

André Gustavo Carlon

**Desenvolvimento de um novo operador para
algoritmos metaheurísticos baseado na
maximização da rigidez aplicado à otimização
de estruturas treliçadas**

Florianópolis/SC-Brasil

2015

André Gustavo Carlon

**Desenvolvimento de um novo operador para algoritmos
metaheurísticos baseado na maximização da rigidez
aplicado à otimização de estruturas treliçadas**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Engenharia Civil pela Universidade Federal de Santa Catarina.

Universidade Federal de Santa Catarina – UFSC

Departamento de Engenharia Civil

Programa de Pós-Graduação em Engenharia Civil

Orientador: Prof. Dr. Rafael Holdorf Lopez

Florianópolis/SC-Brasil

2015

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Carlton, André Gustavo

Desenvolvimento de um novo operador para algoritmos metaheurísticos baseado na maximização da rigidez aplicado à otimização de estruturas treliçadas / André Gustavo Carlton ; orientador, Rafael Holdorf Lopez ; coorientador, Leandro Fleck Fadel Miguel. - Florianópolis, SC, 2015.
105 p.

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

Inclui referências

1. Engenharia Civil. 2. Otimização. 3. Treliças. 4. Algoritmos metaheurísticos. I. Lopez, Rafael Holdorf. II. Miguel, Leandro Fleck Fadel. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Civil. IV. Título.

André Gustavo Carlon


**Desenvolvimento de um novo operador para algoritmos
metaheurísticos baseado na maximização da rigidez
aplicado à otimização de estruturas treliçadas**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Engenharia Civil pela Universidade Federal de Santa Catarina.


Trabalho aprovado. Florianópolis, 20 de agosto de 2015:




Prof. Dr. Glicério Trichês
Coordenador do Programa



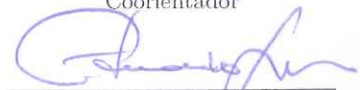
Prof. Dr. Rafael Holdorf
Lopez
Orientador




Prof. Dr. Leandro Fleck
Fadel Miguel
Coorientador



Prof. Dr. Marco Antônio
Luersen
Membro Externo



Prof. Dr. Eduardo Alberto
Fancello
Examinador



Prof. Dr. Wellison José
Santana Gomes
Examinador

Agradecimentos

Agradeço ao professor Rafael Lopez, meu orientador, por sempre me indicar o gradiente a ser seguido e ao professor Leandro Miguel por todo o apoio desde os tempos de graduação. Agradeço também a todos os meus professores da graduação e do mestrado por todo o conhecimento transmitido.

Resumo

Um operador baseado na maximização da rigidez é implementado com o objetivo de melhorar a convergência de algoritmos metaheurísticos na otimização simultânea dimensional, de forma e topológica de treliças com variáveis mistas. O problema estudado apresenta não-convexidade, o que justifica o uso de algoritmos heurísticos na sua solução. O operador faz uso da informação do gradiente do trabalho em relação às variáveis de forma, as únicas contínuas, para buscar uma estrutura mais rígida. O gradiente do trabalho é significativamente menos custoso de ser obtido computacionalmente do que o gradiente da função objetivo e por isso é empregado no presente trabalho. A busca unidimensional é realizada minimizando-se o trabalho e o peso simultaneamente. O algoritmo de competição imperialista *ICA*, um algoritmo metaheurístico de população, é usado para testar o novo operador. Três problemas clássicos de topologia inicial são estudados com e sem o operador e os resultados obtidos são comparados com a literatura. O operador se mostrou eficiente dependendo do problema estudado sendo especialmente eficaz quando restrições de deslocamento máximo se encontram ativas.

Palavras-chaves: otimização, treliças, metaheurísticos.

Abstract

A operator based on the maximization of stiffness is implemented with the objective of improving the convergence of metaheuristic algorithms on the simultaneous optimization of dimensions, shape and topology of trusses with mixed variables. The use of metaheuristics is justified by the nonconvexity of the problem. The proposed operator uses the gradient of the work by the shape variables - the ones treated as continuous on the present work - to search for a stiffer structure. The use of the gradient of the work is justified for its low cost to evaluate. The unidimensional search on the direction of the work gradient is realized through the minimization of a combined function of the penalized weight and the work absorbed by the truss. The ICA, a population based metaheuristic, is employed to test the new operator. Three ground structure benchmark problems are analysed with and without the operator and the results are compared. The proposed operator proved to be efficient depending on the studied problem, specially when the stiffness of the structure is more important for its performance.

Key-words: optimization, trusses, metaheuristics.

Lista de ilustrações

Figura 1 – Exemplo de treliça de Michell. Fonte:Hegemier e Prager(1969)	24
Figura 2 – Tipos de otimização estrutural de treliças.	25
Figura 3 – Região não convexa possuindo dois mínimos locais sendo um deles global.	27
Figura 4 – Organograma do <i>ICA</i>	44
Figura 5 – Pseudocódigo do Algoritmo de Competição Imperialista.	45
Figura 6 – Geração dos impérios.	46
Figura 7 – Representação do movimento de uma colônia. Fonte:Atashpaz-Gargari e Lucas(2007)	46
Figura 8 – Colônia dominando imperialista. Fonte:Atashpaz-Gargari e Lucas(2007)	47
Figura 9 – Impérios competindo por colônias. Fonte:Atashpaz-Gargari e Lucas(2007)	48
Figura 10 – Função clássica estudada.	49
Figura 11 – Exemplo do <i>ICA</i> para um problema clássico.	50
Figura 12 – Configuração final do exemplo do <i>ICA</i> para um problema clássico - Década 20.	51
Figura 13 – Organograma do Algoritmo de Competição Imperialista (<i>ICA</i>) com o Operador de Forma (OF).	53
Figura 14 – Organograma do Operador de Forma onde t são as décadas.	59
Figura 15 – Pseudocódigo do Operador de Forma.	60
Figura 16 – Treliça inicial do primeiro exemplo de busca local com os sentidos do gradiente de forma ilustrados.	61
Figura 17 – Treliça do primeiro exemplo de busca local após operação de forma.	61
Figura 18 – Representação logarítmica na direção do gradiente de forma no primeiro exemplo de busca local	62

Figura 19 – Representação logarítmica da função Z em relação ao passo(in) no primeiro exemplo.	62
Figura 20 – Função Z na busca unidimensional do primeiro exemplo.	64
Figura 21 – Busca unidimensional no primeiro exemplo de busca local.	64
Figura 22 – Representação logarítmica na direção do gradiente de forma no segundo exemplo de busca local.	65
Figura 23 – Representação logarítmica da função Z em relação ao passo(in) no segundo exemplo.	65
Figura 24 – Treliça inicial do segundo exemplo de busca local com as direções do gradiente de forma ilustradas.	66
Figura 25 – Treliça do segundo exemplo de busca local após operação de forma.	66
Figura 26 – Função Z na busca unidimensional do segundo exemplo.	67
Figura 27 – Busca unidimensional no segundo exemplo de busca local.	68
Figura 28 – Topologia inicial da treliça de 10 elementos em polegadas.	70
Figura 29 – Execuções do <i>ICA</i> para a treliça de 10 barras. Mínimos em linha contínua e médias em tracejado.	71
Figura 30 – Solução ótima da treliça de 10 barras encontrada pelo <i>ICA</i> em polegadas.	71
Figura 31 – Execuções do <i>ICA-OF</i> para a treliça de 10 barras. Mínimos em linha contínua e médias em tracejado	72
Figura 32 – Solução ótima da treliça de 10 barras encontrada pelo <i>ICA-OF</i> em polegadas.	72
Figura 33 – Topologia inicial da treliça de 15 barras.	75
Figura 34 – Solução ótima da treliça de 15 barras encontrada pelo <i>ICA</i> em polegadas.	78
Figura 35 – Solução ótima da treliça de 15 barras encontrada pelo <i>ICA-OF</i> em polegadas.	78
Figura 36 – Execuções do <i>ICA</i> para a treliça de 15 barras. Mínimos em linha contínua e médias em tracejado.	79

Figura 37 – Execuções do <i>ICA-OF</i> para a treliça de 15 barras. Mínimos em linha contínua e médias em tracejado.	79
Figura 38 – Topologia inicial da treliça de 25 elementos em polegadas em vista isométrica.	84
Figura 39 – Topologia inicial da treliça de 25 elementos em polegadas vista no plano x-y.	84
Figura 40 – Execuções do <i>ICA</i> para a torre treliçada. Mínimos em linha contínua e médias em tracejado.	85
Figura 41 – Execuções do <i>ICA-OF</i> para a torre treliçada. Mínimos em linha contínua e médias em tracejado.	85
Figura 42 – Solução ótima da treliça de 25 elementos encontrada pelo <i>ICA</i> em polegadas vista no plano x-y.	86
Figura 43 – Solução ótima da treliça de 25 elementos encontrada pelo <i>ICA-OF</i> em polegadas vista no plano x-y.	87
Figura 44 – Solução ótima da treliça de 25 elementos encontrada pelo <i>ICA</i> em polegadas em vista isométrica.	87
Figura 45 – Solução ótima da treliça de 25 elementos encontrada pelo <i>ICA-OF</i> em polegadas em vista isométrica.	90
Figura 46 – Exemplos de conjuntos convexos(A) e não convexos(B). Fonte:Arora(2014)	102

Lista de tabelas

Tabela 1	– Dados do primeiro exemplo de busca local.	63
Tabela 2	– Possíveis áreas das seções transversais para os elementos da treliça de 10 barras em in^2	69
Tabela 3	– Parâmetros do problema da treliça de 10 barras. . .	70
Tabela 4	– Parâmetros do <i>ICA</i> e do <i>ICA-OF</i> para o problema da treliça de 10 barras.	73
Tabela 5	– Comparação entre o desempenho dos algoritmos na solução da treliça de 10 barras.	73
Tabela 6	– Comparação entre os melhores resultados para a treliça de 10 barras. Áreas em in^2 e coordenadas em in	74
Tabela 7	– Limites das variáveis de forma para o problema da treliça de 15 barras.	76
Tabela 8	– Possíveis áreas das seções transversais para os elementos da treliça de 15 barras em in^2	76
Tabela 9	– Parâmetros do problema da treliça de 15 barras. . .	76
Tabela 10	– Parâmetros do <i>ICA</i> e do <i>ICA-OF</i> para o problema da treliça de 15 barras	77
Tabela 11	– Resultados para otimização dimensional, de forma e topológica da treliça de 15 barras.	77
Tabela 12	– Comparação entre os melhores resultados para a treliça de 15 barras. Áreas em in^2 e coordenadas em in	80
Tabela 13	– Parâmetros do problema da treliça de 25 barras. . .	81
Tabela 14	– Parâmetros do <i>ICA</i> e do <i>ICA-OF</i> para o problema da treliça de 25 barras.	82
Tabela 15	– Possíveis áreas das seções transversais para os elementos da treliça de 25 barras em in^2	82
Tabela 16	– Carregamentos da torre de 25 barras.	83
Tabela 17	– Limites das variáveis de forma.	83

Tabela 18 – Posições dos nós para a torre de 25 barras.	83
Tabela 19 – Grupos de elementos com áreas iguais para a torre de 25 barras.	83
Tabela 20 – Comparação entre o desempenho dos algoritmos na solução da torre de 25 barras.	88
Tabela 21 – Comparação entre os melhores resultados para a treliça de 25 barras. Áreas em in^2 e coordenadas em in	88

Lista de abreviaturas e siglas

<i>ICA</i>	<i>Imperialist competitive algorithm</i> ou algoritmo competitivo imperialista
OF	Operador de forma
<i>ICA-OF</i>	Algoritmo de competição imperialista acoplado ao operador de forma
<i>OFE</i>	<i>Objective function evaluation</i> ou avaliações da função objetivo
<i>PSO</i>	<i>Particle swarm optimization</i> ou otimização por enxame de partículas
<i>GA</i>	<i>Genetic algorithm</i> ou algoritmo genético
SI	Sistema internacional de unidades
<i>IS</i>	<i>Imperial System</i> ou sistema imperial de unidades

Lista de símbolos

\mathbf{x}	Vetor de variáveis
P	Peso da treliça
W	Trabalho
A	Área da seção transversal
γ	Peso específico
l	Comprimento
\mathbf{g}	Vetor de restrições de desigualdade
\mathbf{h}	Vetor de restrições de igualdade
ξ	Variável de forma:(ex.: uma coordenada de um nó)
E	Módulo de elasticidade do material
τ	Penalização
f	Função objetivo
$f_{penalizada}$	Função objetivo penalizada
Z	Função objetivo da busca local
η	Ângulo de assimilação do <i>ICA</i>
β	Coefficiente de assimilação do <i>ICA</i>
\mathbf{F}	Vetor de forças
\mathbf{K}	Matriz de rigidez
\mathbf{u}	Vetor de deslocamentos nodais
\mathbf{x}^*	Vetor de variáveis no ponto ótimo

S	Região restrita, onde nenhuma restrição está violada
σ	Desvio padrão
φ	Passo das variáveis de forma

Sumário

1	INTRODUÇÃO	23
1.1	Motivação	28
1.2	Objetivo	29
1.3	Organização do trabalho	29
2	APRESENTAÇÃO DO PROBLEMA	31
2.1	Revisão histórica	31
2.2	Formulação matemática	36
2.3	Otimização estrutural de treliças	36
2.3.1	Otimização dimensional	37
2.3.2	Otimização de forma	38
2.3.3	Otimização topológica	38
2.3.4	Restrições	39
2.4	Otimização global/local	40
2.4.1	Otimização global	40
2.4.2	Otimização local	41
3	ALGORITMO DE COMPETIÇÃO IMPERIALISTA	43
3.1	Exemplo numérico	49
4	OPERADOR DE FORMA	53
4.1	Exemplo do operador de forma sem violação de restrições	58
4.2	Exemplo do operador de forma para início penalizado	63
5	EXEMPLOS NUMÉRICOS	69
5.1	Treliça de 10 barras	69
5.2	Treliça de 15 barras	74
5.3	Torre treliçada de 25 barras	81

Conclusão 91

REFERÊNCIAS 93

APÊNDICES 99

**APÊNDICE A – REVISÃO DE CONCEITOS DE
OTIMIZAÇÃO 101**

A.1 Mínimos 101

A.1.1 Mínimo global 101

A.1.2 Mínimo local 101

A.2 Convexidade 102

1 Introdução

A definição da estrutura mais eficiente é um problema comum na engenharia de estruturas. Em geral o engenheiro faz uso da sua experiência como projetista para definir a geometria e realiza o dimensionamento dos elementos estruturais baseado nesta. Algumas correções podem ser realizadas ao longo do processo, mas dificilmente se poderia afirmar que a estrutura final obtida seria a estrutura ótima para o problema que se busca resolver. Como depende da intuição do projetista, a ineficiência do processo convencional cresce conforme se aumenta a complexidade do problema. Um dos problemas comuns na engenharia de estruturas é o da busca pela treliça mais eficiente sendo este o objeto de estudo do primeiro trabalho publicado em otimização estrutural, o artigo de Michell (1904 apud Hegemier e Prager(2), 1969). As treliças são estruturas compostas por elementos rotulados e que apresentam a propriedade de vencer grandes vãos com economia, transmitindo as forças aplicadas até os apoios através de esforços internos de compressão e tração. Entretanto, a distribuição dos esforços, assim como os deslocamentos e as deformações, são dependentes das variáveis de projeto, o que dificulta uma resolução analítica do problema, principalmente quando se consideram como variáveis as posições dos nós e quais elementos ligam estes. Segundo Michell, a treliça ótima possuiria infinitos elementos, estando esta idealmente próxima de uma estrutura contínua onde os elementos se encontram na direção das linhas de deformação principais de um campo de deslocamentos que tenha todas as deformações constantes e iguais à $\varepsilon_{max} = \sigma_{max}/E_{material}$. Uma treliça ótima de Michell transmitindo um esforço P até a sua fundação rígida é apresentada na Figura 1.

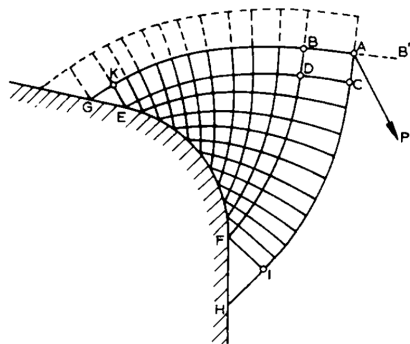


Figura 1 – Exemplo de treliça de Michell. Fonte:Hegemier e Prager(2).

Embora a estrutura ótima seja contínua, uma análise estrutural numérica usualmente depende de uma discretização deste contínuo que possibilite uma manipulação matricial do problema. Diversos métodos foram desenvolvidos que buscam minimizar o resíduo da discretização e encontrar numericamente resultados que se aproximem do comportamento da estrutura no contínuo, sendo o problema de falha local um exemplo da necessidade destes estudos(3, 4). O número de nós e elementos no qual se discretiza uma estrutura contínua influencia diretamente o custo da análise estrutural, sendo a modelagem de treliça pelo método dos deslocamentos um dos métodos mais baratos computacionalmente por resultar em sistemas lineares de ordem baixa. Nos anos 60, motivados pelos avanços computacionais da época, pesquisadores implementaram métodos de programação linear para solucionar problemas de otimização de treliças. Nestas pesquisas se buscava encontrar uma treliça eficiente definida em função de um número de variáveis a partir de uma topologia inicial. Formalizou-se, assim, a otimização de treliças em 3 tipos: otimização dimensional, otimização topológica e otimização de forma(5, 6, 7). A otimização dimensional é entendida como a busca pelas dimensões ótimas das seções transversais dos elementos, como apresentado na Figura 2b. A otimização simultânea dimensional e topológica envolve também a possibilidade da retirada de elementos do sistema, ou de mudanças no tipo de conectividade entre elementos

e é ilustrada na Figura 2c. Por fim, quando se consideram também como variáveis as coordenadas das posições dos nós, caracteriza-se um problema de otimização simultânea dimensional, topológica e de forma, como se observa na Figura 2d.

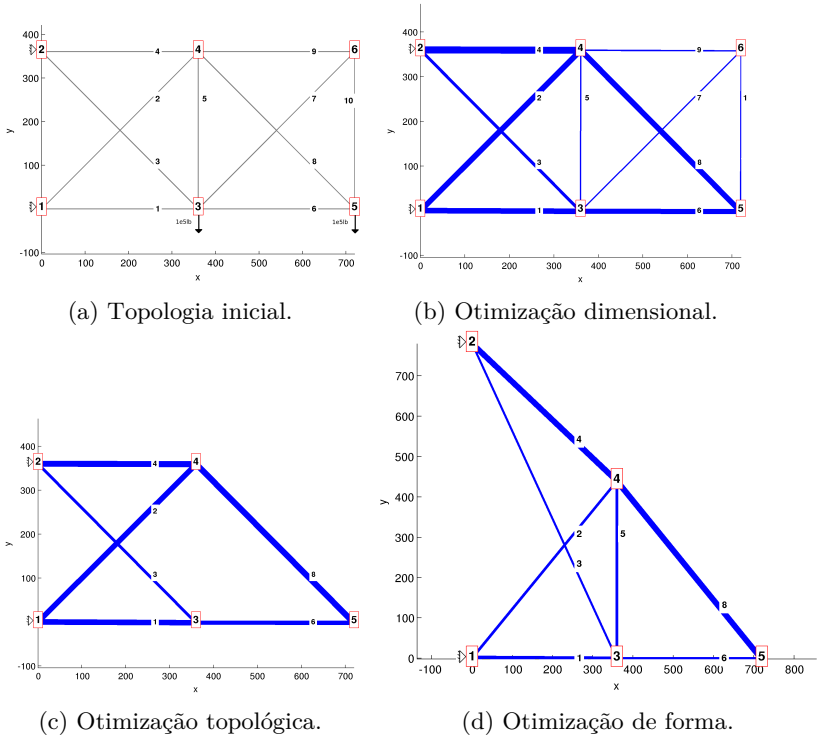


Figura 2 – Tipos de otimização estrutural de treliças.

Para cada tipo de otimização que é realizada simultaneamente, acumulam-se as respectivas variáveis, aumentando as dimensões do espaço de busca. Dorn, Gomory e Greenberg (1964 apud Rozvany(8), 2011) foram os primeiros a utilizar métodos baseados em gradientes para otimizar as áreas das seções transversais dos elementos da treliça e a topologia da mesma. Dobbs e Felton(7) publicaram um artigo no qual pela primeira vez se realizou otimização topológica, de forma e dimensional, considerando múltiplos casos de carregamento e restrição de

flambagem. Neste artigo, a otimização de forma foi realizada em passos alternados com a otimização dimensional e topológica, resultando em estruturas mais eficientes do que apenas se trabalhando com otimização dimensional e topológica. Entretanto, estes autores consideraram as possíveis áreas para as seções transversais dentro de um domínio contínuo; quando na prática do dimensionamento de treliças, principalmente metálicas, o projetista deve escolher os perfis para o seu projeto dentre os vários disponíveis em um catálogo de fabricante. Considerando-se as possíveis áreas dentre as de um catálogo, caracteriza-se um problema de domínio discreto. Logo, sendo a posição dos nós tratada como variável contínua, conclui-se que o problema em questão é um problema de variáveis mistas. Dada a dificuldade em lidar com problemas de variáveis mistas, poucos trabalhos abordam este tipo de problema, sendo que a maioria se limita à otimização dimensional. Dentre as publicações que abordam a otimização simultânea dimensional, de forma e topológica com variáveis discretas, observa-se principalmente o emprego de algoritmos metaheurísticos, como no caso de Miguel *et al*(9), Balling *et al*(10), Tang *et al*(11) e Wu e Chow(12).

Os algoritmos metaheurísticos são heurísticas construídas para a solução de problemas de otimização formalizados como tal e que o fazem através de analogias com situações de convergência(13). Os algoritmos heurísticos em geral apresentam a característica de não depender da informação do gradiente da função que buscam minimizar nem do gradiente das restrições. Isto porque os heurísticos fazem uso apenas do valor da função objetivo em determinados pontos, o que possibilita o tratamento de problemas com domínios discretos. Entretanto, para domínios contínuos, em geral os heurísticos apresentam uma convergência significativamente mais lenta do que métodos baseados em gradientes(14). Na prática uma pior convergência se traduz em um custo computacional mais elevado para alcançar um resultado ótimo. Os métodos baseados em gradientes e hessiana, por outro lado, são usualmente construídos para convergir para mínimos locais e garantem a solução ótima global desde que se prove que a função objetivo e as restrições são convexas

nos seus domínios. Caso se prove a convexidade de um problema de otimização, garante-se que há apenas um mínimo e que este é o mínimo global. Os conceitos de mínimo local, mínimo global e convexidade são definidos no Apêndice A e ilustrados na Figura 3.

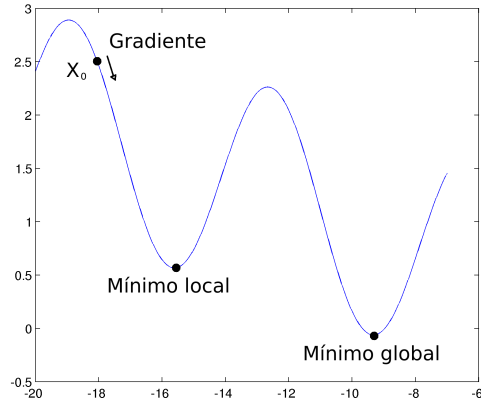


Figura 3 – Região não convexa possuindo dois mínimos locais sendo um deles global.

Metaheurísticos de população, como o algoritmo genético (*GA*), algoritmo de competição imperialista (*ICA*) e o algoritmo de enxame de partículas (*PSO*) apresentam mecanismos em suas rotinas que dividem o custo computacional empregado entre exploração do espaço de busca por regiões promissoras (busca global) e a convergência para os mínimos locais destas (busca local). É importante notar que, dada a natureza dos heurísticos, não se pode garantir que os ótimos encontrados satisfaçam as condições necessárias e suficientes para que sejam ótimos locais. Segundo Achtziger(15) o problema de otimização simultânea dimensional, de forma e topológica com variáveis mistas é não-convexo e um algoritmo de busca robusto e capaz de lidar com tal problema se faz necessário. Como uma das características dos metaheurísticos é a robustez na solução de problemas não-convexos(14), estes se mostram uma possível alternativa para a otimização simultânea dimensional, de forma e topológica de

treliças com variáveis mistas.

1.1 Motivação

Embora os algoritmos metaheurísticos se mostrem eficientes na busca global, estes em geral exigem um custo computacional alto para convergir na busca por mínimos locais(14), o que muitas vezes inviabiliza o uso dos mesmos na solução de problemas práticos de engenharia. Alguns algoritmos híbridos vem sendo desenvolvidos a partir de metaheurísticos com a finalidade de melhorá-los em termos da convergência e/ou redução de custo computacional. Neste sentido os algoritmos metaheurísticos híbridos podem ser metaheurístico/metaheurístico(16) ou metaheurístico/determinístico(17). Outros algoritmos híbridos, não baseados em metaheurísticos, seriam do tipo reinício/ determinístico(18) e Pincus/determinístico(19, 20). Alguns pesquisadores hibridizam metaheurísticos incluindo operadores de outros metaheurísticos nas rotinas destes, como por exemplo nos trabalhos de Hao, Wang e Huang(21) e Jiang, Zhang e Xie(22) que empregaram operadores mutação e *cross-over* do algoritmo genético (*GA*) no algoritmo de enxame de partículas (*PSO*).

Para acelerar a convergência de algoritmos metaheurísticos na otimização estrutural de treliças, propõe-se o desenvolvimento de um operador baseado na maximização da rigidez. Espera-se que o operador proposto com base em uma análise de sensibilidade melhore a convergência do *ICA*. Hegemier e Prager(2) provaram, manipulando as equações de Michell, que a estrutura ótima para um determinado problema de treliça de Michell seria aquela que fosse mais rígida para um volume constante de material e que satisfizesse todas as restrições. Desta forma é possível buscar a treliça ótima através da solução do problema análogo ao do mínimo peso, que seria o do mínimo trabalho absorvido pela estrutura, mantendo-se um volume de material empregado constante. Conforme apresentado no Capítulo 4, é possível obter um gradiente numérico de baixo custo computacional manipulando-se a derivada do

trabalho sofrido pela treliça em função das variáveis de forma. Assim, buscando minimizar o trabalho absorvido pela treliça, o operador de forma deve acelerar a convergência do *ICA* para uma forma mais rígida. Embora proponha-se implementar o operador de forma (OF) no presente trabalho para funcionar em conjunto com o *ICA*, não se pode dizer que o algoritmo *ICA-OF* seja híbrido, pois o operador OF é inovador e não originário de outro metaheurístico.

Sendo o problema de otimização de treliças usualmente não-convexo, um novo operador que auxilie metaheurísticos na sua solução é um incremento no estado da arte da literatura.

1.2 Objetivo

O objetivo principal deste trabalho é desenvolver um operador para acelerar a convergência de algoritmos metaheurísticos na otimização dimensional, de forma e topológica de estruturas treliçadas na presença de variáveis contínuas e discretas.

Como objetivos específicos, tem-se:

- Desenvolver um operador de busca de forma baseado na maximização da rigidez;
- Adaptar o proposto operador ao algoritmo de competição imperialista (*ICA*);
- Gerar uma base de comparação estudando exemplos clássicos com o *ICA* e com o proposto *ICA-OF*.

1.3 Organização do trabalho

No presente trabalho os capítulos são organizados da seguinte forma:

Capítulo 1 - *Introdução*. Contextualiza-se o problema que se busca resolver e propõe-se o método para a solução deste. Discutem-se os aspectos de ineditismo do trabalho e objetivos a ser alcançados.

Capítulo 2 - *Apresentação do problema*. Neste capítulo é realizada uma revisão histórica seguida da formulação matemática adotada para o problema de otimização de treliças. São introduzidos conceitos de otimização como otimização dimensional, de forma e topológica e suas formulações. Discutem-se as restrições e como as suas respectivas penalizações serão aplicadas. Por último se apresentam as noções de otimização global e local, assim como um esquema simplificado do funcionamento do algoritmo *ICA-OF* empregando o *ICA* e o operador de forma.

Capítulo 3 - *Algoritmo de Competição Imperialista*. Neste capítulo o *ICA* tem o seu funcionamento explicado em todas as suas etapas. Um exemplo da aplicação do *ICA* para a solução de um problema clássico é apresentado.

Capítulo 4 - *Operador de Forma*. O Operador de Forma (OF) é introduzido e embasado matematicamente. Apresentam-se exemplos do seu funcionamento.

Capítulo 5 - *Exemplos numéricos*. Treliças de 10 e 15 barras e uma torre treliçada de 25 barras são estudadas usando o *ICA* com e sem o operador e os resultados são comparados com a literatura.

Conclusão - O desempenho do operador é avaliado observando-se os resultados obtidos. Discute-se se os objetivos do trabalho foram alcançados e propõe-se sugestões para futuras pesquisas.

2 Apresentação do problema

2.1 Revisão histórica

Dorn, Gomory e Greenberg(5) foram os primeiros a utilizar métodos matemáticos para otimizar não apenas as áreas das seções transversais dos elementos da treliça mas também a topologia da mesma. Sved e Ginos(23) foram os primeiros a estudar a otimização de treliças submetidas a múltiplos casos de carregamento. Sved e Ginos também realizaram otimização topológica e encontraram problemas no ponto em que a área do elemento de treliça é mínima, problema comum na otimização topológica de treliças no qual o algoritmo não consegue diminuir a área da seção transversal sem violar a restrição de tensão máxima, o que impede o algoritmo de retirar o elemento do sistema. Dobbs e Felton(7) publicaram um artigo no qual pela primeira vez se realizou otimização topológica, de forma e dimensional considerando múltiplos casos de carregamento e restrição de flambagem. Neste artigo a otimização de forma foi realizada em passos alternados com a otimização dimensional e topológica, resultando em estruturas mais eficientes do que apenas se trabalhando com otimização dimensional e topológica.

Outros autores ao longo da década de 1960 trabalharam na busca por métodos diversos(6, 24), limitados sempre pela restrita capacidade computacional da época. Vanderplaats e Moses(25), em 1972, conseguiram melhores resultados na otimização de forma tratando as variáveis dimensionais e de forma em sub-espacos de busca diferentes e modelando o otimizador matematicamente de forma que as variáveis dimensionais fossem dependentes das variáveis de forma. Para a busca no sub-espaço das variáveis dimensionais Vanderplaats e Moses usaram um método de *fully-stressed design*, desenvolvido por estes, enquanto para a busca no sub-espaço das variáveis de forma foi usado um método de máxima descida com restrições. Pedersen(26), em 1973, aplicou um método

de programação linear sequencial e *tight-move limits* para otimizar simultaneamente as dimensões e a forma da treliça.

Com o objetivo de tratar não apenas das dimensões, mas também da geometria da treliça em problemas de maior porte, Adeli e Balasubramanyam(27), em 1988, desenvolveram um algoritmo para solucionar o problema da melhor ponte, trazendo à otimização de treliças uma perspectiva mais prática. O algoritmo BTEXPERT desenvolvido por estes consiste de uma série de regras heurísticas especializadas com o objetivo de escolher um entre 4 tipos de treliças para pontes (Pratt, Parker, *Parallel-chord* e K) e as suas respectivas dimensões e seções transversais para um dado problema.

Os métodos numéricos aplicados até então supunham um possível conjunto contínuo de áreas das seções transversais. Kirsch(28), em sua publicação de 1982, foi o primeiro a tratar da otimização de forma com variáveis dimensionais discretas aplicando um método de técnica de reanálise aproximada. Kirsch também dividiu o espaço de busca em sub-espacos distintos como Vaanderplaats, sendo o sub-espaco referente às variáveis dimensionais discreto enquanto o sub-espaco referente às variáveis de forma é contínuo, caracterizando um problema de variáveis mistas.

Embora alguns autores tenham se esforçado em melhorar a eficiência dos algoritmos com o objetivo de tornar a otimização de treliças mais próxima da prática(29), resultados satisfatórios só foram obtidos nos anos 90, quando foram realizados os trabalhos pioneiros na otimização de treliças com o algoritmo genético (*GA*). O *GA* é um algoritmo metaheurístico evolutivo baseado em populações que busca solucionar um problema formulado como um problema de otimização através de uma analogia com o processo de evolução de uma população(14). Tendo seu funcionamento teorizado por Holland(30), em 1975, o *GA* começou a se tornar prático à medida que as universidades foram tendo acesso a mais capacidade computacional. Alguns anos depois, um aluno de Holland, Bethke(31) aprofundou os princípios do que viria a ser o *GA*

e os seus mecanismos. Um dos primeiros trabalhos a usar o *GA* para a otimização estrutural foi o de Goldberg e Samtani(32) em 1986, no qual foram otimizadas as dimensões da treliça de 10 barras. Jenkins(33) em 1991 utilizou o *GA* para encontrar as áreas ótimas para uma treliça do tipo viga para vencer um grande vão de telhado. Rajeev e Krishnamoorthy(34), no ano seguinte, realizaram também apenas otimização dimensional mas restringiram as áreas a um conjunto de possibilidades, o que caracterizaria um problema de variáveis discretas. Hajela, Lee e Lin(35), em 1993, realizaram otimização topológica e dimensional de treliças em duas etapas distintas, escolhendo um conjunto finito de topologias na primeira etapa e as respectivas áreas ótimas na segunda. Rajan(36), no entanto, foi pioneiro ao trabalhar com o *GA* para resolver o problema de otimização de treliças com variáveis dimensionais, topológicas e de forma simultaneamente, em 1995. Rajan realizou a otimização topológica considerando uma variável lógica pra cada elemento da treliça.

Nas últimas décadas os algoritmos metaheurísticos vêm se consolidando como uma alternativa na otimização estrutural, conforme cada vez mais trabalhos são publicados referentes a este assunto. A aplicação dos metaheurísticos se justifica quando problemas reais de engenharia estrutural apresentam não-convexidade e variáveis discretas ou mistas, como por exemplo na otimização de treliças e compósitos laminados(37, 38, 39).

Recentemente, muitos autores tem publicado trabalhos explorando hibridizações de metaheurísticos. Kaveh e Zolghadr(40), por exemplo, desenvolveram um algoritmo híbrido que usa uma busca por sistema carregado (*Charged System Search*), desenvolvido por eles, para evitar que o *Big Bang Big Crush* fique preso em mínimos locais. Xiao *et al*(41) desenvolveram um algoritmo chamado adequação adaptativa (*Adaptative Fitness*) que controla como se aplicam as restrições à função objetivo de cada partícula do otimizador por enxame de partículas (*Particle Swarm Optimization* ou *PSO*), conseguindo uma melhora significativa na convergência. Kaveh e Javadi(16) implementaram um

algoritmo utilizando a busca harmônica (*Harmony Search*) e um raio otimizador (*Ray optimizer*) desenvolvido por eles para melhorar o desempenho de um *PSO*. Torii, Lopez e Luersen(18) empregaram um método de recomeço probabilístico combinado a uma busca local com programação quadrática sequencial para a otimização simultânea dimensional, de forma e topológica de treliças com variáveis contínuas. Este método de recomeço combinado usa uma função de densidade probabilística construída sobre uma memória das execuções anteriores para recomeçar a busca local em uma região ainda não explorada, visando estudar efetivamente o espaço de busca. Miguel *et al* implementaram o método proposto por Pincus(42) para a proposição dos pontos de partida a partir dos quais um método determinístico do tipo Nelder-Mead realiza uma busca de natureza local(20). Este algoritmo híbrido foi desenvolvido para a identificação de falhas em estruturas baseado em dados experimentais obtidos a partir de excitações naturais. O algoritmo proposto se mostrou eficiente na busca por falhas superando o desempenho de metaheurísticos estudados em outras publicações. Nhamage(19), em sua dissertação de mestrado, aperfeiçoou o algoritmo desenvolvido por Miguel *et al* partindo do pressuposto inicial de que nenhum elemento se encontra danificado. A alteração proposta por Nhamage se provou eficiente melhorando a convergência do Pincus-Nelder-Mead.

Alguns autores tem trabalhado com a implementação de gradientes aproximados para acelerar a convergência de algoritmos evolutivos, como Pholdee e Bureerat(17), Pholdee e Bureerat(43) e Lamberti e Papalitere(44). Nos casos citados um pseudo-gradiente foi obtido a partir dos valores da função objetivo e das posições de alguns pontos próximos. Nestes casos o gradiente foi empregado em todas as variáveis; diferente do presente trabalho no qual apenas as variáveis de forma, as únicas contínuas, são alteradas na direção do gradiente. Pholdee e Bureerat(17) trabalharam com algoritmos evolutivos multi objetivo, buscando minimizar simultaneamente a massa e a flexibilidade. Após estudar alguns tipos de algoritmos evolutivos multi objetivo em conjunto com os gradientes aproximados, concluíram que o gradiente melhorava significativamente

a aproximação da fronteira *Pareto*. Pholdee e Bureerat(43) aplicaram uma busca semelhante mas a um algoritmo de aprendizado incremental baseado em populações. Nestes algoritmos cria-se uma distribuição de probabilidades para cada região do espaço e atualiza-se essa matriz de probabilidades conforme se executa a função objetivo nos pontos distribuídos. Lamberti e Pappalettere(44) trabalharam com a busca harmônica (*HS*) e com o *big bang big crush* aplicando pseudo gradientes semelhantes aos de Pholdee, porém realizando a busca unidimensional no sentido oposto ao do gradiente obtido com o arrefecimento simulado.

O operador proposto no presente trabalho usa gradientes numéricos para acelerar a convergência de metaheurísticos para mínimos locais, entretanto apenas na otimização de treliças e manipulando unicamente as variáveis de forma, as únicas contínuas. A idéia de implementar novos operadores a metaheurísticos não é inovadora. Li, Yang e Korejo(45) e Masrom *et al.*(46), por exemplo, implementaram operadores do *GA* adaptativos no *PSO*, cujos parâmetros variam ao longo da execução dependendo de como o algoritmo se comporta, obtendo bons resultados. Alguns autores buscam melhorar os metaheurísticos modificando os próprios operadores destes, como Kaya *et al.*(47) que modificou o operador *cross-over* do *GA* criando o *ring cross-over* com a intenção dar mais variabilidade ao processo e, conseqüentemente, pesquisar mais eficientemente o espaço de busca. Kanwal, Ramesh e Huang(48) desenvolveram um operador mutação que avalia as informações das últimas iterações através de um pseudogradiante com a intenção de adaptar os próprios parâmetros do operador mutação. Assim, conforme um ponto qualquer se aproxima de um mínimo local, aumenta-se o fator de mutação, evitando que o algoritmo fique preso em mínimos locais.

Avaliando-se o bom desempenho dos algoritmos híbridos na literatura, observa-se que há nos metaheurísticos espaço para melhora, o que justifica o desenvolvimento de novos operadores para estes. O presente trabalho é pioneiro por desenvolver e implementar um operador baseado em uma análise de sensibilidade com baixo custo computacional à algoritmos metaheurísticos de população com o objetivo de acelerar a

convergência destes na otimização de treliças.

2.2 Formulação matemática

Convencionou-se formalizar problemas de otimização como a busca por mínimos de funções. A maioria dos pesquisadores chama estas funções de função objetivo(f), sendo estas construídas como função das variáveis do problema(49). Quando se consideram problemas de otimização restritos, incluem-se na formulação equações de restrição de igualdade(\mathbf{h}) e desigualdade(\mathbf{g}). A formulação completa de um problema de otimização com restrições é apresentada nas Equações 2.1.

$$\text{Encontrar } \mathbf{x} = [x_1, x_2, \dots, x_j] \quad (2.1a)$$

$$\text{Que minimiza } f(\mathbf{x}) \quad (2.1b)$$

$$\text{Sujeita à } \mathbf{h}(\mathbf{x}) = 0 \quad (2.1c)$$

$$\text{Sujeita à } \mathbf{g}(\mathbf{x}) \leq 0 \quad (2.1d)$$

para um problema de j variáveis.

2.3 Otimização estrutural de treliças

O problema de otimização de treliças é formulado no presente trabalho como a busca pela estrutura mais leve. Desta forma, a função objetivo adotada é o peso da treliça, sendo as áreas das seções transversais e posições dos nós as variáveis de projeto sujeitas à restrições de deslocamento, tensão (o tratamento das restrições é discutido na Subseção 2.3.4).

2.3.1 Otimização dimensional

Otimização dimensional é a busca pelas dimensões de uma estrutura que resultem no melhor desempenho desta, sendo o desempenho avaliado pela função objetivo. No caso de treliças a otimização dimensional é geralmente a busca pelas áreas das seções transversais que melhor se adequem à topologia inicial. A formulação matemática do problema de otimização dimensional é apresentada nas Equações 2.2.

$$\text{Encontrar } \mathbf{x} = [A_1, A_2, \dots, A_m] \quad (2.2a)$$

$$\text{Que minimiza } P(\mathbf{x}) = \sum_{i=1}^m \gamma_i l_i A_i \quad (2.2b)$$

$$\text{Sujeita à } \mathbf{g}(\mathbf{x}) \leq 0 \quad (2.2c)$$

onde:

\mathbf{x} - o vetor contendo as variáveis do problema, no caso as áreas das seções transversais;

m - o número de elementos da treliça;

A_i - a área da seção transversal do i -ésimo elemento;

P - o peso da treliça inteira. É a função objetivo que se busca minimizar;

γ_i - o peso específico do i -ésimo elemento;

l_i - o comprimento do i -ésimo elemento;

\mathbf{g} - o vetor de restrições de desigualdade do problema.

Como na presente modelagem do problema da otimização de treliças não são consideradas restrições de igualdade \mathbf{h} , estas serão suprimidas nas fórmulas.

É provado na literatura que a otimização dimensional com variáveis contínuas pode ser resolvida satisfatoriamente com métodos matemáticos por ser um problema convexo(15, 50). A aplicação do *ICA* para a solução do problema de otimização dimensional já foi explorada por alguns pesquisadores como Talahari *et al*(51), Kaveh(52) e Sabour *et al*(53).

2.3.2 Otimização de forma

Otimização de forma é a busca pela geometria da estrutura que consiga transferir as solicitações da maneira mais eficiente aos seus apoios. A eficiência das treliças é medida através da função objetivo que é usualmente o peso próprio da estrutura. Neste trabalho a otimização de forma é realizada tratando-se algumas coordenadas de alguns nós como variáveis (ξ_i), assim, cabe ao otimizador encontrar as posições dos nós que resultem no menor peso próprio. A otimização de k variáveis de forma é apresentada nas Equações 2.3.

$$\text{Encontrar } \mathbf{x} = [\xi_1, \xi_2, \dots, \xi_k] \quad (2.3a)$$

$$\text{Que minimiza } P(\mathbf{x}) = \sum_{i=1}^m \gamma_i l_i(\mathbf{x}) A_i \quad (2.3b)$$

$$\text{Sujeita à } \mathbf{g}(\mathbf{x}) \leq 0 \quad (2.3c)$$

onde ξ_k é a posição do k -ésimo grau de liberdade definido como variável.

2.3.3 Otimização topológica

Otimização topológica é a busca pela estrutura mais eficiente através de mudanças na topologia da estrutura. No presente trabalho, a otimização topológica se resume à retirada de elementos e nós do sistema. Para tal, adiciona-se uma porcentagem de zeros no possível conjunto discreto de áreas a ser escolhidas na otimização dimensional. Desta

forma, caso algum elemento tenha área zero ele é eliminado do sistema. Esta abordagem é possível pois as áreas das seções transversais são tratadas como variáveis discretas. Caso, durante este processo, algum nó não esteja conectado a nenhum elemento, este também é retirado do sistema reduzindo as dimensões do sistema e evitando a singularidade da matriz de rigidez. O problema de otimização simultânea dimensional, de forma e topológica é formulado matematicamente como apresentado nas Equações 2.4.

$$\text{Encontrar } \mathbf{x} = [A_1, A_2, \dots, A_m, \xi_1, \xi_2, \dots, \xi_k] \quad (2.4a)$$

$$\text{Que minimiza } P(\mathbf{x}) = \sum_{i=1}^m \gamma_i l_i(\mathbf{x}) A_i \quad (2.4b)$$

$$\text{Sujeita à } \mathbf{g}(\mathbf{x}) \leq 0 \quad (2.4c)$$

2.3.4 Restrições

Foram impostas restrições específicas para cada problema, sendo estas principalmente valores máximos para tensões e deslocamentos. A penalização por restrições violadas é aplicada através de uma função de penalização τ que multiplica a função objetivo. Caso nenhuma restrição seja violada esta função retorna o valor $\tau = 0$. Todas as restrições são normalizadas na forma $\mathbf{g} = \sqrt{S_{encontrada}/S_{admissivel}} - 1$ sendo S qualquer valor que tenha um máximo admissível. A raiz quadrada é aplicada para reforçar penalizações de ordem pequena, como 10^{-6} , cuja raiz quadrada seria 10^{-3} . A função penalizada composta como uma combinação da função objetivo \mathbf{P} e da função de penalização τ para as violações de restrições foi definida como apresentado na Equação 2.5a.

$$\mathbf{f}_{penalizada}(\mathbf{x}) = \mathbf{P}(\mathbf{x})(1 + C \tau(\mathbf{x})) \quad (2.5a)$$

$$\tau(\mathbf{x}) = \sum \max(\mathbf{g}(\mathbf{x}), 0) \quad (2.5b)$$

onde C é um fator arbitrado a ser calibrado de acordo com o problema. Caso C seja muito pequeno o algoritmo pode encontrar soluções com violações de restrição, por outro lado, caso seja muito grande pode prejudicar a busca na região onde as restrições se encontram ativas. No presente trabalho um valor de 15, obtido empiricamente, foi adotado para C em todos os problemas.

A estaticidade de cada treliça é verificada através do número de condicionamento recíproco (comando `rcond` na linguagem MATLAB) da matriz de rigidez. Se o número de condicionamento recíproco da matriz for menor do que a precisão relativa de *floating-point* a matriz é mal condicionada e existem autovalores de ordens de grandeza muito diferentes. Como no presente trabalho o método dos deslocamentos é aplicado, a equação $\mathbf{F}=\mathbf{K}\mathbf{u}$ é empregada na análise estrutural, sendo \mathbf{F} o vetor de forças, \mathbf{K} a matriz de rigidez e \mathbf{u} o vetor de deslocamentos nodais. Caso se conclua que a matriz de rigidez \mathbf{K} é mal condicionada, uma pequena perturbação no vetor de forças \mathbf{F} causaria uma grande variação no vetor de deslocamentos \mathbf{u} e a matriz \mathbf{K} se encontraria muito próxima da singularidade(54). Nestes casos um peso arbitrado de 10^9lb é definido como o peso da treliça, garantindo-se que para cada problema este valor seja sempre maior do que outros valores iso ou hiperestáticos.

2.4 Otimização global/local

2.4.1 Otimização global

A exploração do espaço de busca por regiões promissoras é uma das dificuldades encontradas nos métodos matemáticos. Já os algoritmos metaheurísticos, usualmente algoritmos de população, conseguem uma convergência aceitável mesmo para problemas não-convexos, dada a capacidade computacional necessária. No presente trabalho, o algoritmo de competição imperialista (*ICA*), desenvolvido por Atashpaz-Gargari e Lucas(55), foi aplicado com a finalidade de realizar a busca global, ou seja, explorar o espaço de busca por regiões onde se espera encontrar o mínimo global. O funcionamento do *ICA* é explicado no capítulo 3.

2.4.2 Otimização local

Como os metaheurísticos, em geral, não fazem uso de derivadas da função objetivo, estes requerem um alto custo computacional para convergir para mínimos locais(14). Para acelerar a convergência do *ICA* propõe-se aqui um operador de forma. Este operador busca melhorar a geometria da treliça estudada dando passos nas variáveis de forma no sentido oposto ao do gradiente de forma. Este gradiente é um vetor no espaço k -dimensional formado pelas k variáveis de forma e aponta para a direção aonde o trabalho realizado pela treliça tem maior acréscimo, como empregado por Torii e Biondini(56). Como o subespaço de busca das variáveis de forma é contínuo, uma análise de sensibilidade empregando um gradiente obtido por diferenças finitas é possível. Estas equações são obtidas a partir da manipulação de equações apresentadas por Bendsoe e Sigmund(57) para a otimização estrutural através da minimização da compliância. Embora Bendsoe e Sigmund tenham formulado o método para a otimização de estruturas contínuas discretizadas em um grande número de elementos finitos, situação próxima da estrutura contínua, no presente trabalho emprega-se o mesmo na análise de sensibilidade de treliças modeladas como nós e um número pequeno de elementos unidimensionais. Como provado por Hegemier e Prager(2), estudando as equações de Michell, uma treliça de Michell ótima com infinitos elementos e $\sigma = \sigma_{crit}$ em todo o seu espaço, é mais rígida do que qualquer outra treliça com o mesmo volume de material. Embora não se possa afirmar categoricamente que esta afirmação seja válida para o caso estudado no presente trabalho, espera-se que a treliça ótima que se busca encontrar esteja próxima de ser a mais rígida com o material empregado. Assim, buscando a treliça mais rígida se estaria convergindo para o ótimo global. O menor custo computacional justifica o uso do gradiente do trabalho ao invés de empregar o gradiente da função objetivo.

3 *Algoritmo de competição imperialista*

O Algoritmo de competição imperialista (*ICA*) é um algoritmo metaheurístico desenvolvido por Atashpaz-Gargari e Lucas(55) para solucionar problemas de otimização através de uma analogia com países em uma competição imperialista. Esta analogia é feita associando-se o valor da função objetivo em cada ponto ao poder do país que se localiza no ponto estudado de tal maneira que um maior poder é associado à um menor valor da função objetivo. Os países então competem entre si a cada década até o critério de parada ser atingido. No fim da execução o país mais forte corresponde à melhor solução encontrada e a sua posição representa o vetor de solução do problema proposto. Um organograma do *ICA* é apresentado na Figura 4 e o pseudocódigo do mesmo na Figura 5.

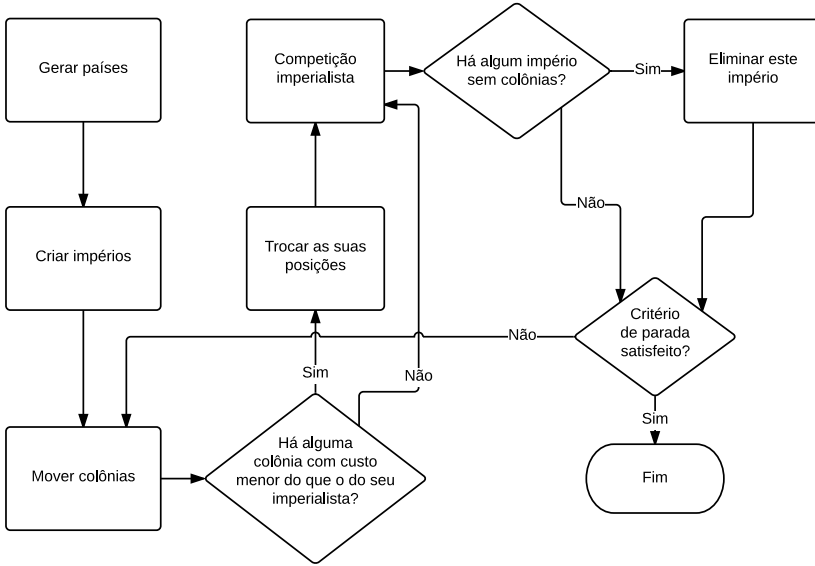


Figura 4 – Organograma do ICA.

Gerar países

Nesta etapa do processo de otimização o ICA distribui aleatoriamente um determinado número de pontos no espaço de busca restrito pelas restrições laterais, ou seja, os limites máximos e mínimos de cada variável. Cada ponto no espaço j -dimensional formado pelas j variáveis será considerado um país. No caso do presente trabalho cada ponto no espaço representa uma treliça diferente.

Criar impérios

Após os países serem distribuídos no espaço a função objetivo é avaliada para cada um destes, sendo o poder de cada país associado ao seu valor da função objetivo. Os países mais poderosos, então, serão aqueles cujas treliças que estes representam sejam as mais leves e que não violam nenhuma restrição. Um certo número dos países mais poderosos é convertido em imperialistas e todos os outros em colônias. As colônias são divididas aleatoriamente entre os imperialistas em número proporcional

-
1. Selecione um número aleatório de pontos da função objetivo e inicialize os impérios.
 2. Mova as colônias na direção dos seus imperialistas.
 3. Se alguma colônia de algum império tiver um valor menor do que o do seu imperialista, troque a posição destes.
 4. Compute o custo total de todos os impérios (Relacionado ao poder tanto dos imperialistas quanto das suas colônias).
 5. Pegue a colônia mais fraca do império mais fraco e a dê para algum dos impérios mais fortes.
 6. Elimine os impérios sem colônias.
 7. Se o critério de parada for satisfeito, pare. Se não, volte a mover as colônias.
-

Figura 5 – Pseudocódigo do Algoritmo de Competição Imperialista.

ao poder de cada um formando os impérios. A representação desta etapa é apresentada na Figura 6 onde a função objetivo é representada em escala crescente de azul à amarelo e cada império com uma cor diferente.

Mover colônias

Sendo os imperialistas mais poderosos do que as colônias, se objetiva aproximar as colônias dos seus imperialistas, explorando o espaço de busca através de um movimento estocástico. Sendo a posição de cada país no espaço j -dimensional a composição das j variáveis do problema, quando uma colônia se aproxima do imperialista, a treliça representada por esta se transforma buscando se tornar mais próxima da treliça representada pelo seu imperialista. O movimento da colônia em direção ao seu imperialista tem natureza estocástica, sendo desviado da direção do imperialista em um ângulo $\theta \sim U(-\eta, \eta)$ sendo η um parâmetro do algoritmo chamado ângulo de assimilação, não maior

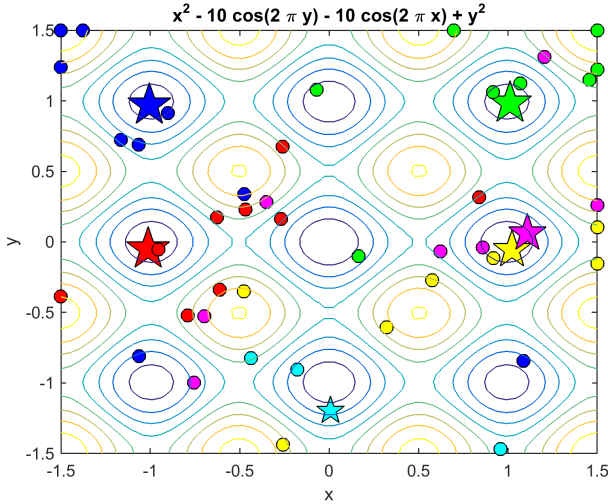


Figura 6 – Geração dos impérios.

do que 90° . De forma semelhante o tamanho do passo da colônia na direção desviada é também função de uma variável estocástica dada por $s \sim U(0, \beta \times d)$ sendo d a distância entre a colônia e o imperialista e β um parâmetro do algoritmo chamado de coeficiente de assimilação. O movimento de uma colônia em relação ao do seu imperialista é representado na Figura 7.

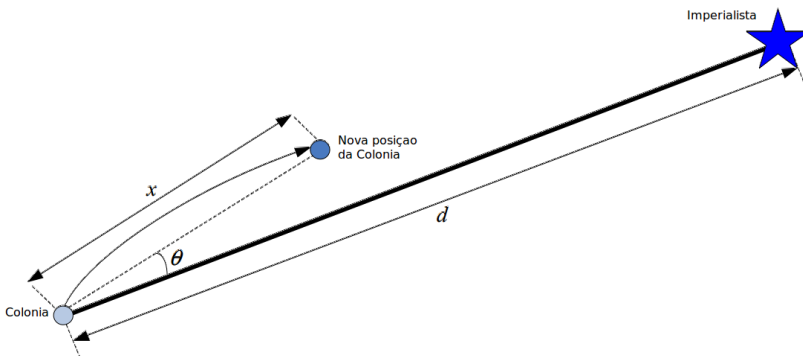


Figura 7 – Representação do movimento de uma colônia.
Fonte: Atashpaz-Gargari e Lucas(55).

Se neste processo alguma colônia se mover para um ponto onde a treliça seja mais leve do que a da sua imperialista e no qual nenhuma restrição (tensão, deformação, estaticidade) esteja violada, esta se torna o novo imperialista e o imperialista se torna uma colônia, como representado na Figura 8.

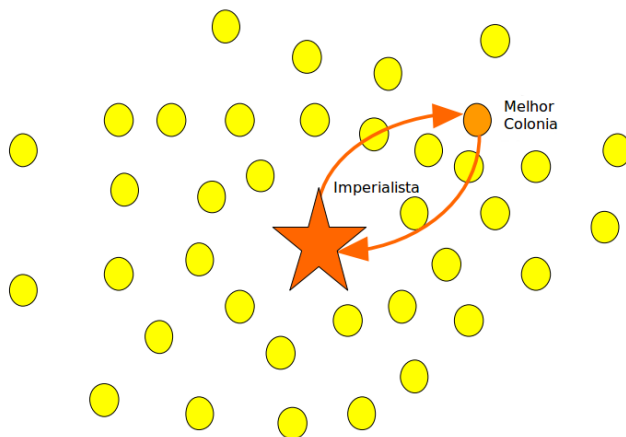


Figura 8 – Colônia dominando imperialista. Fonte: Atashpaz-Gargari e Lucas(55).

Competição Imperialista

Durante a competição imperialista o império mais fraco perde uma colônia para algum dos outros impérios e passará a partir de então a se mover na direção destes. Caso algum império fique sem colônias este é desfeito e o seu imperialista é também colonizado por algum dos outros impérios. A potência de cada império é definida como a potência do seu imperialista somada à uma fração pequena da soma da potência de cada uma das suas colônias.

Revolução

Uma certa porcentagem das colônias mais fracas de cada império tem a sua posição redistribuída aleatoriamente no espaço de busca a

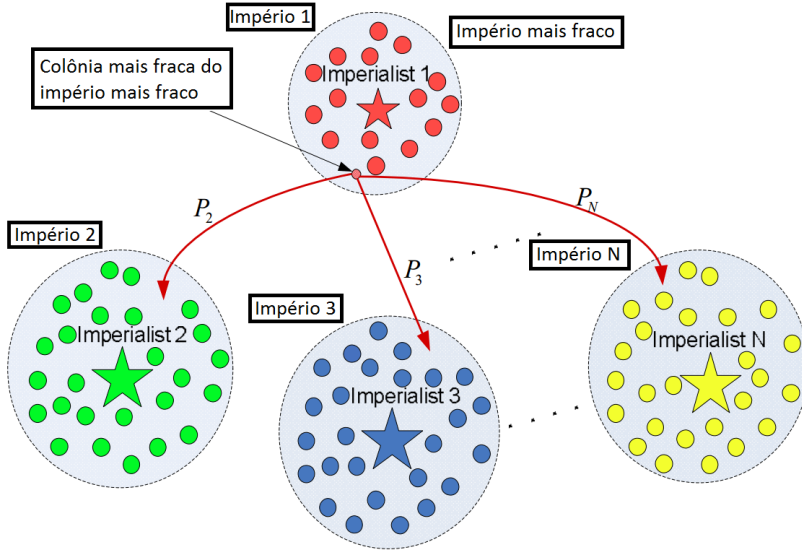


Figura 9 – Impérios competindo por colônias. Fonte:Atashpaz-Gargari e Lucas(55).

cada iteração, com o objetivo de explorar melhor o espaço de busca. Esta porcentagem é definida como parâmetro do algoritmo e decresce ao longo do desenvolvimento do algoritmo, sendo a cada iteração multiplicada por uma constante menor que um. A revolução na *ICA* emula uma mudança brusca na cultura de um país, representada matematicamente pela sua mudança de posição no espaço.

Critério de parada

O critério de parada pode ser escolhido de acordo com as necessidades. No presente trabalho um limite para o número de avaliações da função objetivo (*OFE*) foi estabelecido para cada problema. Caso o critério de parada não seja satisfeito o algoritmo volta a mover as colônias e a seguir a rotina do algoritmo.

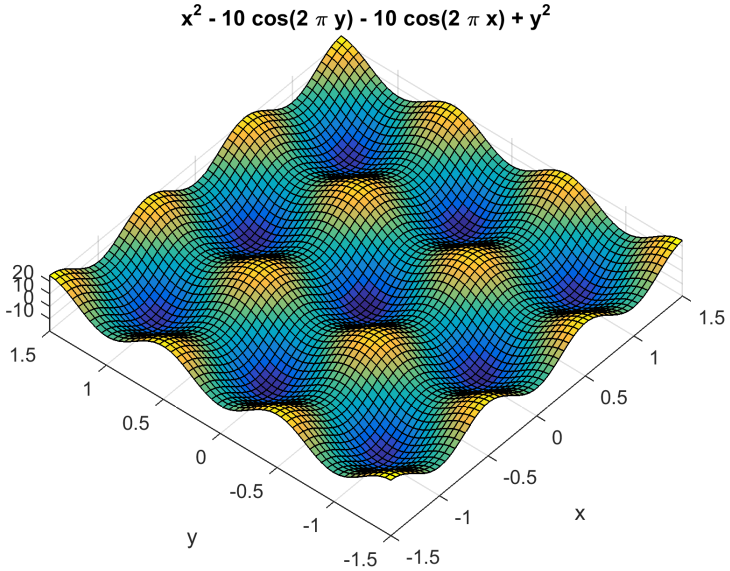


Figura 10 – Função clássica estudada.

3.1 Exemplo numérico

Para ilustrar o funcionamento do *ICA* um exemplo numérico será apresentado no qual se buscará o mínimo da função $F(x, y) = x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y)$ que apresenta vários mínimos locais e tem o seu mínimo global na origem, de valor -20. Foram utilizados para o exemplo 50 países, 6 imperialistas, e 30 décadas e o espaço de busca foi limitado em x e y de -1,5 a 1,5, conforme ilustrado na Figura 10. Na primeira década de busca o *ICA* encontrou o melhor valor de -18,45, no ponto $(-1,02 ; -0,05)$, representado na Figura 11a; através de uma estrela vermelha. Na segunda década pode-se observar que houve movimentação das colônias e que dois imperialistas melhoraram as suas posições, representados na Figura 11b em ciano e azul. Na década 5, apresentada na Figura 11c, o melhor imperialista melhorou a sua posição para -18,87 em $(-0,98 ; 0,02)$. Na década 9 houve uma troca de posições do melhor imperialista que saltou da região do mínimo local $(-1; 0)$

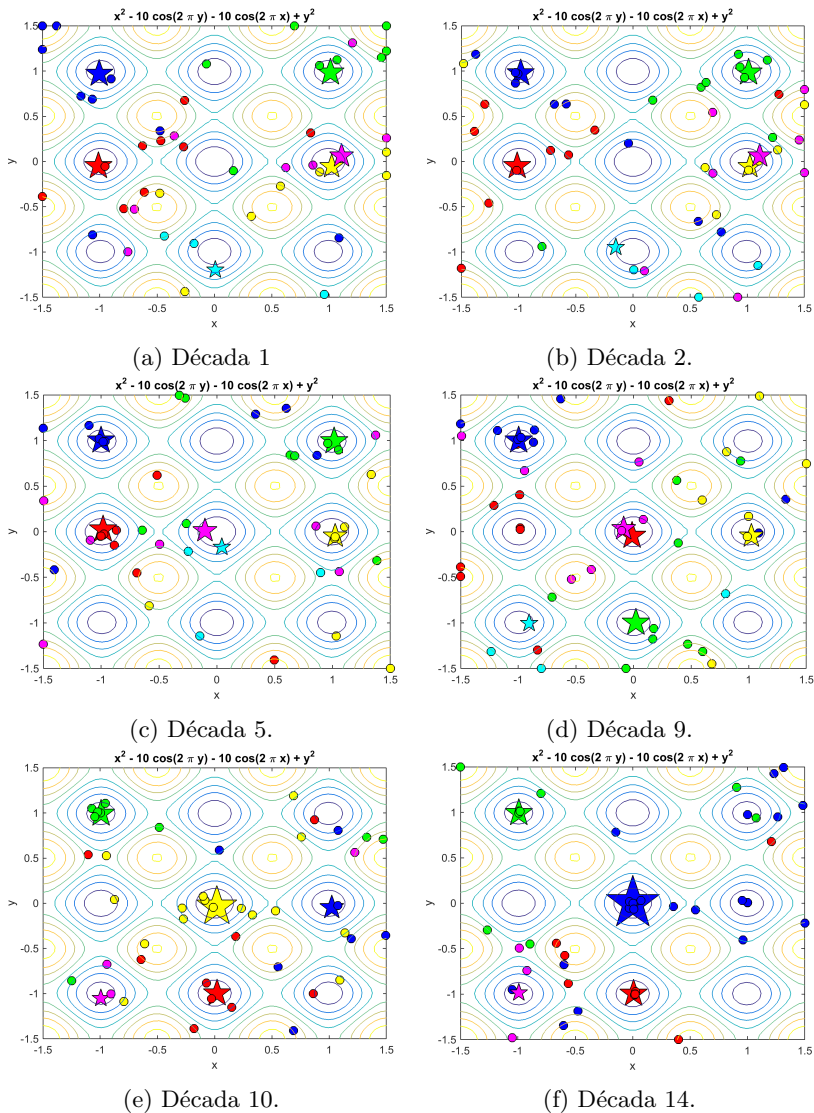


Figura 11 – Exemplo do ICA para um problema clássico.

para a região do mínimo global na origem, como pode ser observado na Figura 11d. Já na década seguinte uma colônia amarela encontrou o ponto $(0,02 ; -0,04)$ de valor $-19,65$, representado na Figura 11e por uma estrela amarela por ter se tornado imperialista. O império amarelo se tornou nesta década o mais poderoso, de forma que, a partir deste momento, os outros impérios tenderão a perder colônias para o amarelo ao invés do vermelho. Percebe-se que os imperialistas tendem a se posicionar em mínimos locais e que o império ciano foi eliminado e suas colônias absorvidas pelos outros impérios. Na década 14 o império azul se fundiu com o amarelo tornando-se o império mais poderoso. Nesta década o imperialista já se encontra na posição $(0,00041; 0,00041)$, com valor de $-19,9999$, bastante próximo do ótimo global.

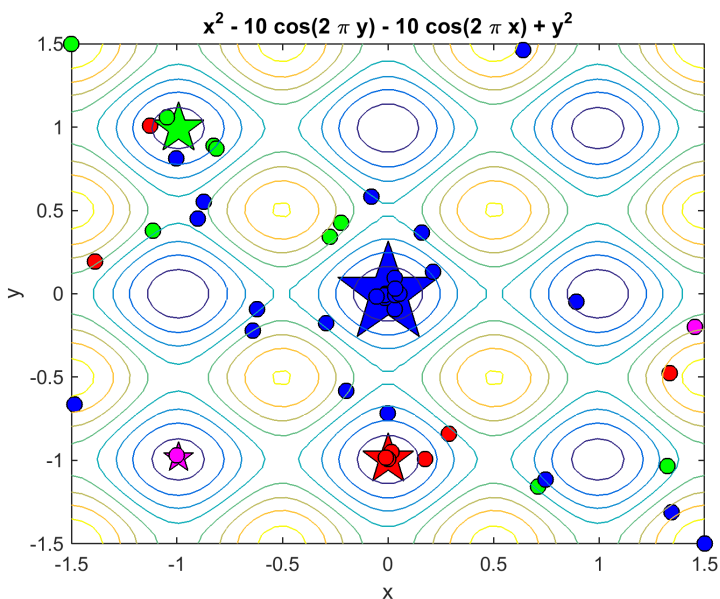


Figura 12 – Configuração final do exemplo do ICA para um problema clássico - Década 20.

Ao fim das 20 décadas, os imperialistas e colônias se encontravam como apresentado na Figura 12. O imperialista azul, o melhor

imperialista, encontrou o ponto (0,00039; 0,0004) de valor -19,9999.

4 Operador de forma

O Operador de Forma é chamado pelo algoritmo metaheurístico, tem como entrada um vetor posição no espaço de busca e retorna outro ao final do processo. No presente trabalho, o *ICA* tem uma probabilidade de chamar o operador a cada década, chamando o operador para avaliar apenas os imperialistas. Então, o operador tem como entrada a posição de um imperialista e, manipulando apenas as variáveis de forma, retorna ao *ICA* uma posição referente a uma estrutura mais rígida. Caso o operador não encontre nenhuma configuração geométrica mais rígida, este retorna a posição inicial ao *ICA*. A Figura 13 ilustra o funcionamento do *ICA* com o operador e quando que este é chamado. A probabilidade de o operador ser chamado para cada imperialista varia com as décadas do

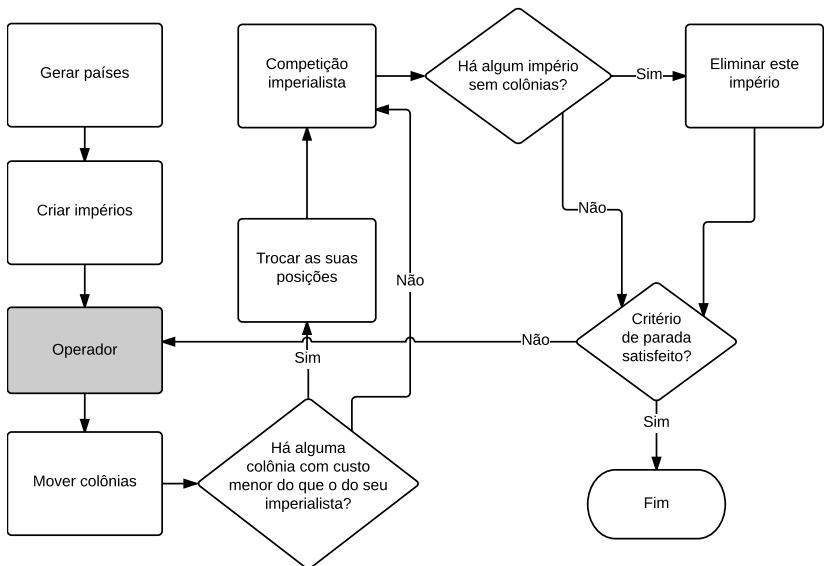


Figura 13 – Organograma do Algoritmo de Competição Imperialista (*ICA*) com o Operador de Forma (*OF*).

ICA. O passo das variáveis de forma é dado no sentido oposto ao do gradiente de forma, obtido a partir das derivadas parciais do trabalho que as forças externas exercem sobre a treliça em relação a cada uma das variáveis de forma. O trabalho que a treliça sofre em função das variáveis é apresentado na Equação 4.1, notando-se que é admitido que as forças aplicadas nos nós não dependem das variáveis de projeto.

$$W(\mathbf{x}_{forma}) = \mathbf{u}^T(\mathbf{x}_{forma}) \cdot \mathbf{F} \quad (4.1)$$

sendo W o trabalho realizado pelas forças externas, \mathbf{u} o vetor de deslocamentos nodais e \mathbf{F} o vetor de forças.

Sabendo-se que o problema da treliça de menor volume é análogo ao problema da treliça mais rígida(2), este pode ser reformulado de forma a se substituir o peso na função objetivo pelo trabalho sofrido pela treliça. A validade desta analogia, entretanto, depende de algumas condições, como o problema apresentar um único caso de carregamento, não serem consideradas instabilidades e estar próximo da situação na qual todas os elementos estão no limite da tensão admissível (*Fully Stressed Design*).

A Equação 2.4 pode ser reescrita, desta vez em função do trabalho, como apresentado nas Equações 4.2.

Encontrar $\mathbf{x}_{forma} = [\xi_1, \xi_2, \dots, \xi_k]$ para

$$\mathbf{x}_{dimensional} = [A_1, A_2, \dots, A_m] \text{ fixo} \quad (4.2a)$$

$$\text{Que minimiza } W(\mathbf{x}_{forma}) = \mathbf{u}^T(\mathbf{x}_{forma}) \cdot \mathbf{F} \quad (4.2b)$$

$$\text{Sujeita à } \mathbf{g}(\mathbf{x}_{forma}) \leq 0 \quad (4.2c)$$

O operador de forma busca minimizar o trabalho absorvido pela treliça dando um passo arbitrário no sentido oposto ao do gradiente $\partial W / \partial \mathbf{x}_{forma}$ com as variáveis de forma.

Como o método dos deslocamentos é empregado, a equação $\mathbf{F} = \mathbf{K} \cdot \mathbf{u}$, que relaciona forças e deslocamentos através de uma matriz de rigidez, é utilizada para a análise estrutural. Substituindo $\mathbf{F} = \mathbf{K} \cdot \mathbf{u}$ na Equação 4.1 tem-se a Equação 4.3.

$$W(\mathbf{x}_{forma}) = \mathbf{u}^T(\mathbf{x}_{forma}) \cdot \mathbf{K}(\mathbf{x}_{forma}) \cdot \mathbf{u}(\mathbf{x}_{forma}) \quad (4.3)$$

Calculando as derivadas parciais de ambos os lados em relação à i -ésima variável de forma se obtém a Equação 4.4.

$$\frac{\partial W}{\partial \xi_i} = \frac{\partial \mathbf{u}^T}{\partial \xi_i} \cdot \mathbf{K} \cdot \mathbf{u} + \mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} + \mathbf{u}^T \cdot \mathbf{K} \cdot \frac{\partial \mathbf{u}}{\partial \xi_i} \quad (4.4)$$

Das propriedades de transposição de multiplicações entre matrizes e vetores se tem que $\mathbf{a}^T \cdot \mathbf{B} \cdot \mathbf{c} = \mathbf{c}^T \cdot \mathbf{B}^T \cdot \mathbf{a}$ para qualquer par de vetores \mathbf{a} e \mathbf{c} e qualquer matriz \mathbf{B} cujas dimensões sejam compatíveis. Pode-se então notar que o primeiro termo à direita da igualdade da Equação 4.4 ($\frac{\partial \mathbf{u}^T}{\partial \xi_i} \cdot \mathbf{K} \cdot \mathbf{u}$) é idêntico ao terceiro ($\mathbf{u}^T \cdot \mathbf{K} \cdot \frac{\partial \mathbf{u}}{\partial \xi_i}$), visto que \mathbf{K} é simétrica ($\mathbf{K}^T = \mathbf{K}$), e que portanto a Equação 4.4 pode ser reescrita como a Equação 4.5.

$$\frac{\partial W}{\partial \xi_i} = 2 \cdot \mathbf{u}^T \cdot \mathbf{K} \cdot \frac{\partial \mathbf{u}}{\partial \xi_i} + \mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} \quad (4.5)$$

Diferenciando $\mathbf{F} = \mathbf{K} \cdot \mathbf{u}$ em ξ_i e lembrando que considerou-se \mathbf{F} independente de \mathbf{x} , temos a Equação 4.6a e seu desenvolvimento na Equação 4.6b:

$$0 = \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} + \mathbf{K} \cdot \frac{\partial \mathbf{u}}{\partial \xi_i} \quad (4.6a)$$

$$\mathbf{K} \cdot \frac{\partial \mathbf{u}}{\partial \xi_i} = - \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} \quad (4.6b)$$

Pode-se notar, observando a Equação 4.6b, que diferenciar o deslocamento nodal em relação às variáveis de forma é bem mais custoso computacionalmente do que diferenciar a matriz de rigidez, visto que a derivada parcial por diferenças finitas exigiria $k + 1$ inversões da matriz de rigidez para k variáveis de forma. Já a derivada parcial numérica da matriz de rigidez em relação às variáveis de forma demanda menor custo computacional para ser calculada, exigindo apenas uma inversão da matriz de rigidez e k montagens da mesma. Substituindo a Equação 4.6b na Equação 4.5 obtêm-se a Equação 4.7.

$$\frac{\partial W}{\partial \xi_i} = -2 \cdot \mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} + \mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} \quad (4.7)$$

E consequentemente:

$$\frac{\partial W}{\partial \xi_i} = -\mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u} \quad (4.8)$$

O vetor formado por $\partial \mathbf{K} / \partial \xi_i$ é aproximado por diferenças finitas impondo-se uma perturbação em cada variável de forma isoladamente e avaliando-se a diferença na matriz de rigidez. Uma vez que a derivada parcial do trabalho em relação à cada variável de forma é obtida se pode montar o gradiente de forma.

$$\vec{\nabla}_{forma} = \frac{\partial W}{\partial \xi_i} = \left\{ \frac{\partial W}{\partial \xi_1}, \frac{\partial W}{\partial \xi_2}, \dots, \frac{\partial W}{\partial \xi_k} \right\} \quad (4.9)$$

sendo k o número de variáveis de forma.

A Equação 4.10 apresenta a atualização da posição do vetor de variáveis de forma de um imperialista dado um passo φ em sentido

oposto ao do gradiente de forma normalizado.

$$\xi_i^{dcada+1} = \xi_i^{dcada} - \varphi \left(\frac{\frac{\partial W}{\partial \xi_i^{dcada}}}{\left| \vec{\nabla}_{forma}^{dcada} \right|} \right) \quad (4.10)$$

Para a busca unidimensional na direção do gradiente, ou seja, para a definição do comprimento de passo φ , uma busca utilizando a seção áurea (*golden section search*) é empregada. No método, um pequeno passo inicial arbitrado é aplicado e o valor do trabalho e do peso penalizado da treliça são avaliados. Ambas as informações são utilizadas para avaliar se o passo é aceitável e se um passo maior deve ser dado, de modo que se pode reformular o problema de otimização na busca unidimensional como apresentado na Equação 4.11.

$$\text{Encontrar } \varphi \mid \mathbf{x}_{forma}(\varphi) = [\xi_1(\varphi), \xi_2(\varphi), \dots, \xi_k(\varphi)] \quad (4.11a)$$

$$\text{Que minimiza } Z(\varphi) = W(\varphi) \cdot f_{penalizada}(\varphi) \quad (4.11b)$$

$$\text{Sujeita à } \mathbf{g}(\varphi) \leq 0 \quad (4.11c)$$

$$\text{Sendo } \mathbf{W}(\varphi) = \mathbf{u}^T(\varphi) \cdot \mathbf{F} \quad (4.11d)$$

$$\text{e } f_{penalizada}(\varphi) = \sum_{i=1}^m \gamma_i l_i(\varphi) A_i(\varphi) (1 + C \tau(\varphi)) \quad (4.11e)$$

Como o problema da minimização da flexibilidade é análogo ao do menor peso para volumes mantidos constantes, se avalia a função objetivo penalizada a cada passo para garantir que o volume não esteja

crescendo muito e extrapolando o princípio mencionado. Assim, adotando o produto entre $f_{penalizada}$ e W como critério; garante-se, por exemplo, que um decréscimo de 10% no trabalho pelo último passo dado será aceito caso $f_{penalizada}$ tenha crescido menos de 10%. É importante notar que, considerando $f_{penalizada}$, já se está impondo as restrições. Então, caso um passo viole alguma restrição, a penalização de $f_{penalizada}$ levará o algoritmo de busca pela porção áurea a parar e escolher o passo anterior, não penalizado, como solução. Caso um ponto estudado seja melhor do que o valor anterior, o passo é multiplicado pela razão áurea (1,6180 ou $\frac{1+\sqrt{5}}{2}$) e o algoritmo segue sua rotina. O último valor que apresentou decréscimo de Z é admitido como a nova posição de forma \mathbf{x}_{forma} . É permitido que o valor de $f_{penalizada}$ aumente durante esta etapa pois espera-se que o *ICA* posteriormente encontre as áreas das seções transversais que, combinadas com \mathbf{x}_{forma} encontrado, reduzam $f_{penalizada}$. É interessante notar que a rotina do operador é executada com as variáveis dimensionais fixas e é dependente destas, de forma que a geometria encontrada é relativa a estas variáveis. Assim, conforme as áreas das seções transversais variam, modificam-se o gradiente e o mínimo encontrado.

Um organograma do funcionamento do operador é apresentado na Figura 14.

4.1 Exemplo do operador de forma sem violação de restrições

Um exemplo será apresentado com a finalidade de ilustrar o otimizador local implementado. Neste caso o *ICA* chamou o operador para otimizar a treliça apresentada na Figura 16. A treliça pesa 6.024,67lb e as suas áreas e posições dos nós são representadas na Tabela 1 em in^2 e in respectivamente.

Neste caso, denominado problema de 10 barras na literatura, as 3 variáveis de forma são as posições dos nós 2, 4 e 6 (ξ_{y2} ; ξ_{y4} ; ξ_{y6}) e há duas cargas de 10^5 lb na direção vertical e sentido para baixo nos nós 3 e 5. O

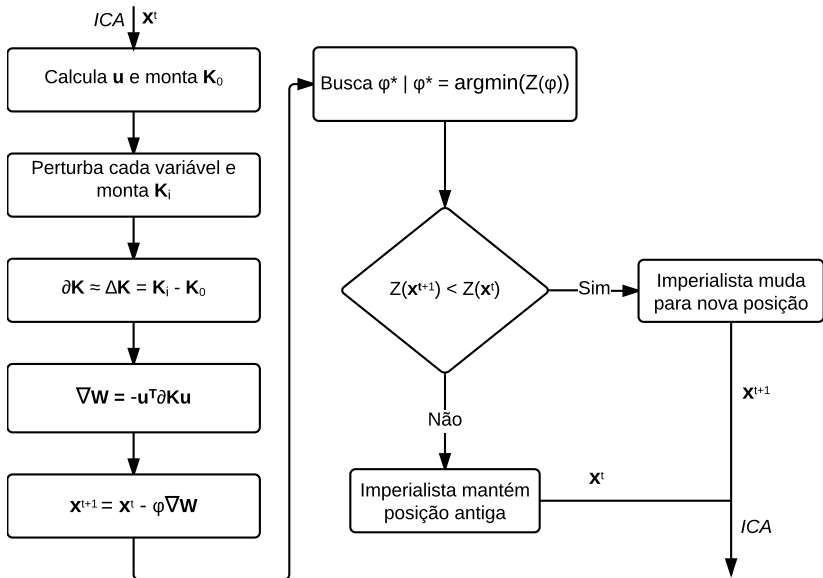


Figura 14 – Organograma do Operador de Forma onde t são as décadas.

gradiente de forma é obtido perturbando-se cada uma destas e avaliando a diferença no trabalho absorvido pela treliça sendo que o gradiente de forma normalizado encontrado foi $\partial W / \partial \xi_i = (-0,9619; 0,2736; 0,0025)$. É interessante notar que a sensibilidade do trabalho à perturbação é muito maior no nó 2 do que nos nós 4 e 6 e com sinais invertidos. Assim a busca de forma tenderá a mover as variáveis de forma no sentido oposto ao do gradiente, subindo o nó 2 e descendo os outros nós. As Figuras 18 e 19 ilustram o trabalho, valor de $f_{penalizada}$ e da função Z da treliça em relação ao passo na direção do gradiente de forma e é possível observar que estes não são convexos. Observando as Figuras 18 e 19 vê-se que estas convergem assintoticamente para o infinito em dois pontos. Estes pontos são pontos de singularidade da matriz de rigidez, onde o trabalho sofrido pela treliça tende ao infinito, sendo o da esquerda referente à um passo negativo e o da direita à um passo positivo. O pico à esquerda do centro representa a situação na qual $\xi_{y2} \approx 0$, ou seja, na qual o nó 2 está muito próximo do nó 1. Da mesma

1. A partir do ponto inicial \mathbf{x}^{dcada} , monte \mathbf{K}_0 e calcule \mathbf{u}_0 .
2. Aplique uma perturbação pequena $\Delta\xi$ individualmente em cada uma das i de k variáveis de forma (ξ_i) e monte k matrizes \mathbf{K}_i .
3. Monte $\frac{\partial \mathbf{K}}{\partial \xi_i} = \frac{\mathbf{K}_i - \mathbf{K}_0}{\Delta\xi}$.
4. Monte o vetor $\vec{\nabla}_{forma}$ com $\frac{\partial W}{\partial \xi_i} = -\mathbf{u}^T \cdot \frac{\partial \mathbf{K}}{\partial \xi_i} \cdot \mathbf{u}$
5. Encontre o comprimento de passo φ que minimize $Z(\mathbf{x}_{forma}(\varphi))$ para

$$\xi_i^{dcada+1} = \xi_i^{dcada} - \varphi \left(\frac{\frac{\partial W}{\partial \xi_i^{dcada}}}{|\vec{\nabla}_{forma}^{dcada}|} \right)$$

6. Compare $Z(\mathbf{x}^{dcada+1})$ e $Z(\mathbf{x}^{dcada})$ e escolha o vetor x que resulte no menor Z .
7. Retorne o valor \mathbf{x} final para o ICA.

Figura 15 – Pseudocódigo do Operador de Forma.

forma o pico à direita representa a situação na qual o nó 4 coincide com o nó 3. Caso se continuasse aumentando o passo positivamente, outro pico apareceria na região onde $\xi_{y6} \approx 0$. Como $\xi_{y6} = 178,5230in$ e $-\partial W/\partial \xi_{y6} = -0,0025$, um passo de $\varphi = 71.409in$ seria necessário.

A busca unidimensional na direção do gradiente de forma custou 19 avaliações da função objetivo sendo o 18º valor avaliado o melhor encontrado. Assim, o passo de forma dado é $\Delta\xi_i = \{106, 1046; -30, 1774; -0, 2761\}in$, de comprimento $\varphi = 110,3129in$. O trabalho (W), $f_{penalizada}$ e Z na busca unidimensional são ilustrados nas Figuras 21 e 20. Na Figura 21 pode-se ver a busca unidimensional partindo do ponto inicial do operador de forma, representado pela Figura 16, e chegando ao 19º ponto,

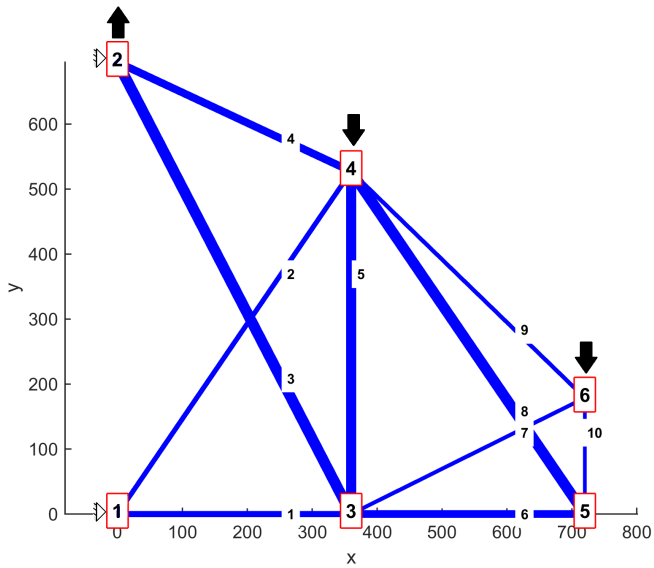


Figura 16 – Treliça inicial do primeiro exemplo de busca local com os sentidos do gradiente de forma ilustrados.

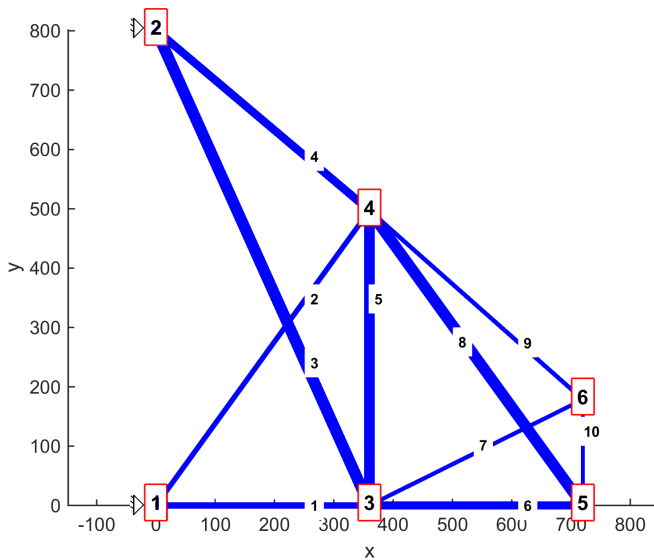
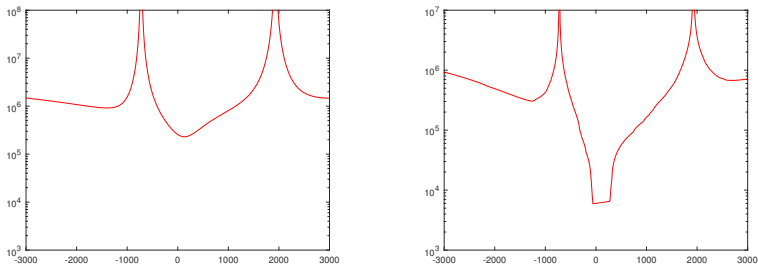


Figura 17 – Treliça do primeiro exemplo de busca local após operação de forma.



(a) $W(\text{lbf.in})$ em relação ao passo(in). (b) $f_{\text{penalizada}}(\text{lb})$ em relação ao passo(in).

Figura 18 – Representação logarítmica na direção do gradiente de forma no primeiro exemplo de busca local

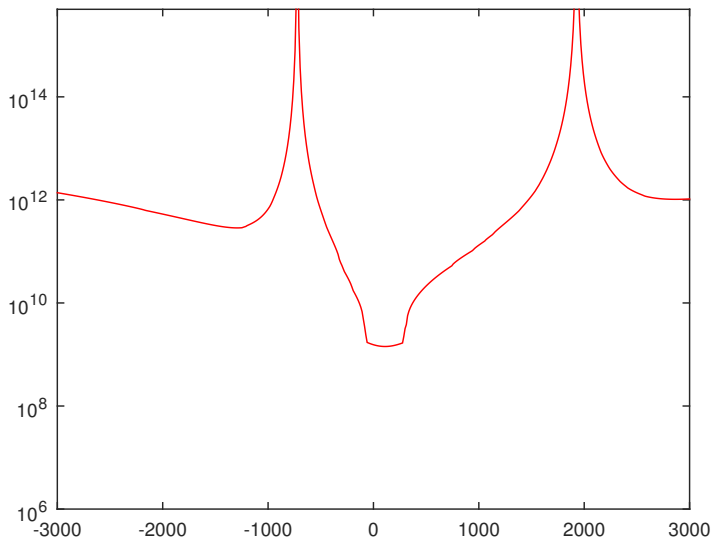


Figura 19 – Representação logarítmica da função Z em relação ao passo(in) no primeiro exemplo.

	Inicial	Final
A_1		7,22
A_2		4,80
A_3		22,90
A_4		11,50
A_5		19,90
A_6		11,50
A_7		3,09
A_8		22,00
A_9		3,38
A_{10}		2,62
ξ_{2y}	695,2413	801,3459
ξ_{4y}	527,3497	497,1723
ξ_{6y}	178,7991	178,5230
$W(\text{lbf.in})$	256.594	230.673
$f_{\text{penalizada}}(\text{lb})$	6.024,67	6.195,53
$Z(10^9\text{lb.lbf.in})$	1,5459	1,4291

Tabela 1 – Dados do primeiro exemplo de busca local.

quando Z apresentou crescimento. O ponto marcado com \times é o 18º e penúltimo ponto, o escolhido para ser retornado ao ICA e representado pela treliça da Figura 17. A treliça final encontrada não viola nenhuma restrição estando em 81,35% do limite de deslocamento máximo no nó 5, direção vertical; em 42,52% do limite de tensão máxima no elemento tracionado 4 e em 63,05% do limite de tensão máxima de compressão no elemento 1.

4.2 Exemplo do operador de forma para início penalizado

Neste exemplo será ilustrado o funcionamento do operador de forma quando a treliça inicial se encontra com restrições violadas. A treliça inicial neste caso é apresentada na Figura 24, estando com 5.704lb de massa e com o deslocamento máximo de 140,5% do permitido para este caso, que seria de $2in$, além de 114,8% da tensão admissível no elemento tracionado de número 5.

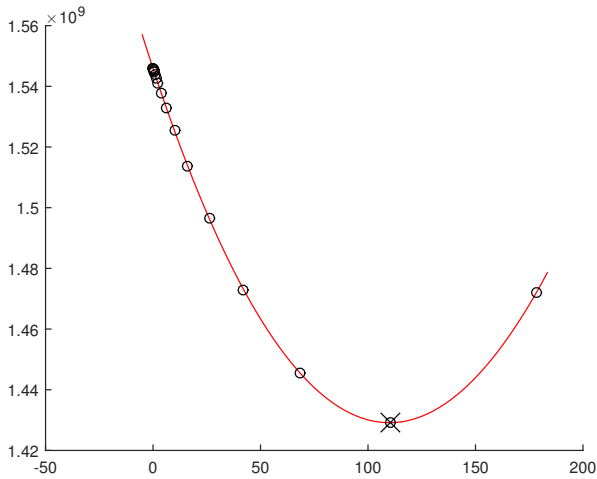
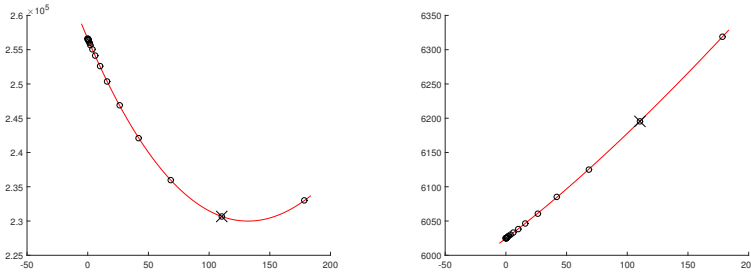


Figura 20 – Função Z na busca unidimensional do primeiro exemplo.

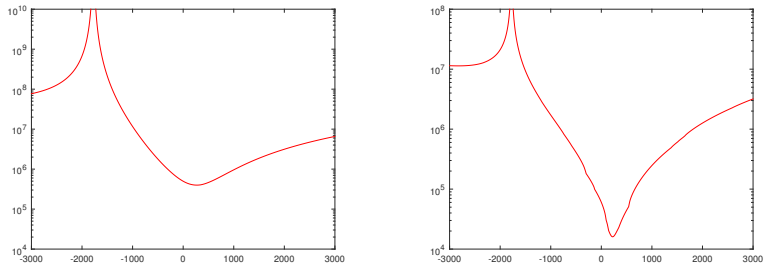


(a) $W(\text{lbf.in})$ em relação ao passo(in) (b) $f_{\text{penalizada}}(\text{lb})$ em relação ao passo(in).

Figura 21 – Busca unidimensional no primeiro exemplo de busca local.

Neste caso, após 19 passos o operador de forma encontrou um gradiente unitário de $\partial W / \partial \xi_i = (-0,0050; -0,3608; 0,9326)$ e um passo de 178,48in resultando em um passo de forma

$\Delta \xi_i = \{0,8909; 64,3982; -166,4615\}in$. Assim, as variáveis de forma mudaram de $\xi_i = \{943,6274; 600,4045; 281,5709\}$ para $\xi_i = \{944,5183;$



(a) $W(\text{lbf.in})$ em relação ao passo(in). (b) $f_{\text{penalizada}}(\text{lb})$ em relação ao passo(in).

Figura 22 – Representação logarítmica na direção do gradiente de forma no segundo exemplo de busca local.

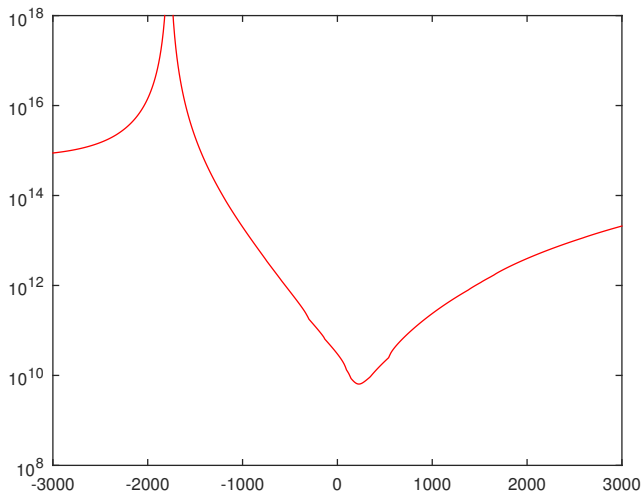


Figura 23 – Representação logarítmica da função Z em relação ao passo(in) no segundo exemplo.

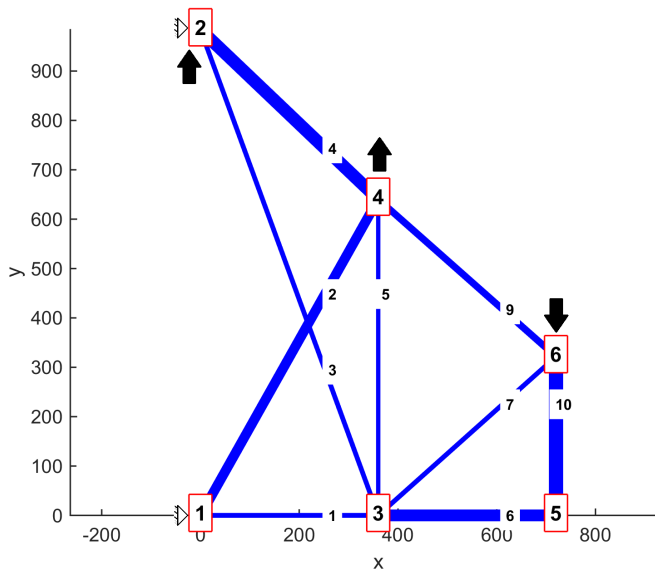


Figura 24 – Treliça inicial do segundo exemplo de busca local com as direções do gradiente de forma ilustradas.

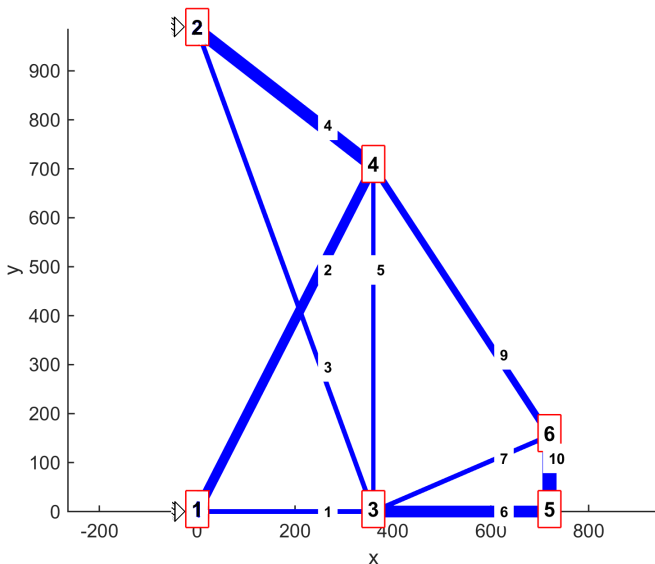


Figura 25 – Treliça do segundo exemplo de busca local após operação de forma.

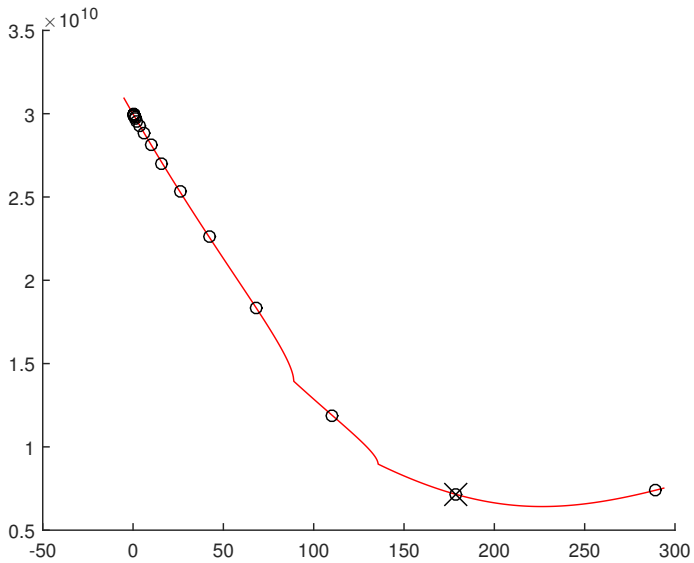
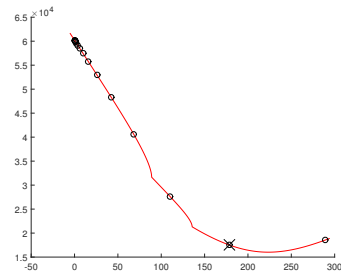
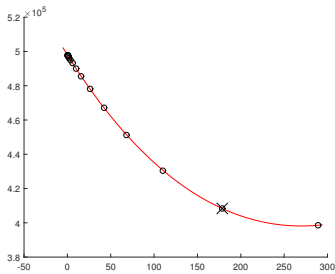


Figura 26 – Função Z na busca unidimensional do segundo exemplo.

664,8028; 115,1094}. A treliça final encontrada, representada pela Figura 25, possui 5.253,5lb de massa, uma redução de 7,9%. O operador também reduziu o deslocamento máximo para 106,67% do limite e a tensão máxima de tração no elemento 5 para 86,497% do valor máximo. O trabalho, que inicialmente era 497.998lbf.in, reduziu para 408.389lbf.in no melhor passo encontrado, redução de 17,99%.



(a) W (lbf.in) em relação ao passo(in). (b) $f_{penalizada}$ (lb) em relação ao passo(in).

Figura 27 – Busca unidimensional no segundo exemplo de busca local.

5 Exemplos numéricos

5.1 Treliça de 10 barras

A treliça de 10 barras é provavelmente o exemplo *benchmark* mais estudado da literatura de otimização de treliças e tem sua topologia inicial apresentada na Figura 28. As variáveis dimensionais são as áreas dos 10 elementos e no presente trabalho devem ser escolhidas de um conjunto de áreas discretas dado pela Tabela 2. As variáveis de forma são 3: as coordenadas y dos nós 2, 4 e 6 e variam continuamente de $180in$ até $1000in$ em relação ao eixo das abscissas que passa nos pontos 1, 3 e 5. A treliça será estudada na faixa de 50.000 avaliações da função objetivo (*OFEs*), a mesma utilizada por Miguel, Lopez e Miguel(9). Será avaliado o desempenho do *ICA* e do *ICA-OF* e os resultados obtidos serão comparados com a literatura. Neste exemplo serão empregados os parâmetros apresentados na Tabela 4. Na Figura 29a são ilustradas as médias e mínimos por década para as 100 execuções do *ICA* em média e a melhor rodagem é apresentada na Figura 29b. Da mesma forma as Figuras 31a e 31b apresentam a convergência do *ICA-OF* ao longo das décadas. A Tabela 3 apresenta os parâmetros do problema.

As Figuras 30 e 32 ilustram as melhores treliças encontradas pelo *ICA* e pelo *ICA-OF* respectivamente.

O resultado de 2.747,05lb encontrado pelo *ICA-OF* não superou

1,62	1,80	2,38	2,62	2,88	3,09	3,13	3,38
3,63	3,84	3,87	4,18	4,49	4,80	4,97	5,12
5,74	7,22	7,97	11,50	13,50	13,90	14,20	15,50
16,00	18,80	19,90	22,00	22,90	26,50	30,00	33,50

Tabela 2 – Possíveis áreas das seções transversais para os elementos da treliça de 10 barras em in^2 .

Parâmetros da treliça	IS	SI
Módulo de elasticidade	10^7 psi	68.948 MPa
Peso específico	0,1 lbf/in ³	27,154 kN/m ³
σ_{adm} de tração	25.000 psi	172,37 MPa
σ_{adm} de compressão	25.000 psi	172,37 MPa
Máx. desl. em y	2 in	5,08 cm

Tabela 3 – Parâmetros do problema da treliça de 10 barras.

o encontrado por Miguel, de 2.705,16lb, entretanto, na média de 100 execuções o algoritmo implementado convergiu mais consistentemente do que o algoritmo dos vagalumes (*FA*), alcançando uma média de 2.822,52lb com desvio padrão de 0,85%. Embora o *ICA* sem a busca local tenha apresentado uma boa convergência para este problema, a busca local reduziu a média de 100 execuções em 9,15lb. As melhores treliças encontradas tanto pelo *ICA* quanto pelo *ICA-OF* alcançaram 99,9999% da restrição de deslocamento máximo de 2in e apresentaram folga nas

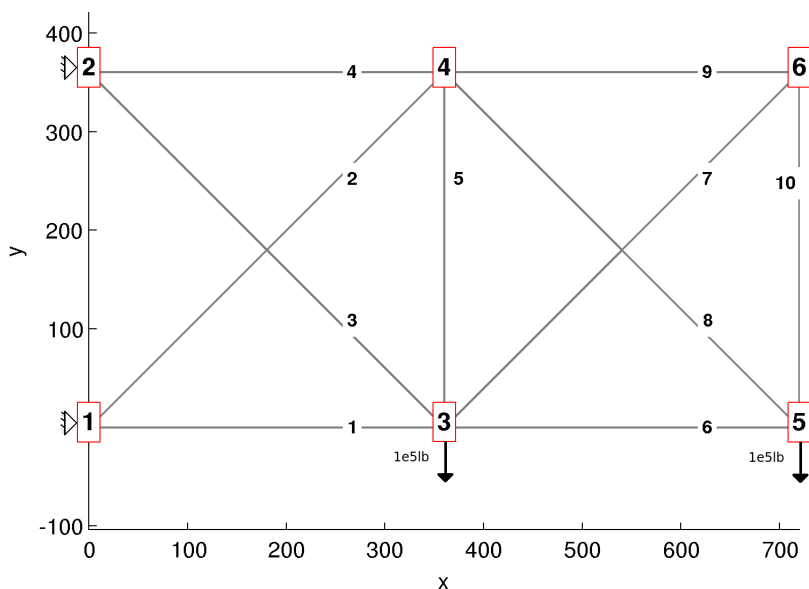
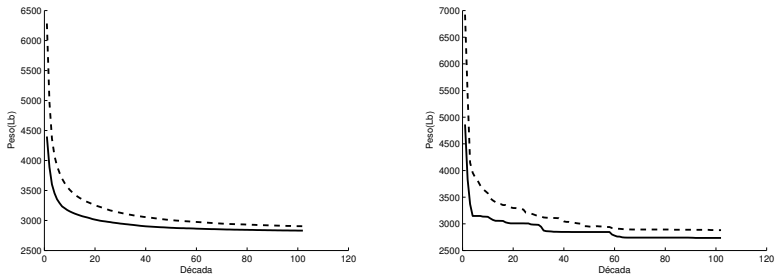


Figura 28 – Topologia inicial da treliça de 10 elementos em polegadas.



(a) Média da convergência em 100 execuções. (b) Convergência da melhor execução.

Figura 29 – Execuções do *ICA* para a treliza de 10 barras. Mínimos em linha contínua e médias em tracejado.

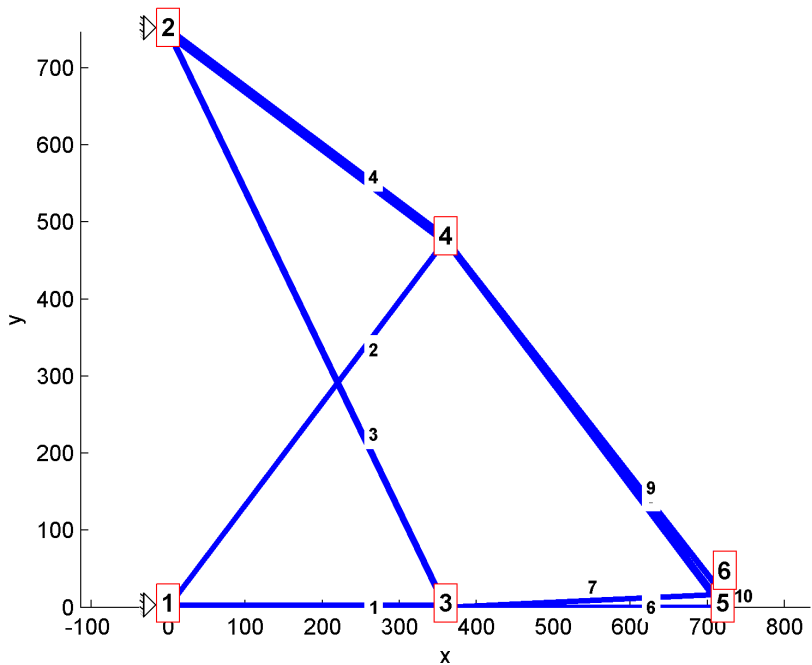
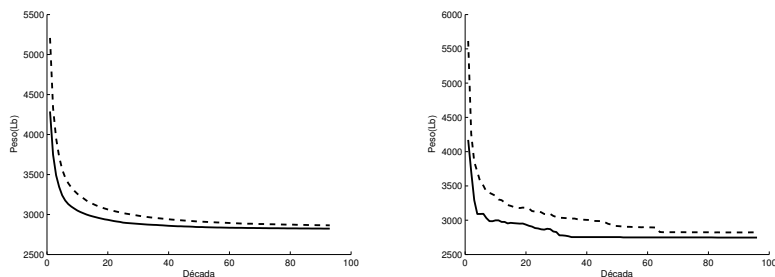


Figura 30 – Solução ótima da treliza de 10 barras encontrada pelo *ICA* em polegadas.



(a) Média da convergência em 100 execuções. (b) Convergência da melhor execução.

Figura 31 – Execuções do *ICA-OF* para a treliça de 10 barras. Mínimos em linha contínua e médias em tracejado

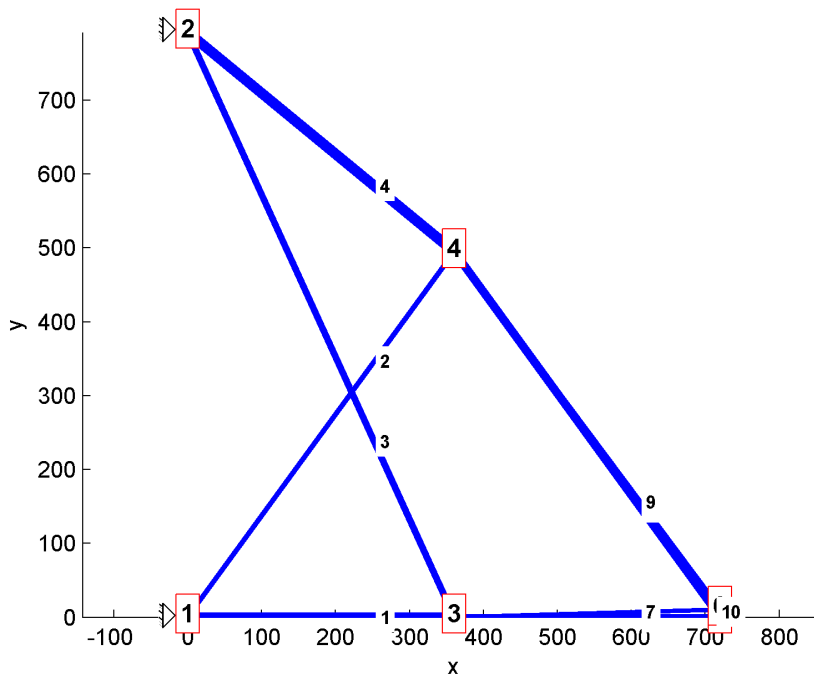


Figura 32 – Solução ótima da treliça de 10 barras encontrada pelo *ICA-OF* em polegadas.

Parâmetro	
Número de colônias (ver seção 3)	490
Número de imperialistas (ver seção 3)	4(0,816%)
Coefficiente de assimilação (ver seção 3)	0,618
Probabilidade de tirar barra da trelça	20%
Prob. de chamar operador no início da execução (ver seção 4)	0,15
Prob. de chamar operador no final da execução(ver seção 4)	0,23

Tabela 4 – Parâmetros do *ICA* e do *ICA-OF* para o problema da trelça de 10 barras.

restrições de tensão máxima. As médias de 100 execuções, respectivos desvios padrão e melhores valores obtidos são apresentados na Tabela 5. Os parâmetros das melhores trelças encontradas são apresentados na Tabela 6.

	Média (lb/kg)	Std(%)	Melhor Valor (lb/kg)
<i>FA</i> (9)	2.893,45/ 1.312,45	2,12	2.705,16/ 1.227,04
<i>ICA</i>	2.831,67/ 1.284,42	1,06	2.736,07/ 1.241,06
<i>ICA-OF</i>	2.822,52/ 1.280.27	0,85	2.747,05/ 1.246,04

Tabela 5 – Comparação entre o desempenho dos algoritmos na solução da trelça de 10 barras.

Alguns autores estudaram o mesmo problema mas não realizaram uma série de execuções. Rajan(36) obteve 3.254lb empregando 3.840 *OFEs* e Balling, Briggs e Gillman(10) encontraram 2.736lb usando 500.000 *OFEs*. Ambos os trabalhos empregaram o algoritmo genético (*GA*). Martini(58) estudou a trelça de 10 barras usando a Busca Harmônica (*HS*) com 4.075 *OFEs* e alcançou um ótimo de $2,91 \times 10^2$ lb. Estando tão distantes do custo computacional empregado no presente trabalho, não se pode comparar os resultados obtidos por Rajan e Martini com os obtidos pelo *ICA*.

O operador implementado melhorou a convergência do *ICA* na

	<i>FA</i> (9)	<i>ICA</i>	<i>ICA-OF</i>
A_1	11,50	11,50	13,50
A_2	2,88	3,84	3,13
A_3	5,74	5,74	5,74
A_4	11,50	13,50	13,50
A_5	0	0	0
A_6	7,22	4,18	4,97
A_7	0	3,84	1,80
A_8	13,50	7,22	7,97
A_9	0	4,80	3,38
A_{10}	0	4,97	4,49
y_2	–	746,9695	791,9156
y_4	–	476,9946	494,7041
y_6	–	16,9618	10,1034
Massa(lb/kg)	2.705,16/ 1.227,04	2.736,07/ 1.241,06	2.747,05/ 1.246,04
Trabalho(lbf.in)	–	399.413	398.340

Tabela 6 – Comparação entre os melhores resultados para a treliça de 10 barras. Áreas em in^2 e coordenadas em in .

solução do problema da treliça de 10 barras, mesmo o *ICA* já apresentando uma boa convergência. É interessante notar como, mesmo tendo uma média de execuções consistentemente mais baixa do que Miguel, o *ICA* e o *ICA-OF* não alcançaram o menor mínimo alcançado por Miguel. Uma possível explicação é a escolha conservadora dos parâmetros do algoritmo, como o coeficiente de assimilação, definido como 0,618. Quanto menor este valor, mais lentamente e consistentemente as colônias se movem na direção dos seus imperialistas.

5.2 Treliça de 15 barras

A treliça de 15 barras é um problema semelhante à treliça de 10 barras por ser uma estrutura em balanço, entretanto as dimensões são menores assim como as possíveis áreas de seções transversais. A topologia inicial da treliça é apresentada na Figura 33. Os parâmetros

do problema são apresentados na Tabela 9. Nota-se que este problema não apresenta restrições de deslocamento máximo. As áreas das seções transversais devem ser obtidas da Tabela 8 e as posições dos nós são tratadas como variáveis contínuas restritas aos intervalos apresentados na Tabela 7. É importante notar que a posição de x_6 é idêntica à de x_2 assim como x_7 é igual à x_3 . O problema apresenta 23 variáveis: 15 dimensionais e discretas e 8 de forma e contínuas.

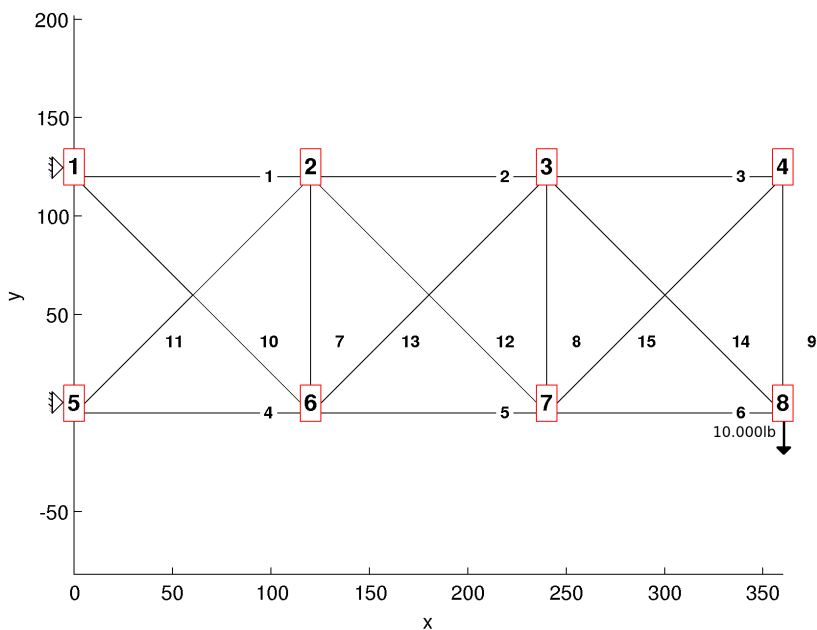


Figura 33 – Topologia inicial da treliça de 15 barras.

Este problema foi estudado usando o algoritmo genético por Wu e Chow(12) como um problema dimensional e também como um problema dimensional e de forma simultaneamente, em ambos os casos considerando restrição de flambagem. Tang, Tong e Gu(11) aplicaram otimização simultânea dimensional e de forma, e também otimização simultânea dimensional, de forma e topológica, aplicando o algoritmo genético, entretanto, desconsiderando a flambagem. Miguel, Lopez e Miguel(9) estudaram todos os casos estudados por Tang, Tong e Gu e

100in	$\leq x_2 \leq$	140in
220in	$\leq x_3 \leq$	260in
100in	$\leq y_2 \leq$	140in
100in	$\leq y_3 \leq$	140in
50in	$\leq y_4 \leq$	90in
-20in	$\leq y_6 \leq$	20in
-20in	$\leq y_7 \leq$	20in
20in	$\leq y_8 \leq$	60in

Tabela 7 – Limites das variáveis de forma para o problema da treliça de 15 barras.

0,111	0,141	0,174	0,220	0,270	0,287	0,347
0,440	0,539	0,954	1,081	1,174	1,333	1,488
1,764	2,142	2,697	2,800	3,131	3,565	3,813
5,952	6,572	7,192	8,525	9,300	10,850	13,330
4,805	14,290	17,170	19,180			

Tabela 8 – Possíveis áreas das seções transversais para os elementos da treliça de 15 barras em in^2 .

por Wu e Chow com o algoritmo dos vagalumes (*FA*), além de estudar o caso de otimização simultânea dimensional, de forma e topológica com restrições de flambagem. Será estudado o caso da otimização simultânea dimensional, de forma e topológica sem restrição de flambagem na faixa de 8.000 *OFEs*, possibilitando a comparação com o trabalho de Miguel, Lopez e Miguel e com o de Tang, Tong e Gu.

Parâmetros da treliça	<i>IS</i>	SI
Módulo de elasticidade	10^7 psi	68.948 MPa
Peso específico	0,1 lbf/in ³	27,154 kN/m ³
Tensão admissível de tração	25.000 psi	172,37 MPa
Tensão admissível de compressão	25.000 psi	172,37 MPa

Tabela 9 – Parâmetros do problema da treliça de 15 barras.

Foram realizadas 100 execuções do problema utilizando o *ICA* e 100 usando o *ICA-OF* e serão calculadas as médias e desvios padrão

Parâmetro	
Número de colônias (ver seção 3)	53
Número de imperialistas (ver seção 3)	2(3,77%)
Coefficiente de assimilação (ver seção 3)	0,95
Probabilidade de tirar barra da treliça	20%
Prob. de chamar operador no início da execução (ver seção 4)	0,40
Prob. de chamar operador no final da execução (ver seção 4)	0,02

Tabela 10 – Parâmetros do *ICA* e do *ICA-OF* para o problema da treliça de 15 barras

Algoritmo	Média(lb/kg)	Std(%)	Melhor Valor(lb/kg)
<i>FA</i> (9)	82,64/37,48	2,96	75,5/34,24
<i>ICA</i>	87,34/39,62	7,64	75,60/34,29
<i>ICA-OF</i>	84,78/38,46	5,80	73,49/33,33

Tabela 11 – Resultados para otimização dimensional, de forma e topológica da treliça de 15 barras.

dos resultados finais obtidos com o objetivo de avaliar o desempenho do algoritmo proposto e compará-lo com a literatura.

Com a configuração apresentada na Tabela ?? o *ICA-OF* empregou em média 25,07% dos desenvolvimentos da função objetivo na busca local, sendo que em 54,65% das vezes que foi chamada a busca local retornou um resultado melhor do que o inicial. Na Figura 36a são ilustradas as médias e mínimos por década para as 100 execuções do *ICA* em média e a melhor rodagem é apresentada na Figura 36b. Da mesma forma as Figuras 37a e 37b apresentam a convergência do *ICA-OF* ao longo das décadas.

A melhor solução encontrada pelo *ICA*, representada na Figura 34, e a encontrada pelo *ICA-OF*, representada na Figura 35, são apresentadas na Tabela 12. A treliça encontrada pelo *ICA* se encontrava no limite da restrição de tensão máxima, com o elemento 9 tracionado em

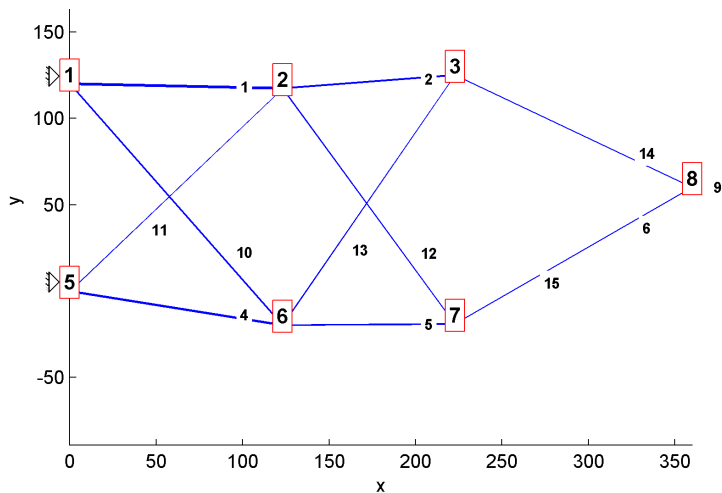


Figura 34 – Solução ótima da treliça de 15 barras encontrada pelo *ICA* em polegadas.

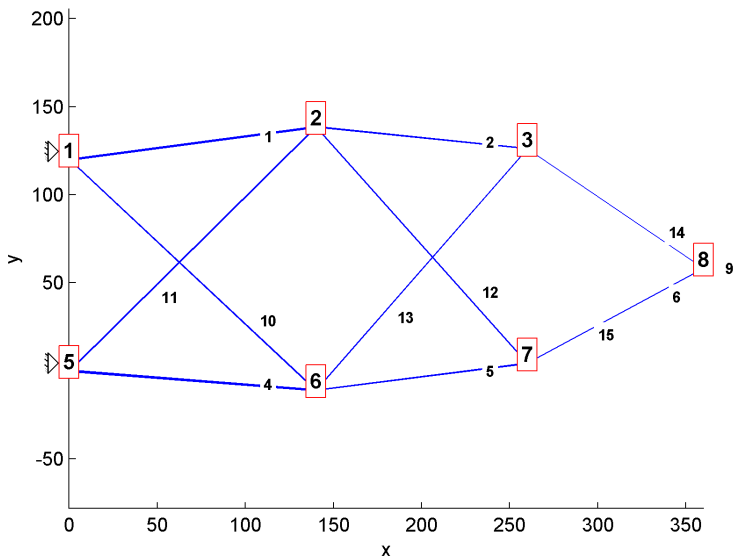
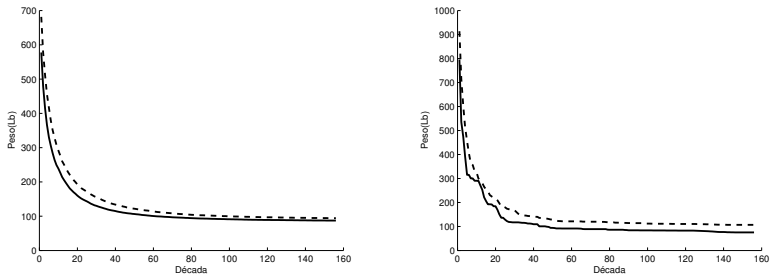
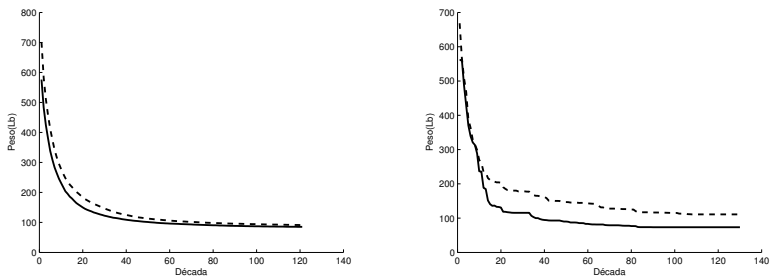


Figura 35 – Solução ótima da treliça de 15 barras encontrada pelo *ICA-OF* em polegadas.



(a) Média da convergência em 100 execuções. (b) Convergência da melhor execução.

Figura 36 – Execuções do *ICA* para a treliça de 15 barras. Mínimos em linha contínua e médias em tracejado.



(a) Média da convergência em 100 execuções. (b) Convergência da melhor execução.

Figura 37 – Execuções do *ICA-OF* para a treliça de 15 barras. Mínimos em linha contínua e médias em tracejado.

	<i>GA</i> (11)	<i>FA</i> (9)	<i>ICA</i>	<i>ICA-OF</i>
A_1	1,081	0,954	0,954	0,954
A_2	0,539	0,539	0,539	0,539
A_3	0,287	0,220	0	0,270
A_4	0,954	0,954	1,081	0,954
A_5	0,954	0,539	0,539	0,440
A_6	0,220	0,220	0,440	0,141
A_7	0,111	0,111	0	0
A_8	0,111	0,111	0	0
A_9	0,287	0,287	3,813	0,347
A_{10}	0,220	0,440	0,539	0,440
A_{11}	0,440	0,440	0,287	0,440
A_{12}	0,440	0,220	0,270	0,174
A_{13}	0,111	0,220	0,270	0,270
A_{14}	0,220	0,270	0,440	0,141
A_{15}	0,347	0,220	0,111	0,270
x_2 e x_6	133,612	114,967	122,979	140,000
x_3 e x_7	234,752	247,040	222,746	260,000
y_2	100,449	125,919	117,338	138,547
y_3	104,738	111,067	125,048	126,170
y_4	73,762	58,298	60,000	58,130
y_6	10,067	-17,564	-19,713	-10,985
y_7	-1,339	-5,821	-19,137	4,266
y_8	50,402	31,465	60,000	58,120
Massa	79,82/	75,5/	75,60/	73,49/
(lb/kg)	36,21	34,24	34,29	33,33
Trabalho	–	–	42.214	40.708
(lbf.in)				

Tabela 12 – Comparação entre os melhores resultados para a treliça de 15 barras. Áreas em in^2 e coordenadas em in .

99,2939% do limite de 25ksi(172,37MPa) e o elemento 4 comprimido com 99,9991% do limite. Já a encontrada pelo *ICA-OF* se encontrava mais próxima do limite, com o elemento tracionado 8 a 99,9996% da tensão máxima e o elemento comprimido 5 a 99,9998%.

Embora o *ICA-OF* não tenha alcançado a média que o *FA* alcançou, esse conseguiu encontrar um mínimo menor do que o *FA*, sendo o

menor valor observado para este problema na literatura. Observando a Tabela 11 é possível se concluir que o operador de forma melhorou a média de 100 execuções do *ICA*, além diminuir o desvio padrão e o mínimo encontrado, comprovando a sua eficiência na solução do problema proposto.

5.3 Torre treliçada de 25 barras

A torre de 25 barras é um conhecido problema de treliças espaciais. A sua topologia inicial está representada nas Figuras 38 e 39. As Tabelas 13 e 18 apresentam respectivamente os parâmetros do problema e as posições dos nós na topologia inicial. Como o problema apresenta dois planos de simetria em $x - z$ e $y - z$, as áreas dos 25 elementos são definidas por 8 variáveis dimensionais agrupadas como na Tabela 19 a ser escolhidas de um conjunto de áreas discretas obtidas da Tabela 15. As 5 variáveis de forma são contínuas e seus limites estão definidos na Tabela 17. Devido à dupla simetria, as posições dos nós 3,5,6,7,9 e 10 dependem também das variáveis de forma. As forças aplicadas na treliça são apresentados na Tabela 16. A treliça será estudada na faixa de 6.000 *OFEs*, a mesma utilizada por Miguel, Lopez e Miguel(9), tanto com o *ICA* como com o *ICA-OF*, e os resultados obtidos serão comparados com a literatura. Em ambos os casos serão empregados os parâmetros apresentados na Tabela 14.

Parâmetros da treliça	<i>IS</i>	SI
Módulo de elasticidade	10^7 psi	68.948 MPa
Peso específico	0,1 lbf/in ³	27,154 kN/m ³
Tensão admissível de tração	40.000 psi	275,79 MPa
Tensão admissível de compressão	40.000 psi	275,79 MPa
Máximo deslocamento na direção y	0,35 in	0,889 cm

Tabela 13 – Parâmetros do problema da treliça de 25 barras.

Parâmetro	
Número de colônias (ver seção 3)	214
Número de imperialistas (ver seção 3)	3(1,4019%)
Coefficiente de assimilação (ver seção 3)	1,06
Probabilidade de tirar barra da treliça	20%
Prob. de chamar operador no início da execução (ver seção 4)	0,15
Prob. de chamar operador no final da execução (ver seção 4)	0,50

Tabela 14 – Parâmetros do *ICA* e do *ICA-OF* para o problema da treliça de 25 barras.

0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0	1,1
1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2,0	2,1	2,2
2,3	2,4	2,5	2,6	2,8	3,0	3,2	3,4			

Tabela 15 – Possíveis áreas das seções transversais para os elementos da treliça de 25 barras em in².

Na Figura 40a são ilustradas as médias e mínimos por década para as 100 execuções do *ICA* em média e a melhor rodagem é apresentada na Figura 40b. Da mesma forma as Figuras 41a e 41b apresentam a convergência do *ICA-OF* ao longo das décadas. As treliças ótimas encontradas pelo *ICA* e pelo *ICA-OF* são apresentadas nas Figuras 42 e 43 em vista no plano x-y e nas Figuras 44 e 45 em vista isométrica.

Nó	F_x (lbf/kN)	F_y (lbf/kN)	F_z (lbf/kN)
1	1.000/4,448	-10.000/-44,482	-10.000/-44,482
2	0	-10.000/-44,482	-10.000/-44,482
3	500/2,224	0	0
6	600/2,669	0	0

Tabela 16 – Carregamentos da torre de 25 barras.

20in	$\leq x_4 \leq$	60in
40in	$\leq x_8 \leq$	80in
40in	$\leq y_4 \leq$	80in
100in	$\leq y_8 \leq$	140in
90in	$\leq z_4 \leq$	130in

Tabela 17 – Limites das variáveis de forma.

Nó	x(in)	y(in)	z(in)
1	-37,5	0	200
2	37,5	0	200
3	-37,5	37,5	100
4	37,5	37,5	100
5	37,5	-37,5	100
6	-37,5	-37,5	100
7	-100	100	0
8	100	100	0
9	100	-100	0
10	-100	-100	0

Tabela 18 – Posições dos nós para a torre de 25 barras.

Grupo	Elemento
A1	1
A2	2, 3, 4
A3	6, 7, 8, 9
A4	10, 11
A5	12, 13
A6	14, 15, 16, 17
A7	18, 19, 20, 21
A8	22, 23, 24, 25

Tabela 19 – Grupos de elementos com áreas iguais para a torre de 25 barras.

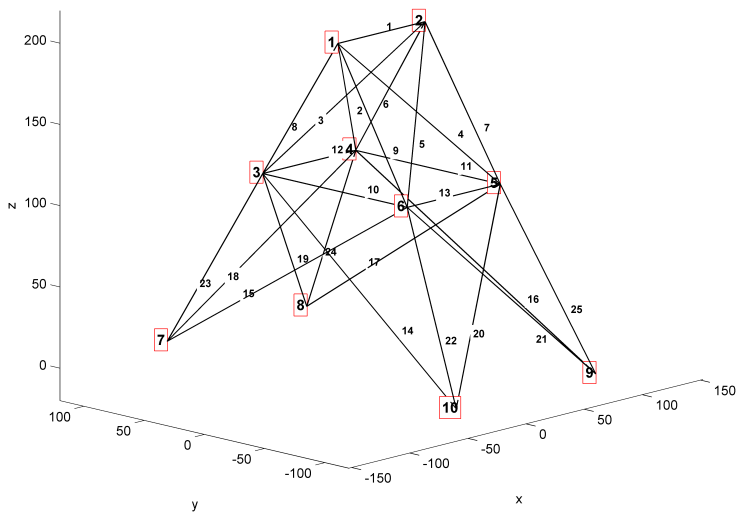


Figura 38 – Topologia inicial da treliça de 25 elementos em polegadas em vista isométrica.

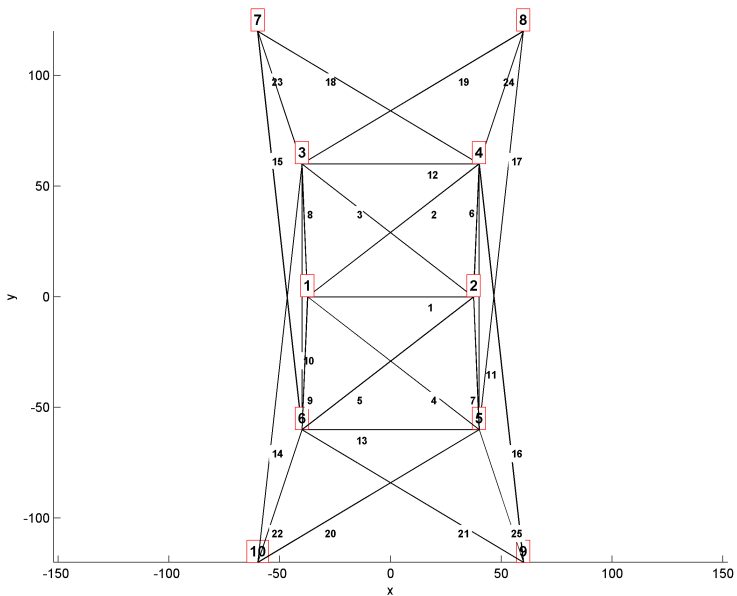
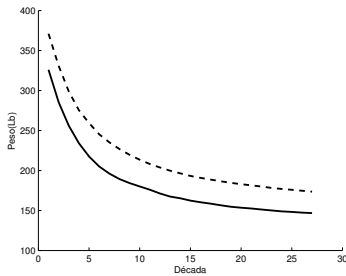
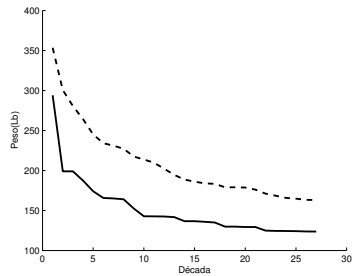


Figura 39 – Topologia inicial da treliça de 25 elementos em polegadas em vista no plano x-y.

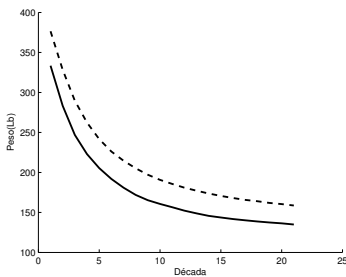


(a) Média da convergência em 100 execuções.

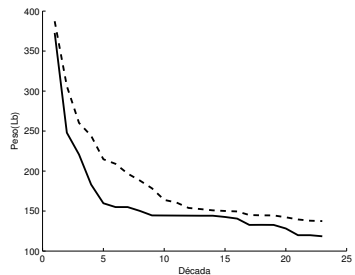


(b) Convergência da melhor execução.

Figura 40 – Execuções do *ICA* para a torre treliçada. Mínimos em linha contínua e médias em tracejado.



(a) Média da convergência em 100 execuções.



(b) Convergência da melhor execução

Figura 41 – Execuções do *ICA-OF* para a torre treliçada. Mínimos em linha contínua e médias em tracejado.

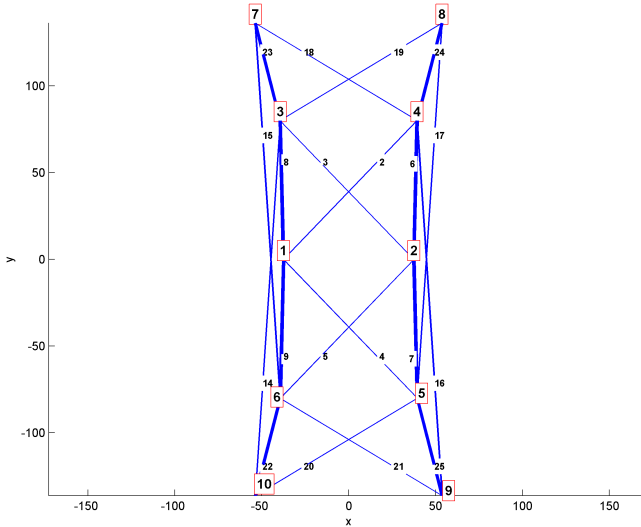


Figura 42 – Solução ótima da treliça de 25 elementos encontrada pelo *ICA* em polegadas vista no plano x - y .

Estudando o mesmo problema, Wu e Chow(12) encontraram um peso ótimo de 136,2lb empregando o algoritmo genético. Tang, Tong e Gu(11) desenvolveram um algoritmo genético melhorado e empregando o mesmo na otimização da torre de 25 elementos obtiveram um peso mínimo de 114,74lb, significativamente melhor do que o de Wu e Chow. O *ICA* encontrou um mínimo de 123,73lb, enquanto o *ICA-OF* se aproximou mais do mínimo global, chegando a 118,80lb. Nem Wu e Chow; nem Tang, Tong e Gu apresentaram a média e desvio padrão de uma série de execuções para possibilitar uma comparação mais aprofundada do desempenho dos algoritmos na solução do problema proposto. A Tabela 20 apresenta a comparação entre as médias e desvios padrões obtidos no presente trabalho com os obtidos por Miguel, Lopez e Miguel(9). A Tabela 21 apresenta as melhores soluções encontradas para o problema por diferentes autores.

Tanto a solução encontrada pelo *ICA* quanto a encontrada pelo *ICA-OF* estavam próximas do limite de deslocamento máximo de 0,35in (0,8890cm). O *ICA* chegou a 99,60% da restrição enquanto o *ICA-OF*

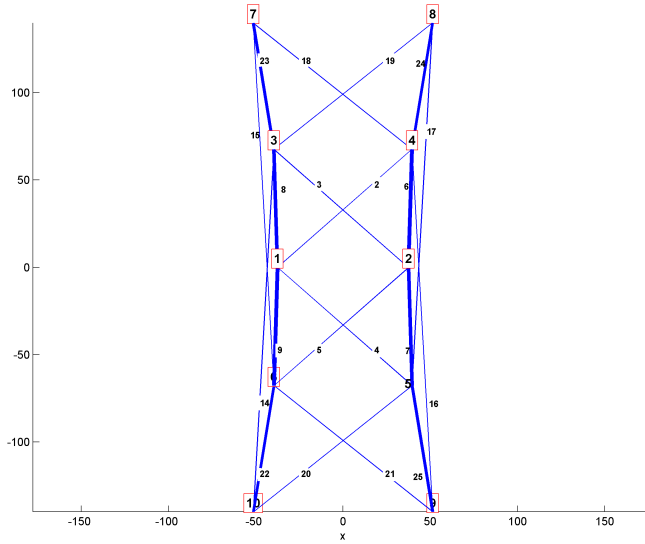


Figura 43 – Solução ótima da treliça de 25 elementos encontrada pelo *ICA-OF* em polegadas vista no plano x-y.

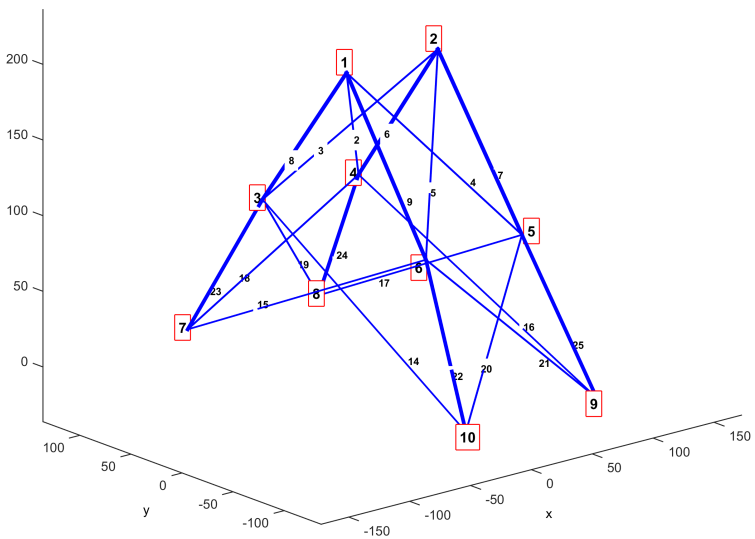


Figura 44 – Solução ótima da treliça de 25 elementos encontrada pelo *ICA* em polegadas em vista isométrica.

	Média(lb/kg)	Std(%)	Melhor Valor(lb/kg)
<i>FA</i> (9)	139,16/63,122	8,0	116,58/52,880
<i>ICA</i>	146,77/66,574	9,64	123,73/56,123
<i>ICA-OF</i>	132,50/60,101	8,87	118,50/53,751

Tabela 20 – Comparação entre o desempenho dos algoritmos na solução da torre de 25 barras.

	<i>GA</i> (11)	<i>FA</i> (9)	<i>ICA</i>	<i>ICA-OF</i>
A_1	0	0	0	0
A_2	0,1	0,1	0,1	0,1
A_3	0,9	1,1	1,0	1,1
A_4	0	0	0	0
A_5	0	0	0	0
A_6	0,1	0,1	0,2	0,1
A_7	0,1	0,1	0,1	0,1
A_8	1,0	0,9	0,9	0,9
x_4	39,91	39,50	39,4401	39,5931
y_4	61,99	64,35	80,0000	67,7187
z_4	118,23	112,87	96,8419	107,5333
x_8	53,13	49,13	53,7663	51,3715
y_8	138,49	134,94	136,1703	140,0000
Massa (lb/kg)	114,74/ 52,045	116,58/ 52,880	123,7273/ 56,123	118,50/ 53,751
Trabalho (lbf.in)	–	–	11.680	11.090

Tabela 21 – Comparação entre os melhores resultados para a treliça de 25 barras. Áreas em in^2 e coordenadas em in .

alcançou 99,69%. Com os parâmetros adotados, a busca de forma consumiu 14,7% do custo computacional relacionado à média das execuções, sendo que, em 82,43% das vezes que foi chamada, retornou uma posição final com Z menor do que a posição original. A busca local se mostrou eficiente na melhora da convergência do *ICA* para o problema da torre de 25 elementos. Estando a restrição de deslocamentos máximos ativa, é esperado que a busca local vá ser eficiente, visto que esta tende a tornar a estrutura mais rígida, reduzindo os deslocamentos. Embora o *ICA*, com os parâmetros utilizados, não tenha alcançado o desempenho de outros metaheurísticos de população como o *GA* empregado por Tang, Tong e Gu(11) e o *FA* empregado por Miguel, Lopez e Miguel(9), a busca local melhorou significativamente a convergência deste, alcançando uma média da função objetivo em 100 execuções 4,79% menor do que Miguel. No entanto, o *ICA-OF* não superou o *FA* nem o *GA* no melhor valor encontrado.

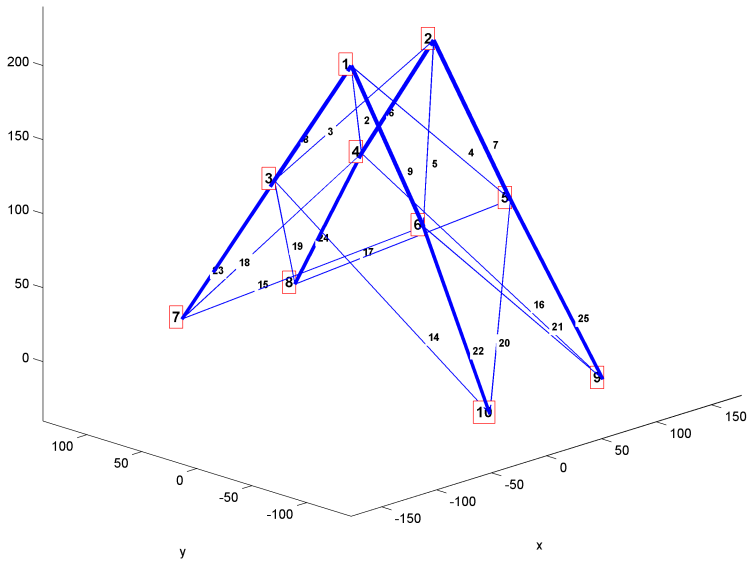


Figura 45 – Solução ótima da treliça de 25 elementos encontrada pelo *ICA-OF* em polegadas em vista isométrica.

Conclusão

O operador de forma proposto foi desenvolvido sobre o código do *ICA* visando melhorar o desempenho deste e três problemas clássicos de otimização estrutural foram estudados e comparados com a literatura. Como o problema que se busca solucionar é complexo e exige algoritmos de otimização robustos para a sua solução, não há muitas referências para comparar o desempenho do algoritmo. Como o trabalho de Miguel, Lopez e Miguel(9) teve bom desempenho empregando o *FA*, e como o autor disponibilizou médias e desvios padrões de 100 execuções, buscou-se reproduzir os mesmo problemas, com o mesmo custo computacional, para possibilitar a comparação. O algoritmo *ICA* foi executado 100 vezes, com e sem o operador, para que ambos os casos possam ser comparados entre si e com o *FA*.

Observando os resultados obtidos é possível se concluir que o *ICA* se mostra eficiente na resolução dos problemas propostos e que o operador de forma implementado melhorou o desempenho do mesmo nos casos estudados. Nos exemplos numéricos 1 e 3 a média alcançada pelo *ICA-OF* foi melhor do que a referência, embora em ambos os casos tenha encontrado mínimos piores do que os encontrados por Miguel, Lopez e Miguel(9) empregando o *FA*. No segundo exemplo o inverso ocorreu e o *ICA-OF* chegou a um mínimo menor do que o *FA* mas com uma média mais alta. A causa destes resultados está provavelmente relacionada às diferenças nas definições dos parâmetros dos algoritmos, afinal o desempenho do algoritmo depende muito da calibração destes.

Uma possível melhora no funcionamento do algoritmo seria o uso da informação da violação da restrição de deslocamento máximo para a ativação da busca local. Assim o algoritmo de busca local seria chamado quando é mais eficiente. Uma probabilidade de movimento dependente do desenvolvimento histórico do ponto estudado durante

a execução também poderia melhorar a eficiência do operador, aumentando a probabilidade de movimento conforme o metaheurístico base se mostrasse preso. Não há dúvidas, entretanto, de que o emprego de metaheurísticos na otimização estrutural tem muito o que desenvolver, sendo este um possível objeto de estudo para futuros trabalhos.

Referências

- 1 MICHELL, A. The limits of economy of material in frame-structures. *Philosophical Magazine Series 6*, v. 8, n. 47, p. 589–597, 1904. Citado na página 23.
- 2 HEGEMIER, G.; PRAGER, W. On Michell trusses. *International Journal of Mechanical Sciences*, Elsevier, v. 11, n. 2, p. 209–215, 1969. Citado 5 vezes nas páginas 23, 24, 28, 41 e 54.
- 3 FANCELLO, E. Topology optimization for minimum mass design considering local failure constraints and contact boundary conditions. *Structural and Multidisciplinary Optimization*, Springer-Verlag, v. 32, n. 3, p. 229–240, 2006. Citado na página 24.
- 4 EMMENDOERFER, H.; FANCELLO, E. A. A level set approach for topology optimization with local stress constraints. *International Journal for Numerical Methods in Engineering*, v. 99, n. 2, p. 129–156, 2014. Citado na página 24.
- 5 DORN, W.; GOMORY, R.; GREENBERG, H. Automatic design of optimal structures. *Journal de Mecanique*, v. 3, p. 25–52, 1964. Citado 3 vezes nas páginas 24, 25 e 31.
- 6 FLERON, P. The minimum weight of trusses. *Bygningsstat. Medd.*, p. 81–96, 1964. Citado 2 vezes nas páginas 24 e 31.
- 7 DOBBS, M.; FELTON, L. Optimization of truss geometry. *Journal of Structural Division - ASCE*, v. 95, p. 2105–2118, 1969. Citado 3 vezes nas páginas 24, 25 e 31.
- 8 ROZVANY, G. I. N. On symmetry and non-uniqueness in exact topology optimization. *Structural and Multidisciplinary Optimization*, v. 43, p. 297–317, 2011. Citado na página 25.
- 9 MIGUEL, L.; LOPEZ, R.; MIGUEL, L. Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm. *Advances in Engineering Software*, v. 56, p. 23–37, 2013. Citado 12 vezes nas páginas 26, 69, 73, 74, 75, 77, 80, 81, 86, 88, 89 e 91.
- 10 BALLING, R. J.; BRIGGS, R. R.; GILLMAN, K. Multiple optimum size/shape/topology designs for skeletal structures using a

genetic algorithm. *Journal of structural engineering*, American Society of Civil Engineers, v. 132, n. 7, p. 1158–1165, 2006. Citado 2 vezes nas páginas 26 e 73.

11 TANG, W.; TONG, L.; GU, Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, v. 62, n. 13, p. 1737–1762, 2005. Citado 6 vezes nas páginas 26, 75, 80, 86, 88 e 89.

12 WU, S.-J.; CHOW, P.-T. Integrated discrete and configuration optimization of trusses using genetic algorithms. *Computers & structures*, Elsevier, v. 55, n. 4, p. 695–702, 1995. Citado 3 vezes nas páginas 26, 75 e 86.

13 BIANCHI, L. *et al.* A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, Springer Netherlands, v. 8, n. 2, p. 239–287, 2009. Citado na página 26.

14 GLOVER, F.; KOCHENBERGER, G. A. *Handbook of metaheuristics*. [S.l.]: Springer Science & Business Media, 2003. Citado 5 vezes nas páginas 26, 27, 28, 32 e 41.

15 ACHTZIGER, W. On simultaneous optimization of truss geometry and topology. *Structural and Multidisciplinary Optimization*, v. 33, n. 4-5, p. 285–304, 2007. Citado 2 vezes nas páginas 27 e 38.

16 KAVEH, A.; JAVADI, S. Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm. *Acta Mechanica*, Springer, v. 225, n. 6, p. 1595–1605, 2014. Citado 2 vezes nas páginas 28 e 33.

17 PHOLDEE, N.; BUREERAT, S. Performance enhancement of multiobjective evolutionary optimisers for truss design using an approximate gradient. *Comput. Struct.*, Pergamon Press, Inc., Elmsford, NY, USA, v. 106-107, p. 115–124, set. 2012. Citado 2 vezes nas páginas 28 e 34.

18 TORII, A. J.; LOPEZ, R. H.; LUERSEN, M. A. A local-restart coupled strategy for simultaneous sizing and geometry truss optimization. *Latin American Journal of Solids and Structures*, scielo, v. 8, p. 335 – 349, 09 2011. Citado 2 vezes nas páginas 28 e 34.

- 19 NHAMAGE, I. *Aperfeiçoamento do algoritmo de otimização híbrido Pincus-Nelder e Mead para detecção de dano em estruturas a partir de dados vibracionais*. Dissertação (Mestrado) — PPGEC/UFSC, 2014. Citado 2 vezes nas páginas 28 e 34.
- 20 MIGUEL, L. F.; LOPEZ, R. H.; MIGUEL, L. F. A hybrid approach for damage detection of structures under operational conditions. *Journal of Sound and Vibration*, v. 332, n. 18, p. 4241–4260, 2013. Cited By 0. Citado 2 vezes nas páginas 28 e 34.
- 21 HAO, Z.-F.; WANG, Z.-G.; HUANG, H. A particle swarm optimization algorithm with crossover operator. In: IEEE. *Machine Learning and Cybernetics, 2007 International Conference on*. China, 2007. v. 2, p. 1036–1040. Citado na página 28.
- 22 JIANG, W.; ZHANG, Y.; XIE, J. A particle swarm optimization algorithm with crossover for vehicle routing problem with time windows. In: IEEE. *Computational Intelligence in Scheduling, 2009. CI-Sched'09. IEEE Symposium on*. USA, 2009. p. 103–106. Citado na página 28.
- 23 SVED, G.; GINOS, Z. Structural optimization under multiple loading. *International of Mechanical Sciences*, v. 10, p. 803–805, 1968. Citado na página 31.
- 24 ROMSTAD, K.; WANG, C. Optimum design of framed structures. *Journal of Structural Division - ASCE*, v. 94, p. 2817–2845, 1968. Citado na página 31.
- 25 VANDERPLAATS, G. N.; MOSES, F. Automated design of trusses for optimum geometry. *Journal of the Structural Division*, ASCE, v. 98, n. 3, p. 671–690, 1972. Citado na página 31.
- 26 PEDERSEN, P. Optimal joint positions for space trusses. *Journal of the Structural Division*, ASCE, v. 99, n. 12, p. 2459–2476, 1973. Citado na página 31.
- 27 ADELI, H.; BALASUBRAMANYAM, K. V. Knowledge-based system for design of bridge trusses. *Journal of Computing in Civil Engineering*, v. 2, n. 1, p. 1–20, 1988. Citado na página 32.
- 28 KIRSCH, U. Synthesis of structural geometry using approximation concepts. *Computers & Structures*, v. 15, p. 305–314, 1982. Citado na página 32.

- 29 ARORA, J.; GOVIL, A. An efficient method for optimal structural design by substructuring. *Computers & Structures*, Elsevier, v. 7, n. 4, p. 507–515, 1977. Citado na página 32.
- 30 HOLAND, J. H. Adaptation in natural and artificial systems. *Ann Arbor: The University of Michigan Press*, 1975. Citado na página 32.
- 31 BETHKE, A. D. Genetic algorithms as function optimizers. *ACM computer science conference*, 1978. Citado na página 32.
- 32 GOLDBERG, D.; SAMTANI, M. Engineering optimization via genetic algorithms. *Ninth Conference on Electronic Computation*, 1986. Citado na página 33.
- 33 JENKINS, W. Towards structural optimization via the genetic algorithm. *Computers & Structures*, v. 40, p. 1321–1327, 1991. Citado na página 33.
- 34 RAJEEV, S.; KRISHNAMOORTHY, C. Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*, v. 118, n. 5, p. 1233–1250, 1992. Citado na página 33.
- 35 HAJELA, P.; LEE, E.; LIN, C.-Y. Genetic algorithms in structural topology optimization. In: *Topology design of structures*. [S.l.]: Springer, 1993. p. 117–133. Citado na página 33.
- 36 RAJAN, S. Sizing, shape, and topology design optimization of trusses using genetic algorithm. *Journal of Structural Engineering*, American Society of Civil Engineers, v. 121, n. 10, p. 1480–1487, 1995. Citado 2 vezes nas páginas 33 e 73.
- 37 LOPEZ, R. H.; LUERSEN, M. A.; CURSI, J. E. S. d. Optimization of hybrid laminated composites using a genetic algorithm. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, scielo, v. 31, p. 269 – 278, 09 2009. Citado na página 33.
- 38 LOPEZ, R.; LUERSEN, M.; CURSI, E. Optimization of laminated composites considering different failure criteria. *Composites Part B: Engineering*, v. 40, n. 8, p. 731 – 740, 2009. Citado na página 33.
- 39 KOIDE, R. M.; FRANÇA, G. v. Z. d.; LUERSEN, M. A. An ant colony algorithm applied to lay-up optimization of laminated composite plates. *Latin American Journal of Solids and Structures*, scielo, v. 10, p. 491 – 504, 05 2013. Citado na página 33.

- 40 KAVEH, A.; ZOLGHADR, A. Truss optimization with natural frequency constraints using a hybridized css-bbbc algorithm with trap recognition capability. *Computers & Structures*, Elsevier, v. 102, p. 14–27, 2012. Citado na página 33.
- 41 XIAO, A. *et al.* Fitness estimation based particle swarm optimization algorithm for layout design of truss structures. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2014, p. 1–11, 2014. Citado na página 33.
- 42 PINCUS, M. Letter to the editor—a closed form solution of certain programming problems. *Operations Research*, v. 16, n. 3, p. 690–694, 1968. Citado na página 34.
- 43 PHOLDEE, N.; BUREERAT, S. Hybrid real-code population-based incremental learning and approximate gradients for multi-objective truss design. *Engineering Optimization*, Taylor & Francis, v. 46, n. 8, p. 1032–1051, 2014. Citado 2 vezes nas páginas 34 e 35.
- 44 LAMBERTI, L.; PAPPALLETTERE, C. Truss weight minimization using hybrid harmony search and big bang-big crunch algorithms. *Metaheuristic Applications in Structures and Infrastructures*, p. 207–240, 2013. Citado 2 vezes nas páginas 34 e 35.
- 45 LI, C.; YANG, S.; KOREJO, I. An adaptive mutation operator for particle swarm optimization. *MIC 2008*, 2008. Citado na página 35.
- 46 MASROM, S. *et al.* Hybridization of particle swarm optimization with adaptive genetic algorithm operators. In: IEEE. *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*. Malásia, 2013. p. 153–158. Citado na página 35.
- 47 KAYA, Y. *et al.* A novel crossover operator for genetic algorithms: ring crossover. *Computing Research Repository*, 2011. Citado na página 35.
- 48 KANWAL, M. S.; RAMESH, A. S.; HUANG, L. A. A novel pseudoderivative-based mutation operator for real-coded adaptive genetic algorithms. *F1000Research*, Faculty of 1000 Ltd, v. 2, 2013. Citado na página 35.
- 49 ARORA, J. *Introduction to Optimum Design*. [S.l.]: Elsevier Science, 2004. Citado na página 36.

- 50 TORII, A. J.; LOPEZ, R. H.; LUERSEN, M. A. A local-restart coupled strategy for simultaneous sizing and geometry truss optimization. *Latin American Journal of Solids and Structures*, v. 8, n. 3, p. 335–349, 2011. Citado na página 38.
- 51 TALATAHARI, S.; KAVEH, A.; SHEIKHOESLAMI, R. Chaotic imperialist competitive algorithm for optimum design of truss structures. *Structural and Multidisciplinary Optimization*, Springer, v. 46, n. 3, p. 355–367, 2012. Citado na página 38.
- 52 KAVEH, A.; TALATAHARI, S. Optimum design of skeletal structures using imperialist competitive algorithm. *Computers & structures*, Elsevier, v. 88, n. 21, p. 1220–1229, 2010. Citado na página 38.
- 53 SABOUR, M.; ESKANDAR, H.; SALEHI, P. Imperialist competitive ant colony algorithm for truss structures. *World Applied Sciences Journal*, v. 12, n. 1, p. 94–105, 2011. Citado na página 38.
- 54 CHENEY, E.; KINCAID, D. *Numerical mathematics and computing*. [S.l.]: Cengage Learning, 2012. Citado na página 40.
- 55 ATASHPAZ-GARGARI, E.; LUCAS, C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: IEEE. *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. Singapura, 2007. p. 4661–4667. Citado 5 vezes nas páginas 40, 43, 46, 47 e 48.
- 56 TORII, A.; BIONDINI, F. A simple geometry optimization method for statically indeterminate trusses. In: *8th World Congress on Structural And Multidisciplinary Optimization*. Portugal: [s.n.], 2009. Citado na página 41.
- 57 BENDSOE, M. P.; SIGMUND, O. *Topology optimization: theory, methods and applications*. [S.l.]: Springer, 2003. Citado na página 41.
- 58 MARTINI, K. Harmony search method for multimodal size, shape, and topology optimization of structural frameworks. *Journal of Structural Engineering*, American Society of Civil Engineers, v. 137, n. 11, p. 1332–1339, 2011. Citado na página 73.
- 59 ARORA, J. *Introduction to Optimum Design*. 2. ed. Amsterdam: Elsevier, 2004. Citado 2 vezes nas páginas 101 e 102.

Apêndices

APÊNDICE A – Revisão de conceitos de otimização

Na literatura de otimização são definidos dois tipos de mínimo, mínimo global e mínimo local, como pode ser visto na Figura 3(59).

A.1 Mínimos

A.1.1 Mínimo global

O mínimo global é a solução do problema que se busca resolver: o(s) ponto(s) no espaço n-dimensional formado pelas n variáveis no qual a função objetivo apresenta o seu menor valor, sem violar nenhuma restrição. Um problema de otimização pode apresentar vários mínimos globais desde que estes tenham o mesmo custo. O mínimo global é representado como \mathbf{x}^* e é definido na Equação A.1a. O conjunto S referido na Equação A.1a é o conjunto formado pelos pontos pertencentes à região restrita (sem violações de restrições) do espaço n-dimensional antes mencionado. O conjunto S é definido na Equação A.1b:

$$f(\mathbf{x}^*) < f(\mathbf{x}) \forall \mathbf{x} \in S \quad (\text{A.1a})$$

$$S = \{\mathbf{x} | h_j(\mathbf{x}) = 0, j = 1 \text{ a } p; g_i(\mathbf{x}) \leq 0; i = 1 \text{ a } m\} \quad (\text{A.1b})$$

onde h_j representa as p restrições de igualdade e g_i representa as m restrições de desigualdade.

A.1.2 Mínimo local

Um mínimo local é um ponto no qual a função objetivo tem valor menor do que qualquer outro ponto da vizinhança deste pertencente

à região restrita. Desta forma o mínimo local é definido como \mathbf{x}^* na Equação A.2a e a vizinhança como N na Equação A.2b.

$$f(\mathbf{x}^*) < f(\mathbf{x}) \forall \mathbf{x} \in N \quad (\text{A.2a})$$

$$N = \{\mathbf{x} | \mathbf{x} \in S \text{ com } \|\mathbf{x} - \mathbf{x}^*\| < \delta\} \quad (\text{A.2b})$$

para um δ pequeno. Caso o problema estudado não seja convexo, não se pode garantir que qualquer mínimo local seja também um mínimo global. Um problema recorrente na aplicação de métodos de direção de descida para problemas de engenharia é o de o algoritmo ficar preso em um mínimo local, distante do mínimo global.

A.2 Convexidade

Um conjunto S qualquer é dito convexo se para quaisquer dois pontos pertencentes à este, possa-se afirmar que todos os pontos pertencentes à reta que os une também pertençam ao conjunto S (59).

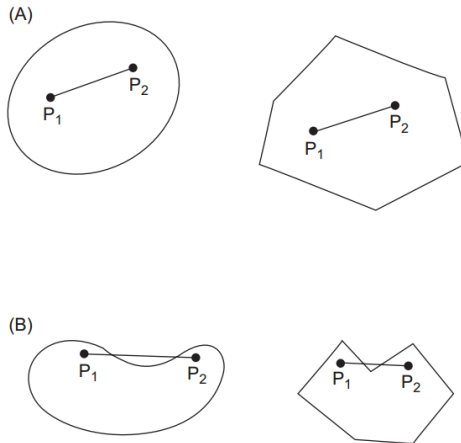


Figura 46 – Exemplos de conjuntos convexos(A) e não convexos(B).
Fonte:Arora(59)

O segmento de linha que une quaisquer dois pontos x_1 e x_2 é dado pela Equação A.3

$$x = \alpha x_2 + (1 - \alpha)x_1; \quad 0 \leq \alpha \leq 1 \quad (\text{A.3})$$

Então, se para qualquer par de pontos x_1 e x_2 pertencentes a S , todos os pontos x da Equação A.3 pertencerem também à S , como apresentado na Equação A.4, pode-se afirmar que S é um conjunto convexo.

$$x(\alpha) \in S \quad \forall 0 \leq \alpha \leq 1 \quad (\text{A.4})$$

Um problema de otimização definido por:

Encontrar $\mathbf{x} = [x_1, x_2, \dots, x_j]$

Que minimiza $f(\mathbf{x})$

Sujeita à $\mathbf{h}(\mathbf{x}) = 0$

Sujeita à $\mathbf{g}(\mathbf{x}) \leq 0$

será convexo se f e g forem convexos em seus domínios. Todas as restrições de igualdade h são convexas.

Caso algum problema de otimização seja convexo, pode-se ter certeza que existe um mínimo local e que este é também o mínimo global do domínio. Desta forma, problemas convexas podem ser resolvidos com um baixo custo computacional empregando-se métodos baseados em hessiana e gradiente.