

# Enhanced Firefly Algorithm with Implicit Movement

Ronald Bartz<sup>1</sup>✉, Sierk Fiebig<sup>2</sup>, Thilo Franke<sup>2</sup>, Paul Falkenberg<sup>1</sup>,  
and Joachim Axmann<sup>1</sup>

<sup>1</sup> Technische Universität Braunschweig, Braunschweig, Germany  
{r.bartz,p.falkenberg,j.axmann}@tu-braunschweig.de

<sup>2</sup> Volkswagen AG, Braunschweig, Germany  
{sierk.fiebig,thilo.franke}@volkswagen.de

**Abstract.** The question for an optimal solution to a certain real-world problem often turns into a complex optimization problem. The sizing of the cross sections for bars of a truss structure is generally hampered by interdependencies. This prevents local search methods from finding a sufficient optimum. For those issues, there is a demand for fast and reliable global optimization algorithms. The Firefly Algorithm is a swarm-intelligence-based method frequently used for solving multi-modal optimization problems. The algorithm maintains a set of individuals, each corresponding to a point within the solution space. During the optimization process, the individuals move within the solution space under certain rules in order to find the global optimum. This paper presents an enhancement of the Firefly Algorithm by an implicit backward Euler movement. Therefore, in each iteration a linear system of equations must be solved to determine the new positions of the individuals. To evaluate the performance of the implicit movement, it is applied to continuous benchmark optimization functions. The optimization process is compared to the basic Firefly Algorithm to specify the effect of implicit movement. Furthermore, a discrete parameter optimization of a ten-bar truss in the sense of a weight reduction is carried out. The optimization results are compared to the basic Firefly Algorithm as well as to the results of four state-of-the-art algorithms. The implicit movement provides an intuitive and an easy to implement modification of the Firefly Algorithm. Simulation results show that the implicit movement causes a significant improvement in the convergence behavior compared to the basic Firefly Algorithm and outperforms state-of-the-art algorithms in terms of the solution quality and convergence behavior. Due to its generality, the proposed implicit movement can be implemented to several swarm-intelligence-based algorithms and offers a promising universal approach for enhancement.

**Keywords:** Stochastic optimization · Swarm intelligence · Firefly algorithm · Sizing

## 1 Introduction

Optimization is a subset of mathematics that deals with determining system parameters that minimizes or maximizes one or more functions under certain conditions. The optimization problem is often bounded by constraints. The constraints can be explicit,

such as by a restriction of the parameter range, or implicit by functions, which may not leave a certain range of values depending on the parameters. In engineering or economics, objectives and constraints are often determined by non-linear functions. These non-linearities yield a parameter response with multi-modal landscape. For such an optimization problem, local search algorithms, like hill climbing or Nelder-Mead downhill simplex methods are usually not able to determine a global optimum. Therefore, global optimization algorithms like stochastic methods are appropriate. [1, 2]

Stochastic methods are randomized search methods, which are usually based on no mathematical foundation. They represent iterative methods whose course is determined by certain stochastics in addition to logical conditions. Because these iterative optimization algorithms are independent of the system to be optimized, they are eligible for optimization problems in which the objective function or constraints are mathematically decoupled from the decision variables. Most stochastic procedures are inspired and abstracted by nature, like the Genetic Algorithm (GA) [3], the Ant Colony Optimization (ACO) [4] or the Particle Swarm Optimization (PSO) [5].

Another frequently used stochastic algorithm is the swarm-intelligence-based Firefly Algorithm (FA), developed in 2008 by Xin-She Yang [6]. The FA is used successfully in a wide range of fields like economic dispatch problems [7, 8], scheduling problems [9] and structural optimization problems [10, 11]. Preliminary studies show that the FA outperforms existing algorithms in many test cases with multi-modal landscape [12, 13].

In this paper, we first give a brief outline of the basic Firefly Algorithm. Then, we introduce the implicit movement within the Firefly Algorithm to enhance its convergence behavior and solution quality. Afterwards, the modified algorithm is applied to continues benchmark optimization functions as well as to a discrete parameter optimization of a ten-bar truss in the sense of a weight reduction. Finally, we will validate the performance of the Firefly Algorithm with implicit movement and propose topics for further research.

## 2 Basic Firefly Algorithm

The Firefly Algorithm (FA) by Yang [6] is inspired by the characteristic behavior of fireflies. The specific feature of these insects is the ability to glow, with which they attract their conspecific. The FA transfers the demand of fireflies to move toward bright glowing mates in a swarm-intelligence-based optimization algorithm. The  $d$  to be optimized parameters form a hyperspace  $\mathbb{R}^d$  in which the  $n$  individuals are located. According to the position vector  $\mathbf{x}_i$ , an individual  $i$  has an objective function value  $f$  whose quality determines the brightness  $I_i$  of the individual. In terms of a maximization problem, the brightness  $I_i$  can simply be chosen as

$$I_i(\mathbf{x}_i) \propto f(\mathbf{x}_i). \quad (1)$$

During the optimization process, each individual  $i$  moves within the solution space toward every brighter (by means of  $I_j > I_i$ ) individual  $j$ , with the intention to improve their own objective function value. The step size toward another individual is up to the attractiveness of this individual. The attractiveness  $\beta_{ij}$  of the individual  $j$  for the individual  $i$  depends on the distance  $r_{ij}$  of the two individuals and is determined by

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}. \tag{2}$$

The parameter  $\beta_0$  corresponds to the attractiveness at a distance of  $r_{ij} = 0$  and  $\gamma$  is the light absorption coefficient. The distance is commonly the Cartesian distance

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|. \tag{3}$$

For an individual  $i$ , its movement toward a brighter individual is determined by

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j^t - \mathbf{x}_i^t) + \alpha_t \epsilon_i^t \tag{4}$$

The third term corresponds to a random movement in space with the step size  $\alpha_t$  and a vector  $\epsilon_i^t$  with random numbers drawn from a Gaussian or uniform distribution. The pseudo-code of the basic Firefly Algorithm is given in Table 1.

**Table 1.** Pseudo code: Firefly Algorithm [6]

---

Firefly Algorithm

---

*Define Objective Function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$*   
*Generate initial population of fireflies  $\mathbf{x}_i (i = 1, 2, \dots, n)$*   
*Light intensity  $I_i$  at  $\mathbf{x}_i$  is determined by  $f(\mathbf{x}_i)$*   
*Define light absorption coefficient  $\gamma$*   
**while** ( $t < \text{Max Generation}$ )  
    **for** ( $i = 1:n$  all  $n$  fireflies)  
        **for** ( $j = 1:n$  all  $n$  fireflies)  
            **if** ( $I_j > I_i$ )  
                 $\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}$   
                 $\mathbf{x}_i = \mathbf{x}_i + \beta_{ij}(\mathbf{x}_j - \mathbf{x}_i) + \alpha \epsilon$   
            **end if**  
        **end for**  $j$   
        *Calculate new Lightintensity  $I_i$  at  $\mathbf{x}_i$*   
    **end for**  $i$   
    *Rank the fireflies and find the current best*  
**end while**  
*Postprocess*

---

### 3 Firefly Algorithm with Implicit Movement

The Firefly Algorithm without random movement correlates to a diffusion process of the form

$$\frac{\partial X}{\partial t} = \beta L(X) \quad (5)$$

with  $L(x_i) = (x_j^t - x_i^t)$ . The diffusion Eq. (5) can be integrated with an explicit Euler scheme, yielding:

$$X^{t+1} = X^t + \beta dt L(X^t), \quad (6)$$

which corresponds to the movement Eq. (4) in the basic Firefly Algorithm without random movement:

$$x_i^{t+1} = x_i^t + \beta (x_j^t - x_i^t). \quad (7)$$

The position of an individual at iteration  $t+1$  is therefore dependent on the positions of the other individuals at iteration  $t$ . To improve the convergence behavior of the FA, Eq. (5) is now solved by an implicit integration. The position of an individual at iteration  $t+1$  thus depends on the positions of the other individuals at iteration  $t+1$ . The integration of Eq. (5) by the implicit backward Euler method is  $X^{t+1} = X^t + \beta dt L(X^{t+1})$ , which leads to the linear system of equation (LSE):

$$\begin{aligned} (E - \beta dt L) X^{t+1} &= X^t \\ K X^{t+1} &= X^t \end{aligned} \quad (8)$$

with the identity matrix  $E$ . The size of the matrix  $K = E - \beta dt L$  equals  $\mathbb{R}^{(n-d) \times (n-d)}$ , with the number of individuals  $n$  and the number of the to be optimized parameters  $d$ . After solving the LSE, a random movement of the individuals takes place:

$$x_i = x_i + \alpha \varepsilon \quad (9)$$

The pseudocode of the Firefly Algorithm with implicit movement is given in Table 2.

**Table 2.** Pseudo code: implicit Firefly Algorithm

Implicit Firefly Algorithm

---

```

Define Objective Function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ 
Generate initial population of fireflies  $\mathbf{x}_i (i = 1, 2, \dots, n)$ 
Light intensity  $I_i$  at  $\mathbf{x}_i$  is determined by  $f(\mathbf{x}_i)$ 
Define light absorption coefficient  $\gamma$ 
while ( $t < \text{Max Generation}$ )
    Initialize Matrix  $K_{n,d \times n,d} = E$ 
    for ( $i = 1:n$  all  $n$  fireflies)
        for ( $j = 1:n$  all  $n$  fireflies)
            if ( $I_j > I_i$ )
                 $\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}$ 
                add entities to  $K$ 
            end if
        end for  $j$ 
    end for  $i$ 
     $X^{t+1} = K^{-1} X^t$ 
     $X^{t+1} = X^{t+1} + \alpha \epsilon$ 
    Calculate new Lightintensity  $I_i$  at  $\mathbf{x}_i$ 
    Rank the fireflies and find the current best
end while
Postprocess

```

---

## 4 Experiments and Results

In the following, the Firefly Algorithm with an implicit movement is evaluated using benchmark optimization problems. Each optimization is carried out fifty times and the average (AVG) value are formed from these to take account of fluctuations in the course. On the one hand, the Ackley’s function, the Sphere function, the Rosenbrock function and the Lévi#13 function given in Table 3 are optimized. The results are compared with the classical FA in order to determine the impact of the implicit position determination. Furthermore, a discrete parameter optimization of a ten-bar truss shown in Fig. 5 regarding to a weight reduction is carried out. The optimization results are compared to the basic FA and furthermore to the optimization results of the Particle Swarm Optimization (PSO) algorithm, the Ant Colony Optimization (ACO) algorithm, the Artificial Bee Colony (ABC) algorithm and a Genetic Algorithm (GA) adapted from [14–18].

The basic FA as well as the implicit FA are implemented with a population size of  $n = 25$ . The parameters are set to  $\beta_0 = 1.0$ ,  $\gamma = 0.0001$  and  $\alpha = 0.1$ . Thus, the basic

**Table 3.** The function test bed

Name	Formula	n	Search domain	Global min.
Ackley’s	$f(\mathbf{x}) = 20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right) + \exp(1) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$	10	$-5 \leq x_i \leq 5$	$f(\mathbf{x}) = 0$
Sphere	$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$	10	$-5 \leq x_i \leq 5$	$f(\mathbf{x}) = 0$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2) + (x_i - 1)^2\right]$	10	$-5 \leq x_i \leq 5$	$f(\mathbf{x}) = 0$
Lévi#13	$f(\mathbf{x}) = \sin^2(3\pi x_1) + (x_1 - 1)^2 (1 + \sin^2(3\pi x_2)) + (x_2 - 1) (1 + \sin^2(2\pi x_2))$	2	$-10 \leq x_i \leq 10$	$f(\mathbf{x}) = 0$

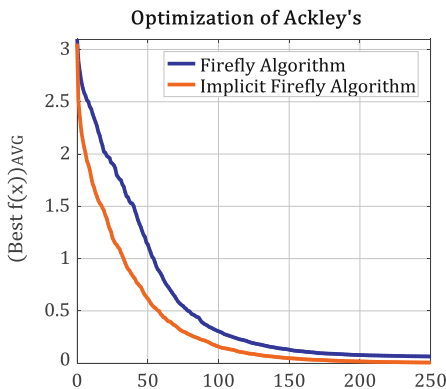
FA and the implicit FA only differ in the sense of explicit and implicit movement. The algorithms are implemented in MATLAB R2016b. The calculations are done on an intel® core™ i5-6200U CPU with 2.40 GHz. The evaluation of the LSE in each iteration of the implicit Firefly Algorithm are done with the standard *mdivide*-operator in MATLAB. The calculation time of the LSE is between 4.50e-06 s and 6.36e-04 s in all optimizations carried out in this study.

### 4.1 Optimization Results of Benchmark Test Functions

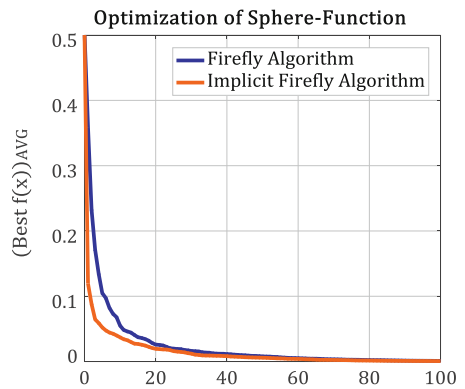
The optimization results of the test functions given in Table 3 are shown in Figs. 1, 2, 3 and 4. In order to obtain representative results, the best objective function values found within the respective iteration are averaged over the fifty optimization runs. The results show that the Firefly Algorithm with implicit movement has a superior convergence behavior. Especially in the first iterations, the objective function value is minimized significantly faster. Table 4 shows the averaged values (AVG) and the standard deviations (STD) of the optimizations in the last iteration. The Firefly Algorithm achieves better mean objective function values, whereby on the average a better optimization result is achieved. The lower standard deviation for all four test functions verifies the superior reliability of the FA with implicit movement. Note from Figs. 1, 2, 3 and 4 that in every iteration, the Firefly Algorithm with implicit movement provides a superior averaged best objective function value.

**Table 4.** Comparison of FA and implicit FA optimization results of the test functions

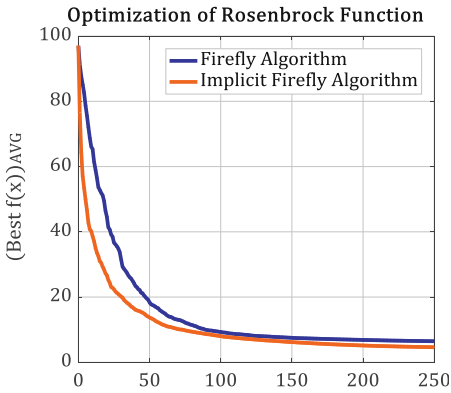
	Ackley's		Sphere		Rosenbrock		Lévi#13	
	AVG	STD	AVG	STD	AVG	STD	AVG	STD
FA	0.0615	0.1607	0.0043	0.0019	9.4771	12.012	1.4e-03	2.5e-03
Impl. FA	0.0149	0.0017	0.0021	0.0005	4.8449	1.0180	5.5e-04	1.7e-04



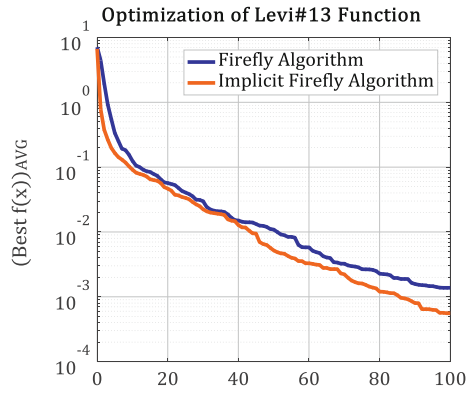
**Fig. 1.** Comparison of convergence rates for Ackley's Function of FA and implicit FA



**Fig. 2.** Comparison of convergence rates for Sphere Function of FA and implicit FA



**Fig. 3.** Comparison of convergence rates for Rosenbrock Function of FA and implicit FA



**Fig. 4.** Comparison of convergence rates for Lévi#13 Function of FA and implicit FA

### 4.2 Optimization Results of a Planar 10-Bar Truss

For the evaluation of the developed optimization algorithm, a discrete parameter optimization problem of a planar 10-bar truss is carried out. This test problem has already been used in different studies and, accordingly, there are comparative values for other optimization algorithms. The planar structure to be optimized is shown in Fig. 5. The parameters to be optimized are the ten bar cross sections with regard to a weight reduction while maintaining the restrictions defined by a maximum permissible stress of  $172.369\text{ MPa}$  and by a maximum node deflection in the vertical and horizontal direction of  $\pm 50.8\text{ mm}$ . The structure is subjected by the external force  $F = 444.82\text{ kN}$  at node 2 and 4. The bar cross-sectional areas are selected from a set of 42 discrete predefined values given in Table 5. The density of the material is  $2767.99\text{ kg/mm}^3$  and the modulus of elasticity is  $68947.6\text{ MPa}$ .

**Table 5.** Predefined set of possible cross-sectional areas for the 10-bar-truss [ $\text{mm}^2$ ]

1045.16	1161.29	1283.87	1374.19	1535.48	1690.32	1696.77	1858.06
1890.32	1993.54	2019.35	2180.64	2238.71	2290.32	2341.93	2477.41
2496.77	2503.22	2696.77	2722.58	2896.77	2961.28	3096.77	3206.45
3303.22	3703.22	4658.06	5141.93	7419.34	8709.66	8967.72	9161.27
9999.98	10322.56	10903.20	12129.01	12838.68	14193.52	14774.16	17096.74
19354.80	21612.86						

The averaged optimization processes out of 50 optimizations of the Firefly Algorithm and the implicit Firefly Algorithm are shown in Fig. 7. The corresponding courses of the standard deviations of the weight are shown in Fig. 8. To compare the optimization results with other algorithms, the optimization processes in Figs. 7 and 8 are indicated by the number of performed calculations. One calculation corresponds to

a stress and deformation calculation of the truss. It can be seen from Figs. 7 and 8 that the implicit movement provides a much better convergence behavior. The averaged best weights obtained after 2500 performed calculations are 2504 kg by the implicit Firefly Algorithm and 2556 kg by the basic Firefly Algorithm respectively. Including the standard deviation after 2500 performed calculations of 11.48 Kg and 65.71 kg as shown in Fig. 8, the Firefly Algorithm with implicit movement achieves a distinctly better reliability.

The best permissible parameter combination found within the 50 optimizations compared to the results of state-of-the art algorithms are given in Table 6. The number of calculations corresponds to the number of stress and deformation calculations necessary to reach the indicated weight. The calculation numbers are to be interpreted in such a way that no improvement to the listed weight was possible in subsequent

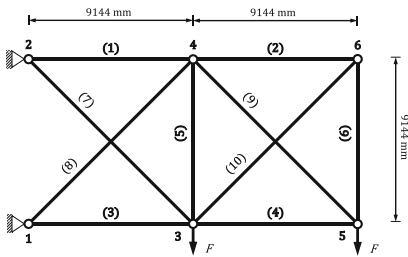


Fig. 5. Schematic of the planar 10-bar truss structure

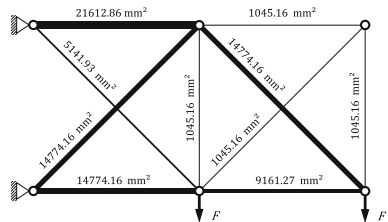


Fig. 6. Schematic of the optimized planar 10-bar truss structure

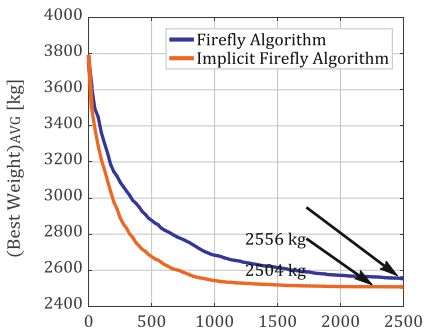


Fig. 7. Comparison of convergence rates for ten-bar truss of FA and implicit FA

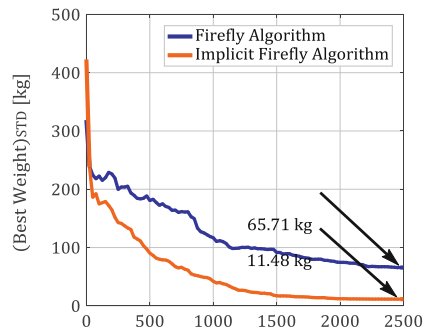


Fig. 8. Comparison of standard deviation for ten-bar truss of FA and implicit FA



iterations. It can be seen in Table 6 that both, the basic Firefly Algorithm and the implicit Firefly Algorithm, outperform the reviewed state-of-the-art algorithms. The implicit Firefly Algorithm achieves the best weight of 2490.55 kg within 2125 calculations and is therefore superior to the basic Firefly Algorithm where 3750 calculations are necessary. The truss structure corresponding to the weight of 2490.55 kg is shown in Fig. 6.

**Table 6.** Comparison of FA and implicit FA optimization results with literature for the ten-bar-truss problem

Algorithm	PSO [14]	ACO [15]	ABC [16]	GA [17]	FA	Impl. FA
Weight [kg]	2507.00	2490.55	2490.55	2520.56	2490.55	2490.55
#calculations performed	50000	10000	25800	2000	3750	2125
A1 [mm <sup>2</sup> ]	19354.80	21612.86	21612.86	19354.80	21612.86	21612.86
A2 [mm <sup>2</sup> ]	1045.16	1045.16	1045.16	1045.16	1045.16	1045.16
A3 [mm <sup>2</sup> ]	14774.16	14774.16	14774.16	17096.74	14774.16	14774.16
A4 [mm <sup>2</sup> ]	8709.66	9161.27	9161.27	8709.66	9161.27	9161.27
A5 [mm <sup>2</sup> ]	1045.16	1045.16	1045.16	1045.16	1045.16	1045.16
A6 [mm <sup>2</sup> ]	1045.16	1045.16	1045.16	1045.16	1045.16	1045.16
A7 [mm <sup>2</sup> ]	5141.93	5141.93	5141.93	4658.06	5141.93	5141.93
A8 [mm <sup>2</sup> ]	17096.74	14774.16	14774.16	14774.16	14774.16	14774.16
A9 [mm <sup>2</sup> ]	14193.52	14193.52	14193.52	14193.52	14193.52	14193.52
A10 [mm <sup>2</sup> ]	1161.29	1045.16	1045.16	1045.16	1045.16	1045.16

## 5 Conclusion

The Firefly Algorithm is an optimization algorithm, which is typically applied to global optimization problems. We successfully improved the Firefly Algorithm by using an implicit determination of the changes in individual positions. The system of equations that must be solved for the implicit position determination is efficiently solvable due to its sparseness. The proposed modification therefore has no remarkable effect on the runtime.

The implicit Firefly Algorithm has been tested by means of test functions and has also been validated by a discrete optimization of a ten-bar truss. Compared to the basic Firefly Algorithm, the implicit Firefly Algorithm showed a much better convergence behavior and better reliability. In addition, the optimization of the ten-bar-truss showed that the implicit Firefly Algorithm outperforms state-of-the-art methods reviewed in this study both in calculation time and quality.

The proposed algorithm presents an easy to implement and mathematically intuitive modification of the basic Firefly Algorithm. Due to its generality, the proposed implicit movement can be implemented to several swarm-intelligence-based algorithms and offers a promising universal approach for enhancement. Within the scope of this study

no disadvantages of the implicit movement could be identified. In addition to the implementation in further swarm-intelligence-based algorithms, supplementary optimization problems should be investigated in order to ensure the validity of the statements made.

## References

1. Yang, X.-S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, Cambridge (2010)
2. Yang, X.-S., Deb, S.: Engineering optimisation by cuckoo search. *Int. J. Math. Model. Numer. Optimisation* **1**(4), 330–343 (2010)
3. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston (1989)
4. Dorigo, M.: *Optimization learning and natural algorithms*, Mailand: Ph.D Thesis Dip. Elettronico, Politecnico di Milano (1992)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
6. Yang, X.-S.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press (2008)
7. Apostolopoulos, T., Vlachos, A.: Application of the firefly algorithm for solving the economic emissions load dispatch problem. *Int. J. Comb.* **23** (2011)
8. Yang, X.-S., Hosseini, S.S.S., Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl. Soft Comput.* **12**(3), 1180–1186 (2012)
9. Sayadi, M.K., Ramezani, R., Ghaffari-Nasab, N.: A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *Int. J. Ind. Eng. Comput.* **1**(1), 1–10 (2010)
10. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Mixed variable structural optimization using firefly algorithm. *Comput. Struct.* **89**(23–24), 2325–2336 (2011)
11. Talatahari, S., Gandomi, A.H., Yun, G.J.: Optimum design of tower structures using Firefly Algorithm. *Struct. Des. Tall Special Buildings* **23**(5), 350–361 (2014)
12. Gomes, H.M.: A firefly metaheuristic structural size and shape optimisation with natural frequency constraints. *Int. J. Metaheuristics* **2**(1), 38–55 (2012)
13. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **18**(1), 89–98 (2013)
14. Li, L.J., Huang, Z.B., Liu, F.: A heuristic particle swarm optimization method for truss structures with discrete variables. *Comput. Struct.* **87**(7–8), 435–443 (2009)
15. Camp, C.V., Bichon, B.J.: Design of space trusses using ant colony optimization. *J. Struct. Eng.* **130**(5), 741–751 (2004)
16. Sonmez, M.: Discrete optimum design of truss structures using artificial bee colony algorithm. *Struct. Multi. Optim.* **43**(1), 85–97 (2011)
17. Camp, C., Pezeshk, S., Cao, G.: Optimized design of two-dimensional structures using a genetic algorithm. *J. Struct. Eng.* **124**(5), 551–559 (1998)
18. Hasançebi, O., Azad, S.K.: Adaptive dimensional search: a new metaheuristic algorithm for discrete truss sizing optimization. *Comput. Struct.* **154**, 1–16 (2015)