

Estudo de Sensibilidade do Algoritmo de Colônia de Vagalumes para um Problema de Engenharia Envolvendo Dimensionamento de Treliças

L. L. M. PEREIRA¹, D. C. SANTOS², M. H. M. MORAES³,
G. M. GONÇALVES FILHO⁴, E. M. ANCIOTO JUNIOR⁵,
W. M. PEREIRA JUNIOR^{6*} e M. J. P. DANTAS⁷

Recebido em 8 de setembro de 2020 / Aceito em 14 de setembro de 2020

RESUMO. A treliça é uma estrutura triangular rígida, com resistência aos esforços normais, podendo ser utilizada em telhados, mezaninos, torres de energia, de telecomunicações e pontes. Logo é possível afirmar que esse sistema estrutural apresenta uma grande relevância no cenário da engenharia de estruturas. Nesta pesquisa é utilizado um método probabilístico de otimização global baseado em inteligência coletiva ou inteligência de enxame, com aplicações promissoras em diversos campos das ciências aplicadas, o Algoritmo de Colônia de Vagalumes (ACV), para determinação do peso mínimo de uma treliça de *benchmark*. Foi conduzida uma análise de sensibilidade com os parâmetros do algoritmo como: população (N_{pop}), número de iterações (N_{gen}), parâmetro de aleatoriedade α , fator de atratividade β e parâmetro de absorção de luz (γ). A treliça utilizada nos testes foi uma estrutura de *benchmark* com 10 barras e essa foi otimizada obtendo um valor de peso mínimo em torno de 2284 kg, tal valor quando comparado a outros trabalhos da literatura mostram a efetividade do método adotado nesse trabalho. O *software* utilizado para as implementações e simulação das treliças foi o MATLAB®.

Palavras-chave: otimização, treliças planas, estruturas metálicas, colônia de vagalumes.

*Autor correspondente: Wanderlei Malaquias Pereira Junior – E-mail: wanderlei.junior@ufg.br

¹Pontifícia Universidade Católica de Goiás (PUCGO), GO, Brasil – E-mail: luanalillia.m@gmail.com
<https://orcid.org/0000-0003-3551-2017>

²Pontifícia Universidade Católica de Goiás (PUCGO), GO, Brasil – E-mail: douglas775@live.com
<https://orcid.org/0000-0001-9715-6527>

³Universidade Federal de Catalão (UFCAT), GO, Brasil – E-mail: matheus.h.h@hotmail.com <https://orcid.org/0000-0002-7285-1344>

⁴Universidade Federal de Catalão (UFCAT), GO, Brasil – E-mail: geraldomagela1905@hotmail.com
<https://orcid.org/0000-0001-9704-9000>

⁵Pontifícia Universidade Católica de Goiás (PUCGO), GO, Brasil – E-mail: junior.ancioto@gmail.com
<https://orcid.org/0000-0001-7803-0937>

⁶Universidade Federal de Catalão (UFCAT), GO, Brasil – E-mail: wanderlei.junior@ufg.br <https://orcid.org/0000-0002-7404-3666>

⁷Pontifícia Universidade Católica de Goiás (PUCGO), GO, Brasil – E-mail: mjpdantas@gmail.com
<http://orcid.org/0000-0002-1998-1801>

1 INTRODUÇÃO

A tomada de decisões em vários campos da ciência é comumente realizada com o uso de conceitos de otimização. Sendo que essa tomada de decisão é baseada pela busca de soluções para uma determinada função, denominada Função Objetivo (FO), sendo que essa solução normalmente extremiza (valores de máximo ou mínimo) a função de referência, respeitando um determinado conjunto de restrições que variam de acordo com o problema estudado [20].

Os algoritmos de otimização são divididos em diversos subgrupos podendo destacar os seguintes: (a) métodos determinísticos; e (b) métodos probabilísticos. Os métodos determinísticos são caracterizados por um modelo matemático que determina os resultados, exatamente, a partir das condições iniciais, ou seja, dada uma certa entrada o mesmo apresentará uma única saída repetidamente. Já os métodos probabilísticos se caracterizam por permitirem diferentes pontos de entrada, podendo então gerar diferentes pontos de saída, ou seja, ao contrário dos métodos determinísticos, estes não ficarão presos em uma única solução.

Nesse artigo é utilizado o Algoritmo de Colônia de Vagalumes (ACV), classificado como probabilístico e baseado no conceito de inteligência de enxames. O mesmo se caracteriza por ser um método de otimização global com aplicações bem sucedidas em diversos campos das ciências aplicadas [12, 14, 17, 23, 28, 34, 35].

Apesar dessas diversas aplicações o desempenho desses algoritmos é fortemente afetado pelos parâmetros de entrada do algoritmo. Portanto, testes de sensibilidade paramétrica tornam-se instrumentos de interesse para nortear a escolha desses parâmetros em determinados campos da ciência, como a escolha de parâmetros do modelo de otimização na engenharia estrutural.

Especificamente em projetos de engenharia estrutural, um fator bastante importante no planejamento é o custo, por conta disso, a maior parte dos problemas de otimização de engenharia tem como finalidade encontrar uma solução que permita utilizar a menor quantidade possível de recursos, sem que haja perda em sua produtividade. Logo no caso de projetos estruturais é muito comum a utilização de funções objetivo que estudem o custo [1, 19, 21, 26] ou peso total da estrutura [5, 7, 13, 22].

Em função dessa necessidade de verificação da funcionalidade dos algoritmos de otimização, os problemas de *benchmark*, ou testes de *benchmark*, são desenvolvidos de forma a compará-los e distingui-los (um algoritmo do outro). Esses testes são de suma importância, uma vez que fornecem parâmetros que ajudam como referência para análises do desempenho de um algoritmo em um determinado problema.

Portanto, nesse artigo objetiva-se o uso de técnicas numéricas baseadas em inteligência de enxame para otimizar o peso de uma estrutura de treliça de *benchmark* de forma a criar subsídios técnicos para uma análise de sensibilidade paramétrica. Deve-se salientar que tal tipo de problema é bastante tradicional na área de engenharia de estruturas.

O artigo está subdividido em cinco seções. A seção 2 é destinada a descrição do algoritmo otimizador utilizado nas análises de sensibilidade paramétrica. A seção 3 diz respeito aos procedimentos de análise e as seções 4 e 5 são destinadas à análise dos resultados e conclusões, respectivamente.

2 O ALGORITMO DE COLÔNIA DE VAGALUMES

A Inteligência de Enxames (IE) - também conhecida como inteligência coletiva ou de colônias - pode ser definida como a capacidade que um grupo de indivíduos tem de encontrar, em esforço conjunto, a solução para um determinado problema. Isto significa que os mesmos compartilham informações, podendo assim dizer que possuem consistência cognitiva, permitindo a capacidade de deduzir e modificar o espaço local. Com isso, podemos dizer que tais modificações acontecem como movimentações em um espaço sócio cognitivo.

Um dos métodos percursoros desse modelo de otimização é o PSO (*Particle Swarm Optimization*), proposto em 1995 pelo psicólogo social James Kennedy e o engenheiro elétrico Russel C. Eberhart. Inicialmente o mesmo foi desenvolvido para solucionar problemas de otimização com variáveis contínuas.

Desde sua autoria o PSO vem sofrendo modificações [10, 30] e influenciando a criação de novos métodos como, por exemplo, o algoritmo de Colônia de Vagalumes (ACV).

O ACV foi desenvolvido por Xin-She Yang [31] e é inspirado na bioluminescência produzida por esses tipos de insetos e as interações ocorridas entre os indivíduos devido a esse fator. Segundo [31] para construção do ACV são necessárias as seguintes bases: (a) Presume-se que todos os vagalumes têm um único sexo, e independentemente de possuir um único sexo são atraídos entre si; (b) A capacidade de atração de cada vagalume é proporcional ao seu brilho, e esse diminui de acordo com o aumento da distância entre os indivíduos da população; (c) No caso de não existência de nenhum vagalume mais brilhante que outros, os vagalumes irão se mover aleatoriamente; e (d) A intensidade de cada vagalume é determinada pela avaliação da Função Objetivo.

O algoritmo 1 apresenta o fluxograma de funcionamento do ACV e suas demais particularidades. Sendo que N_{pop} é dado como o tamanho da população, N_{gen} é o número de iterações ou gerações, β , γ e α são parâmetros internos do algoritmo.

A Equação 2.1 apresenta o movimento da população n -dimensional de vagalumes. O valor de β é dado pela Equação 2.2 e é tido como a atratividade percebida pelo vagalume; β_0 é tido como a atratividade para uma distância $r_{ij}=0$ (entre os vagalumes i e j) sendo que r_{ij} (ver Equação 2.3) é dado como a distância Euclidiana entre os vagalumes i e j ; γ é o parâmetro de absorção da luz, sendo $\gamma \in [0, \infty)$; e k é a k -ésima componente do vetor de variáveis de projeto X .

$$X_{i,k}^{t+1} = X_{i,k}^t + \beta(X_{j,k}^t - X_{i,k}^t) + \alpha(rand_k - 0,50) \quad (2.1)$$

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \cong \frac{\beta_0}{1 + \gamma r_{ij}^2} \quad (2.2)$$

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{i=0}^k (X_{i,k} - X_{j,k})^2} \quad (2.3)$$

Algoritmo 1 Algoritmo de Colônia de Vagalumes

Entrada: $\beta, \gamma, \alpha, Npop, Ngen$
Saída: Variáveis de projeto que minimizam ou maximização a FO

```

1 início
2   Gera população inicial  $X_k$ 
3   while  $t_i Ngen$  do
4      $FO^t \leftarrow X^t$ 
5     Ranqueia a população de vagalumes (Melhor FO para pior FO)
6     for  $i=Npop$  do
7       for  $j=Npop$  do
8         if  $FO_i^t > FO_j^t$  then
9           | Movimenta o vagalume conforme Equação 2.1
10          end
11        end
12      end
13    end
14    retorna  $FO(X_k^{Ngen})$ 
15 fim
```

[6] afirma que o parâmetro β é uma característica primordial do algoritmo visto que esse é um dos fatores responsáveis pela característica de diversidade dos indivíduos. Caso β igual a zero o algoritmo se transforma em um modelo de caminhada randômica. [9] diz que na maioria das aplicações o parâmetro γ pode variar entre 0,01 e 100. O parâmetro α , por sua vez, corresponde à parte probabilística do processo de otimização, sendo $\alpha \in [0, \infty)$ e na maioria dos casos tem seu valor atribuído entre 0 e 1. Segundo [32] o parâmetro α é tido como passo do algoritmo e o termo $rand_k$ é um vetor de números aleatórios que pode variar entre 0 e 1. Já [31] pondera que em situações onde deseja-se melhorar a diversidade dos indivíduos da população pode-se alterar o valor de α estático para uma função α de atualização dinâmica.

A Figura 1 apresenta a topologia de movimentação dos vagalumes. Tal morfologia é dita como topologia de estrela ou topologia totalmente atraída [29] pois todos os vagalumes do sistema tem interconexão, sendo que essas partículas podem se comunicar de forma direta [3].

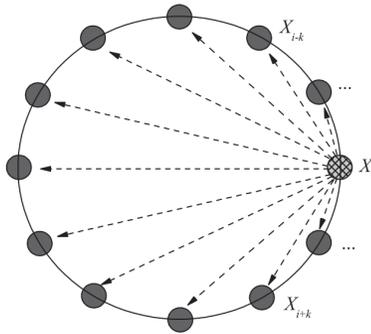


Figura 1: Esquema genérico da topologia de comunicação entre os vagalumes adaptada de [29].

Para o algoritmo ACV foram utilizadas duas modificações em relação à proposta original de [31]. A primeira modificação diz respeito ao termo de atratividade do algoritmo. Nesse caso a variável β teve seu parâmetro r_{ij} modificado para uma distância Euclidiana normalizada, conforme Equação 2.4. Vale ressaltar que X_{max} e X_{min} equivalem aos limites da variável de projeto do problema.

$$r_{ij} = \frac{\|X_i - X_j\|}{\|X_{max} - X_{min}\|} \tag{2.4}$$

Além disso também foi introduzido uma aleatoriedade no cálculo do segundo termo da Equação 2.1. Tal situação implica que em todos os passos do processo iterativo o valor de β não se aproxime de zero, evitando assim situações de transformação do algoritmo em um modelo de caminhada randômica.

A segunda modificação diz respeito a terceira parcela da Equação 2.1 que foi modificada conforme proposto em [11]. Nesse caso o vetor randômico foi modificado para uma distribuição uniforme de -1 até 1. Para o parâmetro α foi adotada uma função de atualização dinâmica gerando assim um processo de decaimento dessa variável em questão. Portanto a nova função de movimento é dada pela Equação 2.5.

$$X_{i,k}^{t+1} = X_{i,k}^t + \beta rand_k(X_{j,k}^t - X_{i,k}^t) + \alpha(rand_k) \tag{2.5}$$

$$\alpha^{t+1} = 0,98\alpha^t \tag{2.6}$$

A utilização de um modelo de redução dinâmica do parâmetro α permite que o algoritmo trabalhe com uma maior diversificação nas iterações iniciais, ou seja, uma busca global. E quando o mesmo estiver em um estado de convergência global [33] o parâmetro α estará sendo reduzido a cada iteração, criando assim um critério automático de busca local. Tal modificação colabora para uma melhoria de convergência do algoritmo e pode ser vista também em [9].

O algoritmo ACV, apesar de novo, tem sido amplamente utilizado em vários campos da ciência [24,35], para resolução de problemas de engenharia [25,27,36] e em especial o uso no estudo de treliças [8,9].

3 PROCEDIMENTO PARA ANÁLISE DE SENSIBILIDADE DO ACV

Nesse trabalho são aplicados testes de sensibilidade ao algoritmo ACV apresentado na seção 2. Como base para esses testes foi utilizado um problema de *benchmark* [2, 8, 15, 27] voltado à Engenharia Civil, sendo este direcionado ao dimensionamento e verificação de treliças metálicas. Neste sentido são apresentados os procedimentos para verificação de sensibilidade paramétrica do algoritmo proposto por [31].

O problema base para análise de sensibilidade é descrito na Figura 2 e as propriedades dessa treliça são observadas a seguir: (a) Densidade do material da treliça $\rho=2770 \text{ kg/m}^3$; (b) Módulo de elasticidade do material $E=69 \text{ GPa}$; (c) A tensão admissível (σ_{lim}) dos membros da treliça é de $\pm 172 \text{ MPa}$; (d) O limite de deslocamento vertical (δ_{lim}) nas conexões da estrutura é de $\pm 5,08 \text{ cm}$, e (e) Considera-se a condição de carregamento da estrutura treliça plana conforme a Equação 3.1, em que o primeiro subíndice representa o nó e o segundo subíndice (y) o grau de liberdade referente à direção vertical.

$$P_{2y} = P_{4y} = -450 \text{ kN} \quad (3.1)$$

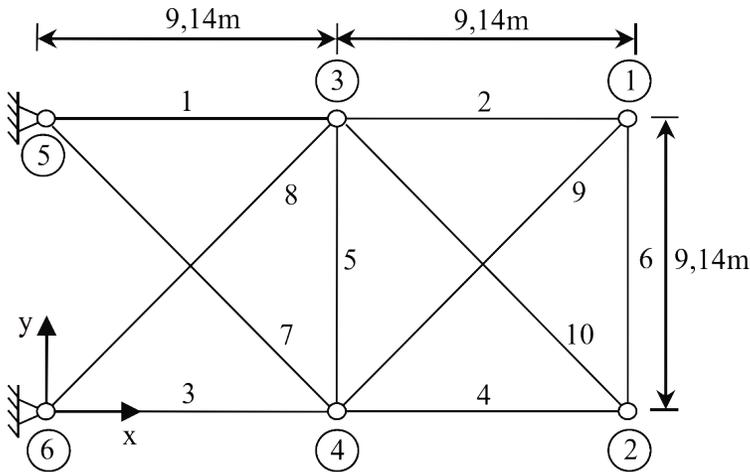


Figura 2: Treliça de *benchmark* de 10 barras [2].

A função a ser minimizada ou Função Objetivo (FO) do problema é dada pela Equação 3.2 e tal problema está sujeito às restrições dadas pelas Equações 3.3 a 3.5. As Equações 3.3 e 3.4 contemplam restrições de tensão e limite de deslocamentos vertical, respectivamente, enquanto a Equação 3.5 contempla os limites laterais da área das barras que, nesse caso, é dada como a variável de projeto da otimização.

$$W = \sum_{i=1}^m A_i L_i \rho \tag{3.2}$$

$$\frac{\sigma_i}{\sigma_{lim}} - 1 \leq 0 \tag{3.3}$$

$$\frac{\delta_{nó}}{\delta_{lim}} - 1 \leq 0 \tag{3.4}$$

$$A_{min} \leq A_i \leq A_{max} \tag{3.5}$$

Em que W é o peso total da treliça, A_i denota a área da seção transversal da barra i (m indica o total de barras) da treliça, L_i é o comprimento da barra i , σ_i é tensão em cada uma das barras da treliça e $\delta_{nó}$ é o deslocamento vertical dos nós da treliça determinado por meio de uma análise estrutural de caráter linear, portanto não foram consideradas não linearidades no material e geometria da peça estudada. A_{min} e A_{max} são os limites laterais da área da barra de cada elemento, visto que são as variáveis de projeto do problema. O valor de área (A_i) adotado para os extremos das barras é o mesmo adotado por [2] sendo esses valores de 0,645 cm² e 226 cm² para área mínima e máxima, respectivamente.

As variações dos parâmetros internos do algoritmo ACV seguem as mesmas proposições de [6]. O teste padrão ou zero é dado pela seguinte padronização: (a) número da população ($Npop$) igual a 100 indivíduos; (b) número de geração ($Ngen$) igual a 500; (c) parâmetro de aleatoriedade (α) igual a 0,9; (d) fator de atratividade (β) igual a 0,8; e (e) parâmetro de absorção de luz (γ) igual a 1. Cada parâmetro do ACV foi permutado a fim de analisar sua influência na resposta final. A Tabela 1 apresentada a variação de cada um dos parâmetros do método de forma a permitir uma análise de sensibilidade.

Tabela 1: Variação dos parâmetros do algoritmo.

$Npop$	20	40	60	80	100	120	140	160	180	200
$Ngen$	200	400	600	800	1000	1200	1400	1600	1800	2000
α	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
β	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
γ	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00

Para gerar os resultados de sensibilidade paramétrica a rotina de otimização foi executada vinte vezes com os parâmetros fixados de forma a criar distribuições de probabilidade da resposta. No caso desse problema de engenharia em específico a solução mínima de projeto que atende as restrições pré-estabelecidas será comparada com trabalhos já desenvolvidos na área que são: [2, 4, 15, 16]. O *software* utilizado para as implementações foi o MATLAB® [18].

4 RESULTADOS E DISCUSSÕES DA ANÁLISE DE SENSIBILIDADE

Nesta seção são apresentados os resultados e as análises dos parâmetros do algoritmo ACV. No caso, foram realizadas aplicações para um problema de *benchmark* envolvendo dimensionamento de treliças.

4.1 Estudo da influência do tamanho da população e do número de gerações

O primeiro estudo trata da influência da população na resposta final do algoritmo. A Figura 3 apresenta os resultados da análise de sensibilidade do valor da FO para essas variações as quais foram apresentadas anteriormente na Tabela 1 da seção 3.

Analisando os resultados da Figura 3 foi possível observar que a parametrização que resultou no maior peso da treliça corresponde à menor população do algoritmo ($N_{pop} = 20$), fato o qual já era esperado nesse tipo de problema. Para população correspondente a esses 20 indivíduos o desvio padrão do peso da treliça foi de $4,73 \times 10^{-2}$, enquanto na simulação de 200 indivíduos o valor do desvio padrão foi de $2,20 \times 10^{-3}$ com um peso mínimo de 2284,5631 kg. Para uma população de 120 indivíduos o peso obtido foi 2284,5617 kg, sendo este o menor peso encontrado na variação deste parâmetro, totalizando um desvio padrão de $4,22 \times 10^{-3}$.

Ainda foi possível detectar na análise dos resultados, que os valores de população acima de 120 indivíduos não influíram em melhorias significativas na redução do peso total da estrutura.

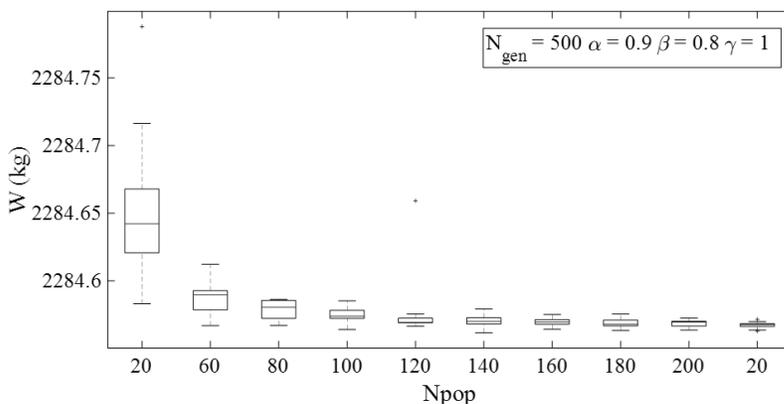


Figura 3: Variação da FO em função do parâmetro N_{pop} .

Analisando a Figura 4 é possível perceber que o ACV encontrou soluções em torno de 2280 kg a partir de 400 gerações. Para 400 gerações o desvio padrão das amostras foi de $3,11 \times 10^{-1}$, enquanto para um N_{gen} igual ou superior a 600 o desvio padrão foi inferior a ordem de $9,00 \times 10^{-5}$, um resultado o qual é bastante satisfatório do ponto de vista estatístico, visto que é possível afirmar que nessas vinte repetições o algoritmo atingiu valor mínimo em todas as análises.

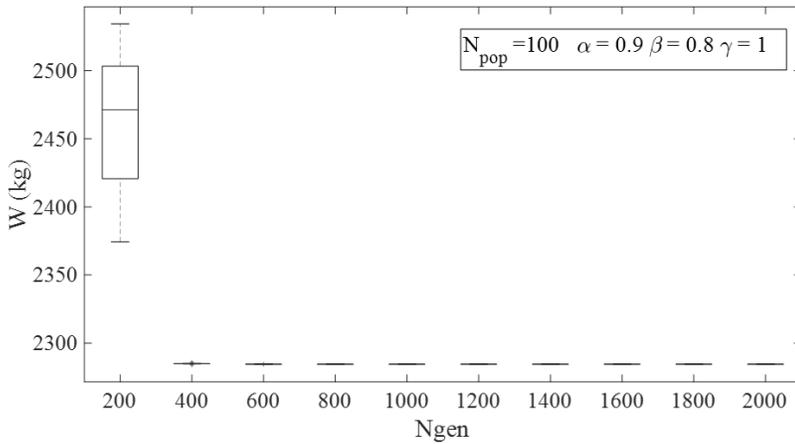


Figura 4: Variação da FO em função do parâmetro N_{gen} .

4.2 Estudo da influência dos parâmetros internos do ACV

Nessa seção são apresentados os resultados de sensibilidade para os parâmetros internos do algoritmo de otimização ACV.

A Figura 5 apresenta os resultados referentes a análise do parâmetro α . A melhor solução encontrada foi com um valor de $\alpha = 0,10$, pois o mesmo obteve um valor mínimo equivalente a 2284,5613 kg com uma amplitude e desvio padrão de $2,00 \times 10^{-4}$ e $4,89 \times 10^{-5}$, respectivamente. O valor de α com a pior solução no cenário analisado foi para $\alpha = 1,00$ com peso mínimo igual a 2284,5678 kg e uma amplitude de $1,22 \times 10^{-2}$.

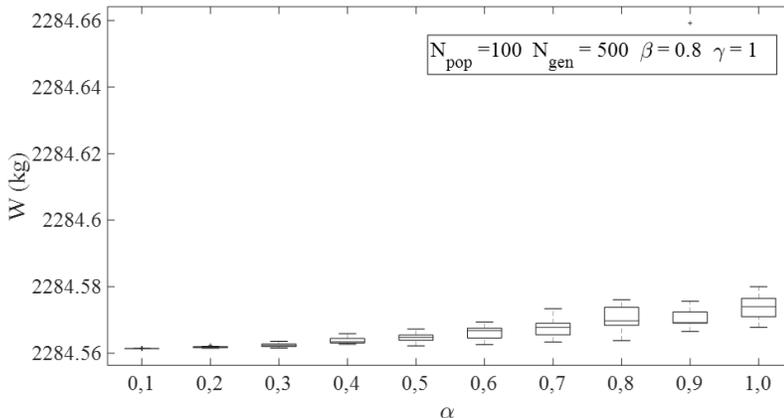


Figura 5: Gráfico da variação da FO em função do parâmetro α .

Nos testes com variação de α , foi possível estabelecer um padrão de comportamento observando as variações do valor final da FO. Verificando a Figura 5 pode-se dizer que quanto maior o valor

de α maior a aleatoriedade introduzida no sistema de partículas, portanto a tendência encontrada é de uma maior dispersão dos resultados em torno da média das amostras. Logo confirma-se aqui a afirmação proposta por Yang [31] de que em análises com o uso de ACV o ideal é que no início do processo de busca utilize-se valores de α mais altos e a medida que o processo iterativo continue esse valor de α tenha reduções, pois espera-se que se o sistema entre em processo de convergência tornando a busca pela melhor resposta um processo de busca local.

A Figura 6 apresenta os resultados obtidos para as variações do parâmetro β . Ao analisar os resultados, observa-se que a melhor solução encontrada foi para uma variação $\beta = 0,50$, pois o mesmo obteve um valor mínimo de 2284,5640 kg com um desvio padrão igual a $5,10 \times 10^{-3}$. Embora seja o melhor resultado da série tal simulação gerou um *outlier* de 2284,5874 kg. Para valores de β maiores que 0,50 o resultado do menor valor de FO se aproxima do peso de 2284,5670 kg com um desvio padrão inferior a $4,00 \times 10^{-3}$.

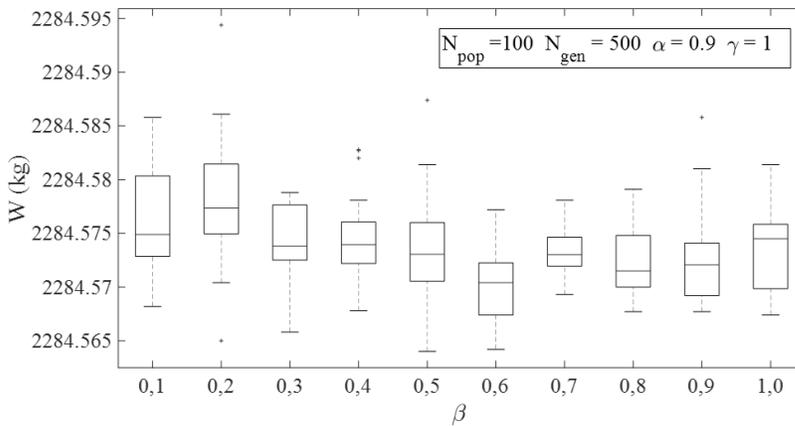


Figura 6: Gráfico da variação da FO em função do parâmetro β .

Conforme [31], a intensidade da luz de um vagalume é fundamental para o ACV, pois tal característica pondera a influência da distância entre os vagalumes no processo de geração de novos indivíduos, ou seja, a intensidade de sua luz decresce à medida que sua distância aumenta. Com isso, pode-se dizer que os dois parâmetros, fator de atratividade (β) e fator de absorção de luz (γ) estão correlacionados.

Na Figura 7 são apresentadas as respostas da FO em relação às variações do parâmetro γ . Ao se observar os resultados, pode-se afirmar que o ACV obteve a melhor solução com $\gamma = 0,2$, pois este atingiu o menor valor, com peso igual à 2284,5637 kg, amplitude de $1,70 \times 10^{-2}$ e um desvio padrão de $4,38 \times 10^{-3}$. Resultados similares foram obtidos para um valor de $\gamma = 1$, sendo encontrado para este um peso mínimo de 2284,5639 kg, com amplitude de $2,14 \times 10^{-2}$ e um desvio de $4,87 \times 10^{-3}$, porém nessa variação de $\gamma = 1$ foi encontrado um *outlier* de 2284,5853 kg.

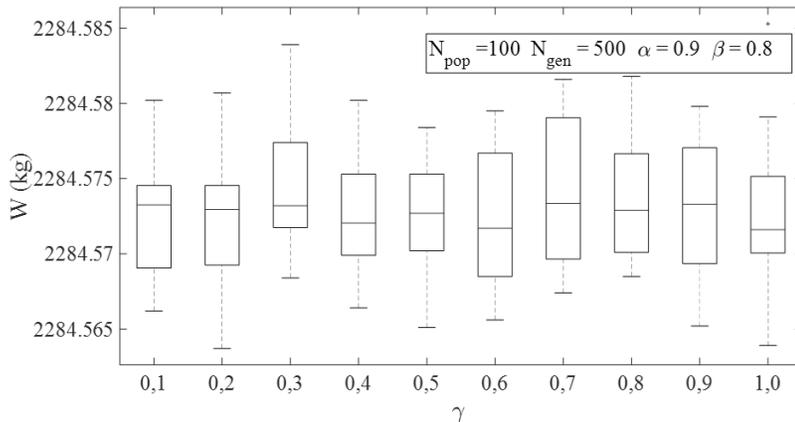


Figura 7: Gráfico da variação da FO em função do parâmetro γ .

4.3 Comparativo de resultados

Para comparativo de resultados criou-se a Tabela 2 onde foi possível comparar os resultados obtidos nesse trabalho com outras bibliografias. Tal tabela apresenta as soluções ou valores de área A_i de cada barra a partir do melhor resultado encontrado de trabalhos da literatura que relatam a análise do problema de *benchmark* de uma treliça de 10 barras.

Para estabelecer uma comparação entre os resultados o algoritmo ACV foi executado por 20 vezes com a seguinte configuração: (a) número da população (N_{pop}) igual a 120 indivíduos; (b) número de geração (N_{gen}) igual a 600; (c) parâmetro de aleatoriedade (α) igual a 0,9; (d) fator de atratividade (β) igual a 0,8; e (e) parâmetro de absorção de luz (γ) igual a 1. A configuração foi obtida como ideal após a análise de sensibilidade e recomendação de outras fontes bibliográficas.

Em uma breve análise desses resultados, pode-se concluir que os resultados obtidos pelo ACV foram satisfatórios quando comparados aos resultados obtidos em outras bibliografias. Quando comparado ao valor obtido no trabalho de [2] é possível afirmar que os resultados alcançados na (FO), com algoritmo do ACV, ultrapassaram em apenas 1,09%.

Para as outras referências como [4, 15, 16] o valor mínimo encontrado no ACV foi superior em cerca de 8,00% fato que permite dizer que o mesmo é um algoritmo bastante eficiente para tal tipo de problema estrutural.

Ainda sobre o resultado referente ao ACV é possível concluir que na melhor resposta das 20 repetições o algoritmo já obteve um padrão de convergência global com cerca 350 iterações. Tal situação é verificada na Figura 8.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Em suma, o artigo verificou a viabilidade da utilização de técnicas de otimização bio-inspirada baseadas em inteligência de enxame para determinação do peso mínimo em uma treliça de

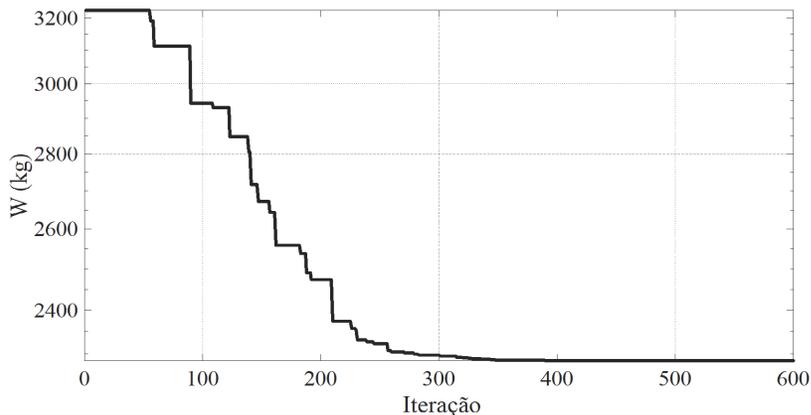


Figura 8: Comportamento da convergência para o *setup* ideal.

Tabela 2: Comparativo entre os diâmetros das barras e valor de FO.

Barras e FO	ACV	PSO [16]	PSO [15]	AGA [2]	TBLO [4]
A1 (cm^2)	195,00	215,90	193,50	197,00	216,10
A2 (cm^2)	1,00	0,70	10,50	0,65	10,50
A3 (cm^2)	148,00	149,50	193,50	180,00	147,70
A4 (cm^2)	98,00	99,90	87,10	87,90	91,60
A5 (cm^2)	1,00	23,50	10,50	0,65	10,50
A6 (cm^2)	3,00	0,80	11,60	0,65	10,50
A7 (cm^2)	48,00	54,10	74,20	51,00	51,40
A8 (cm^2)	134,00	150,60	121,30	125,00	147,70
A9 (cm^2)	1,00	1,20	11,60	0,65	10,50
A10 (cm^2)	138,00	148,50	141,90	124,00	141,90
FO (kg)	2284,56	2508,10	2509,30	2260,00	2490,60

benchmark, com base em uma análise de sensibilidade dos parâmetros. Perspectiva a qual foi reconhecida nos resultados obtidos. Portanto, pode-se afirmar que a ferramenta de otimização bio-inspirada é promissora para aplicações com foco em problemas de engenharia.

Baseado nos resultados obtidos, pode-se destacar que, conforme o número de gerações aumenta, por consequência, a população é renovada gerando melhores indivíduos, tornando maior a chance de alcançar a solução objetivo, comportamento esse que é comum entre os algoritmos bio-inspirados. Foi possível detectar que valor acima de 600 gerações já apresentavam um valor mínimo satisfatório e estabilizado do ponto de vista da análise estatística realizada, visto que o desvio padrão e a amplitude permaneciam constantes. Em relação ao parâmetro N_{pop} , verificou-se que sua influência provocou maiores dispersões para um número de população inferior a 120 indivíduos.

É importante salientar que tanto o número da população como o número de gerações são fatores que influenciam diretamente no aumento do gasto computacional por estar diretamente relacionado ao número de avaliações da FO, que nesse caso se trata de um problema não linear relacionado ao dimensionamento de treliças. Para o padrão de 600 gerações o número de avaliações da função objetivo foi de $6,01 \times 10^4$, enquanto que para soluções com 1000 gerações e 2000 gerações esse valor chegou ao total de $1,001 \times 10^5$ e $2,001 \times 10^5$ avaliações da Função Objetivo, respectivamente.

Sobre as variações dos parâmetros β e γ , pode-se afirmar que as diferenças entre as parametrizações foram relativamente baixas, não sendo possível estabelecer um padrão de influência desses na resposta da FO. Sobre o parâmetro α foi encontrado um padrão do tipo quanto maior o valor de α maior a dispersão dos resultados em torno da média. Apesar disso, ressalta-se que esses parâmetros internos em algoritmos bio-inspirados são essenciais para convergência em problemas complexos.

A variação observável nos resultados obtidos em cada parâmetro com o algoritmo ACV acontece porque seu deslocamento no espaço de busca consegue assumir que soluções consideradas satisfatórias se percam ao longo do processo repetido, ou seja, não é assegurado que os resultados encontrados pelo algoritmo sejam sempre melhores em relação ao ótimo da função objetivo (FO). Porém, ainda assim, o algoritmo ACV tem um estado de centralização global atingindo soluções satisfatórias para esse problema de *benchmark* em específico.

Sobre os testes executados com o algoritmo ACV, para a treliça de *benchmark* de 10 barras, pode-se concluir que o método de otimização utilizado gerou resultados satisfatórios na maioria das repetições. Em termos de eficiência, detectou-se que os melhores resultados assumidos para as barras em cada parâmetro se mantiveram replicados, como também aproximados aos dos trabalhos comparados da literatura como pode-se notar na Tabela 2. É importante ressaltar que as soluções geradas pelo ACV foram melhores que os trabalhos de [4, 15, 16]. Com isso, podemos afirmar que o uso do algoritmo colônia de vagalumes é promissor para problemas aplicáveis à engenharia.

É possível verificar que o algoritmo obteve desempenho satisfatório para um problema de *benchmark* de engenharia onde a FO está relacionada ao cálculo do peso mínimo quando comparado ao valor médio de 2442,00 kg encontrado em outras bibliografias. Sendo que a variação entre o melhor e pior valor foram de 1,06% a 8,96% quando comparada à média desses outros autores.

Agradecimentos

Os autores agradecem à CAPES (Coordenação de Aperfeiçoamento de pessoal de Nível Superior) por subsidiar essa pesquisa.

ABSTRACT. The truss is a rigid triangular structure, with resistance to normal load, and can be used on roofs, mezzanines, energy towers, telecommunications and bridges. Soon it is possible to state that this structural system has great relevance in the structural engine-

ering scenario. This research uses a probabilistic method of global optimization based on collective intelligence or swarm intelligence, with promising applications in several fields of applied sciences, the Firefly Algorithm (FA), to determine the minimum weight of a benchmark truss. A sensitivity analysis was carried out with the parameters of the algorithm, such as the population (N_{pop}), the number of iterations (N_{gen}), randomness parameter α , attractiveness factor β , and light absorption parameter (γ). The truss used in the tests was a benchmark structure with 10 bars, and this was optimized, obtaining a minimum weight value of around 2284 kg. When compared to other works in the literature, such value shows the effectiveness of the method adopted in this work. The software used for the implementations and simulation of the trusses was MATLAB®.

Keywords: optimization, trusses, steel structures, firefly algorithm.

REFERÊNCIAS

- [1] G. A & K. PD. Design optimization of reinforced concrete structures. *Comput Concr* 2006;3:313–34, 3 (2006), 313–334. doi:10.12989/CAC.2006.3.5.313.
- [2] F. Aminifar, F. Aminifar & D. Nazapour. Optimal design of truss structures via an augmented genetic algorithm. *Turkish Journal of Engineering and Environmental Sciences*, 37(1) (2013), 56–68.
- [3] S.A.M. Arruda. “Aplicação da otimização por enxame de partículas com topologia ”multi-ring”na estimação de parâmetros de linhas de transmissão”. Master’s thesis, Universidade Federal de Goiás (2015).
- [4] C. Camp & M. Farshchin. Design of space trusses using modified teaching–learning based optimization. *Engineering Structures*, 62-63 (2014), 87–97. doi:10.1016/j.engstruct.2014.01.020. URL <https://doi.org/10.1016/j.engstruct.2014.01.020>.
- [5] C.V. Camp, B.J. Bichon & S.P. Stovall. Design of Steel Frames Using Ant Colony Optimization. *Journal of Structural Engineering*, 131(3) (2005), 369–379. doi:10.1061/(asce)0733-9445(2005)131:3(369). URL [https://doi.org/10.1061/\(asce\)0733-9445\(2005\)131:3\(369\)](https://doi.org/10.1061/(asce)0733-9445(2005)131:3(369)).
- [6] C.C. Carvalho. “Desenvolvimento de um algoritmo de otimização evolutivo auto-adaptativo para a resolução de problemas de otimização com variáveis mistas”. Master’s thesis, Universidade Federal de Goiás (2018).
- [7] N.D. Cesare & M. Domaszewski. A new hybrid topology optimization method based on I-PR-PSO and ESO. Application to continuum structural mechanics. *Computers & Structures*, 212 (2019), 311–326. doi:10.1016/j.compstruc.2018.11.006. URL <https://doi.org/10.1016/j.compstruc.2018.11.006>.
- [8] B. Farshi & A. Alinia-ziazi. Sizing optimization of truss structures by method of centers and force formulation. *International Journal of Solids and Structures*, 47(18-19) (2010), 2508–2524. doi:10.1016/j.ijsolstr.2010.05.009. URL <https://doi.org/10.1016/j.ijsolstr.2010.05.009>.
- [9] A.H. Gandomi, X.S. Yang & A.H. Alavi. Mixed variable structural optimization using Firefly Algorithm. *Computers & Structures*, 89(23-24) (2011), 2325–2336. doi:10.1016/j.compstruc.2011.08.002. URL <https://doi.org/10.1016/j.compstruc.2011.08.002>.

- [10] K. Gholami & E. Dehnavi. A modified particle swarm optimization algorithm for scheduling renewable generation in a micro-grid under load uncertainty. *Applied Soft Computing*, **78** (2019), 496–514. doi:10.1016/j.asoc.2019.02.042. URL <https://doi.org/10.1016/j.asoc.2019.02.042>.
- [11] S.M.K. Heris. Firefly algorithm. Disponível em <https://yarpiz.com/259/ypea112-firefly-algorithm> (2015). Yarpz Team.
- [12] A. Jaafari, S.V.R. Termeh & D.T. Bui. Genetic and firefly metaheuristic algorithms for an optimized neuro-fuzzy prediction modeling of wildfire probability. *Journal of Environmental Management*, **243** (2019), 358–369. doi:10.1016/j.jenvman.2019.04.117. URL <https://doi.org/10.1016/j.jenvman.2019.04.117>.
- [13] S. Jeong, H.K. Seong, C.W. Kim & J. Yoo. Structural design considering the uncertainty of load positions using the phase field design method. *Finite Elements in Analysis and Design*, **161** (2019), 1–15. doi:10.1016/j.finel.2019.04.002. URL <https://doi.org/10.1016/j.finel.2019.04.002>.
- [14] S. Khalifehzadeh & M. Fakhrazad. A modified firefly algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity. *Computers & Industrial Engineering*, **133** (2019), 42–56. doi:10.1016/j.cie.2019.04.048. URL <https://doi.org/10.1016/j.cie.2019.04.048>.
- [15] L. Li, Z. Huang & F. Liu. A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers & Structures*, **87**(7-8) (2009), 435–443. doi:10.1016/j.compstruc.2009.01.004. URL <https://doi.org/10.1016/j.compstruc.2009.01.004>.
- [16] L. Li, Z. Huang, F. Liu & Q. Wu. A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, **85**(7-8) (2007), 340–349. doi:10.1016/j.compstruc.2006.11.020. URL <https://doi.org/10.1016/j.compstruc.2006.11.020>.
- [17] S. Łukasik & S. Żak. Firefly Algorithm for Continuous Constrained Optimization Tasks. In “Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems”. Springer Berlin Heidelberg (2009), pp. 97–106. doi:10.1007/978-3-642-04441-0_8. URL https://doi.org/10.1007/978-3-642-04441-0_8.
- [18] MathWorks. Technical report, Matrix Laboratory (MATLAB) R2019a v.1.8.0 181, <https://www.mathworks.com/products/matlab.html> (2019). URL <https://www.mathworks.com/products/matlab.html>.
- [19] S.S. Mousavi, M. Nikkhah & S. Zare. A comparative study of two meta-heuristic algorithms in optimizing cost of reinforced concrete segmental lining. *Journal of Mining and Environment*, (Online First) (2018). doi:10.22044/jme.2018.7159.1566. URL <http://doi.org/10.22044/jme.2018.7159.1566>.
- [20] J.D. Pintér. Global Optimization: Software, Test Problems, and Applications. In “Nonconvex Optimization and Its Applications”. Springer US (2002), pp. 515–569. doi:10.1007/978-1-4757-5362-2_15. URL https://doi.org/10.1007/978-1-4757-5362-2_15.
- [21] S. Rana, N. Islam, R. Ahsan & S.N. Ghani. Application of evolutionary operation to the minimum cost design of continuous prestressed concrete bridge structure. *Engineering Structures*, **46** (2013), 38–48.

- doi:10.1016/j.engstruct.2012.07.017. URL <https://doi.org/10.1016/j.engstruct.2012.07.017>.
- [22] M. Salar, M.R. Ghasemi & B. Dizangian. Practical optimization of deployable and scissor-like structures using a fast GA method. *Frontiers of Structural and Civil Engineering*, **13**(3) (2018), 557–568. doi:10.1007/s11709-018-0497-z. URL <https://doi.org/10.1007/s11709-018-0497-z>.
- [23] G.L. Scalia, R. Micale, A. Giallanza & G. Marannano. Firefly algorithm based upon slicing structure encoding for unequal facility layout problem. *International Journal of Industrial Engineering Computations*, (2019), 349–360. doi:10.5267/j.ijiec.2019.2.003. URL <https://doi.org/10.5267/j.ijiec.2019.2.003>.
- [24] B. Selvakumar & K. Muneeswaran. Firefly algorithm based feature selection for network intrusion detection. *Computers & Security*, **81** (2019), 148–155. doi:10.1016/j.cose.2018.11.005. URL <https://doi.org/10.1016/j.cose.2018.11.005>.
- [25] R. Sheikholeslami, B.G. Khalili, A. Sadollah & J. Kim. Optimization of reinforced concrete retaining walls via hybrid firefly algorithm with upper bound strategy. *KSCIE Journal of Civil Engineering*, **20**(6) (2015), 2428–2438. doi:10.1007/s12205-015-1163-9. URL <https://doi.org/10.1007/s12205-015-1163-9>.
- [26] R.M. Singh. Design of Barrages with Genetic Algorithm Based Embedded Simulation Optimization Approach. *Water Resources Management*, **25**(2) (2010), 409–429. doi:10.1007/s11269-010-9706-9. URL <https://doi.org/10.1007/s11269-010-9706-9>.
- [27] S. Talatahari, A.H. Gandomi & G.J. Yun. Optimum design of tower structures using Firefly Algorithm. *The Structural Design of Tall and Special Buildings*, **23**(5) (2012), 350–361. doi:10.1002/tal.1043. URL <https://doi.org/10.1002/tal.1043>.
- [28] P. Vishal & A.R. Babu. Firefly Algorithm for Intelligent Context-Aware Sensor Deployment Problem in Wireless Sensor Network. *Journal of Circuits, Systems and Computers*, **28**(06) (2019), 1950094. doi:10.1142/s0218126619500944. URL <https://doi.org/10.1142/s0218126619500944>.
- [29] H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu & Z. Cui. Firefly algorithm with neighborhood attraction. *Information Sciences*, **382-383** (2017), 374–387. doi:10.1016/j.ins.2016.12.024. URL <https://doi.org/10.1016/j.ins.2016.12.024>.
- [30] G. Xu, B. Liu, J. Song, S. Xiao & A. Wu. Multiobjective sorting-based learning particle swarm optimization for continuous optimization. *Natural Computing*, **18**(2) (2016), 313–331. doi:10.1007/s11047-016-9548-3. URL <https://doi.org/10.1007/s11047-016-9548-3>.
- [31] X.S. Yang. “Nature-Inspired Metaheuristic Algorithms”. Luniver Press (2008).
- [32] X.S. Yang. Firefly Algorithm, Stochastic Test Functions and Design Optimisation. *International Journal of Bio-Inspired Computation*, **2**(2) (2010), 78–84.
- [33] S. Yu, S. Zhu, Y. Ma & D. Mao. A variable step size firefly algorithm for numerical optimization. *Applied Mathematics and Computation*, **263** (2015), 214–220. doi:10.1016/j.amc.2015.04.065. URL <https://doi.org/10.1016/j.amc.2015.04.065>.

- [34] J. Zhang, Y. Zhang & Z. Ma. In silico Prediction of Human Secretory Proteins in Plasma Based on Discrete Firefly Optimization and Application to Cancer Biomarkers Identification. *Frontiers in Genetics*, **10** (2019). doi:10.3389/fgene.2019.00542. URL <https://doi.org/10.3389/fgene.2019.00542>.
- [35] X. Zhang, W. Jia, X. Guan, G. Xu, J. Chen & Y. Zhu. Optimized deployment of a radar network based on an improved firefly algorithm. *Frontiers of Information Technology & Electronic Engineering*, **20**(3) (2019), 425–437. doi:10.1631/fitee.1800749. URL <https://doi.org/10.1631/fitee.1800749>.
- [36] G.D. Zhou, M.X. Xie, T.H. Yi & H.N. Li. Optimal wireless sensor network configuration for structural monitoring using automatic-learning firefly algorithm. *Advances in Structural Engineering*, **22**(4) (2018), 907–918. doi:10.1177/1369433218797074. URL <https://doi.org/10.1177/1369433218797074>.