionamento estrutural é relevante pois oferece soluções de projeto que agregam economia e segurança. Além disso, esses métodos tornam possível automatizar o processo repetitivo de verificação dos elementos estruturais.

Os problemas de otimização estrutural existentes na literatura podem ser classificados em três vertentes: otimização paramétrica (ou dimensional), otimização de forma (ou geométrica) e otimização topológica. Na otimização paramétrica, as dimensões das seções transversais dos elementos são otimizadas, sem que haja alterações na forma da estrutura ou mudança de elementos. Quando as grandezas que representam a geometria da estrutura são tratadas como variáveis de projeto, tem-se um problema de otimização de forma. Já nos problemas de otimização topológica, busca-se a melhor distribuição de material dentro do domínio de projeto, através da remoção de elementos que não contribuem efetivamente para o desempenho estrutural (Silva, 2015).

Por serem métodos iterativos e de natureza estocástica, a cada execução de algoritmo heurístico são obtidos resultados diferentes para um mesmo problema. Ainda, cada algoritmo heurístico pode ser eficiente para um problema e ineficiente para outro. Esses algoritmos podem ser baseados em uma determinada população, como por exemplo, vaga-lumes (*Firefly Algorithm – FA*) e baleias beluga (*Beluga Whale Optimization - BWO*) e jubarte (*Whale Optimization Algorithm - WOA*).

Visto isso, com as frequentes atualizações de algoritmos desenvolvidos com as proposições de novos algoritmos é necessário conhecer o desempenho dos mesmos, apontando seus pontos fortes e possíveis limitações.

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo principal avaliar o desempenho de estruturas treliçadas através da implementação do algoritmo meta-heurístico da baleia beluga (*Beluga Whale Optimization - BWO*), uma versão modificada do algoritmo da baleia jubarte (*Whale Optimization Algorithm - WOA*).

1.2 OBJETIVOS ESPECÍFICOS

Para cumprir o objetivo geral deste trabalho pode-se elencar os objetivos específicos:

- Validar o algoritmo através das funções **benchmark** e problemas clássicos de engenharia;
- Refinar os parâmetros de busca do algoritmo;
- Realizar uma análise comparativa entre os resultados dos algoritmos da literatura existente e o BWO, para as funções **benchmark** e problemas clássicos de engenharia;
- Aplicar os algoritmos BWO, WOA e FA para as estruturas treliçadas (tamanho, forma e topologia) e para a torre de transmissão (tamanho e forma);
- Gerar comparativos probabilísticos entre as simulações realizadas nos algoritmos para as estruturas treliçadas e para a torre de transmissão.

1.3 TRABALHOS ANTERIORES

Métodos para otimização de estruturas treliçadas, utilizando o tamanho das barras como variável a ser otimizada, são totalmente estabelecidos na literatura conforme Hemp (1973) e Adeli e Kamal (1991).

O Algoritmo Firefly tem sido amplamente utilizado para pesquisar valores ótimos de estruturas treliçadas. O mesmo foi desenvolvido por Yang (2009) e utiliza o comportamento do vaga-lume como base para o algoritmo.

Bez (2011) estudou funções de treliças a serem otimizadas utilizando como parâmetro a massa total da estrutura. O autor utilizou o Firefly Algorithm (FA), por

meio do Software MATLAB na resolução de três diferentes estruturas, uma treliça plana e duas espaciais, obtendo valores que representam a eficiência nesses três casos.

Um estudo com o algoritmo Firefly também foi realizado por Miguel e Lopez (2013) em treliças planas e espaciais, com variação de tamanho, forma e topologia para as estruturas. A eficácia do FA em resolver a otimização simultânea de tamanho, forma e topologia de treliças é demonstrada por meio de problemas de referência, cujos resultados são semelhantes ou até melhores do que os relatados na literatura, produzindo custos computacionais menores.

Já o algoritmo das baleias WOA, que é baseado no comportamento das baleias Jubarte, foi publicado por Lewis e Miraji (2016) considerando 29 funções **benchmark**, problemas clássicos de engenharia e treliças, validando a capacidade do mesmo em encontrar os pontos ótimos destas funções. De acordo com Lewis e Miraji (2016) o algoritmo se mostrou competitivo quando comparado com os existentes na meta-heurística e com outros métodos convencionais.

Recentemente houve a publicação do algoritmo BWO criado por Li, Meng e Zong (2022), também baseado na natureza, porém com foco no comportamento da baleia beluga. O algoritmo é uma versão modificada do WOA, foram realizadas pelos autores implementações de algumas funções **benchmark** e problemas de engenharia, porém o mesmo não apresenta validação para estruturas treliçadas e torre de transmissão.

De acordo com Li, Meng e Zong (2022), o BWO é um algoritmo competitivo na resolução de problemas de otimização unimodal e multimodal, pois em relação aos algoritmos comparados dentro do artigo, o BWO é o primeiro na análise de escalabilidade de funções **benchmark** entre os algoritmos meta-heurísticos apresentados por eles.

1.4 JUSTIFICATIVA

Os problemas de otimização visam encontrar os valores extremos para uma determinada função. No caso dos algoritmos heurísticos a busca pode ocorrer para localizar pontos que minimizem a função objetivo, apresentando assim o valor ótimo para uma determinada situação.

O presente trabalho apresenta a validação com algumas funções restritas e outras sem restrições, para problemas de estruturas treliçadas planas, espacial e uma torre de transmissão, utilizando o algoritmo da baleia beluga para verificação dos valores ótimos das funções mencionadas.

A relevância do trabalho está em analisar a eficiência do algoritmo BWO, principalmente quando aplicado aos modelos de otimização de treliças e de torre de transmissão, considerando a importância que essas estruturas possuem em um sistema de distribuição de energia elétrica, a otimização de seus pesos traz um impacto econômico e ecológico, melhorando a eficácia e a segurança destas construções.

2 OTIMIZAÇÃO

Neste capítulo são abordados os conceitos relacionados a otimização computacional e as pesquisas recentes, tendo como foco principal as meta-heurísticas dos algoritmos da baleia, chamados de WOA e BWO. Muitos trabalhos acadêmicos têm sido desenvolvidos na área de otimização de estruturas, verificando a eficiência de algoritmos meta-heurísticos para resolução de problemas de engenharia. As aplicações de modelos de otimização no dimensionamento estrutural tornam-se relevantes ao fornecer soluções que agregam economia e segurança.

2.1 OTIMIZAÇÃO COMPUTACIONAL

Conforme Martha (2010), a utilização de computadores é de grande relevância na otimização de estruturas, devido a agilidade para resolver operações matemáticas, a análise de estruturas passou a ser feita computacionalmente em quase todos escritórios de cálculo estrutural.

A otimização pode ser caracterizada como um processo de busca por valores mínimos ou máximos de uma determinada função em um domínio de projeto. Em geral, a resolução desses problemas é realizada com o auxílio de metodologias implementadas computacionalmente, com características diferentes, mas com um objetivo em comum, encontrar o ponto ótimo do problema. Os algoritmos em geral são desenvolvidos e aperfeiçoados pensando-se em dois fatores: precisão dos resultados e velocidade de convergência (Assis, 2021).

O processo de otimização estrutural geralmente busca a diminuição de custos dos elementos, o que pode ser correlacionado com a função objetivo, de modo que esta descreva a massa da estrutura em função das variáveis de projeto. Os algoritmos de otimização podem ser separados em dois grandes grupos: determinísticos e heurísticos (Saka e Geem, 2013).

O algoritmo caracterizado por uma regra determinística, apresenta uma mesma solução para o problema por ele executado. Grande parte destes algoritmos faz uso de informações de derivadas da função objetivo e das restrições aplicadas, exigindo os valores dos gradientes (derivada de primeira ordem) da função objetivo e, em alguns casos, da matriz hessiana (derivada de segunda ordem) (Arora, 2017).

Souza, Amaral e Almeida (2019) comentam que a grande disponibilidade de recursos computacionais tem impulsionado o desenvolvimento das técnicas de otimização para auxílio na tomada de decisões de projeto. Os métodos de programação matemática, mais tradicionais, exigem domínios convexos e continuidade das variáveis de projeto, da função objetivo e das restrições, porém algumas vezes os problemas reais não se enquadram nessa perspectiva. Além disso, a necessidade de avaliação da precisão da função objetivo e das restrições em relação às variáveis de projeto muitas vezes dificulta sua aplicação. Desse modo, os métodos heurísticos surgiram como alternativa de formulação e implementação mais simples para resolução de problemas.

Conforme Saka e Geem (2013), a técnica de otimização, feita através de algoritmos meta-heurísticos, não requer obter as derivadas das funções. Estes são fundamentados em paradigmas de busca aleatória baseados na simulação de fenômenos naturais. São métodos não tradicionais de busca e otimização, muito adequados e eficazes para encontrar a solução ótima de problemas de otimização.

2.3 MÉTODOS HEURÍSTICOS

Algoritmos meta-heurísticos fundamentados em fenômenos naturais têm sido cada vez mais utilizados nos últimos anos, essa imitação de processos naturais é um dos motivos de sua disseminação, além da sua capacidade em resolver problemas de otimização global. As meta-heurísticas também são formas eficientes de produzir soluções para problemas complexos através de tentativas em um tempo razoavelmente curto, melhorando o custo computacional (Yang, 2011).

Para Miguel e Lopez (2013), os algoritmos meta-heurísticos são adequados para resolver problemas de otimização. As vantagens conhecidas do algoritmo podem ser descritas como, eles não requerem informações de gradiente e podem ser aplicados a problemas nos quais o gradiente é difícil de obter ou simplesmente não existe, não ficam presos em mínimos locais se sintonizados corretamente, podem ser utilizados em funções não suaves ou descontínuas, fornecem um conjunto de soluções ótimas em vez de uma única solução, e por último eles podem ser empregados para resolver problemas de otimização de variáveis mistas, isto é, envolvendo problemas de maior complexidade.

Li, Meng e Zong (2022), afirmam ainda que quatro características podem ser encontradas em algoritmos meta-heurísticos, a simplicidade, a flexibilidade, o mecanismo livre de derivação e prevenção de ótimos locais.

Geralmente, os algoritmos meta-heurísticos são desenvolvidos para solucionar problemas sem restrição. Para resolver problemas de otimização com restrições se exige uma adaptação, que normalmente é atendida pela inserção de funções de penalidade, que incrementam o valor da função objetivo na medida em que as restrições são violadas (Saka e Geem, 2013).

Os algoritmos de otimização podem ser classificados conforme o seu processo de busca. De acordo com Talbi (2009), os métodos baseados em um único indivíduo tendem a realizar buscas locais enquanto que algoritmos que utilizam populações de diferentes amostras tendem a realizar uma busca mais ampla, chamada de busca global.

A diversificação e a intensificação são aspectos relevantes em questão de desempenho para os algoritmos meta-heurísticos. A exploração do domínio das variáveis é fornecida pelo fator de diversificação. Porém, diversificações abrangentes podem dificultar a convergência da função objetivo, pois mais pontos no domínio de projeto serão analisados. Já o fator de intensificação é quem controla a “prática” que o algoritmo adquire a cada iteração, buscando descartar as soluções não ideais para a função em estudo (Yang, 2009).

Nos últimos anos, um grande número de algoritmos meta-heurísticos surgiu como solução de problemas no processo de otimização. Além disso, os algoritmos meta-heurísticos híbridos inspirados na física ou na química também obtiveram convergência para problemas de otimização (Li, Meng e Zong, 2022).

2.3.1 Busca global

O método de busca global tem por objetivo avaliar, de forma mais ampla, o domínio de projeto do problema, tem a habilidade de visitar diferentes regiões em busca do mínimo local. Esse processo diminui a chance de o algoritmo retornar um mínimo local como solução e não encontrar o valor ótimo para o problema (Assis, 2021).

Ainda de acordo com o autor, os algoritmos de busca global partem de populações iniciais de amostras que estão espalhadas por todo domínio de projeto,

fazendo com que mais regiões desse domínio possam ser exploradas, aumentando a chance de encontrar mínimos isolados.

A exploração de um determinado domínio buscando por regiões promissoras é uma das dificuldades encontradas nos métodos matemáticos. Já os algoritmos meta-heurísticos, baseados em população, conseguem uma convergência aceitável mesmo para problemas não-convexos, dada a capacidade computacional necessária (Carlton, 2015).

2.3.2 Busca local

Um processo iterativo que tende a convergir para um determinado valor, seja ele de mínimo ou máximo, em um determinado domínio de projeto, pode ser chamando de busca local (Ribeiro, 2006).

Algoritmos que fazem busca local, representam a capacidade de refinar a busca em uma determinada região para encontrar o mínimo local. Muitos algoritmos matemáticos se enquadram nessa categoria, em geral, esses métodos precisam de funções diferenciáveis, utilizando o gradiente ou a Hessiana para encontrar o próximo ponto no processo iterativo (Assis, 2021).

2.4 ALGORITMOS POR ENXAME DE PARTÍCULAS (SWARM)

Desenvolvido por James Kennedy e Russel Eberhartem em 1995, o algoritmo foi inspirado no comportamento e na dinâmica dos movimentos dos pássaros, insetos e peixes. Foi originalmente desenvolvido para problemas de otimização com variáveis contínuas (Lacerda, 2007).

A otimização por enxame de partículas é baseada na maneira pela busca de alimentos de algumas populações, como por exemplo, pássaros, peixes, formigas, e gerou toda uma classe de algoritmos cuja ideia básica é fazer com que as partículas se “movam” pelo domínio de projeto. O algoritmo simula várias partículas se movendo ao mesmo tempo pela região buscando o mínimo global. A diferença entre os algoritmos é a maneira que as partículas se movem e a interação entre elas (Arora, 2017).

No caso do algoritmo heurístico mais simples (*Particle Swarm Optimization - PSO*), que é baseado em buscar a melhor solução em um determinado domínio, e as

partículas seguem o melhor ponto de projeto já encontrado até o momento. O enxame evolui por meio de competição e cooperação entre a população e as partículas utilizam a sua experiência e dos outros indivíduos para a busca pelo ponto ótimo (Arora, 2017).

Segundo Rao (2009), no algoritmo PSO a partícula representa um indivíduo dentro do conjunto do enxame. Cada indivíduo se comporta de maneira independente do outro, porém quando ele encontra um caminho razoável em direção ao alimento ele alerta os outros indivíduos sobre sua posição, fazendo com que se movam em sua direção.

2.4.1 Estrutura dos algoritmos SWARM

Conforme Lopez (2022) o movimento baseado em seguir algo é dividido em duas porções: a parte “determinística” e a parte “aleatória”. A parcela aleatória serve para aumentar a diversidade populacional e a parcela determinística é a que propriamente segue o ponto especificado no domínio. A função que mostra a busca pelo novo ponto é apresentada através da equação 01 abaixo.

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + c_1 \xi_1 (\mathbf{X}_{target} - \mathbf{X}_i^{(t)}) + c_2 \xi_2 \quad (01)$$

Onde:

$\mathbf{X}_i^{(t+1)}$ é a próxima iteração;

$\mathbf{X}_i^{(t)}$ é a iteração atual;

$c_1 \xi_1$ é a parcela determinística;

\mathbf{X}_{target} é o melhor ponto (alvo);

$c_2 \xi_2$ é a parcela aleatória.

Sendo que X_i representa a posição da partícula, os parâmetros ξ_1 e ξ_2 são variáveis aleatórias entre 0 e 1, c_1 e c_2 são os coeficientes que multiplicam as parcelas determinística e aleatória respectivamente, participando da definição de tamanho do passo. Os parâmetros c_1 e c_2 podem ser constantes ou diminuir à medida que as iterações vão acontecendo. Neste caso, as perturbações seriam menores à medida que aumenta o número de iterações melhorando a busca local do algoritmo (Lopez, 2022).

2.5 ALGORITMO FA

Originado por Yang em 2009, o algoritmo de otimização do vaga-lume é baseado em enxames e imita o comportamento dos vaga-lumes para atrair parceiros e possíveis presas.

Os agentes de pesquisa possuem uma atratividade relativa, conforme com a distância entre eles, e é proporcional ao seu brilho. O brilho do vaga-lume é função da distância do agente de pesquisa em relação ao ótimo global da função objetivo. O desempenho do algoritmo foi avaliado diante da otimização de funções multimodais e os resultados apontaram que o FA tem um potencial de eficiência melhor que outros algoritmos, tais como, por exemplo, o PSO (Souza, 2021).

A atratividade β de um determinado vaga-lume, julgada por outro vaga-lume a uma distância r_f , pode ser definida pela equação 02 (Yang, 2010):

$$\beta = \beta_0 e^{-\gamma \cdot r_f^2} \quad (02)$$

Onde:

β_0 é a atratividade quando $r_f=0$;

γ é um coeficiente de absorção de luz.

A distância entre quaisquer dois vaga-lumes i e j localizados nas posições \mathbf{X}_i e \mathbf{X}_j é dada na equação 03 (Yang, 2010).

$$r_{fij} = \|\mathbf{X}_i - \mathbf{X}_j\| = \sqrt{\sum_{k=1}^d (\mathbf{X}_{i,k} - \mathbf{X}_{j,k})^2} \quad (03)$$

Sendo que $X_{i,k}$ é o k -ésimo componente da coordenada espacial X_i do i -ésimo vaga-lume.

Yang (2010) diz que, o movimento de um vaga-lume i , atraído por outro vaga-lume mais brilhante j , é descrito pela equação 04.

$$X_{i+1} = X_i + \beta_0 e^{-\gamma \cdot r_f^2} (\mathbf{X}_j - \mathbf{X}_i) + \alpha_r \cdot \epsilon_i \quad (04)$$

O 2º termo da direita é relativo à atração, e o último termo é referente à randomização, sendo α_r um parâmetro de randomização e ϵ_i um vetor de números aleatórios extraídos de uma distribuição gaussiana ou uniforme.

Normalmente, na implementação do algoritmo FA, utiliza-se $\beta_0=1$; $\alpha_r \in [0,1]$ e $\epsilon_i = rand - 1/2$, onde *rand* é um gerador de números aleatórios distribuído uniformemente no intervalo de 0 a 1 (Yang, 2010).

2.5.1 Pseudocódigo FA

Yang (2009) desenvolveu o pseudocódigo do algoritmo de otimização do vaga-lume, apresentado na Figura 1.

Figura 1– Pseudocódigo FA

```

Função objetivo  $f(\mathbf{X})$ ,  $\mathbf{X} = (x_1 \dots x_d)^T$ 
Gerar população inicial de  $n$  vaga-lumes  $\mathbf{X}_i$  ( $i = 1, 2, \dots, n$ )
A intensidade da luz  $I_i$  em  $\mathbf{X}_i$  é determinada por  $f(\mathbf{X}_i)$ 
Definir coeficiente de absorção de luz  $\gamma$ 
enquanto ( $t < \text{MaxGeneration}$ )
  para  $i = 1: n$  para todos os  $n$  vaga-lumes
    para  $j = 1: i$  para todos  $n$  vaga-lumes
      se ( $I_j > I_i$ ), Move vaga-lume  $i$  em direção a  $j$  na dimensão  $d$ ; fim se
      A atratividade varia com a distância  $r$  via  $\exp[-\gamma r]$ 
      Avalie novas soluções e atualize a intensidade da luz
    fim para  $j$ 
  fim para  $i$ 
  Classifique os vaga-lumes e encontre os melhores do momento
fim enquanto
Resultados pós-processamento e visualização

```

Fonte: Adaptado de Yang (2009)

2.6 ALGORITMO WOA

Criado em 2016 por Mirjalili e Lewis, o algoritmo de otimização das baleias é baseado em enxames e imita o comportamento de caça das baleias jubarte. Este item do presente trabalho é todo baseado em Mirjalili e Lewis (2016).

Esta espécie de baleia utiliza um método chamado de rede de bolhas para realizar a caça e se alimentar, já que as mesmas preferem caçar os cardumes de peixes pequenos próximos à superfície. As baleias jubarte mergulham aproximadamente 12 metros para baixo e, em seguida, começam a criar bolhas em forma de espiral ao redor do cardume e na sequência nadam em direção à superfície, que é denominado “espiral ascendente”. Esse movimento faz com que o cardume fique agrupado e posteriormente as baleias nadam em direção às presas no estágio de “espiral com duplas rotações”. A Figura 2 apresenta a rede de bolhas.

Figura 2 - Comportamento de alimentação com rede de bolhas de baleias jubarte



Fonte: Mirjalili e Lewis (2016)

O algoritmo das baleias é dividido em três fases: operador das presas circundantes, operador de neutralização das bolhas e operador de busca pela presa. Essas etapas são descritas na sequência.

2.6.1 Fase 1: Operador de presas circundantes

A primeira fase da caça é quando as baleias localizam as presas e circundam elas. De forma equivalente, o WOA assume que a melhor solução é a presa mais próxima das presas e por isso a baleia mais próxima é definida como o ponto a ser seguido, atualizando a posição das demais baleias em relação a essa. Esse

comportamento está vinculado à busca global do algoritmo e é representado pelas equações 05 a 08.

$$D = |C \cdot X^*(t) - X(t)| \quad (05)$$

$$X^{(t+1)} = X^*(t) - A \cdot D \quad (06)$$

$$A = 2a \cdot r - a \quad (07)$$

$$C = 2 \cdot r \quad (08)$$

Onde:

t é a iteração atual;

A e C são vetores de coeficientes;

X^* é a posição do vetor da melhor solução encontrada até o momento;

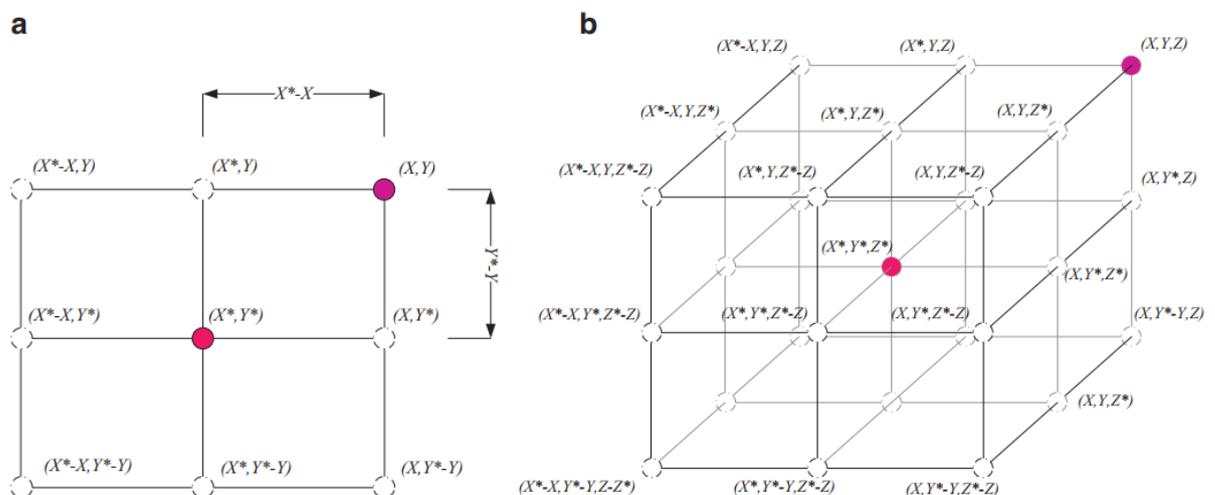
X é o vetor posição;

a é um valor de 2 a 0 que decresce linearmente (tanto na fase de busca global quanto na fase de busca local);

r é um vetor aleatório em $[0, 1]$;

As Figuras 3 **a** e **b** ilustram a lógica por trás das equações apresentadas para duas e três dimensões e mostram que a equação permite que qualquer agente de pesquisa atualize sua posição na vizinhança da melhor solução atual e simulam o cerco da presa.

Figura 3– **(a)** Vetores de posição 2D e **(b)** 3D e suas próximas localizações possíveis (X^* é a melhor solução obtida até o momento)



O mesmo conceito pode ser estendido para um problema com n dimensões, e os agentes de pesquisa se moverão em hipercubos em torno da melhor solução obtida até o momento.

2.6.2 Fase 2: Operador de rede de bolhas

Nesta etapa as baleias nadam em torno das presas, subindo em direção a superfície em círculos e diminuindo o raio do nado, formando um espiral. Por isso, essa fase é considerada como a fase de “*exploitation*”, ou seja, de busca local. Para representar esse comportamento, o algoritmo assume uma probabilidade de 50% para a escolha do mecanismo de subida entre o mecanismo de encolhimento ou espiral, para realizar a atualização da posição das baleias. As Figuras 4 **a** e **b** e equação 09 representam essa fase.

$$\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{X}^*(t) - A \cdot D & \text{se } p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{X}^*(t) & \text{se } p \geq 0.5 \end{cases} \quad (09)$$

Onde:

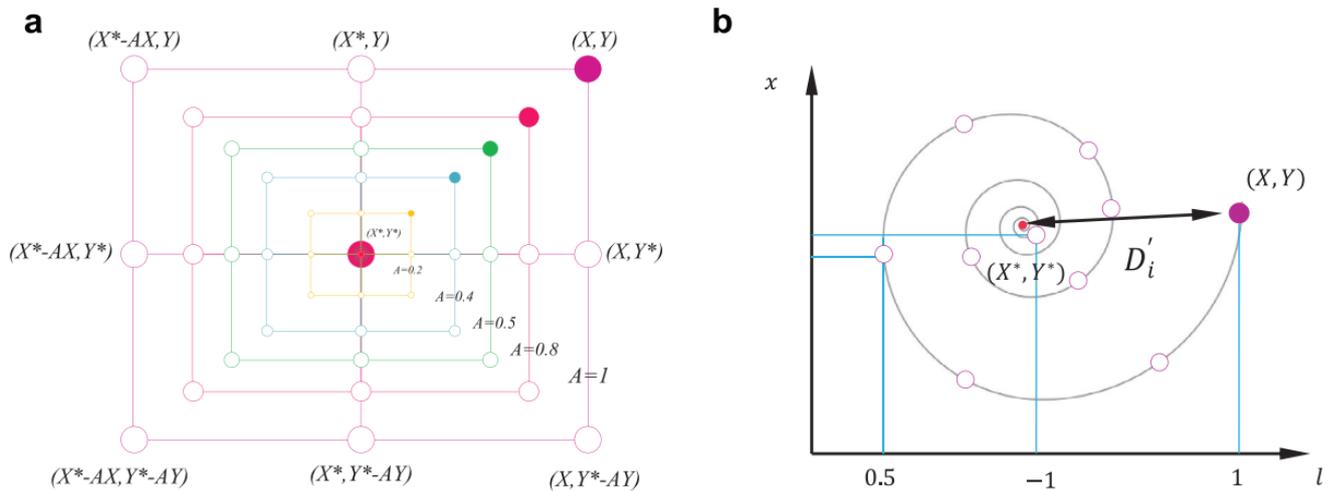
D é a distância da i -ésima baleia até a presa;

b é a constante que define o formato do espiral logarítmico;

l é um valor aleatório no intervalo $[-1, 1]$;

p é um valor aleatório no intervalo $[0, 1]$.

Figura 4– Mecanismo de busca em rede de bolhas implementado no WOA (X^* é a melhor solução obtida até o momento): **(a)** mecanismo de encolhimento e **(b)** posição de atualização da espiral



Fonte: Mirjalili e Lewis (2016)

Além de utilizar o método da rede de bolhas, as baleias jubarte procuram por presas aleatoriamente e o modelo matemático para essa hipótese é apresentado na fase 3.

2.6.3 Fase 3: Operador de busca pela presa

Nesta fase, a atualização da posição da baleia mais próxima é realizada com base em uma baleia aleatória, ao invés de escolher a posição da melhor baleia. Desta forma, o vetor $A > 1$ ressalta o processo de “*exploration*”, enfatizando a busca global. O modelo matemático para esta etapa é apresentado a seguir com base nas equações 10 e 11 e a Figura 5 representa essa etapa.

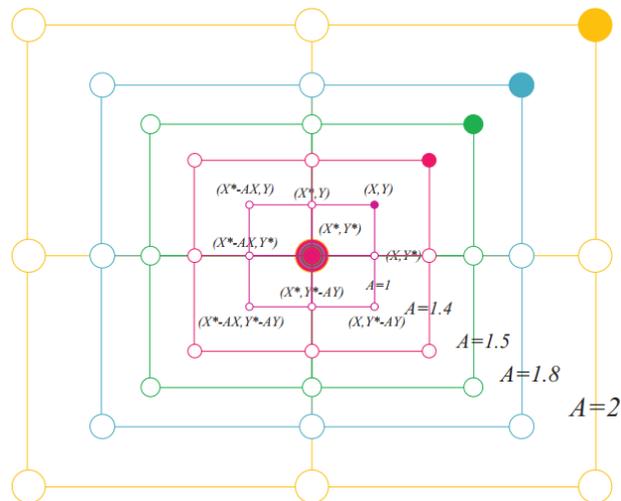
$$D = |C \cdot X_{rand} - X| \quad (10)$$

$$X^{(t+1)} = X_{rand} - A \cdot D \quad (11)$$

Onde:

X_{rand} é um vetor de posição aleatória (uma baleia aleatória) escolhido da população atual.

Figura 5– Mecanismo de “*exploration*” implementado no WOA (X^* é a baleia escolhida aleatoriamente)



Fonte: Mirjalili e Lewis (2016)

O algoritmo WOA utiliza os operadores de presas circundantes e o método de ataque de rede de bolhas para realizar a busca local dos valores ótimos. Já o operador de busca pela presa trabalha na fase de “*exploration*”, ou seja, busca global.

2.6.4 Pseudocódigo WOA

Mirjalili e Lewis (2016) desenvolveram o pseudocódigo do algoritmo de otimização das baleias jubarte, apresentado na sequência, através da Figura 6.

Figura 6– Pseudocódigo WOA

```

Inicialize a população de baleias  $X_i = (i = 1, 2, \dots, n)$ 
Calcule a função fitness para cada agente de pesquisa
 $X^*$  = melhor agente de pesquisa
enquanto ( $t <$  número máximo de iterações)
  para cada agente de pesquisa
    Atualize  $a$ ,  $A$ ,  $C$ ,  $l$  e  $p$ 
    se1 ( $p < 0,5$ )
      se2 ( $|A| < 1$ )
        Atualize a posição de cada agente de pesquisa
      senão se2 ( $|A| \geq 1$ )
        Selecione aleatoriamente um agente de pesquisa ( $X_{rand}$ )
        Atualize a posição de cada agente de pesquisa
      fim se2
    senão se1 ( $p \geq 0,5$ )
      Atualize a posição de cada agente de pesquisa
    fim if1
  fim para
  Checar se cada agente de pesquisa saiu do domínio e altere-o
  Calcule a função fitness para cada agente de busca
  Atualize  $X^*$  se for a melhor solução
   $t = t + 1$ 
fim enquanto
retorna  $X^*$ 

```

Fonte: Adaptado de Mirjalili e Lewis (2016)

2.7 ALGORITMO BWO

Criado por Li, Meng e Zong (2022) esse algoritmo de otimização é baseado em enxames e imita o comportamento de caça das baleias beluga. Este item do presente trabalho é todo baseado em em Li, Meng e Zong (2022).

A baleia beluga têm visão aguçada e capacidade auditiva, e se movem para caçar através do som.

A baleia beluga pode nadar lateralmente com barbatana peitoral levantada que estende a barbatana peitoral para longe do corpo, de modo que, a aleta fica

perpendicular ao corpo. Elas podem mergulhar ou emergir de maneira sincronizada ou espelhada ao nadar, conhecida como moagem. Também podem liberar bolhas do orifício de sopro e nadar rapidamente para longe de outros animais. Além disso, eles podem ser vistos brincando, nadando, vocalizando um ao redor do outro.

Inspirado no comportamento da baleia beluga, incluindo o nado, caça e queda de baleias, foi desenvolvido um novo algoritmo meta-heurístico chamado de otimização da baleia beluga (BWO). Semelhante a outras meta-heurísticas, o BWO contém a fase de busca global e a busca local. Para modelar os comportamentos, as baleias belugas são consideradas como agentes de busca que podem se mover no domínio de projeto alterando seus vetores de posição.

Além disso, a probabilidade de queda de baleia é considerada mudando as posições de baleias beluga. Devido ao mecanismo baseado na população de BWO, as baleias beluga são consideradas agentes de busca, enquanto cada baleia é uma possível solução, que é atualizada durante as iterações.

A matriz para as posições dos agentes de busca é modelada como a equação 12:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} \\ \cdot & \cdot & \cdot \\ x_{n,1} & \cdots & x_{n,d} \end{bmatrix} \quad (12)$$

Onde:

n é o tamanho da população de baleias beluga;

d representa a dimensão das variáveis de projeto.

Para cada iteração, os valores da função de avaliação (fitness) correspondente são armazenados conforme a equação 13:

$$\mathbf{F}_X = \begin{bmatrix} f(x_{1,1} & \cdots & x_{1,d}) \\ \cdot & \cdot & \cdot \\ f(x_{n,1} & \cdots & x_{n,d}) \end{bmatrix} \quad (13)$$

O algoritmo BWO pode mudar da exploração global para a exploração local, dependendo do fator de equilíbrio Bf , que é modelado conforme a equação 14.

$$Bf = B_0(1 - T/2T_{max}) \quad (14)$$

Onde:

T é a iteração atual;

T_{max} é a iteração máxima;

B_0 é uma variável aleatória que assume valores no intervalo (0, 1) em cada iteração.

A fase de busca global acontece quando o fator de equilíbrio Bf é maior que 0,5 enquanto a fase de busca local acontece quando Bf é menor ou igual a 0,5. Com o aumento da iteração T , a faixa de flutuação de Bf é reduzida de (0, 1) a (0, 0,5), ilustrando a mudança significativa de probabilidades para a fase de busca local e global, enquanto a probabilidade de fase de busca local é aumentada com a iteração cada vez maior T .

2.7.1 Fase de busca global (“exploration”)

A fase de “exploration” do algoritmo BWO é estabelecida considerando o comportamento de nado da baleia beluga. A baleia beluga pode realizar comportamentos sob diferentes posturas, como o par de duas baleias beluga nadando juntas com modo sincronizado ou espelhado. Assim, as posições dos agentes de pesquisa são determinadas pelo modo de nado adotado pelas baleias e são atualizadas nas equações 15 e 16.

$$\mathbf{X}_{ij}^{T+1} = \mathbf{X}_{i,p_j}^T + (\mathbf{X}_{r,p_1}^T - \mathbf{X}_{i,p_j}^T)(1 + r_1)\text{sen}(2\pi r_2) \quad (15)$$

$$\mathbf{X}_{ij}^{T+1} = \mathbf{X}_{i,p_j}^T + (\mathbf{X}_{r,p_1}^T - \mathbf{X}_{i,p_j}^T)(1 + r_1)\text{cos}(2\pi r_2) \quad (16)$$

Onde:

T é a iteração atual;

\mathbf{X}_{ij}^{T+1} é a nova posição para a i -ésima baleia beluga na j -ésima dimensão;

p_j ($j = 1, 2, \dots, d$) é um número aleatório selecionado;

\mathbf{X}_{i,p_j}^T é a posição da baleia beluga em relação a p_j

$X_{r,p1}^T$ e $X_{i,pj}^T$ são as posições atuais para i-ésima e r-ésima baleia beluga (r é uma baleia beluga selecionada aleatoriamente), r_1 e r_2 são números aleatórios no intervalo entre 0 e 1, $sen(2\pi r_2)$ e $cos(2\pi r_2)$ referem-se ao nado espelhado das baleias beluga em direção à superfície.

2.7.2 Fase de busca local (“exploitation”)

A fase de exploração do BWO é inspirada no comportamento de caça das baleias beluga. As baleias beluga atacam compartilhando as informações posições entre si, considerando o melhor candidato. A estratégia de voo de Levy considera a velocidade constante durante o caminho percorrido pela baleia, esse procedimento é introduzido na fase de exploração local do BWO para aumentar a convergência, pois otimiza a busca em ambientes esparsos. Os autores supõem que as baleias podem pegar a presa com a estratégia de voo de Levy (caracterizado pela construção de trajetórias predominantemente curtas). O modelo matemático é expresso através da equação 17.

$$X_i^{T+1} = r_3 X_{best}^T - r_4 X_i^T + C_1 \cdot L_f \cdot (X_r^T - X_i^T) \quad (17)$$

Onde:

T é a iteração atual;

X_r^T e X_i^T são as posições para baleia aleatória e a i-ésima baleia beluga;

X_i^{T+1} é a nova posição para a i-ésima baleia beluga;

X_{best}^T é a melhor baleia;

r_3 e r_4 são variáveis aleatórias entre 0 e 1;

$C_1 = 2r_4 \left(1 - \frac{T}{T_{max}}\right)$, é a força do salto que mede a intensidade do voo de Levy;

$L_F = 0,05 \cdot \frac{\mu \cdot \sigma}{|v|^{1/\beta}}$, é a função do voo de Levy;

$$\sigma = \left(\frac{\Gamma(1+\beta) \cdot sen(\pi\beta/2)}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta}$$

Sendo: μ e v são números aleatórios normalmente distribuídos, Γ é constante e igual a 1,5.

2.7.3 Queda da baleia

Durante a migração e alimentação, as baleias beluga são ameaçadas por outras espécies. A maioria das baleias beluga são inteligentes e podem escapar dessas ameaças compartilhando informações entre si. No entanto, um pequeno número de baleias não sobrevive e cai no fundo do mar. O fenômeno é chamado de “queda da baleia”.

Para modelar o comportamento da queda das baleias em cada iteração, é utilizada uma probabilidade de queda da baleia dos indivíduos presentes na população, simulando pequenas mudanças nos grupos. Assume-se que essas baleias beluga se mudaram para outro lugar ou foram abatidas e caíram no mar profundo. Para garantir o número de tamanho da população constante, as posições das baleias beluga e tamanho do passo de queda são usados para estabelecer a posição atualizada. O modelo matemático é expresso através da equação 18.

$$\mathbf{X}_i^{T+1} = r_5 \mathbf{X}_i^T - r_6 \mathbf{X}_r^T - r_7 \mathbf{X}_{step} \quad (18)$$

Onde: r_5, r_6, r_7 são números aleatórios entre (0, 1) e \mathbf{X}_{step} é o tamanho do passo da queda da baleia estabelecido conforme a equação 19.

$$\mathbf{X}_{step} = (u_b - l_b) \exp(-C_2 T / T_{max}) \quad (19)$$

Neste modelo, a probabilidade de queda da baleia (Wf) é calculada como uma função linear com base na equação 20.

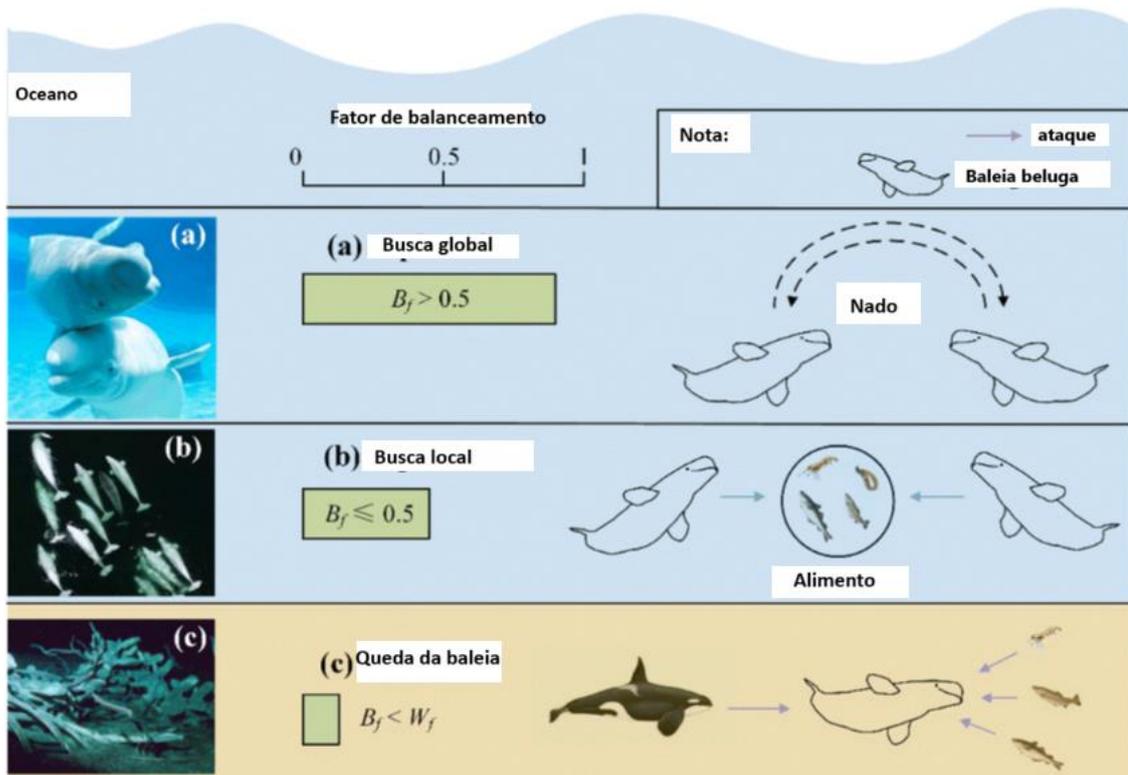
$$Wf = 0,1 - \frac{0,05T}{T_{max}} \quad (20)$$

A probabilidade da queda da baleia é reduzida de 0,1 na primeira iteração para 0,05 na última iteração, indicando que quando as baleias estão mais próximas do alimento durante o processo de otimização, o perigo diminui.

2.7.4 Funcionamento do algoritmo BWO

O BWO consiste em três principais fases: a fase de exploração global simulando o comportamento de natação, a fase de exploração local com comportamento predatório, e a fase de queda da baleia. Durante o processo de otimização, a fase de queda da baleia é implementada quando a fase de exploração global e local são concluídas em cada iteração. As Figuras 7 **a**, **b** e **c** representam todas essas etapas.

Figura 7 - Comportamentos das baleias beluga, (a) natação, correspondente à fase de busca global; (b) caça, correspondente à fase de busca local, (c) queda da baleia, para o estágio de queda



Fonte: Adaptado de Li, Meng e Zong (2022)

2.7.5 Pseudocódigo BWO

Li, Meng e Zong (2022) desenvolveram o pseudocódigo do algoritmo de otimização das baleias beluga, apresentado na Figura 8.

Figura 8 – Pseudocódigo BWO

Entrada: Parâmetros algorítmicos (tamanho da população, iteração máxima)

Resultado: A melhor solução

Inicialize a população e avalie os valores de aptidão, obtenha a melhor solução (X^*)

Enquanto $t \leq T_{max}$ **Faça**

Obtenha a probabilidade de queda da baleia Wf e fator de equilíbrio Bf

Para cada baleia beluga (X_i) **Faça**

Se $Bf(i) > 0,5$

// Na fase de exploração global

Gere p_j ($j = 1,2,\dots,d$) aleatoriamente a partir da dimensão

Escolha uma baleia beluga X_r aleatoriamente

Atualize a nova posição da i -ésima baleia beluga

Senão se $Bf(i) \leq 0,5$

// Na fase de exploração local

Atualize a força do salto aleatório C_1 e calcule a função de voo Levy

Atualize a nova posição da i -ésima baleia beluga

Fim se

Verifique os limites das novas posições e avalie os valores de aptidão

Fim para

Para cada baleia beluga (X_i) **Faça**

// a fase de queda da baleia

Se $Bf(i) \leq Wf$

Atualize o fator de passo $C2$

Calcule o tamanho do passo $Xstep$

Atualize a nova posição da i -ésima baleia beluga

Verifique os limites da nova posição e calcule o valor de aptidão

Fim se

Fim para

Encontre a melhor solução atual X^*

$T=T+1$

Fim enquanto

Produza a melhor solução

Fonte: Adaptado de Li, Meng e Zong (2022)

2.7.6 Diferença entre BWO e WOA

O BWO e o algoritmo de otimização de baleias jubarte (WOA) têm características em comum, são algoritmos baseados em população, possuem fase de busca global e fase de busca local e são inspirados nas baleias.

As diferenças entre BWO e WOA são as seguintes, o comportamento predador de WOA é inspirado no movimento espiral de jubarte, enquanto o

comportamento predador do BWO é inspirado na atualização das posições com base na própria posição procurando a melhor solução alimentar, sem o movimento espiral.

O WOA não tem a queda da baleia, enquanto o BWO tem a fase de queda da baleia para sair do ótimo local. Por último WOA não possui o mecanismo de voo Levy, que é introduzido no BWO, fazendo com que o algoritmo explore melhor o domínio de projeto.

2.8 COMPARAÇÕES ESTATÍSTICAS E PROBABILÍSTICAS

Os métodos heurísticos dependem de um número aleatório e, conseqüentemente, toda vez que um algoritmo é executado, ele normalmente fornece um resultado diferente. Por isso, é necessário analisar alguns parâmetros para possibilitar a comparação entre os desempenhos de algoritmos diferentes (Lopez, 2022).

É necessário executar o algoritmo NIR (número de execuções independentes) vezes, gerando uma amostra de resultados ótimos e então é possível realizar comparações como: comparação estatística e comparação probabilística (Assis, 2021).

De acordo com Gomes et al. (2018) a comparação estatística da amostra obtida apresenta média, desvio padrão, máximos e mínimos encontrados. Para utilizar esse método, o NIR deve ser alto o suficiente para que se tenha mais confiança nos resultados e por isso é importante verificar o gráfico de convergência das estimativas.

Segundo os autores acima a comparação probabilística apresenta a probabilidade de um algoritmo ser melhor que outro, tendo como base um algoritmo de referência. Nesse caso também é necessário rodar o mesmo problema NIR vezes para que seja possível obter a amostra e comparar os resultados.

Para realizar uma comparação probabilística foi utilizado o CPMETRIC (Gomes et. al, 2018), que produz a probabilidade de que um determinado algoritmo encontrar um valor ótimo melhor do que um algoritmo alternativo, em uma única execução. O CPMETRIC necessita de um algoritmo referência, chamado de método 1, e os demais algoritmos utilizados na comparação são chamados de métodos 2, 3 e assim sucessivamente. Os valores por ele fornecidos estão descritos conforme as equações 21 a 23, e são dados como " P_{worst} ", " P_{better} " e " P_{eq} ", que representam

respectivamente a chance de um algoritmo ser pior, melhor ou equivalente ao método referência.

$$P_{better} = P[\{Y_P < Y_E\}] \cong \frac{1}{m_P - m_E} \sum_{k=1}^{m_E} (\sum_{j=1}^{m_P} I(y_P^j < y_E^k)) \quad (21)$$

$$P_{eq} = P[\{Y_P = Y_E\}] \cong \frac{1}{m_P - m_E} \sum_{k=1}^{m_E} (\sum_{j=1}^{m_P} I(y_P^j = y_E^k)) \quad (22)$$

$$P_{worst} = P[\{Y_P > Y_E\}] \cong 1 - P_{better} - P_{eq} \quad (23)$$

Onde:

y_P^j e y_E^k vetores que armazenam os dados das execuções de uma função;

Y_P e Y_E componentes dos vetores vistas como variáveis aleatórias identicamente distribuídas;

m_P e m_E número de execuções.

3 METODOLOGIA

Nesse capítulo é apresentado o método para validação do algoritmo, também é abordada a métrica que possibilita a comparação dos resultados obtidos. Por fim, são apresentadas as funções e as estruturas que são utilizadas para obtenção dos resultados.

3.1 VERIFICAÇÃO DO ALGORITMO

O BWO é um algoritmo recente, publicado no final de 2022. Conforme mencionado anteriormente ele faz uso da estratégia de enxame de partículas para encontrar a solução ótima.

O algoritmo depende de alguns parâmetros que podem ser alterados para verificar como ele se adapta melhor em relação a obtenção dos resultados. No modelo original foram utilizadas trinta rodadas independentes, mil iterações e cinquenta agentes de pesquisa.

O código do algoritmo BWO foi implementado em MATLAB, com os parâmetros e a respectiva função a ser chamada em cada rodada, os valores que são utilizados como referência para o comparativo (mínimo, máximo, média e desvio padrão) foram inseridos posteriormente.

Também foram adicionadas as funções e suas respectivas restrições, bem como seus limites superiores e inferiores. Nas funções com restrição e problemas de engenharia foi utilizado o método de penalidade aditiva. A penalização aditiva em algoritmos heurísticos é utilizada para executar restrições adicionais na função objetivo de um problema de otimização. Uma função de penalização é incorporada através da adição de termos à função objetivo, os quais penalizam as soluções que não atendam às restrições (Carvalho, 2014). O termo de penalização penaliza os pontos no domínio que não respeitam as restrições e transforma um problema com restrições em um problema sem restrições.

O método da pena de morte é uma abordagem que tem proporcionado os melhores resultados tanto em robustez quanto em eficiência computacional. Um número muito alto é utilizado para penalizar os pontos no domínio que não compreende as restrições, evitando que os mesmos estejam presentes no domínio de projeto. As equações 21 e 22 apresentam a formulação do método da penalização aditiva e o valor de p_1 utilizados no código de MATLAB.

Encontrar o melhor valor de projeto pertencente a um conjunto real:

$$X^* \in \mathbb{R}^n$$

Que minimiza a função através da penalização aditiva:

$$J(X) = f(X) + P(X) \quad (21)$$

Sujeito a restrição: $X_l \leq X \leq X_u$

Onde P penaliza os indivíduos que estão no domínio inviável através da equação:

$$P(X) = p_1 \sum_{i=1}^m \max(0, g_i(X)) \quad (22)$$

$$\text{com } p_1 = 10^8$$

Para realizar a validação, os parâmetros alterados são, números de agentes de pesquisa, iterações máximas e execuções independentes (NIR) de cada função.

São estudadas funções conhecidas da literatura para análise e comparação com resultados existentes, validando a funcionalidade do algoritmo em relação as mesmas. Após a validação com as funções **benchmark** são implementados os problemas clássicos de engenharia, estruturas treliçadas e uma torre de transmissão.

Neste trabalho foram realizados estudos para a validação inicial do algoritmo BWO, com funções restritas e irrestritas e problemas clássicos de engenharia, estes foram comparados com os resultados existentes de literaturas.

Para validação do algoritmo BWO em relação as estruturas treliçadas, foram simuladas duas treliças planas, uma espacial e uma torre de transmissão, os problemas também foram implementados e executados nos algoritmos WOA e FA para gerar comparativos probabilísticos.

3.2 PROBLEMAS DE OTIMIZAÇÃO GERAIS

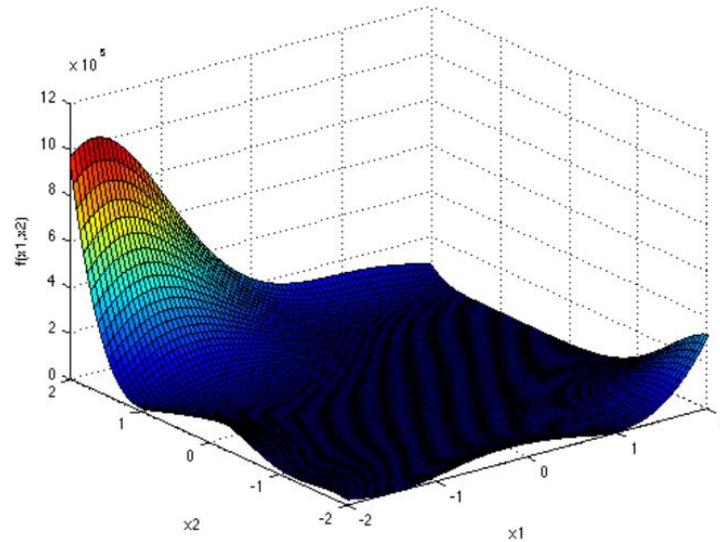
Nesse capítulo são apresentadas algumas funções amplamente conhecidas e que são utilizadas para validar o algoritmo BWO.

3.2.1 Função Goldstein-Price

A função Goldstein-Price possui duas dimensões, vários mínimos locais e é geralmente avaliada no quadrado $x_i \in [-2, 2]$ para todo $i = 1, 2$ conforme a equação 23. A Figura 9 apresenta o gráfico da função apresentada abaixo.

$$f(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (23)$$

Figura 9- Função Goldstein Price



Fonte: Surjanovic, Bingham, 2018

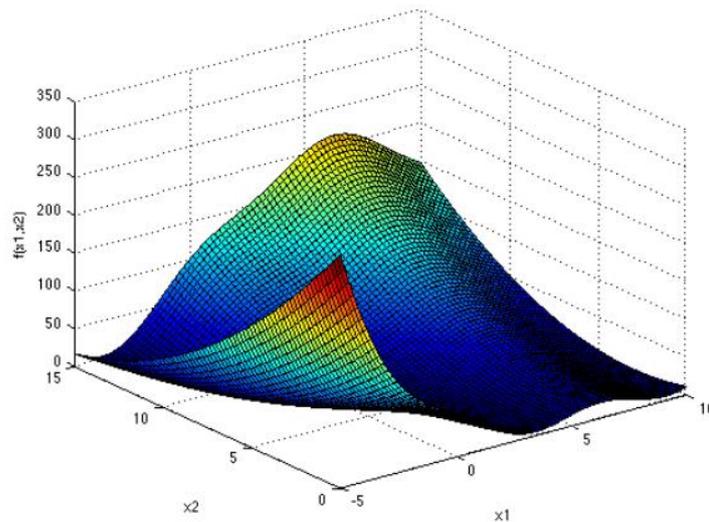
O mínimo global dessa função é no ponto $X^* = (0, -1)$, com função objetivo ótima de $f(X^*) = 3$.

3.2.2 Função Branin

A função Branin possui duas dimensões, com três mínimos globais e geralmente é avaliada no quadrado $x_1 \in [-5, 10]$ e $x_2 \in [0, 15]$. A sua função é apresentada abaixo, conforme a equação 24, com os valores recomendados de a, b, c, e, s e t. A Figura 10 apresenta o gráfico da função.

$$f(X) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t) \cos(x_1) + s \quad (24)$$

Figura 10 – Função Branin



Fonte: Surjanovic, Bingham, 2018

Sendo: $a = 1$; $b = 5,1/(4\pi^2)$; $c = 5/\pi$; $r = 6$; $s = 10$ e $t = 1/(8\pi)$.

Os mínimos globais apresentam o mesmo valor para a função objetivo de $f(X^*) = 0,397887$ com $X^* = (-\pi, 12,275)$, $(\pi, 2,275)$ e $(9,42478, 2,475)$.

3.2.3 Função Shekel

A função Shekel, conforme equações de 26 a 28, possui m mínimos locais, quatro dimensões e geralmente é avaliada no hipercubo $x_i \in [0, 10]$ para todo $i = 1, 2, 3, 4$.

Apresenta-se abaixo a sua função na equação 25, bem como os valores recomendados de m , o vetor β e a matriz C . Sendo β um vetor m -dimensional e C uma matriz 4 por m -dimensional. A Figura 11 demonstra o gráfico em sua forma bidimensional da função.

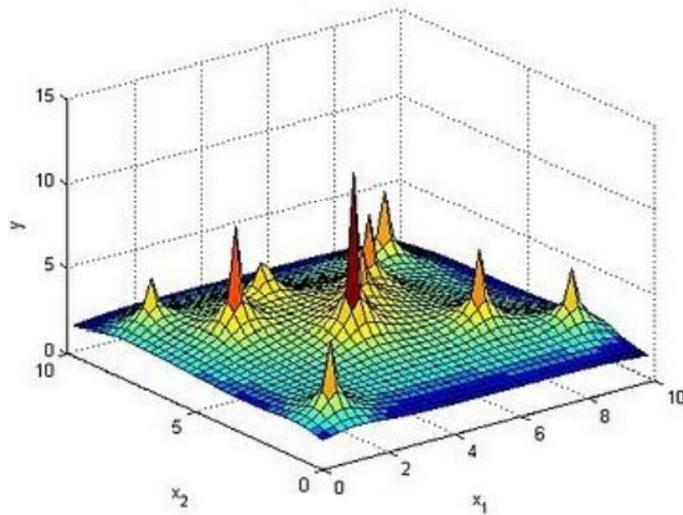
$$f(X) = - \sum_{i=1}^m \left(\sum_{j=1}^4 (x_j - C_{ji})^2 + \beta_i \right)^{-1}, \text{ onde} \quad (25)$$

$$m = 10$$

$$\beta = \frac{1}{10} (1,2,2,4,4,6,3,7,5,5)^T$$

$$C = \begin{pmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{pmatrix}$$

Figura 11– Função Shekel em sua forma bidimensional



Fonte: Surjanovic, Bingham, 2018

Para este problema, os mínimos globais são:

$$\text{para } m = 5: f(X^*) = -10.1532, \text{ com } X^* = (4, 4, 4, 4) \quad (26)$$

$$\text{para } m = 7: f(X^*) = -10.4029, \text{ com } X^* = (4, 4, 4, 4) \quad (27)$$

$$\text{para } m = 10: f(X^*) = -10.5364, \text{ com } X^* = (4, 4, 4, 4) \quad (28)$$

3.2.4 Problema com restrição por *Eggholder* Modificada

Este problema de *Eggholder* foi modificado e proposto por Assis (2021), possui duas dimensões e tem as duas variáveis restritas a valores contínuos no intervalo entre 0 e 1. O autor utilizou uma função de restrição para adaptar da função de *Eggholder*. As equações 29 e 30 apresentam a função objetivo e a função de restrição, respectivamente:

$$f(X) = 2 - 4x_1 - 4x_2 + 4x_1^2 + 4x_2^2 \quad (29)$$

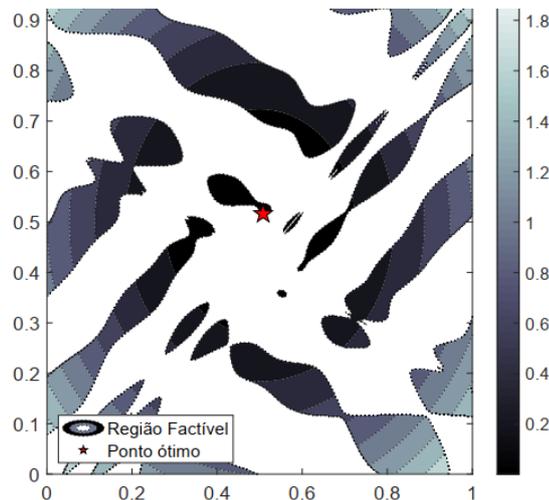
$$g_1(X) = 0.15 + \frac{-(x'_2+47)\sin\left(\sqrt{|x'_2+\frac{x'_1}{2}+47|}\right) - x'_1\sin\left(\sqrt{|x'_1-(x'_2+47)|}\right)}{400} \quad (30)$$

Sendo:

$$\begin{aligned} x'_1 &= (-0.5 + x_1) \cdot 600; \\ x'_2 &= (-0.5 + x_2) \cdot 600. \end{aligned}$$

Esse problema possui diversos mínimos locais que são encontrados em várias regiões do domínio e isso mostra que a função de restrição é complexa, irregular e descontínua. A Figura 12 apresenta as regiões factíveis do problema, bem como seu ponto ótimo.

Figura 12– Problema com restrição *Eggholder* modificada



Fonte: Assis, 2021

Para este problema, o mínimo global é em $X^* = (0,5081, 0,5163)$, com a função objetivo avaliada em $f(X^*) = 0,001333$.

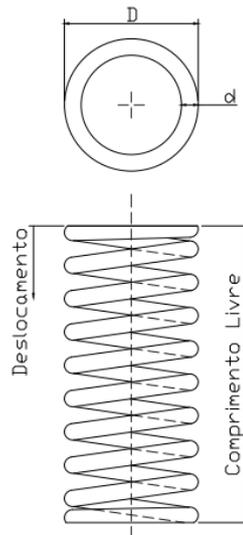
3.2.5 Problema de engenharia: Mola sob tração/compressão

O problema da mola proposto neste item é um problema clássico de engenharia que consiste na minimização do volume de uma mola sujeita a tração/compressão, dado pela função objetivo apresentada através da equação 31, possuindo três dimensões.

$$V = (x_3 - 2)x_2x_1^2 \quad (31)$$

Segundo Assis (2021), este problema foi proposto por Mezura-Montes, Coello e Landa-Becerra (2003) e possui como variáveis de projeto o diâmetro do fio ($d = x_1 \in [0,05; 2]$), diâmetro de cada volta ($D = x_2 \in [0,25; 1,3]$) e o número de espirais ativos ($N = x_3 \in [2; 15]$). A Figura 13 ilustra a representação do problema proposto.

Figura 13– Mola sob tração/compressão



Fonte: Assis, 2021

Neste problema, mola está sujeita a quatro restrições mecânicas de deflexão mínima, tensão de cisalhamento, frequência de oscilação e limite no diâmetro externo, dadas respectivamente pelas equações 32 a 35.

$$g_1(X) = 1 - x_2^3 x_3 \leq 0 \quad (32)$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (33)$$

$$g_3(X) = 1 - \frac{140,45x_1}{x_2^2 x_3} \leq 0 \quad (34)$$

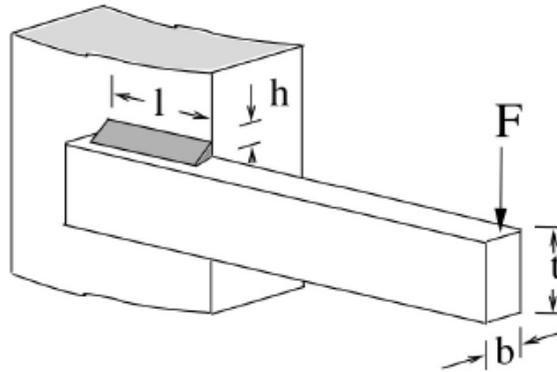
$$g_4(X) = \frac{x_2 + x_1}{1,5} - 1 \leq 0 \quad (35)$$

3.2.6 Problema de engenharia: viga soldada

Ainda conforme Assis (2021), este problema de engenharia trata-se de minimizar o custo de uma viga soldada dada por Mezura-Montes, Coello e Landa-Becerra (2003). Nesse exemplo, as variáveis de projeto são dadas por $h = x_1 \in [0,$

125; 2, 0], $l = x_2 \in [0, 1; 10, 0]$, $t = x_3 \in [0, 1; 10, 0]$ e $b = x_4 \in [0, 1; 2, 0]$. A Figura 14 ilustra o problema.

Figura 14– Viga soldada



Fonte: Carvalho, 2014

A função custo é dada pela equação 36.

$$C(X) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14 + x_2) \quad (36)$$

Sujeita as restrições de tensão de cisalhamento, tensão de flexão, carga de flambagem, deflexão final e travamento lateral da viga, dadas respectivamente pelas equações 37 a 42:

$$g_1(X) = \tau_x - \tau_{max} \leq 0 \quad (37)$$

$$g_2(X) = \sigma_x - \sigma_{max} \leq 0 \quad (38)$$

$$g_3(X) = x_1 - x_4 \leq 0 \quad (39)$$

$$g_4(X) = 0,10471x_1^2 + 0,04811x_3x_4(14 + x_2) - 5 \leq 0 \quad (40)$$

$$g_5(X) = \delta_x - \delta_{max} \leq 0 \quad (41)$$

$$g_6(X) = P - P_c(X) \leq 0 \quad (42)$$

Onde:

$$\tau_x = ((\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2)^{1/2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (x_1x_3/2)^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\}$$

$$\tau_x = \frac{6PL}{x_4x_3^2}$$

$$\delta_x = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(X) = \frac{4,013E \sqrt{x_3^2x_4^6/36}}{L^2} \left(1 - \frac{x_2}{2L} \sqrt{\frac{E}{4G}} \right)$$

Sendo:

$$P = 2721,554 \text{ kg};$$

$$L = 0,3556 \text{ m};$$

$$E = 2,068 \times 10^{11} \text{ Pa};$$

$$G = 8,274 \times 10^{10} \text{ Pa};$$

$$\tau_{max} = 9,377 \times 10^7 \text{ Pa};$$

$$\sigma_{max} = 2,068 \times 10^8 \text{ Pa};$$

$$\delta_{max} = 0,00635 \text{ m}.$$

3.3 TRELIÇAS PROPOSTAS

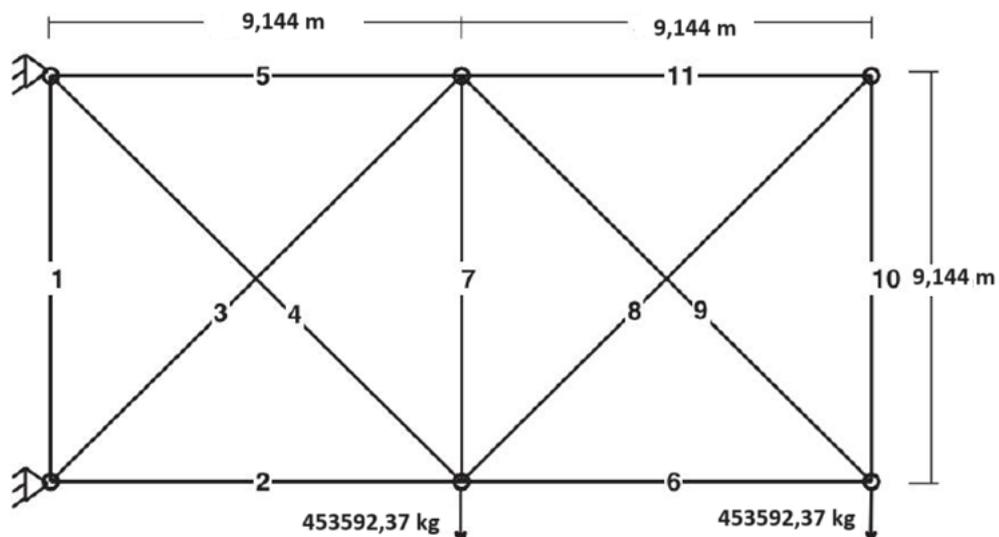
Nesse item são apresentadas as treliças estudadas. As estruturas são otimizadas em relação a dimensão, forma e topologia. Os exemplos utilizados para esse capítulo foram retirados do artigo de Lopez e Miguel (2013).

Os autores comentam que a otimização estrutural para treliças pode ser classificada em três vertentes: otimização dimensional, topológica e de forma. A otimização dimensional procura as dimensões da seção transversal dos elementos de modo a minimizar o volume estrutural. Já a otimização da forma modifica as coordenadas nodais e por fim, a otimização da topologia que é empregada para eliminar os membros desnecessários da estrutura. Esse procedimento de otimização busca o peso estrutural mínimo da treliça submetida a restrições de tensão, deslocamento e estabilidade cinemática.

3.3.1 Treliça plana com onze barras

Esta treliça é utilizada como um problema de referência e tem sido aplicada em muitos dos estudos nesta área. A estrutura deste exemplo é mostrada na Figura 15 e os parâmetros de projeto são dado na Tabela 1. As áreas de seções transversais variam dentro do intervalo de 0,0 a 0,0193548 m² em incrementos de 0,00064516 m².

Figura 15- Exemplo de *benchmark* de treliça de onze barras



Fonte: Adaptado de Miguel e Lopez, 2013

Tabela 1 - Parâmetros para o projeto de treliça de onze barras

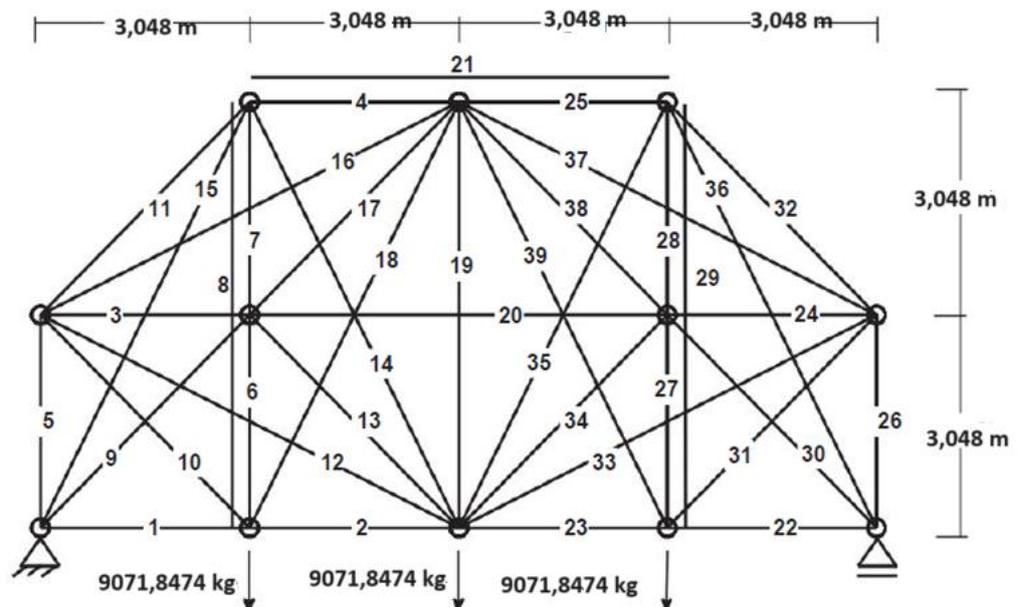
Parâmetro de projeto	Valor
Módulo de elasticidade	$68,95 \cdot 10^9 \text{ Pa}$
Densidade	$2767,99 \text{ kg/m}^3$
Tensão admissível	$132,37 \cdot 10^6 \text{ Pa}$
Tensão admissível na compressão	$132,37 \cdot 10^6 \text{ Pa}$
Deslocamento	$0,0508 \text{ m}$

Fonte: Adaptado de Miguel e Lopez, 2013

3.3.2 Treliça plana de trinta e nove barras

O segundo exemplo de referência é a estrutura de 39 barras, 12 nós, simplesmente apoiadas e de dois níveis, de vão único, mostrada na Figura 16. A resistência admissível é de $1,379 \times 10^8 \text{ Pa}$ e as propriedades do material, bem como, deslocamento são os mesmos utilizados na treliça de onze barras. As áreas de seção transversal variam entre $0,000145 \text{ m}^2$ e $0,00145 \text{ m}^2$ e uma área crítica de $e = 3,2258 \times 10^5 \text{ m}^2$ é considerada.

Figura 16- Exemplo de **benchmark** de treliça de trinta e nove barras



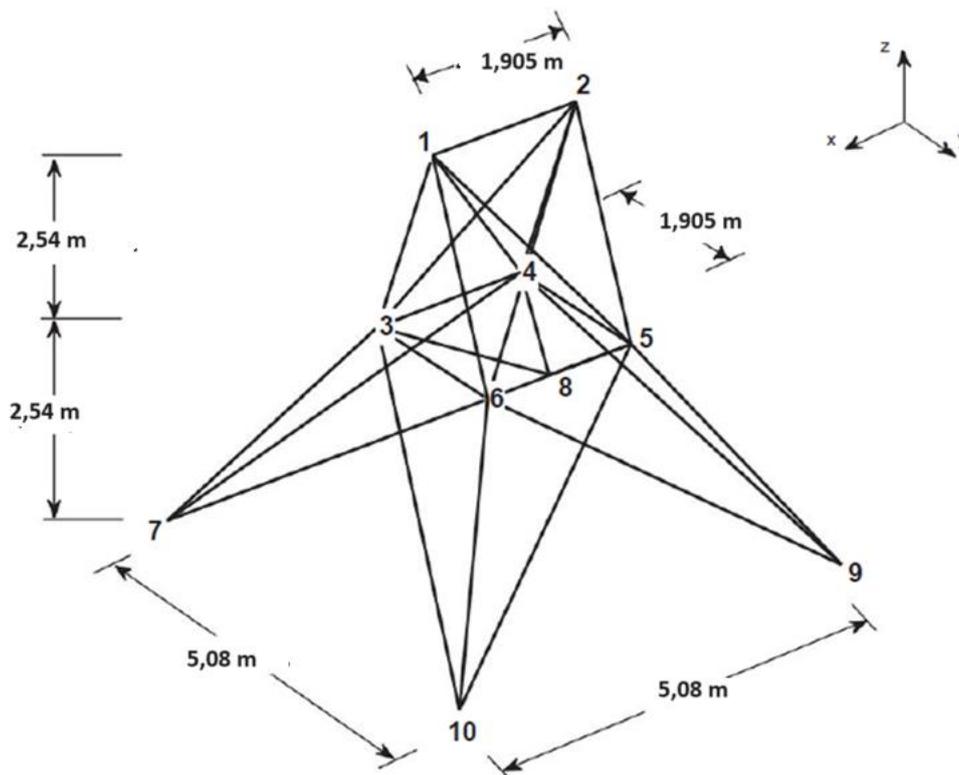
Fonte: Adaptado de Miguel e Lopez, 2013

3.3.3 Treliça espacial vinte e cinco barras

Essa treliça 3D também é frequentemente usada como um problema de referência. A estrutura é apresentada na Figura 17 e os detalhes do carregamento e agrupamentos de membros são dados nas Tabelas 2 e 3.

A tensão admissível é de $2,758 \times 10^8$ Pa em tração e compressão, o módulo de elasticidade e densidade são os mesmos que no problema da treliça de 39 barras e a deflexão máxima permitida é de 0,00889 m em qualquer direção para cada nó. As áreas da seção transversal são escolhidas de um conjunto $D = (0,00254, 0,00508, 0,00762, 0,01016, 0,0127, 0,01524, 0,01778, 0,02032, 0,02286, 0,0254, 0,02794, 0,03048, 0,03302, 0,03556, 0,0381, 0,04064, 0,04318, 0,04572, 0,04826, 0,0508, 0,05334, 0,05588, 0,05842, 0,06096, 0,0635, 0,06604, 0,07112, 0,0762, 0,08128, 0,08636)$ m.

Figura 17 - Exemplo de referência de vinte e cinco barras (treliça 3D)



Fonte: Adaptado de Miguel e Lopez, 2013

Tabela 2– Carregamentos

Nó	P _x (kg)	P _y (kg)	P _z (kg)
1	45,3592	-4,53592	-4,53592
2	0	-4,53592	-4,53592
3	226,796	0	0
4	272,155	0	0

Fonte: Adaptado de Miguel e Lopez, 2013

Tabela 3 - Coordenadas dos nós e agrupamento de membros

Nó	x (m)	y (m)	z (m)	Grupos	Membro
1	-0,9525	0	5,08	A1	1(1,2)
2	-0,9525	0	5,08	A2	2(1, 4), 3(2, 3), 4(1, 5), 5(2,6)
3	-0,9525	-0,9525	2,54	A3	6(2, 5), 7(2, 4), 8(1, 3), 9(1,6)
4	-0,9525	-0,9525	2,54	A4	10(3,6), 11(4,5)
5	-0,9525	-0,9525	2,54	A5	12(3, 4), 13(5,6)
6	-0,9525	-0,9525	2,54	A6	14(3, 10), 15(6,7), 16(4,9), 17(5, 8)
7	-2,54	2,54	0	A7	18(3, 8), 19(4,7), 20(6,9), 21(5,10)
8	-2,54	2,54	0	A8	22(3, 7), 23(4,8), 24(5,9), 25(6,10)
9	-2,54	-2,54	0		
10	-2,54	-2,54	0		

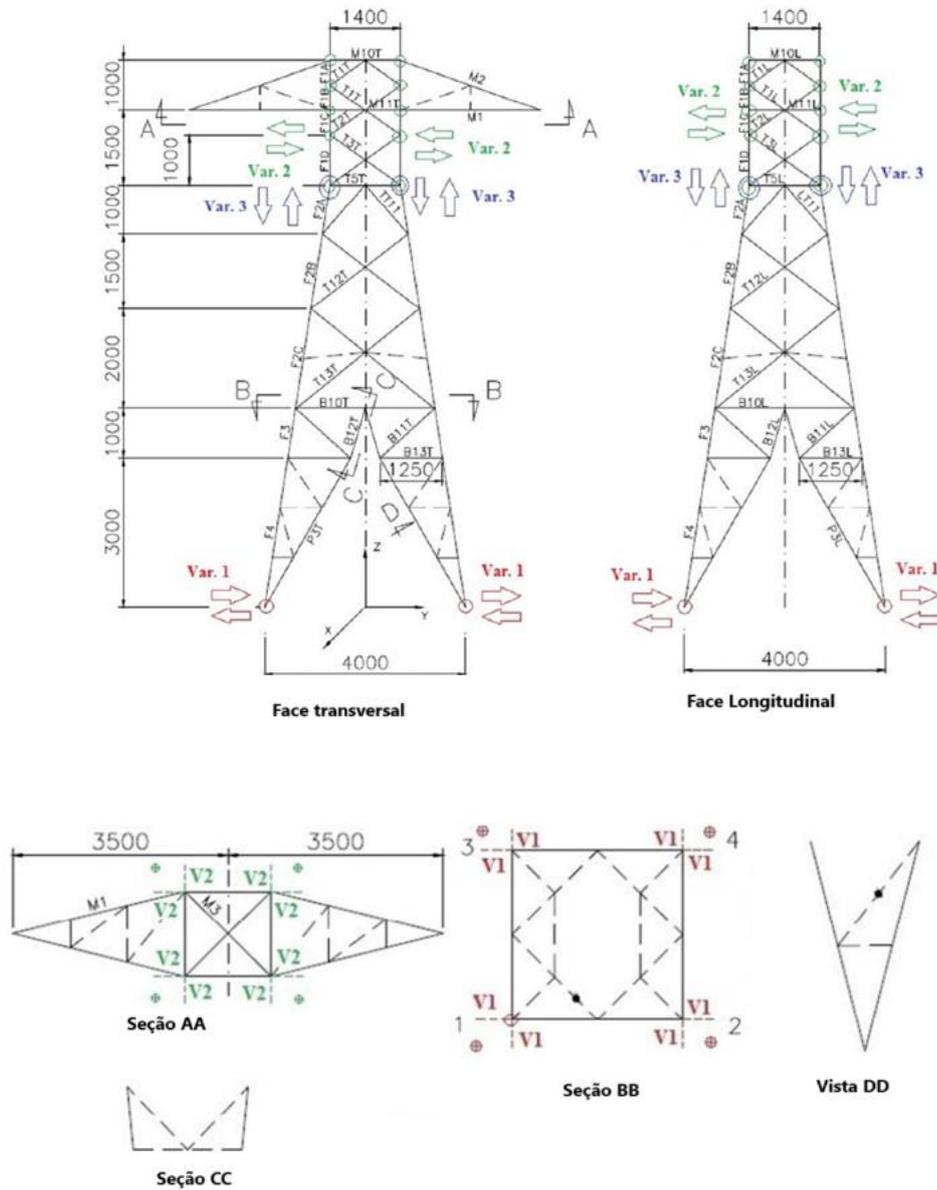
Fonte: Adaptado de Miguel e Lopez, 2013

3.4 TORRE DE CIGRÉ

A torre de Cigré é uma estrutura térrea treliçada que foi publicada no folheto Cigré 387 em 2009. Este trabalho estuda a torre número 1 do folheto. A Figura 18 apresenta o perfil da torre estudada, as barras e membros redundantes, bem como os cortes e as seções das estruturas.

A torre possui altura total de 10 metros e largura da base igual a 4 metros, 166 membros e 76 membros redundantes. As linhas tracejadas representam as barras redundantes e as dimensões estão em milímetros. Foram adotadas restrições de tensão e esbeltez de acordo com a norma ASCE 10/97 (Souza et. al. 2014).

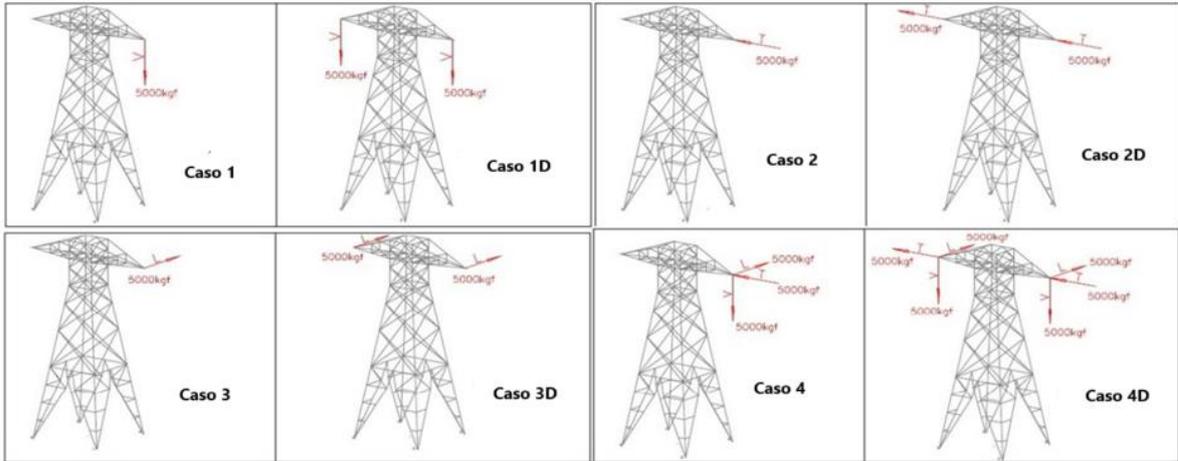
Figura 18- Torre de Cigré adotada para o trabalho



Fonte: Adaptado de Cigré, 2009

A torre foi projetada considerando oito hipóteses de carregamento, que estão apresentadas na Figura 19.

Figura 19- Casos de Carga



Fonte: Adaptado de Cigré (2009)

4 RESULTADOS E DISCUSSÃO

Os modelos de funções, problemas e estruturas apresentados anteriormente são utilizados para analisar e validar o algoritmo BWO em relação aos valores estatísticos encontrados.

4.1 FUNÇÕES *BENCHMARK*

O Quadro 1 apresenta as funções e os valores ótimos encontrados em relação a implementação no algoritmo BWO, utilizando como métrica a obtenção da média de soluções encontradas e desvio padrão a cada iteração. Em uma primeira análise fazendo uso de dez rodadas independentes, cinquenta agentes de pesquisa e número máximo de iterações igual a mil.

Quadro 1 –Valores ótimos das funções com NIR =10

Funções	Valores			
	BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)
Goldstein	3,0684	8,00E-11	3,0684	3,0684
Branin	0,4012	0,0029	0,4005	0,4368
Shekel	-10,2415	1,2562	-10,5364	-1,4484
Eggholder modificada	0,0027	0,0065	0,0013	0,049
Mola	0,0168	0,0018	0,0151	0,0205
Viga soldada	1,2558	1,33E-11	1,2558	1,2558

Fonte: A autora

Já o Quadro 2 apresenta as funções e os valores ótimos encontrados em relação a implementação, porém alterando o parâmetro de NIR, nessa análise o valor utilizado foi igual a cem.

Quadro 2- Valores ótimos das funções mudando NIR =100

Funções	Valores			
	BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)
Goldstein	3,4161	3,14700E-01	3,3848	8,8899
Branin	0,4244	0,3147	0,4011	2,2579
Shekel	-10,2811	1,0444	-10,5364	-2,2661
Eggholder modificada	0,0017	0,0017	0,0013	0,0152
Mola	0,0177	0,0029	0,016	0,04
Viga soldada	1,3229	1,11E-11	1,3229	1,3229

Fonte: A autora

As funções também foram rodadas no algoritmo com número de agentes de pesquisa igual a trinta, número máximo de iterações igual a quinhentos e NIR igual a dez, conforme o Quadro 3, para que o comparativo fosse realizado de modo mais realista, uma vez que, esses são valores adotados pelos testes no algoritmo WOA.

Quadro 3- Valores ótimos das funções com parâmetros WOA

Funções	Valores			
	BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)
Goldstein	3,3837	2,1688	3,1675	36,6953
Branin	0,4039	0,0198	0,4021	0,6236
Shekel	-10,128	1,4481	-10,534	-1,2028
Eggholder modificada	0,0045	0,0057	0,0016	0,0199
Mola	0,0297	0,0291	0,0225	0,2358
Viga soldada	2,6598	0,166	2,5238	3,0373

Fonte: A autora

Percebe-se que para as funções Goldstein e Branin, a implementação com NIR igual a dez chegou mais próximo aos valores ótimos presentes na literatura. Para as funções de Shekel e Eggholder modificada, a implementação com NIR igual a cem se mostrou mais promissora. Já para o problema da mola, os valores obtidos nos Quadros 2 e 3 são muito semelhantes. Por último, para a viga soldada a melhor solução encontra-se fazendo uso dos parâmetros do Quadro 3.

Os resultados apresentados no Quadro 4 como valores ótimos para as simulações das funções **benchmark** também são comparados com os valores apresentados no artigo WOA de Miraji e Lewis (2016), Karaboga e Akay (2009) para o PSO, Souza (2021) para o FA e com trabalho de Assis (2021).

Quadro 4 - Comparativo entre ótimos obtidos no algoritmo BWO e valores existentes da literatura para as funções **benchmark**

Algoritmo	Funções			
	Goldstein	Branin	Shekel (com m=10)	Eggholder modificada
WOA	3	0,3979	-9,34238	-
PSO	3	0,3979	-2,0870079	-
FA	3	0,3979	-	-
Assis (2021)	-	-	-	0,001333
BWO (este trabalho)	3,0684	0,4012	-10,2811	0,0017

Fonte: A autora

Para os problemas clássicos de engenharia, foram utilizados os comparativos com os trabalhos realizados por Miraji e Lewis (2016) no algoritmo WOA, He et. al (2004) para o algoritmo PSO, Gandomi et. al. (2011) para o FA, e Assis (2021), apresentados no Quadro 5.

Quadro 5 - Comparativo entre ótimos obtidos no algoritmo BWO e valores existentes da literatura para os problemas da mola e viga soldada

Algoritmo	Mola	Viga soldada
WOA	0,0127	1,732
PSO	0,01266528	2,381
FA	-	1,7312
Assis (2021)	0,0126	2,6296
BWO (este trabalho)	0,0177	2,6598

Fonte: A autora

Pode-se notar através do comparativo que a maior diferença entre resultados obtidos foi para a função Shekel, que é uma função mais complexa, ocasionando algumas vezes aprisionamento em algum mínimo local. Desse modo, os valores encontram-se em uma margem pequena de desvio e são aceitáveis para validação do algoritmo BWO.

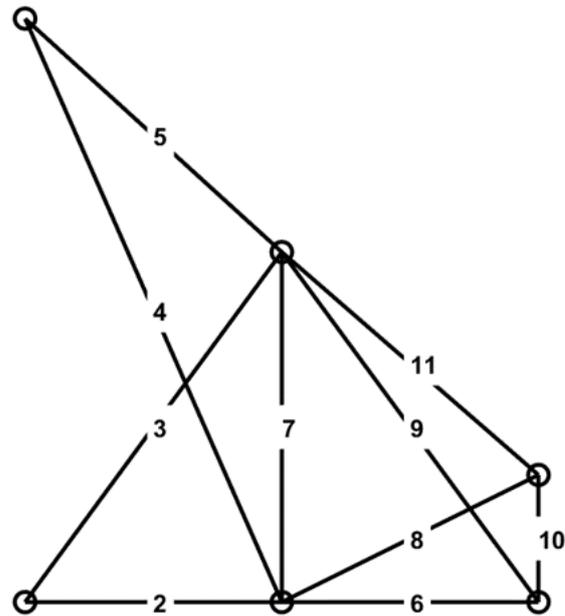
4.2 TRELIÇAS

4.2.1 Treliza plana de onze barras

Neste trabalho para a treliza de 11 barras o problema foi implementado no algoritmo BWO, WOA e FA a fim de verificar sua eficácia e gerar comparativos. Foram realizadas variações no número de iterações máximas, agentes de pesquisa e em alguns casos do NIR. Os Quadros 6, 7 e 8 apresentam os resultados estatísticos obtidos para os testes com os valores de massa e tempo de execução do algoritmo em cada simulação.

A topologia adotada como ótima para o problema é representada na figura 20.

Figura 20- Otimização para a treliça de 11 barras



Fonte: A autora

O Quadro 6 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para cada simulação do algoritmo BWO na treliça de onze barras. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 6- Resultados estatísticos algoritmo BWO - treliça 11 barras

Dados de entrada		Função	Valores (kg)				tempo(s)
Pop=100	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 11 barras	1269	21,1815	1233	1300,5
Dados de entrada		Função	Valores (kg)				tempo(s)
Pop=100	NIR=50		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 11 barras	1264,95	21,114	1260,603	1308,15
Dados de entrada		Função	Valores (kg)				tempo(s)
Pop=80	NIR=100		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=500			Treliça 11 barras	1465,2	65,25	1291,05	1708,02
Dados de entrada		Função	Valores (kg)				tempo(s)
Pop=30	NIR=100		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 11 barras	2042,775	165,15	1552,5	2338,785
Dados de entrada		Função	Valores (kg)				tempo(s)
Pop=100	NIR=100		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 11 barras	1265,85	20,979	1255,788	1330,65

Fonte: A autora

O melhor valor obtido para o BWO é de 1233 kg, que utiliza 100 agentes de pesquisa, 5000 iterações e NIR igual a 10.

O Quadro 7 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo WOA na treliça de onze barras. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 7 - Resultados estatísticos algoritmo WOA - treliça 11 barras

Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=50			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=5000		Treliça 11 barras	1904,4	3,8115	1316,25	2594,25	1513
Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=30			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000		Treliça 11 barras	1636,065	120,15	1409,265	1756,485	9761
Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=50			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000		Treliça 11 barras	1473,93	1,2465	1273,005	1701,495	20255
Dados de entrada	NIR=100	Função	Valores (kg)				tempo(s)
Pop=30			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=5000		Treliça 11 barras	2182,95	444,15	1326,42	3615,39	11721
Dados de entrada	NIR=100	Função	Valores (kg)				tempo(s)
Pop=50			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000		Treliça 11 barras	2078,46	429,3	1419,525	3629,925	16486

Fonte: A autora

O ótimo encontrado para o algoritmo WOA é de 1273,005 kg, utilizando população igual a 50 e número de iterações 50000 e NIR igual a 10.

O Quadro 8 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo FA na treliça de onze barras. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 8 - Resultados estatísticos algoritmo FA - treliça 11 barras

Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=10			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=6000		Treliça 11 barras	1281,69	29,655	1218,6	1321,335	87
Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=10			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=10000		Treliça 11 barras	1296,45	24,7815	1263,96	1342,8	154
Dados de entrada	NIR=10	Função	Valores (kg)				tempo(s)
Pop=20			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=10000		Treliça 11 barras	1290,6	15,624	1264,93	1315,35	308

Fonte: A autora

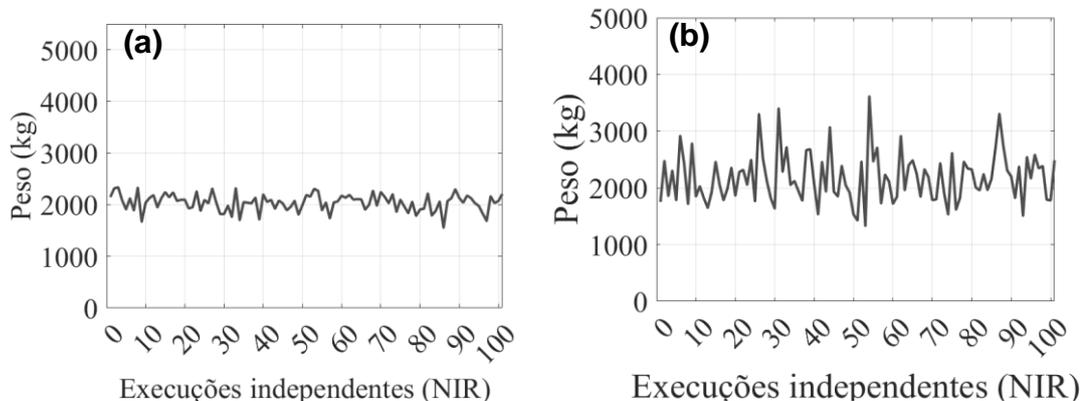
Para o algoritmo FA o melhor valor obtido de massa é de 1218,6 kg, utilizando 10 agentes de pesquisa e 6000 iterações e NIR igual a 10.

O algoritmo FA foi o que encontrou o menor valor de massa e em menor tempo. Os resultados encontrados para o BWO são próximos ao da referência utilizando o FA, se tratando dos menores resultados de média obtidos nos mesmos, a diferença entre eles é de 14,4 kg. Quando comparado o algoritmo FA ao WOA tem-se uma diferença de 54,405 kg. Também pode-se notar que o custo computacional do algoritmo BWO para obter o valor ótimo nesse caso foi menor que o do WOA.

Os gráficos 1 **(a)** e **(b)** mostram a evolução da estrutura treliça plana de 11 barras para cada algoritmo da baleia em 150.000 avaliações da função objetivo.

Gráfico 1 - Evolução para treliça de 11 barras nos algoritmos

(a) BWO (b) WOA



Fonte: A autora

Analisando os gráficos 1 **(a)** e **(b)** para a evolução dos algoritmos WOA e BWO ambos considerando mesma população e iterações máximas, é possível perceber que o algoritmo WOA possui maior desvio padrão, maiores valores de máximos e média quando comparado ao BWO que possui uma menor dispersão de valores de massa para essa estrutura.

O problema também foi implementado no CPMETRIC que é um algoritmo que produz uma comparação entre algoritmos utilizando um dos métodos como referência, mostrando a probabilidade de um ser melhor, equivalente ou pior que o outro.

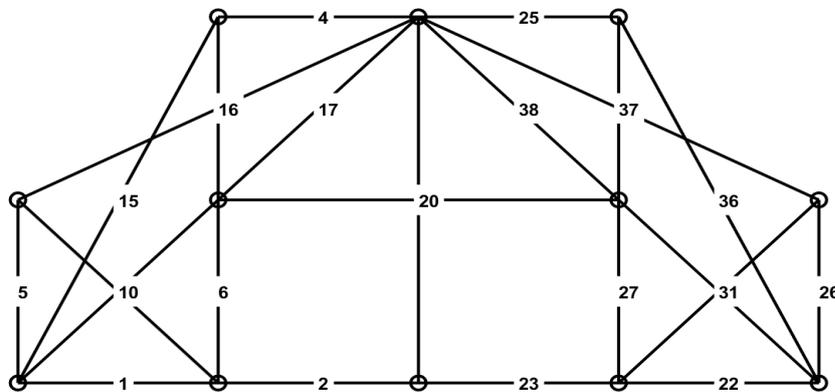
Os resultados das simulações com 30 agentes de pesquisa e 5000 iterações, apresentados nos quadros anteriores para o BWO e WOA foram adicionados ao CPMETRIC, para gerar os comparativos probabilísticos entre os algoritmos utilizados. O método 1 (também chamado de referência) é o WOA, e apresentou uma probabilidade de aproximadamente 48% em ser melhor que o método 2 (utilizando o algoritmo BWO).

4.2.2 Treliça plana de trinta e nove barras

Para a treliça de 39 barras o problema foi implementado no algoritmo BWO, WOA e FA a fim de verificar sua eficácia e gerar comparativos, foram realizadas variações no número de iterações máximas, agentes de pesquisa e em alguns casos do NIR. Os quadros 9, 10 e 11 apresentam os resultados estatísticos obtidos para os testes com os valores de massa e tempo de execução do algoritmo em cada simulação.

A topologia adotada como ótima para o problema é representada na figura 21.

Figura 21- Otimização para a treliça de 39 barras



Fonte: A autora

O Quadro 9 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para cada simulação do algoritmo BWO na treliça plana de trinta e nove barras. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 9- Resultados estatísticos algoritmo BWO - treliça 39 barras

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 39 barras	113,8365	4,194	109,035	120,285
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=80	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 39 barras	112,086	3,735	103,32	118,35
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 39 barras	128,1195	6,219	116,748	138,3165
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 39 barras	114,1335	4,725	107,8515	120,3255
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=100	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 39 barras	110,754	5,3415	101,7675	117,981
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=100	NIR=20		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 39 barras	112,68	5,535	99,657	119,7
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=50		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 39 barras	124,2	5,7195	109,935	135

Fonte: A autora

O melhor valor obtido para o BWO é de 99,657 kg, que utiliza 100 agentes de pesquisa, 50000 iterações e NIR igual a 20.

O Quadro 10 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo WOA na treliça de trinta e nove barras. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 10 - Resultados estatísticos algoritmos WOA - treliça 39 barras

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=5000			Treliça 39 barras	163,1655	26,82	118,4265	194,517
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=5000			Treliça 39 barras	149,751	16,3665	129,9645	186,516
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000			Treliça 39 barras	132,3405	14,9085	95,3415	145,7775
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=80	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000			Treliça 39 barras	134,2395	19,341	96,993	148,4415
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=20		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=50000			Treliça 39 barras	131,6475	21,267	102,645	186,4665
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=50		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=5000			Treliça 39 barras	164,3535	28,6785	93,744	228,051

Fonte: A autora

Para o algoritmo WOA o melhor valor obtido de massa é de 93,744 kg, utilizando 30 agentes de pesquisa, 5000 iterações e NIR igual a 50.

O Quadro 11 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo FA na treliça plana de trinta e nove barras. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 11 - Resultados estatísticos algoritmos FA - treliça 39 barras

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=10	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=500			Treliça de 39 barras	106,2045	9,9585	89,3565	124,227
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=10	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=5000			Treliça de 39 barras	91,1745	4,131	87,7905	100,7955
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=10	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=50000			Treliça de 39 barras	95,0535	6,5115	88,281	104,535

Fonte: A autora

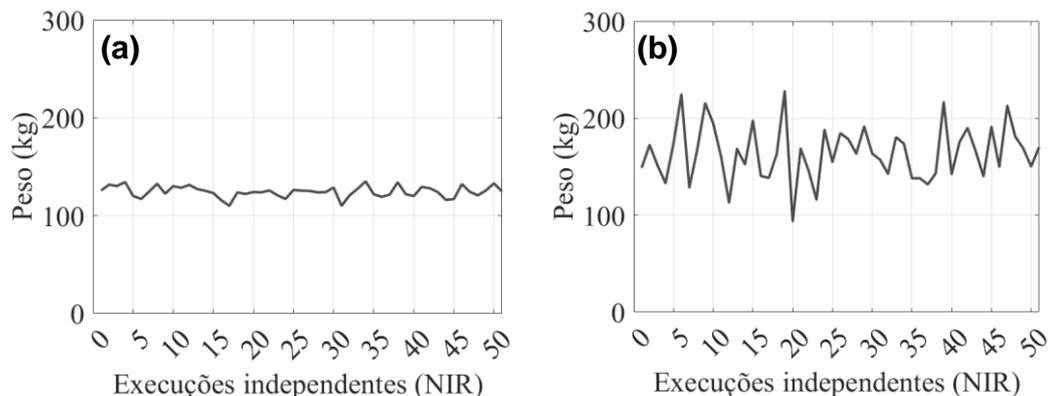
Para o algoritmo FA o melhor valor obtido de massa é de 87,79 kg, utilizando 10 agentes de pesquisa e 5000 iterações e NIR igual a 10. O algoritmo FA foi o que encontrou o menor valor de massa e em menor tempo.

Realizando uma comparação entre o algoritmo FA e BWO para os valores mínimos obtidos percebe-se que a diferença entre eles é de 11,87 kg enquanto entre o FA e WOA é de 5,95 kg. Nessa simulação o WOA atingiu melhor valor mínimo de projeto apresentando também menor custo computacional, porém quando analisados os demais dados estatísticos o BWO indica melhores resultados.

Os gráficos 2 **a** e **b** mostram a evolução da estrutura para a estrutura da treliça plana de 39 barras em relação a cada algoritmo da baleia em 150.000 avaliações da função objetivo.

Gráfico 2 - Evolução para treliça de 39 barras nos algoritmos

(a) BWO (b) WOA



Fonte: A autora

Nesse caso todos os algoritmos tiveram um padrão de evolução quando analisados os valores de média e desvio padrão. É perceptível que quanto maior forem os números de agentes de pesquisa, as iterações máximas e o NIR, maior é o tempo para processamento da estrutura. Quando considerados os mesmos dados de entrada para os algoritmos BWO e WOA, o BWO proporciona um tempo menor de resolução, com melhores valores de média, desvio e máximo.

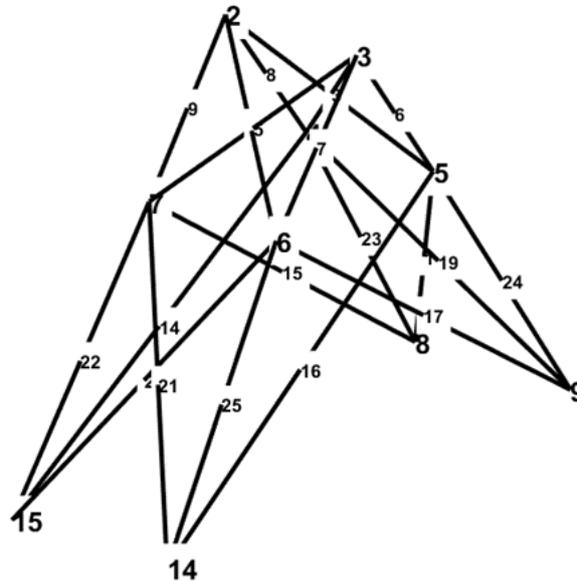
Posterior a análise estes resultados de massa foram implementados no CPMETRIC com os valores das simulações para gerar os comparativos probabilísticos entre os algoritmos WOA e BWO, com 30 agentes de pesquisa e 5000 iterações. Quando considerados os mesmos dados de entrada para os algoritmos (os mesmos utilizados nos gráficos de evolução), o método 1 é o WOA que apresentou uma probabilidade de aproximadamente 6,6% em ser melhor que o método 2 (utilizando o algoritmo BWO).

4.2.3 Treliza espacial de vinte e cinco barras

Para a treliza espacial o problema foi implementado nos algoritmos BWO, WOA e FA para analisar sua eficácia e gerar comparativos, foram realizadas variações no número de iterações máximas e nos agentes de pesquisa. Os quadros abaixo apresentam os resultados estatísticos obtidos para os testes com os valores de massa e tempo de execução do algoritmo em cada simulação.

A topologia adotada como ótima para o problema é representada na figura 22.

Figura 22- Otimização para a treliça espacial de 25 barras



Fonte: A autora

O Quadro 12 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para cada simulação do algoritmo BWO na treliça espacial de vinte e cinco barras. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 12- Resultados estatísticos algoritmo BWO – treliça espacial

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 3D- exemplo 1	75,4605	5,2965	67,419	83,0655
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 3D- exemplo 1	71,451	4,419	65,655	79,515
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=50000			Treliça 3D- exemplo 1	67,959	3,0915	62,0415	73,341
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=50		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=5000			Treliça 3D- exemplo 1	75,6045	5,7735	64,5435	90,207

Fonte: A autora

O melhor valor obtido para o BWO é de 62,04 kg, que utiliza 50 agentes de pesquisa e 50000 iterações com NIR igual a 10.

O Quadro 13 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo WOA na treliça espacial. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 13 - Resultados estatísticos algoritmo WOA – treliça espacial

Dados de entrada		Função	Valores (kg)				
Pop=30			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	Tempo(s)
Iter=5000	NIR=10	Treliça 3D - exemplo 1	67,896	8,2845	59,8995	85,0095	10167
Dados de entrada		Função	Valores (kg)				
Pop=50			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	Tempo(s)
Iter=5000	NIR=10	Treliça 3D - exemplo 1	58,3515	7,3125	58,3515	83,871	16543
Dados de entrada		Função	Valores (kg)				
Pop=50			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	Tempo(s)
Iter=50000	NIR=10	Treliça 3D - exemplo 1	57,51	2,7045	54,5445	64,314	216158
Dados de entrada		Função	Valores (kg)				
Pop=30			WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	Tempo(s)
Iter=5000	NIR=50	Treliça 3D - exemplo 1	71,127	10,0485	57,591	104,0805	54400

Fonte: A autora

Para o algoritmo WOA o menor valor obtido de massa é de 54,54 kg utilizando 50 agentes de pesquisa e 50000 iterações.

O Quadro 14 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo FA na treliça espacial. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 14 - Resultados estatísticos algoritmo FA – treliça espacial

Dados de entrada		Função	Valores (kg)				
Pop=10			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	Tempo(s)
Iter=500	NIR=10	Treliça 3D-exemplo 1	63,774	15,92325	54,774	149,9085	240
Dados de entrada		Função	Valores (kg)				
Pop=10			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	Tempo(s)
Iter=5000	NIR=10	Treliça 3D-exemplo 1	59,6205	10,368	54,5931	192,1581	648
Dados de entrada		Função	Valores (kg)				
Pop=20			FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	Tempo(s)
Iter=5000	NIR=10	Treliça 3D-exemplo 1	58,401	8,424	53,514	155,6235	5280

Fonte: A autora

Para o algoritmo FA o melhor valor obtido de massa é de 53,51 kg utilizando 20 agentes de pesquisa, 5000 iterações e NIR igual a 10.

Para a estrutura espacial de vinte e cinco barras os três algoritmos tiveram convergência dos resultados quando analisados os parâmetros estatísticos e o tempo

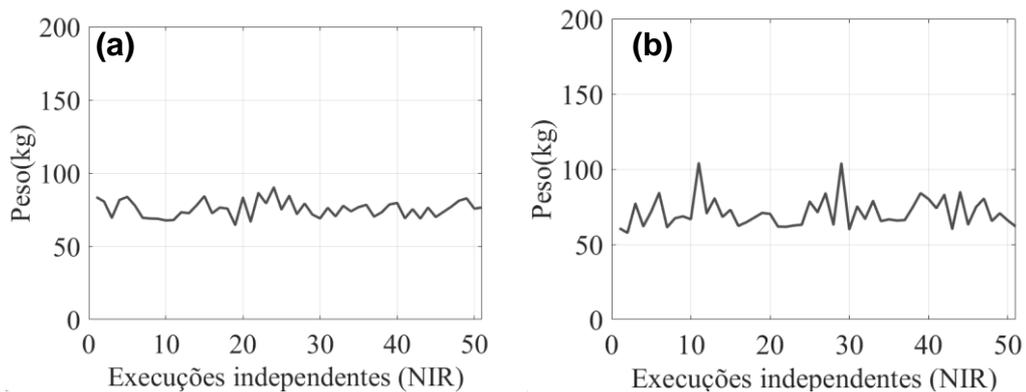
de processamento de modo geral foi maior que nas estruturas planas. Considerando a simulação com melhor valor de projeto para os algoritmos BWO e WOA, o BWO proporciona um tempo menor de resolução, porém o algoritmo WOA entrega os melhores valores estatísticos nessa execução.

O algoritmo FA foi o que encontrou o menor valor de massa e em menor tempo. Os resultados encontrados para o WOA nesse caso são mais próximos ao da referência utilizada no FA, se tratando dos menores resultados de média obtidos nos mesmos, a diferença entre eles é de aproximadamente 1,0 kg, enquanto a diferença entre o FA e BWO é de 8,5 kg.

Os gráficos 3 **a** e **b** mostram a evolução de cada algoritmo da baleia em 150.000 avaliações da função objetivo para a estrutura da treliça espacial de 25 barras.

Gráfico 3- Evolução para treliça espacial de 25 barras nos algoritmos

(a) BWO (b) WOA



Fonte: A autora

Analisando os gráficos 3 **(a)** e **(b)** para a evolução dos algoritmos WOA e BWO ambos considerando mesma população e iterações máximas, é possível perceber que o algoritmo WOA possui melhores valores de mínimo e média quando comparado ao BWO que possui uma menor dispersão de valores de massa para essa estrutura (menor desvio padrão).

Posterior a análise estes resultados de massa foram implementados no CPMETRIC com os valores das simulações utilizadas nos gráficos de evolução anteriores para gerar os comparativos probabilísticos entre os algoritmos utilizados. Quando considerados os mesmos dados de entrada para os algoritmos BWO e WOA (os mesmos utilizados nos gráficos de evolução), o BWO proporciona um tempo menor de resolução, com melhores valores de média, desvio e máximo. O método 1

é algoritmo WOA que apresentou uma probabilidade de aproximadamente 69% em ser melhor que o método 2 (utilizando o algoritmo BWO).

4.3 TORRE DE CIGRÉ

Para o problema da torre de Cigré a implementação foi realizada nos algoritmos BWO, WOA e FA, considerando otimização de tamanho e forma, analisando sua eficácia e gerando comparativos, foram feitas variações no número de iterações máximas, nos agentes de pesquisa e no NIR. Os quadros abaixo apresentam os resultados estatísticos obtidos para os testes com os valores de massa e tempo de execução do algoritmo em cada simulação.

O Quadro 15 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para cada simulação do algoritmo BWO para a torre de transmissão. A primeira coluna mostra os valores adotados para os agentes de pesquisa e iterações máxima, a segunda coluna apresenta as variações do NIR nas respectivas execuções.

Quadro 15- Resultados estatísticos algoritmo BWO – Torre

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=500			TORRE	1825,4	78,32	1710,04	1955,1
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=6000			TORRE	1675,2	73,18	1579,21	1826,7
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=600			TORRE	1762,5	101,33	1603,12	1901,5
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=1000			TORRE	1756,4	83,53	1629,55	1920,2
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=20		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=6000			TORRE	1711,7	39,48	1620,49	1781,7
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=30		BWO (média)	BWO (desvio)	BWO (mínimo)	BWO (máximo)	
Iter=500			TORRE	1811,4	78,01	1590,54	1914,8

Fonte: A autora

O melhor valor obtido para o BWO é de 1579,21 kg, que utiliza 30 agentes de pesquisa, 6000 iterações e NIR igual a 10.

O Quadro 16 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo WOA para a torre de Cigré. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 16 - Resultados estatísticos algoritmos WOA - Torre

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=500			TORRE	1865,4	142,08	1630,6	2160,7
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=6000			TORRE	1695,1	105,79	1549,6	1927,9
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=50	NIR=10		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=1000			TORRE	1662,6	118,54	1467,3	1884,5
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=30	NIR=30		WOA (média)	WOA (desvio)	WOA (mínimo)	WOA(máximo)	
Iter=500			TORRE	1820,4	132,02	1552,1	2085,9

Fonte: A autora

Para o algoritmo WOA o melhor valor obtido de massa é de 1467,3 kg, utilizando 50 agentes de pesquisa, 1000 iterações e NIR igual a 10.

O Quadro 17 apresenta os resultados de média, desvio padrão, mínimo e máximo obtidos para as implementações do algoritmo FA para a torre. As duas primeiras colunas mostram os valores adotados para os agentes de pesquisa e iterações máxima e para o NIR, respectivamente.

Quadro 17 - Resultados estatísticos algoritmos FA - Torre

Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=10	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=500			TORRE	1197,6	45,91	1134,8	1283
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=10	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=5000			TORRE	1037,9	12,74	1015,8	1056,1
Dados de entrada		Função	Valores (kg)				Tempo(s)
Pop=20	NIR=10		FA (média)	FA (desvio)	FA (mínimo)	FA (máximo)	
Iter=5000			TORRE	1018,7	9,02	1005,1	1037,1

Fonte: A autora

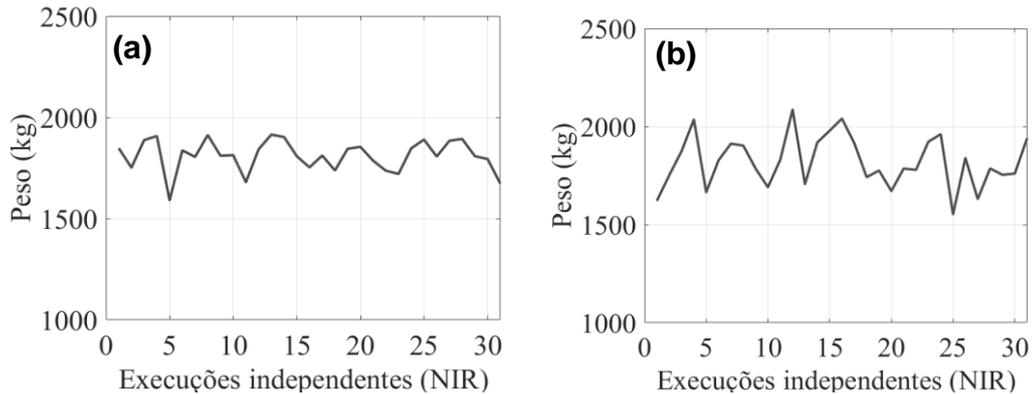
Para o algoritmo FA o melhor valor obtido de massa é de 1005,1 kg, utilizando 20 agentes de pesquisa e 5000 iterações e NIR igual a 10. O algoritmo FA foi o que encontrou o menor valor de massa.

Realizando uma comparação entre o algoritmo FA e BWO para os valores mínimos obtidos percebe-se que a diferença entre eles é de 574 kg enquanto entre o FA e WOA é de 462 kg. Nessa simulação o WOA atingiu melhor valor mínimo e média. Nesse caso os dois algoritmos das baleias obtiveram valores semelhantes, porém representam a maior diferença em relação as estruturas apresentadas quando comparados com a massa (FA x BWO e FA x WOA).

Os gráficos 4 **a** e **b** mostram a evolução da estrutura para a estrutura da torre de Cigré em relação a cada algoritmo da baleia em 15.000 avaliações da função objetivo.

Gráfico 4- Evolução para Torre nos algoritmos

(a) BWO (b) WOA



Fonte: A autora

Analisando os gráficos 4 **(a)** e **(b)** para a evolução dos algoritmos WOA e BWO com o mesmo número de avaliações da função objetivo, é possível perceber que o algoritmo WOA possui melhor solução de projeto, porém possui maiores valores de máximos, média e desvio padrão quando comparado ao BWO para essa estrutura.

Posterior a análise estes resultados de massa foram implementados no CPMETRIC com os valores das simulações utilizadas nos gráficos de evolução acima para gerar os comparativos probabilísticos entre os algoritmos da baleia. Quando considerados os mesmos dados de entrada para os algoritmos BWO e WOA (os mesmos utilizados nos gráficos de evolução), o BWO proporciona um tempo menor de resolução e possui uma probabilidade de 51% em ser melhor que o algoritmo WOA.

CONCLUSÃO

Os algoritmos de otimização estrutural descritos ao longo do trabalho foram implementados a exemplos existentes da literatura, a fim de verificar o procedimento proposto. A otimização foi realizada variando os parâmetros dos algoritmos (população, iterações máximas e número de rodadas independentes) e verificando sua eficácia nas respectivas simulações, sendo notável que uma alteração nos parâmetros pode influenciar nos resultados encontrados.

Conforme proposto inicialmente, as funções foram implementadas em um novo algoritmo da baleia, chamado BWO, para validação primeiramente em funções **benchmark** e problemas clássicos de engenharia, seguido da análise de treliças e da torre de Cigré. Essa etapa permitiu um melhor entendimento em relação a funcionalidade e eficácia do algoritmo em estudo.

Para a análise da otimização das funções **benchmark** e problemas clássicos, observa-se que os resultados obtidos no algoritmo BWO se aproximam aos já estabelecidos em literatura, mostrando certa convergência.

Para as estruturas treliçadas, o algoritmo FA encontrou melhores valores estatísticos em menor tempo de execução. Quando verificados e comparados os algoritmos das baleias, para a treliça de 11 barras, o algoritmo BWO encontrou melhor valor ótimo, bem como, maior probabilidade em ser melhor que o WOA quando aplicados os mesmos parâmetros de entrada no CPMETRIC.

Na estrutura treliçada de 39 barras o algoritmo WOA encontrou uma solução menor de projeto, porém quando analisados os outros dados estatísticos das simulações o BWO aponta melhores resultados, assumindo maior percentual em ser melhor que o WOA quando executados no CPMETRIC.

Já para a treliça espacial, o algoritmo WOA apresentou melhores valores estatísticos, e também uma probabilidade de ser melhor que o algoritmo BWO em relação aos dados quando aplicados no CPMETRIC.

Para a torre de transmissão o algoritmo BWO apresenta maior probabilidade em ser melhor que o WOA quando executados no CPMETRIC. O WOA possui melhor valor de projeto e o BWO encontra melhores valores de desvio padrão.

O algoritmo BWO possui convergência de resultados quando comparado aos outros dois algoritmos testados nas estruturas treliçadas e na torre de transmissão.

Percebe-se também que em 3 das 4 estruturas simuladas (considerando a avaliação do CPMETRIC) ele apresenta uma solução melhor que o algoritmo da baleia jubarte.

O algoritmo da baleia beluga não encontrou os melhores valores de ótimos em todas as situações, porém sabe-se que o melhor algoritmo para todos solucionar todos problemas de otimização não existe, implica em conhecimento prévio, alteração dos parâmetros, para que o otimizador possa chegar mais próximo da solução ótima para um determinado problema, podendo ser mais eficaz em uma função do que em outra.

Por fim, como sugestão de trabalhos futuros propõe-se a implementação e análise de eficácia do algoritmo em modelos reais de estruturas e em outras torres de transmissão, pois os resultados dependem do problema em estudo, assim, trazendo novos modelos de torres para a análise o algoritmo pode-se encontrar pontos ótimos nesses casos.

REFERÊNCIAS

- ADELI, H.; KAMAL, O. Efficient optimization of plane trusses. **Advances in Engineering Software and Workstations**, v. 13, n. 3, p. 116 – 122, 1991. ISSN 0961-3552.
- ARORA, J. S. **Introduction to Optimum Design**. Elsevier. 4^o edição, 2017.
- ASSIS, G. C. de O. **Metodologia baseada em metamodelos para a otimização global de problemas black-box com restrições computacionalmente onerosas**. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Civil, Florianópolis, 2021.
- BEZ, M. P. Otimização de Massa de Estruturas Treliçadas através do Algoritmo Meta-Heurístico Firefly. 2011. 15p. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) - Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.
- BRASIL, G. S. **Otimização de pórticos de aço de galpões por métodos heurísticos**. Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Civil, Porto Alegre, 2022.
- CARLON, A.C. **Desenvolvimento de um novo operador para algoritmos metaheurísticos baseado na maximização da rigidez aplicado à otimização de estruturas treliçadas**. Dissertação (Mestrado em Engenharia Civil) - Universidade Federal de Santa Catarina, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Florianópolis, 2015.
- CARVALHO, E C R. **Solução de problemas de otimização com restrições usando estratégias de penalização adaptativa e um algoritmo do tipo PSO**. Dissertação (Mestrado em Programa de pós-graduação em engenharia civil) - Universidade Federal de Juiz de Fora, Juiz de Fora, 2014.
- GANDOMI, A. H.; YANG, X.-S.; ALAVI, A. H. Mixed variable structural optimization using Firefly. **Computer and Structures- Elsevier**.v.89, p. 2325-2336, 2011.
- GOMES, W. J. S.; BECK, A. T.; LOPEZ, R. H.; MIGUEL, L. F. F. A probabilistic metric for comparing metaheuristic optimization algorithms. **Structural Safety- Elsevier**, v. 70, p. 59–70, 2018.
- HE S, Prempain E, Wu QH. **An improved particle swarm optimizer for mechanical design optimization problems**. *Eng Optimiz* 2004;36(5):585–605.
- HEMP, W. S. **Optimum structures**. Clarendon Press, 1973.
- KARABOGA, D.; AKAY, B. A comparative study of Artificial Bee Colony algorithm **Applied Mathematics and Computation-Elsevier** v. 214, p. 108–132, 2009.

LACERDA E. G. M. A. **Otimização Nuvem de Partículas (Particle Swarm)**. Departamento de Engenharia da Computação e Automação da Universidade Federal do Rio Grande do Norte, 2007.

LI G.; MENG Z.; ZONG C. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. **Knowledge- Based Systems- Elsevier**, v. 251, 2022.

LOPEZ, R. H. **Introdução à otimização por enxame de partículas**. Florianópolis, UFSC, 2022 (Notas de aula).

MARTHA, L. **Análise de estruturas: conceitos e métodos básicos**. Elsevier Brasil, 2010.

MATLAB. **version R2021a**. Natick, Massachusetts: The MathWorks Inc., 2021.

MEZURA-MONTES, Efrén; COELLO, CA Coello; LANDA-BECERRA, Ricardo. Engineering optimization using simple evolutionary algorithm. *In: IEEE. Proceedings [...] 15th IEEE international conference on tools with artificial intelligence.[S.l.: s.n.]*, 2003. p. 149–156.

MIGUEL, L. F.F; LOPEZ, Rafael Holdorf; MIGUEL, Letícia Fleck Fadel. Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. **Advances in Engineering Software - Elsevier**, v. 56, p. 23–37, 2013.

MIRJALILI, S.; LEWIS, A. The Whale Optimization Algorithm. **Advances in Engineering Software - Elsevier.**, v. 95, p. 51–67, 2016.

RAO, S. **Engineering Optimization: Theory and Practice**. 4ª Edição. New Jersey. John Wiley & Sons Inc, 2009.

RIBEIRO, Celso da Cruz Carneiro. **Heurísticas para o Problema das p-Mediana Conectadas**. 2006. Tese (Doutorado) – PUC–Rio.

SAKA, M. P.; GEEM, Z. W. **Mathematical and Metaheuristic Applications in Design Optimization of Steel Frame Structures: An Extensive Review**. Hindawi Publishing Corporation, 2013.

SILVA, L. A. R. **Otimização topológica de treliças considerando o colapso progressivo devido às incertezas intrínsecas e epistêmicas**. 2022. 104 p. Dissertação (Mestrado em Ciências - Engenharia Civil (Engenharia de Estruturas)) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2022.

SOUZA, M. J.; AMARAL, P. F. T., ALMEIDA, S. R. M. Otimização não determinística de treliças. **Revista eletrônica de engenharia civil**. v.15. n.2. p.01-16. Universidade Federal de Goiás, 2019. Disponível em: <https://revistas.ufg.br>. Acesso em: 23 de janeiro de 2023.

SOUZA, R. R. **A procedure for the size, shape and topology optimization of transmission line tower structures**. Dissertação (Mestrado em Programa de pós-

graduação em engenharia civil) - Universidade Federal de Santa Catarina, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Florianópolis, 2016.

SOUZA, R. R., MIGUEL, L. F.F., LOPEZ, R. H., MIGUEL, L. F. F. **On the Size and Shape Optimization of Transmission Line Towers**. 14th Pan-American Congress of Applied Mechanics. March 24-28, 2014, Santiago, Chile.

SOUZA, O. A. P. de **Elaboração de um algoritmo de otimização aplicado a engenharia estrutural: Cicle-Inspired Optimization Algorithm (CIOA)**. Dissertação (Mestrado em Programa de pós-graduação em engenharia civil) - Universidade Federal do Rio Grande do Sul. Porto Alegre, 2021.

SURJANOVIC, S., BINGHAM, D. **Virtual Library of Simulation Experiments: Test Functions and Datasets**. 2018. Disponível em: <http://www.sfu.ca/~ssurjano/>. Acesso em: 05 de dezembro de 2022.

TALBI, El-Ghazali. **Metaheuristics: from design to implementation**. [S.l.]: John Wiley & Sons, 2009. v. 74.

YANG, X-S. Firefly algorithms for multimodal optimization. *In*: INTERNATIONAL SYMPOSIUM ON STOCHASTIC ALGORITHMS, 5., 2009a, Sapporo. **Proceedings [...]** Berlin: Springer, 2009a, p. 169-178.

YANG, X. S. Engineering Optimization. **An Introduction with Metaheuristic Applications**. University of Cambridge. John Wiley & Sons, Inc. Hoboken, New Jersey, 2010.

YANG, X. S. Review of meta-heuristics and generalised evolutionary walk algorithm. **International Journal of Bio-Inspired Computation**, v. 3, n. 2, p. 77-84, 2011.

Working group B2.08, 2009. "Influence of the hyperstatic modeling on the behavior of transmission line lattice structures" **Cigré Brochure** 387.