Gebremedhen *et al.*
Adv. Model. and Simul. in Eng. Sci. *(2020) 7:44*
https://doi.org/10.1186/s40323-020-00183-0

Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE **Open Access**

A firefly algorithm based hybrid method for structural topology optimization

Hailu Shimels Gebremedhen[1*], Dereje Engida Woldemichael[2] and Fakhruldin Mohd Hashim[3]

*Correspondence:
hshimels278@gmail.com
[1] Bahir Dar Institute of Technology, Bahir Dar, Ethiopia
Full list of author information is available at the end of the article

Abstract

In this paper, a firefly algorithm based hybrid algorithm through retaining global convergence of firefly algorithm and ability to generate connected topologies of optimality criteria (OC) method is proposed as an alternative method to solve stress-based topology optimization problems. The lower and upper limit of design variables (0 and 1) were used to find initial material distribution to initialize the firefly algorithm based section of the hybrid algorithm. Input parameters, the number of fireflies, and the number of function evaluations were determined before the implementation of the firefly algorithm to solve formulated problems. Since the direct application of the firefly algorithm cannot generate connected topologies, outputs from the firefly algorithm were used as an initial input material distribution for the OC method. The proposed method was validated using two-dimensional benchmark problems and the results were compared with results using the OC method. Weight percentage reduction, maximum stress-induced, optimal material distribution, and compliance were used to compare results. Results from the proposed method showed that the proposed method can generate connected topologies which are free from the interference of end-users, and only depend on boundary conditions or design variables. From the results, the objective function (weight of the design domain) can be further reduced in the range of 5 to 15% compared to the OC method.

Keywords: Firefly algorithm, FEA, Stress based topology optimization

Introduction

Continuum structural topology optimization as a generalized shape optimization problem that has received extensive attention and considerable progress over the past few years. Different methods have been developed based on this concept. One of the most established methods is a homogenization method where a structure is represented by micro scale void materials. Evaluation and orientation of optimal microstructures is one of the major challenges in homogenization. Another approach, named the SIMP method, originally proposed by Bendsoe [1], has gained a general acceptance due to its conceptual simplicity and computational efficiency. However, the existence of transition elements and local convergence has been a major challenge in SIMP method. Filtering techniques and parameter constraints have been used to address these challenges. Evolutionary Structural Topology Optimization/Bi-directional Structural Topology

Springer Open

Optimization (ESO/BESO) methods are other techniques proposed to solve structural topology optimization [2–4]. Due to truss like optimal layouts it generates, ESO-based methods have been recommended to solve problems having pin-jointed connections. The other variation is level set method which is based on the optimization of implicit interfaces. The boundaries move according to the stress on them. This method is very sensitive to the initial guess and easily caught in local minima [5]. For real world problems, globally optimum solutions are preferable and methods which can generate these solutions are getting more attention for solving optimization problems. Metaheuristic algorithm-based methods are also attractive when solving structural topology optimization problems [5–14].

Firefly algorithm (FA) is a nature-based optimization algorithm developed by Xin-She Yang in late 2007 [15, 16]. It is a stochastic, meta-heuristic algorithm which has been implemented for solving different optimization problems. FA is inspired by the flashing behavior of fireflies, whose lighting behavior attracts mating partners and send warnings of a potential predators. This algorithm has been used to solve different numerical problems [17–21] and shows an outstanding performance in terms of convergence and efficiency over the other nature-inspired algorithms, including genetic and particle-swarm optimization algorithms [19, 22–24]. FA and its variants have been applied for different optimization and engineering problems as shown in Fig. 1.

Principle of firefly algorithm

The FA idealizes several aspects of fireflies in nature. For example, a real firefly flashes in discrete patterns, whereas the modeled fireflies are always glowing. Then, three rules are used to model fireflies behavior and govern the algorithm [15]:

1. The fireflies are unisex which leads fireflies to be attracted to another firefly irrespective of their sex.

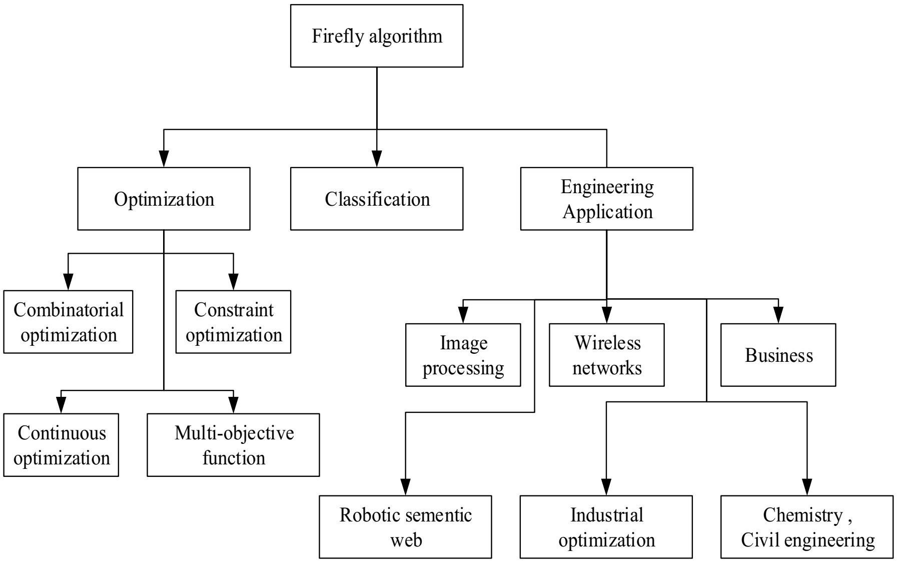


Fig. 1 Taxonomy of firefly application [22]

2. Attractiveness is proportional to the brightness, where both decrease as the distance between two fireflies increase. For any two fireflies within the domain considered, less bright fireflies are attracted to brighter ones. If all fireflies have equal brightness, the fireflies will move randomly.
3. The brightness of a firefly is proportional to the value of the function being maximized and can be considered as the value of the objective function.

Based on these three rules, the basic FA can be summarized by a pseudo code as (Fig. 2):

Step 1. (Generating initial population of solution)

The FA generates a randomly initial population of solutions, x_{ik}.

where $i = 1, 2, \ldots SP$ and $k = 1, 2, 3 \ldots D$.

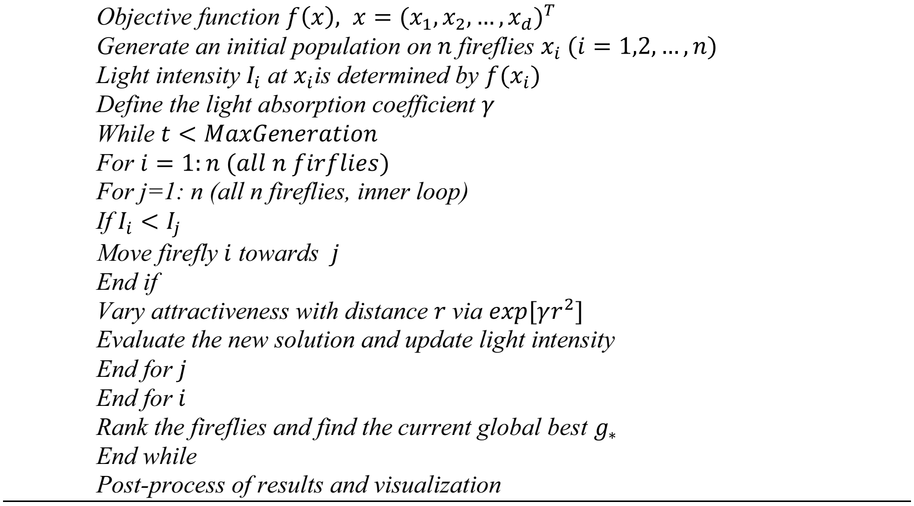
$$\boldsymbol{x_{ik}} = \boldsymbol{l_k} + (\boldsymbol{u_b} - \boldsymbol{l_b}) * \boldsymbol{rand}(\boldsymbol{size}(\boldsymbol{l_b})), \tag{1}$$

where SP is the population size, D is a dimension of the problem, u_b and l_b are the lower and upper limit of the parameter or design variable x_{ik}. After the generation of the initial population, the objective function values for all solutions x_i will be calculated and variable t, which is the number of iteration, is set to 1.

Step 2. (Calculate the new population)

Each solution of the new population is created from the appropriate solution x_i as follows: For each solution x_i, the algorithm examines every solution $x_j, j = 1, 2, 3, \ldots i$, iteratively, starting from $j = 1$. If solution x_j has a higher objective function value than x_i (x_j is brighter than x_i), the parameter values $x_{ik}, k = 1, 2, 3, \ldots, D$ are updated by:

$$x_{ik} = x_{ik} + \beta\left(x_{ik} - x_{jk}\right) + \alpha S_k(rand - 0.5), \tag{2}$$



```
Objective function $f(x),\ x = (x_1, x_2, \ldots, x_d)^T$
Generate an initial population on n fireflies $x_i\ (i = 1,2, \ldots, n)$
Light intensity $I_i$ at $x_i$is determined by $f(x_i)$
Define the light absorption coefficient $\gamma$
While $t < MaxGeneration$
    For $i = 1{:}\, n\ (all\ n\ firflies)$
        For j=1: n (all n fireflies, inner loop)
            If $I_i < I_j$
                Move firefly $i$ towards $j$
            End if
            Vary attractiveness with distance $r$ via $exp[\gamma r^2]$
            Evaluate the new solution and update light intensity
        End for j
    End for i
    Rank the fireflies and find the current global best $g_*$
End while
Post-process of results and visualization
```

Fig. 2 Pseudo code of firefly algorithm [15]

where the first, second and third terms are the Cartesian distances, due to the attraction and randomization term. In the second term, β is a monotonically decreasing exponential function which describes a firefly's attractiveness:

$$\boldsymbol{\beta} = \boldsymbol{\beta}_0 \boldsymbol{e}^{\boldsymbol{\gamma r_{ij}}}, \tag{3}$$

where r_{ij} is the distance between firefly i and firefly j, while β_0 and γ are predetermined algorithm parameters, maximum attractiveness value, and absorption coefficient, respectively. Distance r_{ij} is obtained by the Cartesian distance as:

$$\boldsymbol{r_{ij}} = \left|\boldsymbol{x_i} - \boldsymbol{x_j}\right| = \sqrt{\sum_{\boldsymbol{k}=1}^{\boldsymbol{d}} \left(\boldsymbol{x_{i,k}} - \boldsymbol{x_{j,k}}\right)^2}. \tag{4}$$

Control parameter β_0 describes attractiveness when two fireflies are found at the same point of search space, i.e. at $r = 0$ the variation of attractiveness with increasing distance from a communicated firefly is determined by the control parameter γ.

$\alpha \in [0, 1]$ in the third term in Eq. 2 is a randomization parameter,S_k and $rand_k$ are scaling parameter and random number uniformly distributed between 0 and 1, respectively. The scaling parameters S_k are calculated by:

$$\boldsymbol{S_k} = |\boldsymbol{u_k} - \boldsymbol{l_k}|. \tag{5}$$

In addition, whenever the values of the solution $\boldsymbol{x_{ik}}$ are changed, the FA controls the boundary conditions of created solutions and memorizes the new objective function value instead of the old one. The boundary constraint-handling mechanism used in the FA is given by:

$$\boldsymbol{x_{ik}} = \begin{cases} \boldsymbol{l_k}, & \boldsymbol{if} \quad \boldsymbol{x_{ik}} < \boldsymbol{l_k} \\ \boldsymbol{u_k}, & \boldsymbol{if} \quad \boldsymbol{x_{ik}} > \boldsymbol{l_k} \end{cases}. \tag{6}$$

The last solution obtained by Eq. 6 will be taken as the final solution of the new population which will be transformed the next iteration of the FA.

Step 3. (Reduce the randomization parameter)

The solution quality can be enhanced by reducing the randomization parameter α with a geometric progression reduction scheme which can be expressed as:

$$\boldsymbol{\alpha}(\boldsymbol{t}) = \boldsymbol{\alpha}(\boldsymbol{t} - 1).\boldsymbol{\theta}^{\frac{1}{MCN}}, \tag{7}$$

where MCN is the maximum cycle number, t is the current iteration number, $\theta \in (0, 1]$ is the randomness reduction constant. In most applications, we can use $\theta = 0.95 \sim 0.99$ and $\alpha_0 = 1$ [15].

Step 4. (Record the best solution)

Ranks the fireflies by their light intensity/objectives and memorizes the best solution so far $\boldsymbol{x_{best}}$ and increase the variable $\boldsymbol{t}$ by one.

Step 5. (Check the termination criterion)

If $\boldsymbol{t}$ is equal to the maximum number of iterations then the algorithm is finished, else go to step 2.

In the simplest case for the maximum optimization problem, the brightness I of a firefly at a location x can be chosen *I(x)* $\propto$ *f(x)* However, the attractiveness $\boldsymbol{\beta}$ will be judged by other fireflies. Thus, it varies with the distance r_{ij} between firefly i and j. The distance between any two fireflies i and j at x_i and x_j respectively is the Cartesian distance

$$\boldsymbol{r_{ij}} = \left|\boldsymbol{x_i} - \boldsymbol{x_j}\right| = \sqrt{\sum_{k=1}^{d} \left(\boldsymbol{x_{i,k}} - \boldsymbol{x_{j,k}}\right)^2}. \tag{8}$$

Thus, the light intensity decreases with the distance from its source and the light is also absorbed in the media. The light intensity $I(\gamma)$ varies according to the inverse square law:

$$\boldsymbol{I}(\boldsymbol{\gamma}) = \frac{\boldsymbol{I_s}}{\boldsymbol{r}^2}, \tag{9}$$

where I_s is the intensity at the source. For a given medium with a fixed light absorption coefficient γ, the light intensity I varies with the distance r which can be expressed as:

$$\boldsymbol{I} = \boldsymbol{I_0} \boldsymbol{e}^{-\gamma r}, \tag{10}$$

where I_0 is the original light intensity at zero distance $r \approx 0$. Since a firefly's attractiveness is proportional to the light intensity seen by the adjacent fireflies, which can be defined as:

$$\beta = \boldsymbol{\beta_0} \boldsymbol{e}^{\gamma r^2}, \tag{11}$$

where β_0 is the attractiveness at $r = 0$. The movement of a firefly i to another, more attractive (i.e. brighter) firefly j is determined as:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{\beta_0} \boldsymbol{e}^{-\gamma r^2} \left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \boldsymbol{\alpha} \in_i^t, \tag{12}$$

where the first term is the Cartesian distance between two fireflies, and the second term is due to the attraction. The third term describes the randomization, with α being the randomization parameter having a value of [0, 1] and $\in_i$ is a vector of random numbers drawn from a Gaussian distribution or uniform distribution.

Optimization process

Problem formulation

Based on von Mises stress failure theory a ductile material will fail when the von Mises stress induced in the material is higher than the yield strength of the material. Taking this failure into account, generalized stress constrained topology optimization problem for two- and three-dimensional mass minimization can be defined as:

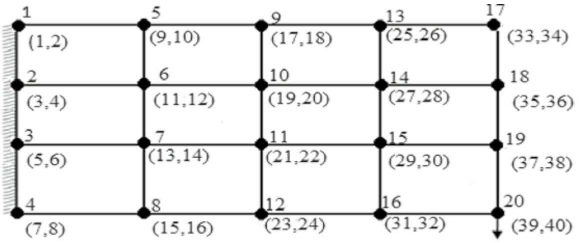


Fig. 3 Discretized design domain

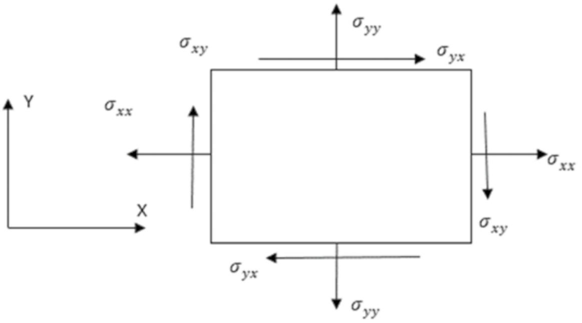


Fig. 4 Two-dimensional element stress state

$$
\begin{aligned}
\underset{X}{Min}\, V &= \sum_{e=1}^{N} x_e^P v_e \\
\text{Subjected to} : \boldsymbol{g}(\boldsymbol{x_e}) &= \frac{\boldsymbol{\sigma^{vm}}}{\boldsymbol{\sigma_{yield}}} < \mathbf{1} \\
: KU &= F \\
0 < x_{min} \leq x_e &\leq 1,
\end{aligned}
\tag{13}
$$

where $\boldsymbol{V}$ is a volume (objective function), $\boldsymbol{N}$ the total number of elements which defines the design domain,$\boldsymbol{e}$ elements within the design domain, $\boldsymbol{ve}$ is volume of each element in the design domain, $\boldsymbol{\sigma_{vm}}$ von Mises stress, $\boldsymbol{\sigma_{yield}}$ is maximum (yield stress). $\boldsymbol{K}$ is a global stiffness matrix, $\boldsymbol{U}$ is a global displacement vector, $\boldsymbol{F}$ a global force vector, $\boldsymbol{x_e}$ is relative density/design variable, $\boldsymbol{x_{min}}$ is the minimum relative density to control the singularity phenomenon associated with the design variable.

Finite element analysis

For 2D problems, all the design domains are assumed to be rectangular and discretized by a square finite element as shown in Fig. 3. Element numbering and the degree of freedoms for each node are also defined in the Figure.

Stress and strain analysis

A two-dimensional stress state consists of three different stress components as shown in Fig. 4, which are the normal stresses $\boldsymbol{\sigma_{yy}}$ and $\boldsymbol{\sigma_{xx}}$ and the shear stress $\boldsymbol{\sigma_{xy}} = \boldsymbol{\sigma_{yx}}$.

Like the stress state, there are three strain components which are directly proportional to the displacements in the respective direction can be expressed as:

$$\boldsymbol{\varepsilon_{xx}} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}, \boldsymbol{\varepsilon_{yy}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{y}}, \boldsymbol{\varepsilon_{xy}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{y}}, \tag{14}$$

where ε_{xx},ε_{xx} and ε_{xy} are the strains in X, Y and XY plane, respectively. u and v are the displacement in X and Y direction, respectively. The stress-induced in the material can be related with the strain as:

$$\{\boldsymbol{\sigma}\} = [\boldsymbol{D}]\{\boldsymbol{\varepsilon}\}, \tag{15}$$

where D is the constitutive matrix, which can be related to the Young's modulus and Poisson's ratio as:

$$[\boldsymbol{D}] = \frac{\boldsymbol{E}}{1 - \boldsymbol{v}^2} \begin{bmatrix} 1 & \boldsymbol{v} & \boldsymbol{0} \\ \boldsymbol{v} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \frac{\boldsymbol{1-v}}{\boldsymbol{2}} \end{bmatrix}, \tag{16}$$

where E is Young's modulus which is a measure of the ratio of axial stress to axial strain in uniaxial tension. Poisson's ratio, respectively. v is the Poisson's ratio which is the negative ratio of lateral strain to axial strain, these values range between 0.25 and 0.35 [25].

Shape function

Shape functions are used to interpolate the displacement field within the design domain. For a linear displacement, the shape functions in a local coordinate, as shown in Fig. 5, can be expressed as shown in Eq. 17. To calculate the stress-induced and elemental stiffness matrix the displacement is approximated using shape functions as:

$$[\boldsymbol{N}] = \begin{bmatrix} \boldsymbol{N1} & \boldsymbol{0} & \boldsymbol{N2} & \boldsymbol{0} & \boldsymbol{N3} & \boldsymbol{0} & \boldsymbol{N4} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N1} & \boldsymbol{0} & \boldsymbol{N2} & \boldsymbol{0} & \boldsymbol{N3} & \boldsymbol{0} & \boldsymbol{N4} \end{bmatrix}, \tag{17}$$

where

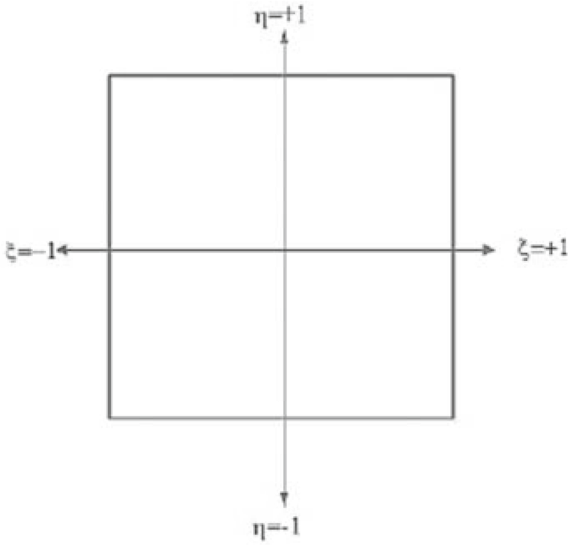


Fig. 5 Square finite element in the Natural coordinate system

$$N1 = \frac{1}{4}(1-\xi)(1-\eta) \quad N3 = \frac{1}{4}(1+\xi)(1+\eta),$$

$$N2 = \frac{1}{4}(1+\xi)(1-\eta) \quad N4 = \frac{1}{4}(1-\xi)(1+\eta).$$

Strain–displacement matrix

The displacement vector can be approximated in terms of the shape functions as:

$$\begin{aligned} &\{\boldsymbol{u}\} = [\boldsymbol{N}]\{\boldsymbol{d}\} \\ &\{\boldsymbol{\varepsilon}\} = [\boldsymbol{\partial}]\{\boldsymbol{u}\} \\ &\{\boldsymbol{\varepsilon}\} = [\boldsymbol{\partial}][\boldsymbol{N}]\{\boldsymbol{d}\}, \end{aligned} \tag{18}$$

where $\{\boldsymbol{d}\}$ is the nodal displacement vector, $\{\boldsymbol{u}\}$ is the displacement vector and $[\boldsymbol{N}]$ is a vector of shape function. For computing the stress, strain, and stiffness the matrix the strain–displacement matrix having a size of 3×8.

The term $[\boldsymbol{\partial}]\{\boldsymbol{N}\}$ is described as a strain displacement matrix which can be expressed as:

$$[\boldsymbol{B}] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}. \tag{19}$$

Solving formulated problems

The proposed hybrid method has two stages for solving the proposed problem: the first stage will generate the values for design variables using the FA algorithm, and generated values of the design variables will be used as input for the second stage, OC method. The design variables at the second stage of the proposed method are updated using the following scheme:

$$\begin{aligned} &\boldsymbol{if}\, \boldsymbol{x_e \beta_e^\eta} \leq \mathbf{max}(\boldsymbol{x_{min}}, \boldsymbol{x_e} - \boldsymbol{m}) \\ &\boldsymbol{x_e^{new}} = \mathbf{max}(\boldsymbol{x_{min}}, \boldsymbol{x_e} - \boldsymbol{m}) \\ &\mathbf{if}\,\mathbf{max}(\mathbf{x_{min}}, \mathbf{x_e} - \mathbf{m}) < \mathbf{x_e}\boldsymbol{\beta}_\mathbf{e}^{\boldsymbol{\eta}} \leq \mathbf{min}(1, \mathbf{x_e} + \mathbf{m}) \\ &\boldsymbol{x_e^{new}} = \boldsymbol{x_e \beta_e^\eta} \\ &\boldsymbol{if}\, \min{(1, \boldsymbol{x_e} + \boldsymbol{m})} < \boldsymbol{x_e \beta_e^\eta} \\ &\boldsymbol{x_e^{new}} = \min{(1, \boldsymbol{x_e} + \boldsymbol{m})}, \end{aligned} \tag{20}$$

where $\boldsymbol{m}$ is a positive limit, which usually takes a value of 0.2, $\boldsymbol{\eta}$ is a numerical damping coefficient with a value of 0.5 [26–28], $\boldsymbol{\beta}$ which will be dependent on the type of problems defined below:

$$\boldsymbol{\beta_e} = \frac{\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}}{\boldsymbol{\lambda} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}}, \tag{21}$$

where λ is a Lagrangian multiplier. Here, the value is found from a bi-sectioning algorithm. The sensitivity analysis for problems defined in Eq. 21 can be calculated as:

$$\frac{\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}}{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}} = \frac{1}{\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\sigma^{vm}}{\sigma_{yield}} - 1\right)}. \tag{22}$$

In this paper, the stress is calculated at the centroid of an element [28–30]. To relate the macro and micro stress levels, local stress interpolation scheme proposed by Duysinx and Sigmund [31] is used as:

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{\boldsymbol{D_e}(\boldsymbol{x})\overline{\boldsymbol{\varepsilon}}(\boldsymbol{x})}{\boldsymbol{x^q}}, \tag{23}$$

where $\sigma(\mathrm{x})$ is local stress at a material point, $\boldsymbol{D_e}(\boldsymbol{x})$ is macroscopic elastic tensor which can be related to the constitutive elasticity tensor $\boldsymbol{D}_0$ by a power-law approach as shown in Eq. 24, $\overline{\boldsymbol{\varepsilon}}(\boldsymbol{x})$ is the average strain of a material point which can be expressed in terms of strain displacement matrix $\boldsymbol{B_e}$ and elemental displacement vector $\boldsymbol{u_e}$. The exponent $\boldsymbol{q} > 1$ is a constant to preserve physical consistency in the material model.

$$\boldsymbol{D_e}(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{P}}\boldsymbol{D}_0, \tag{24}$$

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \boldsymbol{Bu_e}. \tag{25}$$

Substituting Eqs. 24 and 25 into Eq. 23, the stress at any material point with the given design domain can be expressed as:

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{P-q}}\boldsymbol{D}_0\boldsymbol{B_e u_e}. \tag{26}$$

From Eq. 26 the partial derivative of the constraint function in Eq. 22 can be expressed as:

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = (\boldsymbol{p} - \boldsymbol{q})\boldsymbol{x}^{\boldsymbol{P-q-1}}\boldsymbol{D}_0\boldsymbol{B_e u_e} + \boldsymbol{x}^{\boldsymbol{P-q}}\boldsymbol{D}_0\boldsymbol{B_e}\frac{\partial \boldsymbol{u_e}}{\partial \boldsymbol{x}}. \tag{27}$$

From equilibrium equation we have, $\boldsymbol{KU} = \boldsymbol{F}$, differentiating both sides of the equilibrium equation with respect to the design variable yields:

$$\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{x}}\boldsymbol{u} + \boldsymbol{k}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}} = \boldsymbol{0} \rightarrow \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}} = \frac{\boldsymbol{k}}{\boldsymbol{u}}\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{x}} \tag{28}$$

Substituting Eq. 20 for partial derivative of the displacement vector in Eq. 27 yields the sensitivity analysis for stress constraint defined in Eq. 22 as:

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = (\boldsymbol{p} - \boldsymbol{q})\boldsymbol{x}^{\boldsymbol{P-q-1}}\boldsymbol{D}_0\boldsymbol{B_e u_e} - \boldsymbol{x}^{\boldsymbol{P-q}}\boldsymbol{D}_0\boldsymbol{B_e}\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{x}}\frac{\boldsymbol{u}}{k}. \tag{29}$$

Therefore, the sensitivity analysis for stress-based topology optimization becomes:

$$\frac{\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}}{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}} = \frac{\boldsymbol{1}}{(\boldsymbol{p} - \boldsymbol{q})\boldsymbol{x}^{\boldsymbol{P-q-1}}\boldsymbol{D}_0\boldsymbol{B_e u_e} - \boldsymbol{x}^{\boldsymbol{P-q}}\boldsymbol{D}_0\boldsymbol{B_e}\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{x}}\frac{\boldsymbol{u}}{\boldsymbol{k}}} \tag{30}$$

Once the sensitivity analysis for both formulation was completed, a Matlab code is written for the stress based topology optimization and an existing Matlab code was used for compliance [28, 29]

Determination input parameters

Before using FA to solve stressed based topology optimization problems, input parameters must first be determined. These parameters include the number of fireflies, the maximum number of iterations, the randomness parameter $\boldsymbol{\alpha}$ and initial brightness value $\boldsymbol{\beta}$. Among these parameters, the effect of the number of fireflies and the number of iterations were studied, and the best combination of these parameters was selected for further implementation of the algorithm. The following sections will discuss the determination of these parameters (number of iteration and number of fireflies). The value of other parameters was directly adopted from other applications of FA [17, 19].

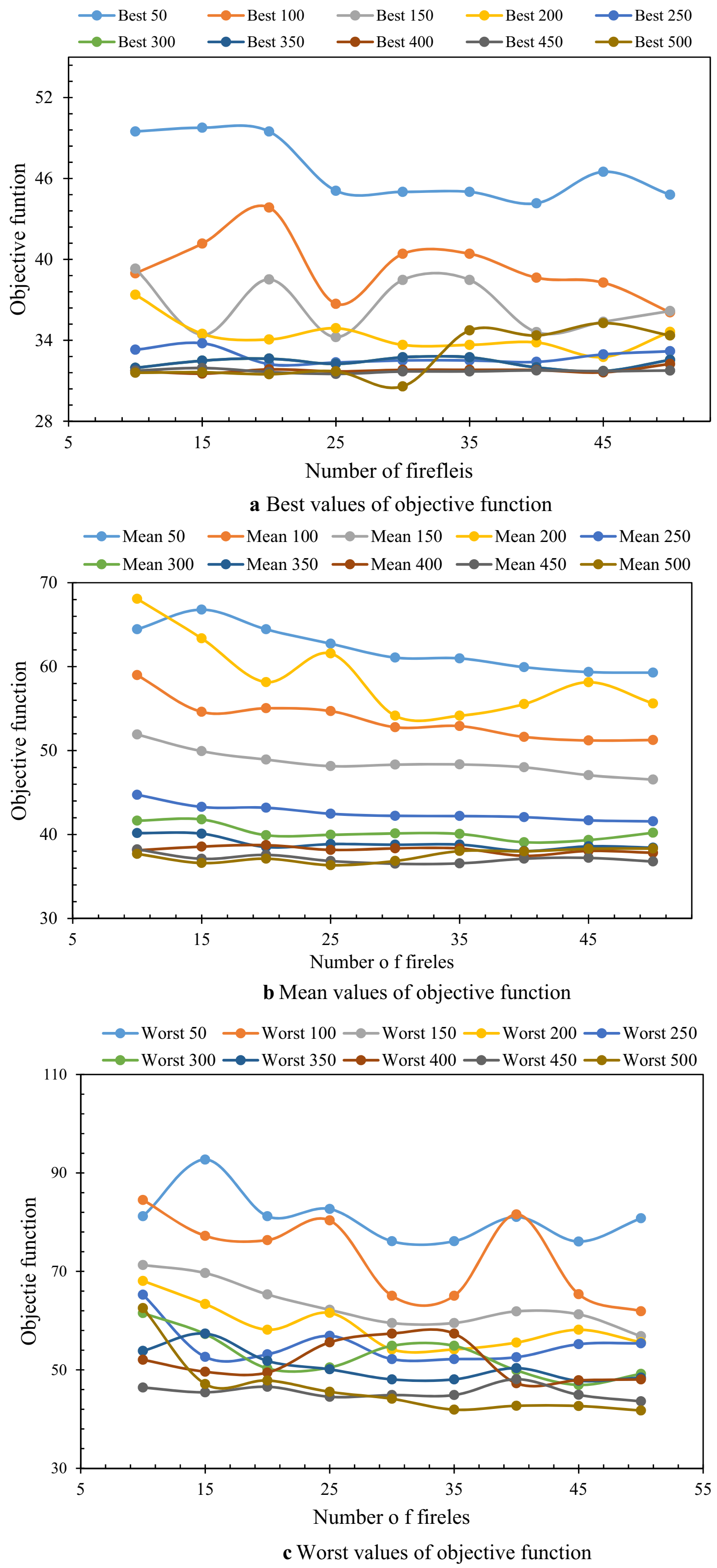
Number of fireflies

To assess the effect of a number of function evaluations on the efficiency of FA, a range between [50 and 500] function evaluations was considered. A design domain was formulated and solved using a FA for the range of function evaluations for 100 test runs. Then, the variation of best, meaning, and worst values of the objective function with function evaluation was studied through plotting these values with the number of fireflies, as shown in Fig. 6. From the figure, the objective function is minimum when the number of function evaluations is 200, and the number of fireflies is 30 as shown in Fig. 6a but the variation of the objective is not stable with the variation of the number of fireflies as shown in Fig. 6b, c. From the figure, it can be noticed that variation of the objective function is less sensitive for the number of fireflies greater than 35. Range of fireflies > 35 and function evaluations > 250 is the optimum range for best values of the objective function.

Number of iterations

An optimization problem based on Eq. 13 was formulated and solved using FA for the range of function evaluations for 100 test runs. Then, the variation of best, mean, and worst values of the objective function with the variation of the number of fireflies was studied through plotting these values of the objective function with the number of fireflies, as shown in Fig. 7. From the figure, it can be noticed that the variation of the objective function is less sensitive to the number of function evaluations when > 350. For function evaluations > 350, the range of the number of fireflies > 30 is the range for best values of the objective function.

Since the direct implementation of the FA yields a topology full of transition elements and highly affected by the checkerboard effect as shown in Fig. 8. A hybrid method was proposed to overcome this issue as described in Fig. 9. One of the challenges in the currently available methods is the dependency of initial material distributions [5]. From the initial topology generated from the firefly algorithm, we can

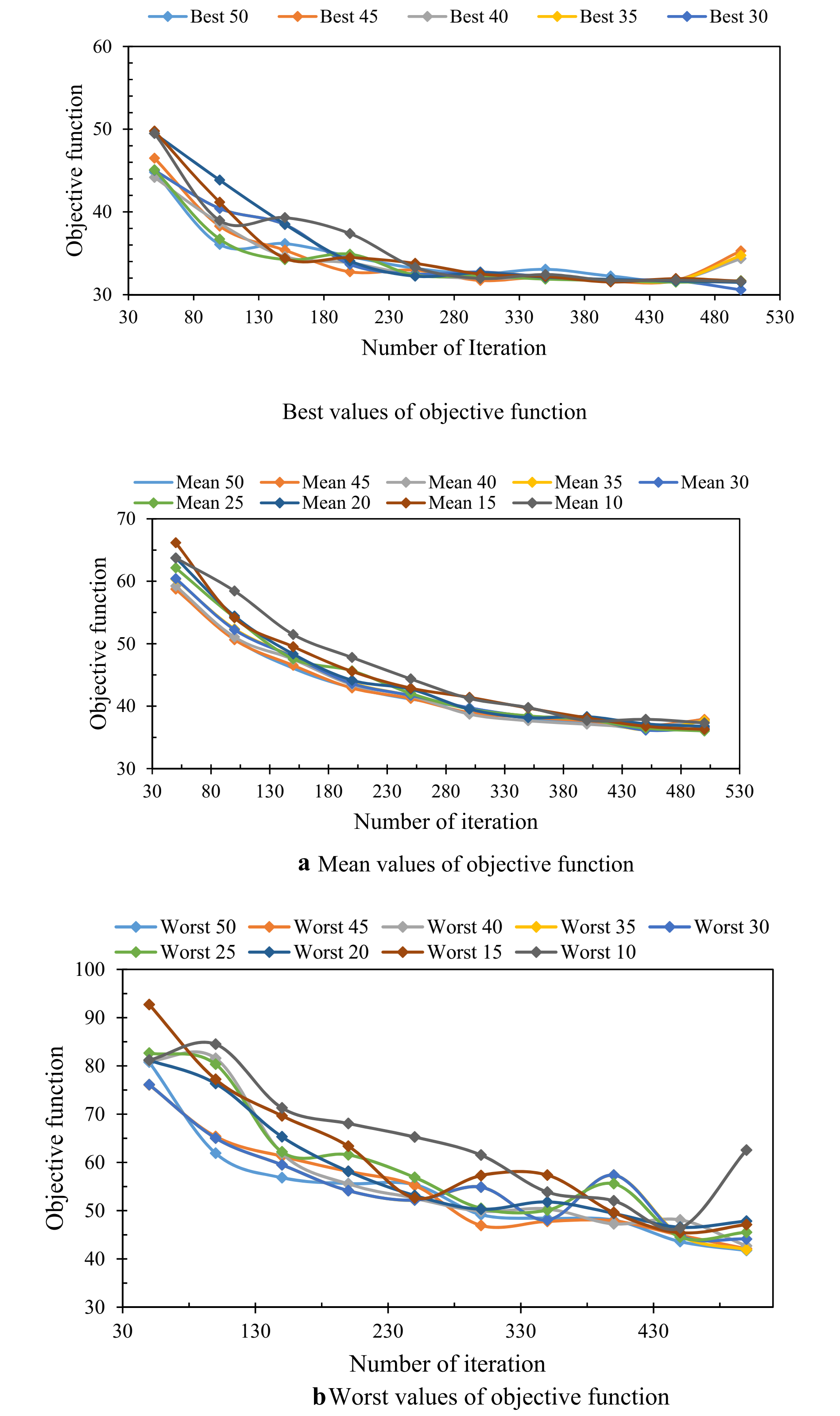


a Best values of objective function

b Mean values of objective function

c Worst values of objective function

Fig. 6 Effect of variation of function evaluation on objective function



Best values of objective function

a Mean values of objective function

b Worst values of objective function

Fig. 7 Effect of number of fireflies on objective function

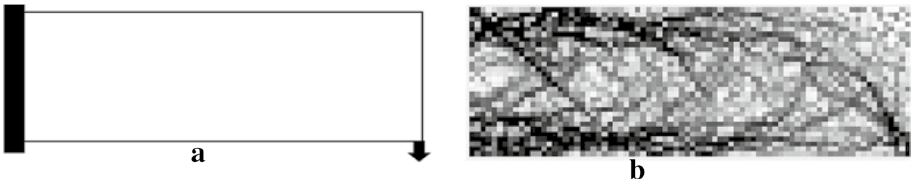


Fig. 8 **a** Design domain and **b** generated material distribution

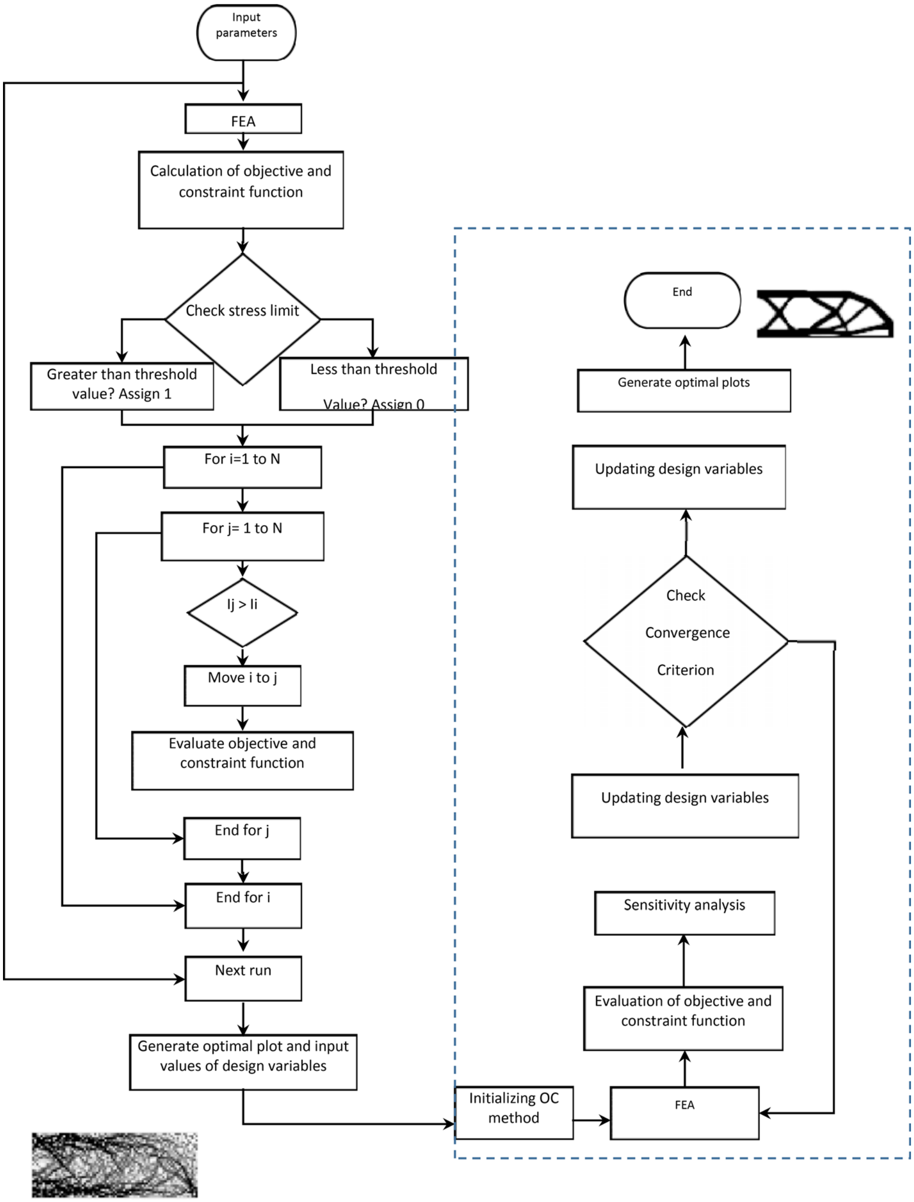

Input parameters
FEA
Calculation of objective and constraint function
Check stress limit
Greater than threshold value? Assign 1
Less than threshold Value? Assign 0
For i=1 to N
For j= 1 to N
Ij > Ii
Move i to j
Evaluate objective and constraint function
End for j
End for i
Next run
Generate optimal plot and input values of design variables
Initializing OC method
FEA
Evaluation of objective and constraint function
Sensitivity analysis
Updating design variables
Check Convergence Criterion
Updating design variables
Generate optimal plots
End

Fig. 9 A schematic diagram of structural topology optimization using discrete firefly

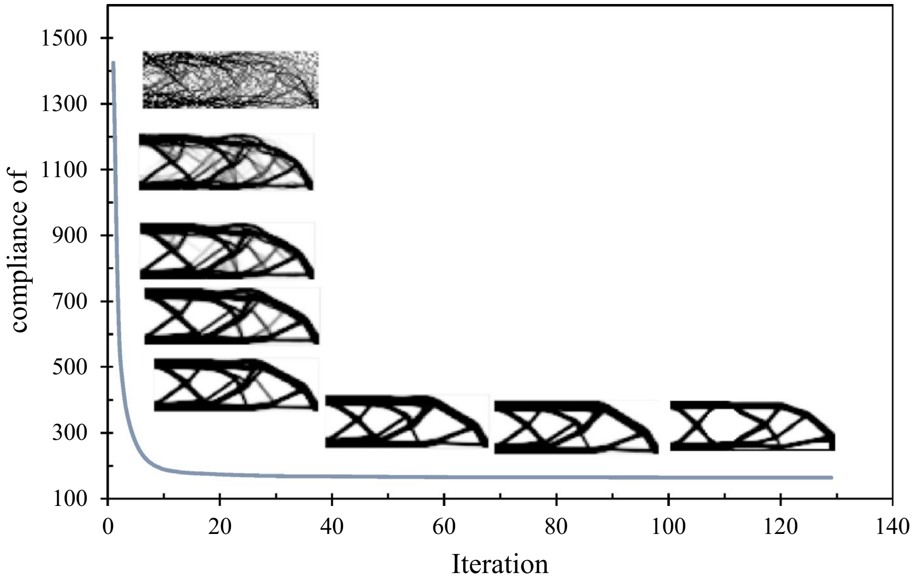


Fig. 10 Convergence history of hybrid method

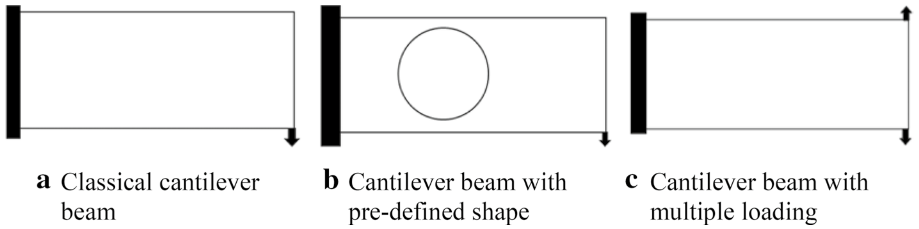

a Classical cantilever beam
b Cantilever beam with pre-defined shape
c Cantilever beam with multiple loading

Fig. 11 Boundary and loading condition

have the best values for the design variables which leads to the best values of the objective function. Even if it generates values of design variables, generated topologies are full of disconnected elements which are unwanted from an engineering perspective. To address this issue in direct implementation of FA and dependence of OC method on initial material, the outputs of the design variable values using FA as an input for the OC method. From the convergence history shown in Fig. 10, the proposed method can generate connected optimal plots.

Result and discussion

The proposed method was used to solve benchmark problems under different discretization size and the results are compared with solutions using an OC method.

Numerical results

Cantilever beam

The first case studies considered was a cantilever beam under loading and boundary conditions defined in Fig. 11.

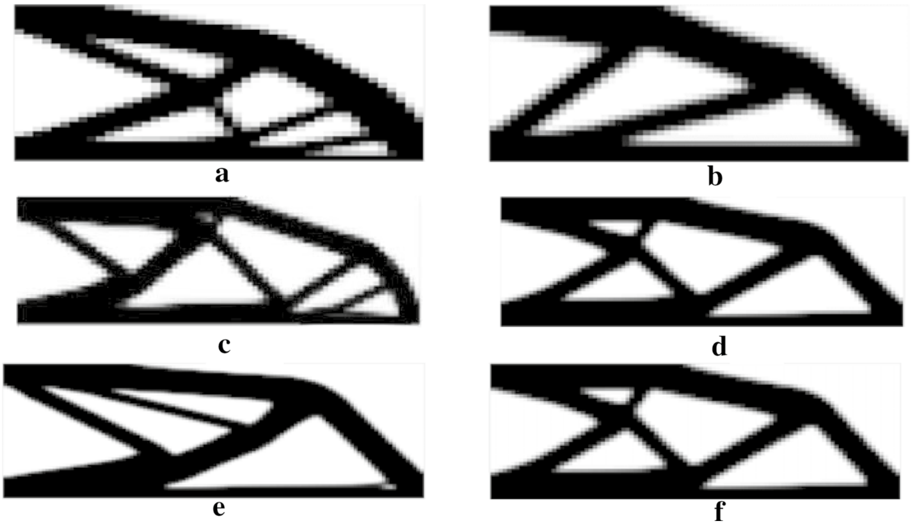


Fig. 12 Optimal material plots for numerous sizes of design domain **a** 45 × 32, **b** 60 × 80, **c** 80 × 30, **d** 100 × 40, **e** 120 × 60 and **f** 200 × 70

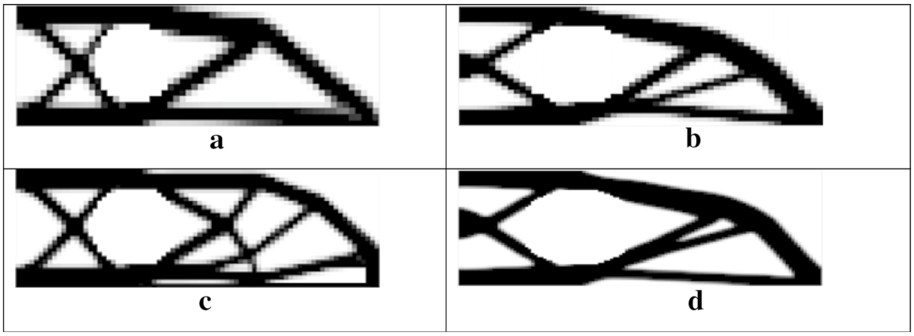


Fig. 13 Optimal material plots for numerous sizes of design domain **a** 40 × 40, **b** 60 × 30, **c** 80 × 30, and **d** 120 × 60

The first study was solved using the proposed method with a design domain and loading conditions defined in Fig. 11a and the generated topologies for different discretization size of the design domain are presented in Fig. 12.

The generated topologies for the Cantilever beam under loading and boundary conditions in Fig. 11b, c are presented in Figs. 13 and 14, respectively.

L-shape beam

The other benchmark problem considered to be solved using the proposed method is an L-shape beam under different boundary and loading conditions. Generated topologies for the design domains using the proposed method are shown in Tables 1 and 2, respectively.

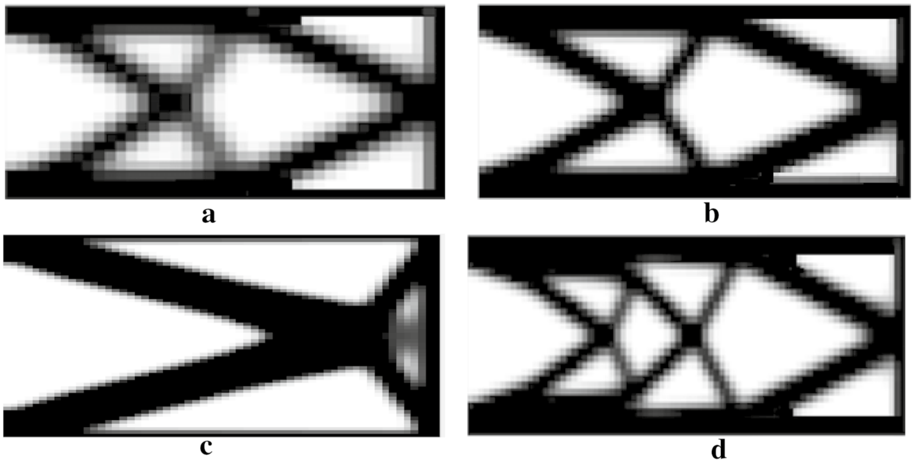
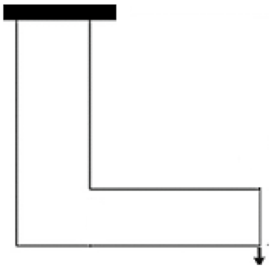
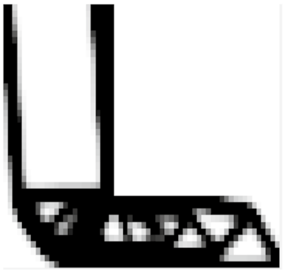
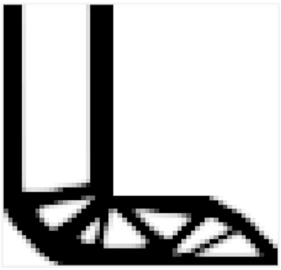
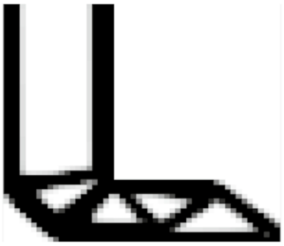
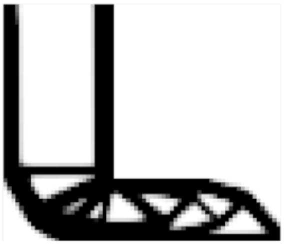


Fig. 14 Optimal material plots for numerous sizes of design domain **a** 40 × 20, **b** 60 × 30, **c** 60 × 60, and **d** 80 × 30

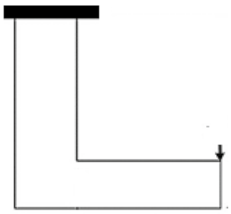
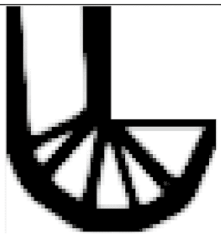
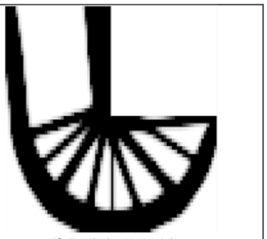
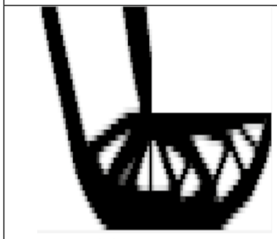
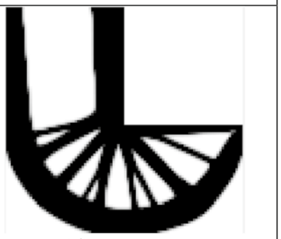
Table 1 Boundary and loading condition with respective topologies under various discretization size L-shape variant 1

 Design domain definition	 (a) 60 *X 40*	 (b) *60 X 60*
	 (c) *50X50*	 (d) *70X60*

Simply supported

The other benchmark problem used for validation of the proposed method was a simply supported beam under the loading and boundary condition defined in Fig. 15 and generated topologies are shown in Fig. 16 under different discretization sizes.

Table 2 Boundary and loading condition with respective topologies under various discretization size L-shape variant 2

 Design domain definition	 (a) 60 *X 60*	 (b) 80 *X 60*
	 (c) 10*0X40*	 (d) 10*0X60*

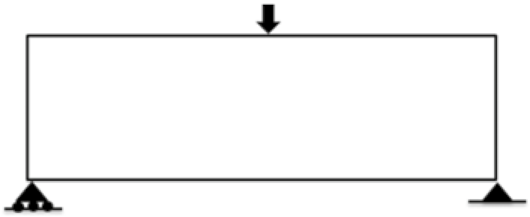


Fig. 15 Definition of boundary and loading conditions

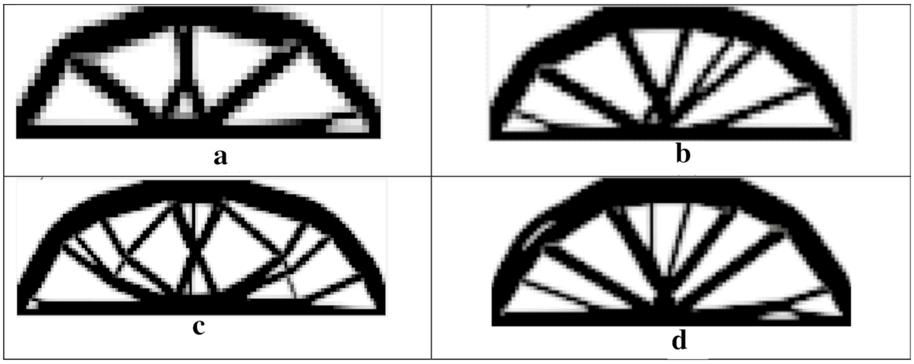


Fig. 16 Optimal material plots for different sizes of design domain **a** 60 × 20, **b** 80 × 30, **c** 120 × 60, and **d** 100 × 40

Comparison of results

The optimization results using the proposed method were compared with results from OC-method topologies. Table 3 shows the composition of generated optimal topologies using the proposed method and optimal topologies using OC based method for

Table 3 Generated optimal topologies using proposed method and optimality criteria (OC) method

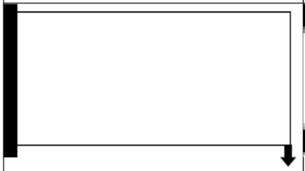
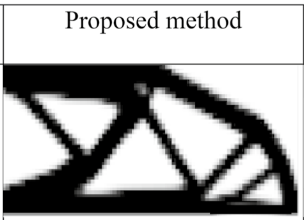
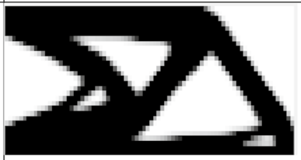
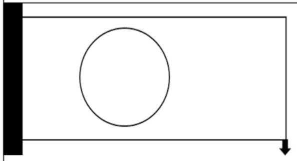
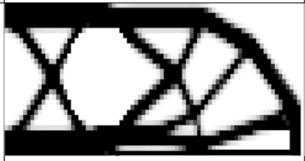
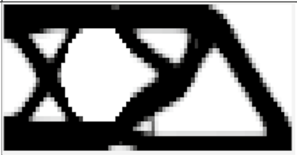
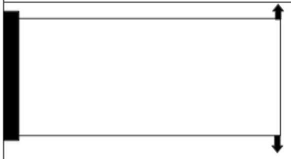
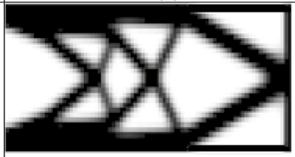
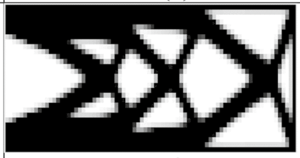
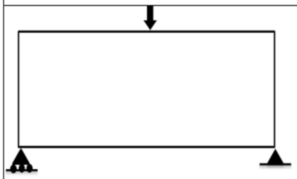
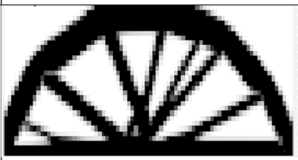
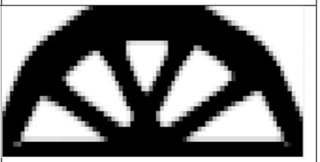
Design domain definitions and discretization size	Proposed method	OC based
	 (a)	 (b)
	 (c)	 (d)
	 (e)	 (f)
	 (g)	 (h)

Table 4 Performance evaluation of proposed method on different design domains

Design domain	Optimal plots					
	Void material (%) proposed	Void material (%) OC based	Solid material (%) proposed	Solid material (%) OC based	Transition material (%) proposed	Transition material (%) OC based
CCA	51.13	36.71	37.54	55.00	12.63	8.29
CCB	54.50	44.67	7.96	8.50	37.54	46.83
CCC	44.50	38.75	13.46	9.33	42.04	51.92
SS	44.67	37.13	14.38	8.96	40.96	53.92
LLL	60.52	54.6	5.04	4.8	34.44	40.6
LLU	47.96	44.05	6.96	5.6	45.07	50.35

Where CCA, CCB, CCC, SS, LLL and LLU indicate classical cantilever beam, Cantilever beam with pre-defined shape, cantilever beam under multiple loading, simply supported beam and loading and boundary conditions of L-shaped beam, respectively

different benchmark problems. Optimal topologies using the proposed method have less weight reduction percentage than the topologies generated using the OC method as shown in Table 4. Even if the weight reduction percentage is less for respective design domains, generated topologies using the proposed method are more complex than the OC method.

Conclusion

This paper presented a firefly algorithm (FA)-based hybrid method for stress-based topology optimization of 2D structures. The advantages of generating globally convergent solutions from FA, and the ability to generate connected topologies of the OC method, are crucial elements in the proposed method. In the proposed method, initial parameters for the SIMP method were determined using FA and also used as an input from the OC-based method, then optimal topologies were generated. The proposed method was validated using different benchmark problems to address global convergence and dependence of optimal material distribution on initial values of design variables of the OC method. Generated topologies and simulation results show the objective function, which is the weight of the design domain, can be further minimized in the range of 5–15%.

Author Contributions

HSG carried out the model development, writing the program code to solve developed model and simulation of results. DEW carried out checking the reporting of results and over all structure of the paper. FMH has done the reading the paper and worked on the language and asscoiated things within the paper. All authors read and approved the final manuscript.

Author details
[1] Bahir Dar Institute of Technology, Bahir Dar, Ethiopia. [2] Addis Ababa Science and Technology University, Addis Ababa, Ethiopia. [3] Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Perak, Malaysia.

Received: 6 June 2020 Accepted: 26 October 2020
Published online: 16 November 2020

References

1. Bendsoe MP. Optimal shape design as a material distribution problem. Struct Optim. 1989;1:193–202.
2. Xie YM, Steven GP. Evolutionary structural optimization. London: Springer; 1997.
3. Querin OM, Steven GP, Xie YM. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. Eng Comput. 1998;15(8):1031–48.
4. Huang X, Xie YM. Evolutionary topology optimization of continuum structures. New Delhi: Wiley; 2010.
5. Wang S, Tai K, Wang MY. An enhanced genetic algorithm for structural topology optimization. Int J Numer Methods Eng. 2006;65(1):18–44.
6. Hadidi A, Azad SK, Azad SK. Structural optimization using artificial bee colony algorithm. In: 2nd international conference on engineering optimization. 2010. p. 6–9.
7. Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. Adv Eng Softw. 2013;59:19–28.
8. Wang SY, Tai K. Structural topology design optimization using genetic algorithms with a bit-array representation. Comput Methods Appl Mech Eng. 2005;194(36):3749–70.
9. Sandgren E, Jensen E. Topological design of structural components using genetic optimization method. 1990.
10. Tejani GG, Kumar S, Gandomi AH. Multi-objective heat transfer search algorithm for truss optimization. Eng Comput. 2019, pp. 1–22.
11. Tejani GG, et al. Multiobjective adaptive symbiotic organisms search for truss optimization problems. Knowl Based Syst. 2018a;161:398–414.
12. Tejani GG, et al. Structural optimization using multi-objective modified adaptive symbiotic organisms search. Expert Syst Appl. 2019;125:425–41.
13. Tejani GG, et al. Topology and size optimization of trusses with static and dynamic bounds by modified symbiotic organisms search. J Comput Civil Eng. 2018b;32(2):04017085.
14. Tejani GG, Savsani VJ, Patel VK. Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. J Comput Des Eng. 2016;3(3):226–49.
15. Yang XS. Nature-inspired metaheuristic algorithms. Frome: Luniver press; 2010a.
16. Yang X-S. Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Berlin: Springer; 2009. p. 169–78.
17. Kumbharana SN, Pandey GM. Solving travelling salesman problem using firefly algorithm. Int J Res Sci Adv Technol. 2013;2(2):53–7.
18. Jati GK. Evolutionary discrete firefly algorithm for travelling salesman problem. In: International conference on adaptive and intelligent systems. Berlin: Springer; 2011. p. 393–403.
19. Lukasik S, Zak S. Firefly algorithm for continuous constrained optimization tasks. In: International conference on computational collective intelligence. Berlin: Springer; 2009. p. 97–106.

20. Apostolopoulos T, Vlachos A. Application of the firefly algorithm for solving the economic emissions load dispatch problem. Int J Comb. 2010, p. 2011.
21. Yang XS. Firefly algorithm, Levy flights and global optimization. In: Research and development in intelligent systems XXVI. London: Springer; 2010. p. 209–18.
22. Fister I, Yang X-S, Brest J. A comprehensive review of firefly algorithms. Swarm Evolut Comput. 2013;13:34–46.
23. Hönig U. A firefly algorithm-based approach for scheduling task graphs in homogeneous systems. Proc Inform. 2010, p. 724.
24. Pal SK, Rai C, Singh AP. Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems. Int J Intell Syst Appl. 2012;4(10):50.
25. Chandrupatla TR, et al. Introduction to finite elements in engineering. Upper Saddle River: Prentice Hall; 2002.
26. Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. Struct Multidiscip Optim. 2014;50(6):1175–96.
27. Andreassen E, et al. Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidiscip Optim. 2011;43:1–16.
28. Sigmund O. A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim. 2001;21:120–7.
29. Erik A, et al. Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidiscip Optim. 2011;43:1–16.
30. Biyikli E, To AC. Proportional topology optimization: a new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in MATLAB. PLoS ONE. 2015;10(12):e0145041.
31. Duysinx P, Sigmund O. New developments in handling stress constraints in optimal material distribution. In: 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. St. Louis, Missouri. 1998; p. 4906.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.