Struct Multidisc Optim
DOI 10.1007/s00158-016-1624-x

RESEARCH PAPER

Modified firefly algorithm for multidimensional optimization in structural design problems

Jui-Sheng Chou[1] **· Ngoc-Tri Ngo**[1,2]

Received: 29 September 2015 / Revised: 16 October 2016 / Accepted: 13 November 2016

Abstract An enhanced nature-inspired metaheuristic optimization algorithm, called the modified firefly algorithm (MFA) is proposed for multidimensional structural design optimization. The MFA incorporates metaheuristic components, namely logistic and Gauss/mouse chaotic maps, adaptive inertia weight, and Lévy flight with a conventional firefly algorithm (FA) to improve its optimization capability. The proposed MFA has several advantages over its traditional FA counterpart. Logistic chaotic maps provide a diverse initial population. Gauss/mouse maps allow the tuning of the FA attractiveness parameter. The adaptive inertia weight controls the local exploitation and the global exploration of the search process. Lévy flight is used in the exploitation of the MFA. The proposed MFA was evaluated by comparing its performance in solving a series of benchmark functions with those of the FA and other well-known optimization algorithms. The efficacy of the MFA was then proven by its solutions to three multidimensional structural design optimization problems; MFA yielded the best solutions among the observed algorithms. Experimental results revealed that the proposed MFA is more efficient and effective than the compared algorithms. Therefore, the MFA serves as an alternative algorithm for solving multidimensional structural design optimization problems.

Keywords Nature-inspired optimization · Structural engineering design · Enhanced firefly algorithm · Chaotic maps · Adaptive inertia weight · Lévy flight

1 Introduction

Most engineering design optimization problems are highly nonlinear, and involve diverse design variables and complex design constraints such as displacements, geometrical configuration, stresses, and load carrying capabilities. Their solutions must satisfy the objectives and complex constraints (Adekanmbi and Green 2015). The structural design process typically involves generating an initial design on the basis of experience and optimizing it with algorithms to yield the desired performance target (Dugré et al. 2016). In numerous optimization applications, the search for optimality is challenging (Sergeyev and Kvasov 2015), and search efficiency is an essential measure of the effectiveness of an optimization algorithm. However, no single algorithm can solve all optimization problems.

Swarm intelligence (SI) and bio-inspired computation have attracted great interest and attention. SI is a collective behavior found in some decentralized and self-organized systems. SI systems consist typically of a population of simple agents interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. In the fields of optimization, computational intelligence, and computer science, bio-inspired algorithms, especially SI-based algorithms, have become very popular and effective (Yang 2014a, b; Roque and Martins 2015). Examples of SI-based algorithms include artificial bee colony

✉ Jui-Sheng Chou
jschou@mail.ntust.edu.tw

Ngoc-Tri Ngo
D10205804@mail.ntust.edu.tw; trinn@dut.udn.vn

1 Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei, Taiwan

2 Faculty of Project Management, The University of Danang - University of Science and Technology, 54, Nguyen Luong Bang, Danang, Vietnam

Published online: 02 December 2016

(ABC), particle swarm optimization (PSO), cuckoo search (CS), and firefly algorithm (FA). The FA is a particularly efficient SI-based algorithm because it has two major advantages over the others: automatic subdivision and the ability to handle multimodality (Yang 2014a, b).

Global optimization methods have been extensively developed over the last three decades; they include genetic algorithms (GA) (Goldberg 1989), PSO (Eberhart and Kennedy 1995), FA (Yang 2008), CS (Yang and Deb 2009), bat algorithms (BA) (Yang 2010a, b), differential evolution (DE) (Storn and Price 1997), DE based on covariance matrix learning and bimodal distribution parameter setting (CoBiDE) (Wang et al. 2014), composite DE (CoDE) (Wang et al. 2011), backtracking search optimization (BSA) (Civicioglu 2013), ABC (Karaboga and Basturk 2007), and multi-start gradient-base algorithms (MSGA) (Peri and Tinti 2012). Yang (2008) developed the FA (Yang 2008), a nature-inspired metaheuristic algorithm that has been successful in solving a wide range of problems in various domains (Chou et al. 2016).

The FA is a stochastic, nature-inspired metaheuristic algorithm that has recently produced excellent solutions to optimization problems (Fister et al. 2015). However, despite its simplicity and efficiency the FA is limited by its tendency to become trapped at local minima owing to premature convergence and a weak global search capability when performing complex multimodal tasks (Fister et al. 2013a, b). Additionally, the convergence rate declines rapidly as the evolution proceeds (Coelho and Mariani 2013). When it reaches a nearly optimal solution, the algorithm stops optimizing, so its accuracy is limited. These weaknesses limit the range of real-world optimization problems to which the FA can be successfully applied.

Therefore, improving convergence speed and avoiding local optima are two crucial tasks in the development of FA-based optimization algorithms. Chaos is an irregular nonlinear phenomenon in the natural world; it is defined as the highly unstable and unpredictable motion of deterministic systems in finite phase space (Hong et al. 2011). Chaotic maps, such as logistic and Gauss/mouse maps have been proven to be effective in tuning the parameters of SI algorithms (Hong et al. 2011; Gandomi et al. 2013a, b; Kazem et al. 2013; Fister et al. 2015). For instance, a logistic chaotic map has been integrated with a GA to select the parameters of a machine learning model (Hong et al. 2011).

Exploration is the process of visiting entirely new regions of a search space whereas exploitation is the process of visiting the regions of a search space that are within the neighborhood of previously visited points (Črepinšek et al. 2013). Balancing exploration with exploitation is the difficult part of the optimization. Adaptive inertia weight (AIW) (Shi and Eberhart 1998) can balance exploration and exploitation in an optimization algorithm. AIW has been successfully combined with the PSO (Shi and Eberhart 1998; Nickabadi et al. 2011). A large AIW facilitates exploration, whereas a small AIW facilitates exploitation. The search capability can be dynamically adjusted by changing the inertia weight.

Lévy flight is a random walk in which the step size is determined using the Lévy distribution. Numerous studies have established that the behavior of various flying animals and insects can be characterized as Lévy flights (Solomon et al. 1994; Weeks et al. 1995; Pavlyukevich 2007). Lévy flight has been used to improve the search capabilities of PSO (Haklı and Uğuz 2014) and of FA (Yang 2010a, b). To the best of the authors' knowledge, literature on the simultaneous integration of FA with the previously mentioned chaotic maps, AIW, and Lévy flight is scarce.

In this work, a standard FA is enhanced to improve its performance in solving certain engineering design optimization problems. The modified FA (MFA) is augmented with chaotic maps, AIW, and Lévy flight. A Gauss/mouse map and a logistic map are used to tune an attractiveness parameter and to provide a diverse initial population for the FA, respectively. The AIW helps to control the local exploitation and the global exploration of the FA. Finally, the Lévy flight facilitates the exploitation search by generating new optimal neighborhoods around the obtained best solution.

The performance of the proposed MFA was compared with those of the standard FA and other well-known optimization algorithms on widely used benchmark functions and on real design problems in structural engineering (namely, the welded beam design problem, the cantilever beam design problem, and the problem of minimization of vertical deflection of an I-beam). Nonparametric tests were then conducted to compare the algorithms fairly.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature. Section 3 presents the components and implementation of the proposed MFA. Section 4 compares the MFA with other well-known optimization algorithms using benchmark functions. Section 5 describes the performance of the MFA on real engineering problems. The final section draws conclusions.

2 Literature review

Optimization algorithms have recently been used in various domains, such as in solving constrained engineering problems (Jaberipour and Khorram 2010; Gandomi et al. 2011; Lamberti and Pappalettere 2011; Saka and Dogan 2012; Saka et al. 2013; Saka and Geem 2013; Alberdi et al. 2015), optimizing structures with unrestricted dynamic shakedown constraints (Benfratello et al. 2015), and optimizing structural topologies (Guirguis et al. 2015; Luo and Tong 2015; Rojas-Labanda and Stolpe 2015). Meng et al. (2015) used four engineering design problems (a speed-reducing design coil, a compression spring, a cantilever beam, and a welded beam) to evaluate the performance of a novel bat algorithm (Meng et al. 2015). Kripka et al. (2015) applied simulated annealing (SA) to optimize the costs of reinforced concrete structures (Kripka et al. 2015).

Alberdi et al. (2015) investigated the optimization of the connective topology of steel moment frames under gravity and wind loads; they used different metaheuristic methods to vary both connection topologies and member sections to minimize cost (Alberdi et al. 2015). They considered four metaheuristic algorithms (harmony search, ant colony optimization, GA and tabu search) that are commonly used for steel frame optimization to optimize connection topology. All algorithms consistently yielded low-cost, feasible designs, as evidenced by the low standard deviations after 100 optimizations were run.

Baykasoğlu and Akpinar (2015) presented an effective weighted superposition attraction (WSA) algorithm for solving constrained globally optimized problems in the design of products such as a tension/compression coil spring, a pressure vessel, a welded beam, and a speed reducer (Baykasoğlu and Akpinar 2015). Computational study indicated the robustness and the effectiveness of the WSA in terms of its results, its level of convergence, and t its capacity to escape premature convergence by escaping local optima and stagnation.

Gandomi et al. (2011) utilized a conventional FA to solve mixed structural optimization problems with both continuous and discrete characteristics (Gandomi et al. 2011). They tested their FA code in solving six structural optimization problems that had been considered in the literature; these were design problems for a welded beam, a pressure vessel, a helical compression spring, a reinforced concrete beam, a stepped cantilever beam, and a car side impact. Their comparisons revealed that the optimization efficiency level of their FA exceeded those of other metaheuristic algorithms, including PSO, GA, SA, and DE.

Zhou et al. (2015) used a multi-objective optimization model that was established using the Latin square design method and a response surface approach. They used nondominated sorting genetic algorithm-II (NSGA-II) to optimize the design of a runflat structure (Zhou et al. 2015). Gomes (2012) conducted a structural mass optimization in terms of shape and size under dynamic constraints (Gomes 2012). In this problem, mass reduction came into strong conflict with lower bounds on frequency constraints because vibration modes were easily changed by shape modifications; for these problems, FA served as an optimization engine that outperformed metaheuristic methods and conventional gradient-based methods.

To improve the performance of FA and to prevent stagnation, Fister et al. (2013a, b) utilized quaternions in their FA to represent individuals (Fister et al. 2013a, b). Tilahun and Ong (2012) modified the random movement of the brightest firefly by generating random directions in which it could move to determine the best direction in which the brightness of that firefly could increase (Tilahun and Ong 2012). Yang (2010a, b) developed a new metaheuristic algorithm that combined Lévy flights and an FA (Yang 2010a, b). Numerical studies have suggested that this Lévy-flight FA was superior to PSO and GA methods in terms of efficiency and success rate.

Baykasoglu (2012) proposed a great deluge algorithm with chaos for solving constrained nonlinear design optimization problems (Baykasoglu 2012), and he confirmed that chaotic maps hold staggering potential in diversifying search. Chaos refers to deterministic nonlinear dynamical behaviors that can be observed in numerous domains, such as communication, automation, and pattern recognition (Holden 1986). Researchers recently combined chaos with heuristic optimization algorithms, such as GA (Gharooni-fard et al. 2010) and chaos optimization algorithms (Yang et al. 2014) to generate hybrid algorithms. Furthermore, the nonrepetitive characteristics of chaotic systems can enable rapid searches (Dos Santos Coelho and Coelho 2009).

In stochastic optimization, algorithms that use chaotic variables rather than random variables are called chaotic optimization algorithms. The most common chaotic maps are the Chebyshev map, the circle map, the iterative map, the intermittency map, the Gauss/mouse map, the logistic map, the Liebovitch map, the sine map, the piecewise map, the Singer map, the sinusoidal map, and the tent map. Various chaotic maps have been developed as alternatives to pseudorandom sequences in optimization algorithms. Logistic maps appear in the nonlinear dynamics of biological populations; logistic maps exhibit chaotic behavior.

Despite the high efficiency of the FA approach, its solutions repeatedly change as it approaches an optimum (Pal et al. 2012; Chou and Ngo 2016). Additionally, designing tuning parameters that improve the convergence of an FA is difficult. A search of the literature shows that no work has fully integrated metaheuristic components, such as chaotic maps, AIW, and Lévy flight into an FA for solving structural design optimization problems. This work, therefore, incorporates these components — chaotic maps, AIW, and Lévy flight — into the conventional FA to improve its optimization.

3 Enhanced firefly algorithm

3.1 Conventional firefly algorithm

The conventional FA, which was developed by Yang (2008) (Yang 2008), is a stochastic, nature-inspired metaheuristic algorithm that is effective in solving optimization problems (Fister et al. 2015). The FA applies the following three idealized rules; (1) each firefly is attracted to other fireflies regardless of gender because all fireflies are unisex; (2) a firefly's attractiveness is proportional to its brightness and decreases as distance increases; additionally, a firefly moves randomly if no other firefly is brighter; and (3) the brightness of a firefly is affected or determined by the search space of the objective function. Figure 1 presents the pseudo-code of the conventional FA based on these rules.

For a maximization problem, the brightness value of a firefly is simply set to be proportional to the value of the objective function. As the attractiveness of a firefly is proportional to the intensity of light that is visible to adjacent fireflies, the attractiveness β of a firefly satisfies Equation (1).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{1}$$

where β is the attractiveness of the firefly; β_0 is the attractiveness of the firefly at $r=0$; r is the distance between the firefly of interest and any other; e is a constant coefficient, and γ is the absorption coefficient.

The distance between any two fireflies i and j at x_i and x_j, respectively, is the Cartesian distance, which is given by Equation (2).

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

where r_{ij} is the distance between any two fireflies i and j at x_i and x_j, respectively; $x_{i,k}$ is the k-th component of spatial coordinate x_i of the i-th firefly; $x_{j,k}$ is the k-th component of spatial coordinate x_j of the j-th firefly, and d is the number of dimensions of the search space.

Objective function $f(x)$, $x = (x_1, ..., x_d)^T$
Generate initial population of fireflies x_i ($i = 1, 2, ..., n$)
Light intensity I_i at x_i is determined by $f(x_i)$
Define light absorption coefficient
while *(t <MaxGeneration)*
for *i = 1 : n all n fireflies*
  for *j = 1 : i all n fireflies*
    if *($I_j > I_i$), Move firefly i towards j in d-dimension;*
end if
    *Attractiveness varies with distance r via exp[−γ*r]*
    Evaluate new solutions and update light intensity
  end for *j*
end for *i*
Rank the fireflies and find the current best
end while
Post-process results and visualization

Fig. 1 Pseudo-code of conventional FA

Equation (3) specifies the movement of the i-th firefly when attracted to the j-th firefly that is more attractive (brighter).

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha^t \operatorname{sign}[rand - 0.5] \tag{3}$$

where x_i^{t+1} is the coordinate of the i-th firefly in the $(t+1)$-th iteration; x_i^t is the coordinate of the i-th firefly in the t-th iteration; x_j^t is the coordinate of the j-th firefly in the t-th iteration; γ is the absorption coefficient, which typically varies from 0.1 to 10 in most applications; β_0 is the attractiveness at $r_{ij}=0$; α^t is a randomization parameter, and rand is a number drawn at random from a Gaussian distribution or from a uniform distribution at time t with a mean value of zero and a standard deviation of one.

Generally, the FA is controlled by three parameters, which are γ, β, and α. When $\beta_0=0$, the movement of the fireflies is a simple random walk. Notably, the randomization terms in the FA are easily extended to other distributions such as chaotic maps and Lévy flights. In the sensitivity analysis of γ, the optimal result was obtained when $\gamma=1$, and $\beta_0=\beta_{\min}=$ the attractiveness at $r_{ij}=0$.

Although the FA and its variants are very efficient in numerous situations, they frequently become trapped at local optima (Fister et al. 2013a, b). Additionally, to set tuning parameters that promote the convergence of the FA is challenging. The FA control parameters must be optimized to balance exploitation with exploration [18]. This work, therefore, incorporates metaheuristic components — chaotic maps, AIW, and Lévy flight — into the conventional FA to improve its search and optimization capability.

3.2 Metaheuristic components in the fine-tuning of the firefly algorithm

The conventional FA utilizes three metaheuristic components in performing a heuristic search that proceeds toward global optimality, namely the initial population, the attractiveness parameter, and the movement. These three terms are adjusted using supplementary components, which are chaotic maps, AIW, and Lévy flight, to improve the performance of the conventional FA. The following subsections elucidate the role of each term.

3.2.1 Generating diversity of initial population using logistic map

The FA uses a typical random method to generate an initial solution. The two major disadvantages of this method are its slow convergence and its tendency to become trapped in local optima owing to low population diversity. To improve the diversity of the initial solution, the logistic map is used to generate a highly diverse initial population in the initial stage of the MFA.

The logistic map that was developed by May (1976) is one of the simplest chaotic maps (May 1976). Logistic mapping provides more diverse initial populations than does random selection and provides a lower probability of premature convergence (Hong et al. 2011). Equation (4) is the formula of the logistic map.

$$X_{n+1} = \eta X_n (1 - X_n), \quad 0 \le X_0 \le 1 \tag{4}$$

where n denotes the number of the individual firefly; X_n is the logistic chaotic value for the n-th firefly, and X_0 donates the initial randomness of fireflies, which is used in the chaotic map for generating initial population of fireflies.

Because Equation (4) is clearly deterministic, it exhibits chaotic dynamics when $\eta = 4.0$ and $X_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. It exhibits strong dependence on initial conditions, and describes travel with ergodicity, irregularity, and pseudorandomness. A slight difference in the initial population causes a substantial difference in its long-term behavior (Liu et al. 2005). In this work, initial firefly positions are generated using the logistic map equation, and parameter η is set to 4.0 in all experiments.

3.2.2 *Tuning attractiveness parameter using gauss/mouse map*

Comparisons have revealed that the Gauss/mouse map (Gandomi et al. 2013b) is the optimal chaotic map for tuning the attractiveness parameter (β) in a conventional FA. Equation (5) describes the Gauss/mouse map that is used in this study instead of the random parameters that are used in a conventional FA.

Gauss/mouse map:

$$\beta_{chaos}^{t} = \begin{cases} 0 & \beta_{chaos}^{t-1} = 0 \\ 1/\beta_{chaos}^{t-1} \bmod(1) & otherwise \end{cases} \tag{5}$$

Equation (1) in a conventional FA is then updated to,

$$\beta = \left(\beta_{chaos}^{t} - \beta_0\right) e^{-\gamma r_{ij}^{2}} + \beta_0 \tag{6}$$

where β_{chaos}^{t} *is* the t-th chaotic number and t is the iteration number.

3.2.3 *Tuning randomization parameter using adaptive inertia weight*

A swarm-based algorithm can be improved by reducing the randomness of its parameters as the iteration proceeds. In the early stages of the search process, a high AIW can boost global exploration (the searching of a new area). However, in the late stages, the AIW, which is reduced in each stage, can improve the local exploitation of the optimal solution (fine tuning of the current search area). Inertia weight affects the convergence to the optimal solution and the execution time of the simulation. The AIW helps to balance the local exploitation and global exploration of the swarm algorithm.

The aforementioned issue is addressed by using a monotonically decreasing function of the inertia weight to change the randomization parameter α from that of a conventional FA. Because the AIW is utilized to tune the randomization parameter α, the distances between fireflies are reduced to maintain α in a reasonable range [Equation (7)].

$$\alpha^t = \alpha_0 \theta^t \tag{7}$$

where α_0 is the initial randomization parameter; α^t is the randomization parameter at the t-th iteration; θ is the randomness reduction constant ($0 < \theta < 1$); and t is the number of iterations. The selected value of θ is 0.9 in this implementation, based on a sensitivity analysis and the literature, and $t \in [0, t_{max}]$, where t_{max} is the maximum number of iterations.

3.2.4 *Controlling metaheuristic firefly movement using Lévy flight*

Random walk theory has a critical role in modern metaheuristic algorithms and stochastic optimization. Lévy flights are random walks in which the step length follows a Lévy distribution. The step lengths have no characteristic scale, meaning that the second moment or even the first moment may diverge and the distribution exhibits self-affine properties. The steps form a random walk process with a power-law step-length distribution with a heavy tail.

Lévy flights are defined such that each jump, regardless of size, takes one unit of time (Geisel et al. 1985). The generation of random numbers using Lévy flights comprises two steps, which are the random selection of a direction and the generation of steps that obey the selected Lévy distribution. In this work, a uniform distribution is utilized to generate a direction. The Mantegna algorithm, which is the most efficient, is used to generate steps from a symmetric Lévy stable distribution. The step length s in the Mantegna algorithm is calculated using Equation (8).

$$\text{Lévy} \sim s = \frac{u}{|v|^{1/\tau}} \tag{8}$$

where u and v are normally distributed, as follows.

$$u \sim N\left(0, \sigma_u^2\right), \quad v \sim N\left(0, \sigma_v^2\right) \tag{9}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\tau)\sin\left(\pi\tau/2\right)}{\Gamma\left[(1+\tau)/2\right]\tau 2^{(\tau-1)/2}} \right\}^{1/\tau}, \quad \sigma_v = 1 \tag{10}$$

where $\Gamma(t)$ is the Gamma function.

$$\Gamma(t) = \int_0^\infty z^{t-1} e^{-z} dz \tag{11}$$

A Lévy walk generates some new solutions around the best solution obtained so far, accelerating the local search. This behavior can be used in optimal search for optimization problems, and preliminary results confirm the potential effectiveness of so doing (Pavlyukevich 2007). Therefore, Equation (3) is revised to Equation (12).

$$x_i^{t+1} = x_i^t + \beta\left(x_j^t - x_i^t\right) + \alpha^t \mathrm{sign}[\mathrm{rand}-0.5] \otimes \mathrm{Lévy} \tag{12}$$

where the second term denotes the attraction; the third term provides the randomization that is associated with the Lévy flights; and α^t is a randomization parameter. The product operator ⊗ indicates entry-wise multiplication. The term sign [rand-0.5] with rand ∈ [0, 1] denotes a random sign or direction when the random step length follows a Lévy distribution.

3.3 Implementation of the modified firefly algorithm

The modifications suggested to conventional FA involve tuning its parameters. Three parameters used in the conventional FA (initial population, attractiveness coefficient, and movement) are adjusted by adding supplementary components — chaotic maps (logistic and Gauss/mouse maps), AIW, and Lévy flight.

Firstly, the logistic chaotic map is used to generate a highly diverse initial population. Next, the attractiveness parameter (β) is tuned using the Gauss/mouse chaotic map. Then, the AIW is used to tune the randomization parameter α, which controls local exploitation and global exploration. Finally, Lévy flight is used to generate new solutions around the optimal solution accelerating the exploitation. Table 1 presents the settings of the proposed MFA, which improve upon those of the conventional FA.

Figure 2 displays the pseudo code for the proposed MFA. The firefly population is initialized by applying a logistic chaotic map. The firefly search process consists of the following steps: (1) calculate the 'AlphaNew' function by multiplying the initial value of parameter α by 0.9^t (AIW); (2) use the chaotic Gauss/mouse map to compute the 'BetaNew' function; (3) execute the 'EvaluateMFA' algorithm to evaluate solution quality in terms of the objective function *f(x);* (4) use the 'SortMFA' function to select the fittest individual in the population; and finally, (5) use the 'MoveMFA' to move the fireflies by Lévy flight toward more attractive individuals in the search space. When the stopping criteria are satisfied, the optimal solution is represented by the firefly with the highest light intensity.

4 Numerical experiments

Ten well-known benchmark functions from the literature were utilized to evaluate the proposed MFA with various levels of complexity and multimodality. These benchmark functions were obtained from the literature (Jamil and Yang 2013) and include multimodal, unimodal, separable, and nonseparable functions. Table 2 formulates the functions, their characteristics (C), search ranges, dimension (D), and optimal values.

The difficulty of a problem generally increases with its dimensionality. Problems of high dimensionality may pose a considerable challenge for almost all optimization algorithms. In this work, problems with three levels of dimensions (i.e., 10, 30, and 50) are used to evaluate the proposed MFA. The MFA may yield fluctuating optima among various runs owing

Table 1 Comparison between MFA and FA optimization methods

Group	Parameter	FA	MFA	Purpose	Setting
Swarm and evolutionary intelligence setting	No. of fireflies	Yes	Yes	Population size	Depends on problem
	Max generation	Yes	Yes	Stopping criteria	Depends on problem
	Logistic chaotic map	N/A	Yes	Generate initial high diverse population of fireflies	Chaotic generation; biotic potential η is set to 4
Brightness	Objective function	Yes	Yes	Calculate the brightness of each firefly	Benchmark functions or the objective function of the problem
Attractiveness	β min	Yes	Yes	Minimum value of attractiveness parameter β	Default value: 0.1
	γ	Yes	Yes	Absorption coefficient	Default value: 1
	Chaotic Gauss/mouse map	N/A	Yes	Automatically tune β parameter	Chaotic generation
Random movement	α	Yes	Yes	Randomness of firefly movement	Default value: $\alpha_o = 0.2$
	Adaptive inertia weight	N/A	Yes	Control the swarm algorithm's local and global exploration capabilities	Default value: $\theta = 0.9$
	Lévy flight	N/A	Yes	Speed up the local search by generating new solutions around the best solution	Default value: $\tau = 1.5$

Fig. 2 Pseudo-code of proposed MFA

Begin

Perform objective function $f(x)$, $x = (x_1, ..., x_d)^T$

Set the search space and the number of generation

Generate initial population of fireflies x_i (i = 1, 2, ..., n) using chaotic map operator (logistic map) in defined search space \\ Initialize Chaotic FA function

while ($t \leq$ MaxGeneration) **do**

(1) AlphaNew() \\ Vary α according to the adaptive inertia weight ($\alpha = \alpha_0 \cdot 0.9^t$)

(2) BetaNew() \\ Vary new β *via* chaotic map operator (Gauss/mouse map)

(3) EvaluateMFA() \\Call the objective function $f(x)$

(4) SortMFA() \\ Sort objective function and find the current best solution

(5) MoveMFA() \\ Random component has altered by Lévy flight

end while

Find the best solution

Evaluate post-process results and visualization

End

to its stochastic characteristics. Therefore, 30 independent runs were performed for each benchmark function to eliminate any stochastic discrepancy. The maximum number of iterations was set to 1000 and the number of fireflies was 200, so the number of function evaluations (FEs) for each run was 200,000. The proposed MFA parameters were initially set to $\alpha = 0.2$, $\beta_{min} = 0.1$, and $\gamma = 1$ as shown in Table 1.

Table 2 Benchmark functions used in experiments

Name	Formulation	C	Range	Optima
Ackey (F1)	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + e$	MN	$[-30, 30]^D$	0
Sphere (F2)	$f_2(x) = \sum_{i=1}^{D} x_i^2$	US	$[-10, 190]^D$	0
Rosenbrock (F3)	$f_3(x) = \sum_{i=1}^{D}\left\{100\left(x_{i+1}-x_i^2\right)^2 + (x_i-1)^2\right\}$	UN	$[-10, 10]^D$	0
Rastrigin (F4)	$f_4(x) = \sum_{i=1}^{D}\left\{x_i^2 - 10\cos(2\pi x_i) + 10\right\}$	MS	$[-5.12, 5.12]^D$	0
Alpine (F5)	$f_5(x) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	M	$[-10, 10]^D$	0
Exponential (F6)	$f_6(x) = -\exp\left(\sum_{i=1}^{D} x_i^2 - 0.5\right)$	U	$[-1, 1]^D$	−1
2^n Minima (F7)	$f_7(x) = \sum_{i=1}^{D}\left(x_i^4 - 16x_i^2 + 5x_i\right)$	M	$[-5, 5]^D$	−78D
Levy and Montavo 2 (F8)	$f_8(x) = 0.1\Big(\sin^2(3\pi x_i) + \sum_{i=1}^{D}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right] + (x_D-1)^2\left[1+\sin^2(2\pi x_D)\right]\Big)$	M	$[-5, 5]^D$	0
Sinusoidal (F9)	$f_9(x) = -\left[2.5\prod_{i=1}^{D}\sin(x_i - 30) + \prod_{i=1}^{D}\sin(5(x_i - 30))\right]$	M	$[-180, 180]^D$	−3.5
Step (F10)	$f_{10}(x) = \sum_{i=1}^{D}\left(\lfloor x_i + 0.5 \rfloor\right)^2$	US	$[-100, 100]^D$	0

C Characteristic, *U* Unimodal, *M* Multimodal, *N* Non-separable, *S* Separable, *D* No. of dimensions

Table 3 Experimental results obtained using FA and MFA

Function	*D*	Optima	FA		MFA	
			Mean	*Stdev.*	*Mean*	*Stdev.*
F1	10	0	6.08E + 00	3.74E-01	**1.27E-05**	1.50E-06
	30	0	1.06E + 01	3.79E-01	**4.22E-05**	2.71E-06
	50	0	1.20E + 01	2.40E-01	**7.26E-05**	4.48E-06
F2	10	0	2.17E + 02	3.84E + 01	**1.26E-09**	2.16E-10
	30	0	2.05E + 03	2.01E + 02	**4.98E-08**	1.02E-08
	50	0	6.35E + 03	5.82E + 02	**2.31E-07**	4.59E-08
F3	10	0	5.22E-04	6.10E-04	**2.47E-03**	5.36E-03
	30	0	5.23E-04	4.34E-04	**1.48E-03**	1.62E-03
	50	0	4.86E-04	5.57E-04	**6.25E-04**	7.51E-04
F4	10	0	2.17E + 01	2.17E + 00	**1.26E + 00**	5.07E-01
	30	0	1.66E + 02	1.08E + 01	**3.50E + 00**	1.10E + 00
	50	0	3.48E + 02	1.55E + 01	**4.76E + 00**	1.22E + 00
F5	10	0	1.44E + 00	2.72E-01	**8.41E-07**	9.76E-08
	30	0	1.46E + 01	6.02E-01	**7.72E-06**	6.48E-07
	50	0	3.03E + 01	1.52E + 00	**4.63E-05**	1.47E-05
F6	10	−1	−9.91E-01	1.80E-03	**−1.00E + 00**	1.04E-08
	30	−1	−9.00E-01	8.82E-03	**−9.88E-01**	1.20E-02
	50	−1	−7.36E-01	1.81E-02	**−9.20E-01**	3.68E-02
F7	10	−78*D*	−7.15E + 02	3.46E + 01	**−7.42E + 02**	1.50E + 01
	30	−78*D*	**−1.88E + 03**	8.81E + 01	−1.82E + 03	5.23E + 01
	50	−78*D*	**−2.87E + 03**	1.09E + 02	−2.65E + 03	8.01E + 01
F8	10	0	1.20E-05	1.07E-05	**00E + 00**	00E + 00
	30	0	9.62E-06	8.33E-06	**00E + 00**	00E + 00
	50	0	1.19E-05	1.13E-05	**00E + 00**	00E + 00
F9	10	−3.50	−1.69E + 00	1.30E-01	**−3.50E + 00**	1.21E-08
	30	−3.50	−1.75E-02	1.60E-02	**−3.50E + 00**	3.98E-07
	50	−3.50	−9.44E-06	1.04E-05	**−3.43E + 00**	1.37E-01
F10	10	0	1.89E + 02	5.59E + 01	**1.33E-09**	2.33E-10
	30	0	2.98E + 03	3.99E + 02	**3.60E-08**	5.18E-09
	50	0	7.53E + 03	6.56E + 02	**1.82E-07**	1.55E-08

Bold values denote the best performance

D is no. of dimensions

The superiority of the MFA was established by comparing the average optimal values and standard deviations of optimal values with those obtained using other algorithms. Any value below 1E-12 was assumed to be zero. A nonparametric test, the Friedman test (Friedman 1937, 1940), which can be applied to multiple algorithms was utilized for this purpose. The Friedman test is a nonparametric procedure that used to test hypotheses concerning two or more algorithms. It is analogous to the repeated-measures analysis of variance in nonparametric statistical procedures, and detects substantial differences between the performances of at least two algorithms (Triguero 2016).

The null hypothesis for the Friedman test is H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$; the median of the population i equals the median of the population j, $i \neq j$, $1 \leq i, j \leq k$. The alternative is H_1: Not H_0. If Friedman's test results in the rejection of the null hypothesis, then a post-hoc test will be carried out to make the concrete pairwise comparisons that produce differences. The null hypothesis states that algorithms are equivalent to each other and their ranks are equal. If the null hypothesis is rejected, then the performances of the algorithms differ

Table 4 Statistical results of nonparametric test of two algorithms

Test	Mean ranking	Statistical difference	Critical difference (*CD*)	*p*-value
MFA	1.167			
FA	1.833	0.666	0.065	0.0002

Table 5 Additional benchmark functions used in experiments

Name	D	Formulation	C	Range	Optima
Colville (F11)	4	$f_{11}(x) = 100\left(x_1^2-x_2\right)^2 + (x_1-1)^2 + (x_3-1)^2 + 90\left(x_3^2-x_4\right)^2 + 10.1(x_2-1)^2 + (x_4-1)^2 + 19.8(x_2-1)(x_4-1)$	MN	$[-10, 10]^D$	0
Zakharov (F12)	10	$f_{12}(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^3 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	US	$[-5, 10]^D$	0
Powell (F13)	24	$f_{13}(x) = \sum_{i=1}^{D/4} \left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + x_{4i})^2 + (x_{4i-2} + 10x_{4i-1})^2 + 10(x_{4i-3} + x_{4i})^2 \right]$	UN	$[-4, 5]^D$	0
Rosenbrock (F3)	30	$f_3(x) = \sum_{i=1}^{D-1} \left\{100\left(x_{i+1}-x_i^2\right)^2 + (x_i-1)^2\right\}$	UN	$[-5.12, 5.12]^D$	0

D Dimension, *C* Characteristic, *U* Unimodal, *M* Multimodal, *N* Non-separable, *S* Separable

statistically. In that case, the Bonferroni-Dunn test (Iman and Davenport 1980) is carried out to detect statistically significant differences between the algorithms and the control algorithm (MFA) by calculating the critical difference (*CD*) between the mean ranks of the tested algorithms. When the statistical difference is higher than the *CD* value, the algorithms differ substantially.

The MFA and a traditional FA were compared under the same conditions. Table 3 presents the statistical results obtained by the MFA and the traditional FA in optimizing ten benchmark functions with three levels of dimensions in 30 runs. The comparison confirms that the MFA yielded more favorable optima in 28 out of 30 tests. In addition, the results of nonparametric tests shown in Table 4 indicate that the *p-values* is 0.0002, revealing the results of the tested algorithms differ significantly at a significance level of 0.1%. The statistical difference between the proposed MFA and the traditional FA exceeded the *CD* as shown in Table 4. Therefore, the MFA is more effective and robust than the FA in solving multidimensional problems.

To confirm the efficiency of the proposed MFA algorithm, complex problems that have been inefficiently optimized previously were solved using the MFA and its performance was compared with those of other well-known algorithms — GA, PSO, DE, and ABC (Karaboga and Akay 2009) — using the same benchmark functions including Colville, Zakharov, Powell, and Rosenbrock (Table 5). To maintain consistency with previous works, all comparison algorithms were set in the same conditions with 500,000 FEs and a population size of 50.

For the GA implementation, a binary-coded standard GA was used (Karaboga and Akay 2009). Single-point crossover with a rate of 0.8 was employed. The mutation rate was 0.01. The generation gap value was 0.9. For DE settings, the real constant which affects the differential variation between two solutions was set to 0.5. The value of crossover rate, which controls the change of the diversity of the population, was chosen to be 0.9 (Karaboga and Akay 2009). PSO was arranged with cognitive and social components of 1.8. The inertia weight, which determines how the previous velocity of the particle influences the velocity in the next iteration, was 0.6 (Karaboga and Akay 2009).

Tables 6 and 7 present the comparisons and the nonparametric tests for the proposed MFA, the GA, PSO, DE, and ABC. The MFA outperformed the other algorithms in terms

Table 6 Comparison of results obtained using MFA and well-known algorithms

Function	D	Optima	Measure	GA (Karaboga and Akay 2009)	PSO (Karaboga and Akay 2009)	DE (Karaboga and Akay 2009)	ABC (Karaboga and Akay 2009)	MFA
Colville (F11)	4	0	*Mean*	0.014938	**0**	0.0409122	0.0929674	**6.38E-10**
			Stdev	0.007364	**0**	0.081979	0.066277	**2.81E-10**
Zakharov (F12)	10	0	*Mean*	0.013355	**0**	**0**	0.0002476	**0**
			Stdev	0.004532	**0**	**0**	0.000183	**0**
Powell (F13)	24	0	*Mean*	9.703771	0.00011004	2.17E-07	0.0031344	**1.34E-09**
			Stdev	1.547983	0.000160	1.36E-07	0.000503	**1.34E-09**
Rosenbrock (F3)	30	0	*Mean*	1.96E + 05	15.088617	18.203938	0.0887707	**1.48E-03**
			Stdev	3.85E + 04	24.170196	5.036187	0.077390	**1.62E-03**

Bold values denote the best performance

Table 7 Statistical results of nonparametric test of five algorithms

Algorithm	Mean ranking	Statistical difference	Critical difference (*CD*)	*p*-value
MFA	1.75		0.510	0.0773
GA	4.50	2.25		
PSO	2.25	0.50		
DE	2.75	1.00		
ABC	3.75	2.00		

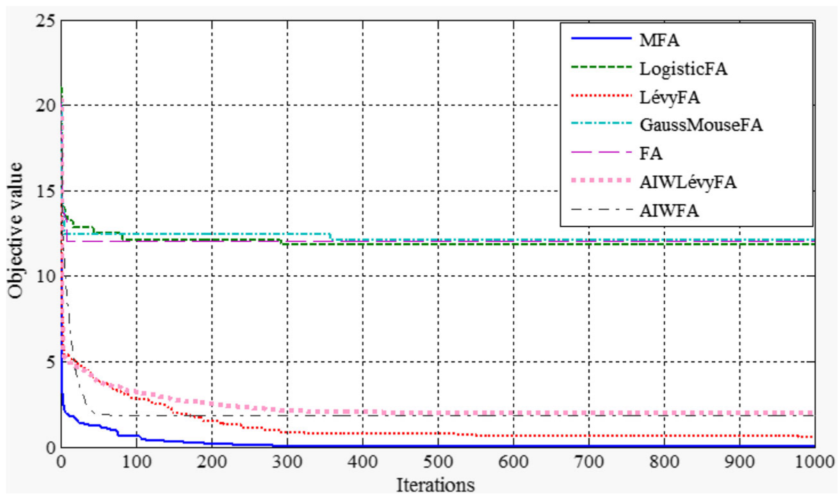

Fig. 3 Convergence curves of algorithms

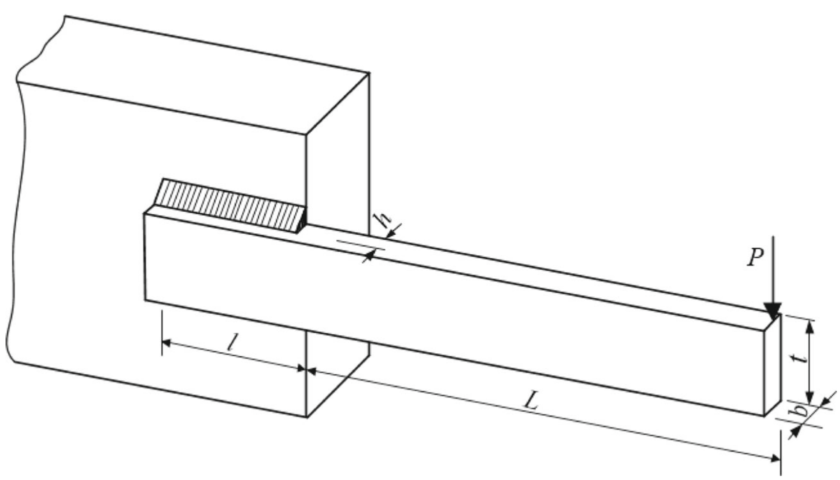

Fig. 4 Geometric parameters of welded beam design problem

L – Overhang length of the beam; *P* – Loading condition; *h* – thickness of the weld; *l* - length of the welded point; *t* – the width of the beam; *b* – the thickness of the beam.

of both mean value and standard deviation (Table 6). The statistical results in Table 7 demonstrate that the MFA is superior to the other algorithms except the PSO at a significance level of 0.10. The comparison proves that integrating the metaheuristic components into the conventional FA enhances its performance.

To identify the critical metaheuristic component that has the greatest impact on the performance of the MFA, a sensitivity analysis was carried out in this work. Each metaheuristic component was integrated with the conventional FA, and the performance after each integration was evaluated. Specifically, LogisticFA was derived by integrating the FA with the logistic chaotic map; GaussMouseFA was derived by integrating the FA with the Gauss/mouse chaotic maps; AIWFA was derived by integrating the FA with the AIW; and LévyFA was derived by integrating the FA and the Lévy flight. The parameters in all of these algorithms were set identically to provide a fair comparison. Figure 3 plots a convergence curve that was obtained from the algorithms in solving the Ackley function (F1), which is a typical benchmark function.

Figure 3 confirms that the MFA was superior to the other algorithms, followed by the LévyFA, and then the AIWFA. This comparison showed that the Lévy flight provides the greatest enhancement to the FA, followed by the AIW. The performance of the AIWLévyFA was evaluated through integration of the FA with Lévy flight and AIW. The simulation results revealed that the AIWLévyFA did not outperform the AIWFA or the LévyFA. This result confirmed the necessity of logistic and Gauss/mouse chaotic maps to enhance the MFA.

5 Structural engineering design problems

This section demonstrates the applicability of the MFA in solving three structural engineering design problems that were taken from the literature. These problems have recently been used by researchers to evaluate the performance of new optimization algorithms (Gandomi et al. 2011; Cheng and Prayogo 2014; Adekanmbi and Green 2015; Baykasoğlu and Ozsoydan 2015; Meng et al. 2015). Optimal solutions that were obtained using the MFA were compared with those obtained using other algorithms that were taken from the literature. To ensure a fair comparison, the maximum numbers of FEs were similar to those in previous works. The details of the application of the algorithms to the problems of interest are as follows.

5.1 Welded beam design

The welded beam design problem is a practical design problem that has been regarded as a benchmark problem (Rao 1996). The cost of fabrication of a welded beam must be minimized under constraints on the shear stress in the weld (τ), the bending stress in the beam (σ), the buckling load on the bar (P_c), the end deflection of the beam (δ), and side constraints. Figure 4 presents the geometrical parameters for the welded beam design problem.

Table 8 Values of constant parameters of welded beam design problem

Parameter	Description	Unit	Value
C1	Unit cost of the welded material	\$/in^3	0.10471
C2	Unit cost of the bar stock	\$/in^3	0.04811
L	Overhang length of the welded beam	in	14
τ_d	Design shear stress of the welded material	psi	13,600
σ_d	Design normal stress of the bar material	psi	30,000
P	Loading condition	lb	6000
δ_d	Design bar end deflection	in	0.25
E	Young's modulus of bar stock	psi	30×10^6
G	Shear modulus of bar stock	psi	12×10^6

Equation (13) provides the objective function of the problem. The design variables of the optimization problem are the thickness of the weld (h), the length of the welded joint (l), the width of the beam (t) and the thickness of the beam (b). Table 8 shows the values of the constant parameters. The bounds on the design variables are $0.125 \le h \le 5$, $0.1 \le l$, $t \le 10$, and $0.1 \le b \le 5$. When the constant parameters are replaced by the values in Table 8, the objective function is given by Equation (28).

$$\text{Objective function}: \quad f(h, l, t, b)$$
$$= (1 + C_1)h^2 l + C_2 tb(L + l) \tag{13}$$

subject to the following constraints;

– Shear stress in the weld (τ):

$$g_1(x) = \tau(x) - \tau_d \le 0 \tag{14}$$

– Bending stress in the beam (σ):

$$g_2(x) = \sigma(x) - \sigma_d \le 0 \tag{15}$$

– Geometric constraints:

$$g_3(x) = h - b \le 0 \tag{16}$$

Table 9 Settings for MFA and statistical results concerning welded beam design problem

Initial settings of MFA		Statistical results			
No. of fireflies	No. of iterations	Best	Mean	Worst	Stdev.
25	2000	1.7249	1.7277	1.7327	0.0024

– Buckling load on the bar (P_c):

$$g_4(x) = P - P_c(x) \le 0 \tag{17}$$

– Deflection of the beam (δ):

$$g_5(x) = \delta(x) - \delta_d \le 0 \tag{18}$$

– Cost constraint:

$$g_6(x) = C_1 h^2 + C_2 tb(14 + l) - 5 \le 0 \tag{19}$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{l}{2R} + (\tau'')^2} \tag{20}$$

$$\tau' = \frac{P}{\sqrt{2}hl} \tag{21}$$

$$\tau'' = \frac{P\left(L + l/2\right)R}{J} \tag{22}$$

$$R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2} \tag{23}$$

$$J = 2\left\{\frac{lh}{\sqrt{2}}\left[\frac{l^2}{12} + \left(\frac{h+t}{2}\right)^2\right]\right\} \tag{24}$$

$$\sigma(x) = \frac{6PL}{bt^2} \tag{25}$$

$$\delta(x) = \frac{4PL^3}{Et^3 b} \tag{26}$$

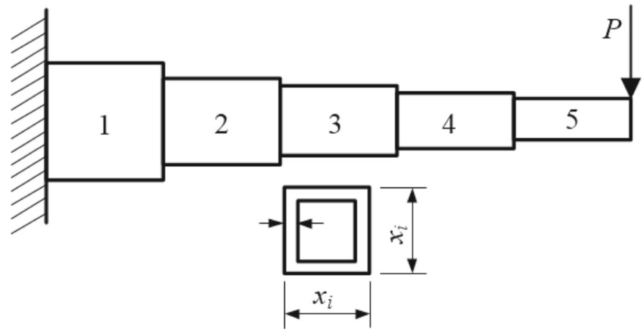

Fig. 5 Geometric parameters of cantilever design problem

$$P_c(x) = \frac{4.013\sqrt{EG\left(t^2b^6/36\right)}}{L^2}\left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}}\right) \tag{27}$$

$$\text{Minimize}: f(h, l, t, b) = 1.1047h^2 l + 0.04811tb(14 + l) \tag{28}$$

The proposed MFA with 25 fireflies yielded the optimal solution within 2000 iterations. Table 9 presents the initial settings of parameters of the MFA and the statistical results that were obtained from 30 runs. The optimal solution for the welded beam design was a fabrication cost of \$ 1.7249 with h=0.2057 (in), l=3.4712 (in), t=9.037 (in); b=0.2057 (in). The ratio between the highest and lowest optimal costs was 1.0045. The mean value and standard deviation were \$1.7277 and \$0.0024, respectively.

Table 10 compares the optimal solutions that were found by the MFA and the other algorithms of interest, including a socio-behavioral model (SBM) (Akhtar et al. 2002), a GA

Table 10 Comparison of optimal solutions for welded beam design problem

Author	Algorithm	Cost (\$)	h (in)	l (in)	t (in)	b (in)	FE
Akhtar et al. (2002)	SBM (Akhtar et al. 2002)	2.4426	0.2407	6.4851	8.2399	0.2497	19,259
Lemonge and Barbosa (2004)	GA (Lemonge and Barbosa 2004)	2.3816	0.2443	6.2117	8.3015	0.2443	320,000
He et al. (2004)	PSO (He et al. 2004)	2.3810	0.2444	6.2175	8.2915	0.2444	30,000
Lee and Geem (2005)	HS (Lee and Geem 2005)	2.3810	0.2442	6.2231	8.2915	0.2443	110,000
Hedar and Fukushima (2006)	FSA (Hedar and Fukushima 2006)	2.3811	0.2444	6.2158	8.2939	0.2444	56,243
Bernardino et al. (2007)	AIS-GA (Bernardino et al. 2007)	2.3812	0.2444	6.2183	8.2912	0.2444	320,000
Mahdavi et al. (2007)	IHS (Mahdavi et al. 2007)	1.7248	0.2057	3.4705	9.0366	0.2057	200,000
Fesanghary et al. (2008)	HHSA (Fesanghary et al. 2008)	1.7248	0.2057	3.4706	9.0368	0.2057	90,000
Zhang et al. (2008)	DSS-MDE (Zhang et al. 2008)	2.3810	0.2444	6.2175	8.2915	0.2444	24,000
Zhang et al. (2009)	RAER (Zhang et al. 2009)	2.3816	0.2443	6.2201	8.2940	0.2444	38,897
Gandomi et al. (2011)	FA (Gandomi et al. 2011)	1.7312	0.2015	3.5620	9.0414	0.2057	50,000
Present study	MSGA (Existing method)	1.7249	0.2057	3.4705	9.0366	0.2057	50,000
Present study	MFA (Proposed method)	1.7249	0.2057	3.4712	9.0370	0.2057	50,000

SBM Socio-behavioral model, *GA* Genetic algorithm, *PSO* Particle swarm optimization, *HS* Harmony search, *FSA* Filter simulated annealing, *AIS-GA* Artificial immune system – genertic algorithm, *IHS* Improved harmony search, *HHSA* a hybrid harmony search algorithm, *DSS-MDE* Dynamic stochastic selection - multimember differential evolution, *FA* Firefly algorithm, *MSGA* Multistart gradient-based algorithm, *RAER* Multiagent evolutionary optimization algorithm, *MFA* Modified firefly algorithm (the proposed algorithm), *FE* Function evaluation

Table 11 Comparison optimal solutions for cantilever beam design problem

Author	Method	f_{min}	x1	x2	x3	x4	x5	FE
Chickermane and Gea (1996)	MMA (Chickermane and Gea 1996)	1.3400	6.0100	5.3000	4.4900	3.4900	2.1500	N/A
Chickermane and Gea (1996)	GCA(I) (Chickermane and Gea 1996)	1.3400	6.0100	5.3000	4.4900	3.4900	2.1500	N/A
Chickermane and Gea (1996)	GCA(II) (Chickermane and Gea 1996)	1.3400	6.0100	5.3000	4.4900	3.4900	2.1500	N/A
Gandomi et al. (2013a, b)	CS (Gandomi et al. 2013a, b)	1.33999	6.0089	5.3049	4.5023	3.5077	2.1504	N/A
Cheng and Prayogo (2014)	SOS (Cheng and Prayogo 2014)	1.33996	6.01878	5.30344	4.49587	3.49896	2.15564	15,000
Present study	MSGA (Existing method)	1.3400	6.0160	5.3092	4.4943	3.5015	2.1527	15,000
Present study	MFA (Proposed method)	1.339957	6.01422	5.31220	4.48929	3.50375	2.15422	15,000

MMA Method of moving asymptotes, *GCA* Generalized convex approximation, *CS* Cuckoo search, *SOS* Symbiotic organisms search, *MSGA* Multistart gradient-based algorithm, *MFA* Modified firefly algorithm, *FE* Function evaluation

(Lemonge and Barbosa 2004), PSO (He et al. 2004), an HS (Lee and Geem 2005), a filter SA (FSA) (Hedar and Fukushima 2006), an artificial immune system – genetic algorithm (AIS-GA) (Bernardino et al. 2007), an improved HS (IHS) (Mahdavi et al. 2007), a hybrid HS algorithm (HHSA) (Fesanghary et al. 2008), a dynamic stochastic selection - multimember DE (DSS-MDE) (Zhang et al. 2008), a multiagent evolutionary optimization algorithm (RAER) (Zhang et al. 2009), a traditional FA (Gandomi et al. 2011), and the MSGA of this study.

Most of the algorithms, but not the IHS (Mahdavi et al. 2007), the HHSA (Fesanghary et al. 2008), the FA (Gandomi et al. 2011), and the MSGA found optimal costs higher than \$2.3. Although the IHS and the HHSA found a cost of \$1.7248, which is similar to that obtained by the proposed MFA, their numbers of FEs — 200,000 and 90,000, respectively — were notably higher than that of the proposed MFA. The MFA and MSGA yielded the similar optimal results of \$1.7249. The comparison results reveal that both the MSGA and the proposed MFA yielded the optimal design results with a fabrication cost of \$1.7249 in 50,000 evaluations. Therefore, the proposed MFA is an efficient algorithm for the welded beam design.

5.2 Cantilever beam design

The objective of the design of a cantilever beam is to minimize the weight of the beam. Fleury and Braibant were the first to consider this problem (Fleury and Braibant 1986). Figure 5 presents the geometrical parameters of the problem of designing a cantilever beam with five elements. Each element has a hollow cross-section of a fixed diameter. The beam is rigidly supported, as displayed in Fig. 5, and a vertical force is applied at the free end of the beam.

The design variables are the heights or widths of the cross-sectional area of the five elements (*i.e.,* x_1, x_2, x_3, x_4, and x_5). The constraints are set to $0.01 \le x_i \le 100$. Equation (29) presents the objective function.

$$\text{Objective function}: f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \quad (29)$$

subject to the constraint:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0 \quad (30)$$

Table 11 compares the performance of the optimal solutions that were obtained from the MFA and other algorithms reported in previous works, namely the method of moving asymptotes (MMA) (Chickermane and Gea 1996), generalized convex approximation I (GCA(I)) (Chickermane and Gea 1996), GCA(II) (Chickermane and Gea 1996), CS (Gandomi et al. 2013a, b), symbiotic organisms search (SOS) (Cheng and Prayogo 2014), and MSGA. The optimal objective value that was obtained using the MMA, GCA (I),

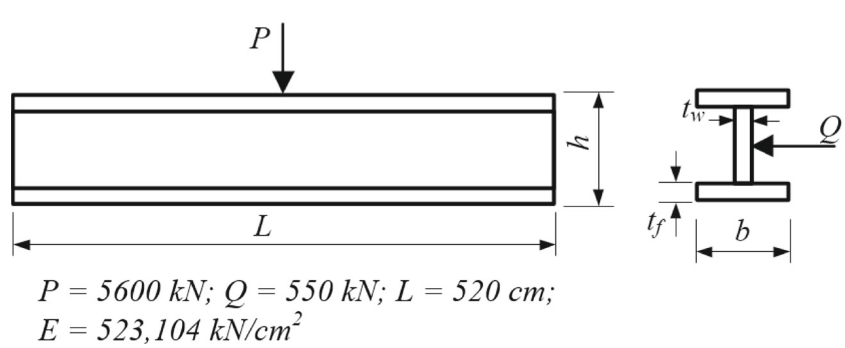

Fig. 6 Geometric parameters of I-beam problem

Table 12 Settings of MFA and statistical results concerning vertical deflection design of I-beam

Initial settings of MFA		Statistical results			
No. of fireflies	No. of iterations	Best	Average	Worst	Stdev.
25	200	0.0034	0.0039	0.0068	0.000716

GCA(II), and MSGA algorithms was 1.34, whereas those obtained using the CS and the SOS were slightly lower at 1.33999 and 1.33996, respectively. The proposed MFA yielded an optimal value of 1.339957 at $x_1 = 6.01422$, $x_2 = 5.31220$, $x_3 = 4.48929$, $x_4 = 3.50375$, and $x_5 = 2.15422$ with 25 fireflies and 600 iterations. The analysis again showed that the MFA outperformed the compared algorithms.

5.3 Design of I-beam

This case was modified from a problem that was presented elsewhere (Gold and Krishnamurty 1997). Figure 6 presents the geometrical parameters in an I-beam problem. The goal is to minimize the vertical deflection of an I-beam. A solution must simultaneously satisfy the cross-sectional area and stress constraints under given loads.

The design problem is to minimize the vertical deflection $f(x) = PL^3/48EI$ of the beam when the length of the beam (L) and modulus of elasticity (E) are 520 (cm) and 523,104 (kN/cm^2), respectively. Equation (31) is the objective function of the problem.

$$\text{Objective function}: f\left(b, h, t_w, t_f\right) = \frac{5000}{\frac{t_w\left(h-2t_f\right)}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2} \quad (31)$$

subject to the following constraints;

– The cross-sectional area is less than 300 cm^2:

$$g_1 = 2bt_w + t_w\left(h-2t_f\right) - 300 \leq 0 \quad (32)$$

– If the allowable bending stress of the beam is 56 kN/cm^2, then the stress constraint is as given by Equation (33)

$$g_2 = \frac{18h \times 10^4}{t_w\left(h-2t_f\right)^3 + 2bt_w\left(4t_f^2 + 3h\left(h-2t_f\right)\right)} + \frac{15h \times 10^3}{\left(h-2t_f\right)t_w^3 + 2t_w b^3} - 56 \leq 0 \quad (33)$$

where the initial design ranges are $10 \leq h \leq 80$, $10 \leq b \leq 50$, and $0.9 \leq t_w, t_f \leq 5$.

To make a fair comparison of the MFA with previously proposed algorithms, the number of FEs was set to 5000 with 25 fireflies and 200 iterations, as shown in Table 12. The statistical results verify that the optimal and average vertical deflections were 0.0034 (cm) and 0.0039 (cm), respectively. The simulation results reveal that the proposed MFA yielded the optimal value of 0.0034 (cm) at $h = 80$ (cm), $b = 50.0000$ (cm), $t_w = 1.7646$ (cm), and $t_f = 5.0000$ (cm).

Table 13 compares the optimal solutions that were found using the MFA and the other algorithms, which were an adaptive response surface method (ARSM) (Wang 2003), an improved ARSM (Wang 2003), CS (Gandomi et al. 2013a, b), SOS (Cheng and Prayogo 2014), and MSGA. The analytical results confirm that both MFA and MSGA yielded the smallest vertical deflection of 0.0034 (cm), followed by SOS, which yielded a deflection of 0.0130 (cm). Therefore, the MFA is one of the most effective algorithms for solving the problem of interest.

6 Conclusions

The proposed MFA was firstly evaluated by comparing its performance to that of a conventional FA and those of other well-known optimization algorithms using well-known benchmark functions. The comparison confirmed that the proposed MFA outperformed the conventional FA and the other algorithms in terms of average optimal values and the standard deviations of most considered benchmark functions. The

Table 13 Comparison of optimal solutions for vertical deflection problem of I-beam

Author	Method	f_{min}	*h (cm)*	*b (cm)*	t_w *(cm)*	t_f *(cm)*	FE
Wang (2003)	ARSM (Wang 2003)	0.0157	80.0000	37.0500	1.7100	2.3100	N/A
Wang (2003)	Improved ARSM (Wang 2003)	0.0131	79.9900	48.4200	0.9000	2.4000	N/A
Gandomi et al. (2013a, b)	CS (Gandomi et al. 2013a, b)	0.0135	80.0000	50.0000	0.9000	2.3216	5000
(Cheng and Prayogo 2014)	SOS (Cheng and Prayogo 2014)	0.0130	80.0000	50.0000	0.9000	2.3217	5000
Present study	MSGA (Existing method)	0.0034	80.0000	50.0000	1.7706	5.0000	5000
Present study	MFA (Proposed method)	0.0034	80.0000	50.0000	1.7646	5.0000	5000

ARSM Adaptive response surface method, *CS* Cuckoo search, *MSGA* Multistart gradient-based algorithm, *MFA* Modified firefly algorithm, *FE* Function evaluation statistical analysis verified that the MFA outperformed the traditional FA in 28 out of 30 tests. Nonparametric test confirmed that the MFA also was superior to the GA, PSO, DE, and ABC in solving the considered benchmark functions.

The MFA was then evaluated by using it to solve three real structural engineering problems The MFA obtained the values of 1.7249 ($), 1.339957 (weight unit), and 0.0034 (cm) for the welded beam, the cantilever beam, and the I-beam design problems, respectively. The MFA yielded more favorable solutions than the other algorithms reported in previous works in optimizing the above structural design problems (Tables 10, 11 and 13). Notably, the MFA and MSGA obtained the same optimial results for the welded beam and I-beam designs (Tables 10 and 13). The comparison indicates that the MFA was superior to other optimization algorithms when used to solve real structural engineering problems.

Although the analytical results demonstrate the superiority of the proposed MFA in solving multidimensional structural design optimization problems, further complex experiments such as truss problems with high dimensions should be conducted in the future. Another potential application could be a standalone interactive system that would deliver optimization solutions through a user-friendly graphical user interface.

References

Adekanmbi O, Green P (2015) Conceptual comparison of population based metaheuristics for engineering problems. Sci World J 2015:9

Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. Eng Optim 34(4):341–354

Alberdi R, Murren P, Khandelwal K (2015) Connection topology optimization of steel moment frames using metaheuristic algorithms. Eng Struct 100:276–292

Baykasoglu A (2012) Design optimization with chaos embedded great deluge algorithm. Appl Soft Comput 12(3):1055–1067

Baykasoğlu A, Akpinar Ş (2015) Weighted superposition attraction (WSA): a swarm intelligence algorithm for optimization problems – part 2: constrained optimization. Appl Soft Comput 37:396–415

Baykasoğlu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164

Benfratello S, Palizzolo L, Tabbuso P (2015) Optimization of structures with unrestricted dynamic shakedown constraints. Struct Multidiscip Optim 52(3):431–445

Bernardino HS, Barbosa IJC, Lemonge A (2007) A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. IEEE Congress on Evolutionary Computation, IEEE, Piscataway, p 646–653

Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112

Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39(5):829–846

Chou J-S, Ngo N-T (2016) Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns. Appl Energy 177:751–770

Chou J-S, Ngo N-T, Pham A-D (2016) Shear strength prediction in reinforced concrete deep beams using nature-inspired metaheuristic support vector regression. J Comput Civ Eng 30(1):04015002

Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144

Coelho L d S, Mariani VC (2013) Improved firefly algorithm approach applied to chiller loading for energy conservation. Energ Buildings 59:273–278

Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):1–33

Dos Santos Coelho L, Coelho AAR (2009) Model-free adaptive control optimization using a chaotic particle swarm approach. Chaos, Solitons Fractals 41(4):2001–2009

Dugré A, Vadean A, Chaussée J (2016) Challenges of using topology optimization for the design of pressurized stiffened panels. Struct Multidiscip Optim 53(2):303–320

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, Piscataway, p 39–43

Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y (2008) Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Comput Methods Appl Mech Eng 197(33–40):3080–3091

Fister I, Fister I Jr, Yang X-S, Brest J (2013a) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46

Fister I, Yang X-S, Brest J, Fister I Jr (2013b) Modified firefly algorithm using quaternion representation. Expert Syst Appl 40(18):7220–7230

Fister I Jr, Perc M, Kamal SM, Fister I (2015) A review of chaos-based firefly algorithms: perspectives and research challenges. Appl Math Comput 252:155–165

Fleury C, Braibant V (1986) Structural optimization: a new dual method using mixed variables. Int J Numer Methods Eng 23(3):409–428

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701

Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Stat 11(1):86–92

Gandomi AH, Yang X-S, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89(23–24):2325–2336

Gandomi A, Yang X-S, Alavi A (2013a) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35

Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013b) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98

Geisel T, Nierwetberg J, Zacherl A (1985) Accelerated diffusion in Josephson junctions and related chaotic systems. Phys Rev Lett 54(7):616–619

Gharooni-fard G, Moein-darbari F, Deldari H, Morvaridi A (2010) Scheduling of scientific workflows using a chaos-genetic algorithm. Procedia Comput Sci 1(1):1445–1454

Gold S, Krishnamurty S (1997) Trade-offs in robust engineering design. Proceeding of the ASME design engineering technical conferences, ASME, New York

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc., Boston

Gomes H (2012) A firefly metaheuristic structural size and shape optimisation with natural frequency constraints. Int J Metaheuristics 2(1):38–85

Guirguis D, Hamza K, Aly M, Hegazi H, Saitou K (2015) Multi-objective topology optimization of multi-component continuum structures via a Kriging-interpolated level set approach. Struct Multidiscip Optim 51(3):733–748

Haklı H, Uğuz H (2014) A novel particle swarm optimization algorithm with Levy flight. Appl Soft Comput 23:333–345

He S, Prempain E, Wu QH (2004) An improved particle swarm optimizer for mechanical design optimization problems. Eng Optim 36(5): 585-605

Hedar A-R, Fukushima M (2006) Derivative-free filter simulated annealing method for constrained continuous global optimization. J Glob Optim 35(4):521–549

Holden AV (1986) Chaos. Manchester University Press, Manchester

Hong W-C, Dong Y, Chen L-Y, Wei S-Y (2011) SVR with hybrid chaotic genetic algorithms for tourism demand forecasting. Appl Soft Comput 11(2):1881–1890

Iman RL, Davenport JM (1980) Approximations of the critical region of the fbietkan statistic. Commun Stat -Theory Methods 9(6):571–595

Jaberipour M, Khorram E (2010) Two improved harmony search algorithms for solving engineering optimization problems. Commun Nonlinear Sci Numer Simul 15(11):3316–3331

Jamil M, Yang X-S (2013) A literature survey of benchmark functions for global optimization problems. Int J Math Model Numer Optim 4(2): 150–194

Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471

Kazem A, Sharifi E, Hussain FK, Saberi M, Hussain OK (2013) Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Appl Soft Comput 13(2):947–958

Kripka M, Medeiros GF, Lemonge ACC (2015) Use of optimization for automatic grouping of beam cross-section dimensions in reinforced concrete building structures. Eng Struct 99:311–318

Lamberti L, Pappalettere C (2011) Metaheuristic design optimization of skeletal structures: a review. Comput Technol Rev 4:1–32

Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194(36–38):3902–3933

Lemonge ACC, Barbosa HJC (2004) An adaptive penalty scheme for genetic algorithms in structural optimization. Int J Numer Methods Eng 59(5):703–736

Liu B, Wang L, Jin Y-H, Tang F, Huang D-X (2005) Improved particle swarm optimization combined with chaos. Chaos, Solitons Fractals 25(5):1261–1271

Luo Q, Tong L (2015) Structural topology optimization for maximum linear buckling loads by using a moving iso-surface threshold method. Struct Multidiscip Optim 52(1):71–90

Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188(2):1567–1579

May RM (1976) Simple mathematical models with very complicated dynamics. Nature 261(5560):459–467

Meng X-B, Gao XZ, Liu Y, Zhang H (2015) A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization. Expert Syst Appl 42(17–18):6350–6364

Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. Appl Soft Comput 11(4):3658–3670

Pal SK, Rai CS, Singh AP (2012) Comparative study of firefly algorithm and particle swarm optimization for noisy Non-linear optimization problems. Int J Intell Syst Appl 4(10):50–57

Pavlyukevich I (2007) Lévy flights, non-local search and simulated annealing. J Comput Phys 226(2):1830–1844

Peri D, Tinti F (2012) A multistart gradient-based algorithm with surrogate model for global optimization. Commun Appl Ind Math 3(1):1-22

Rao SS (1996) Engineering optimization: theory and practice. John Wiley & Sons, Chichester

Rojas-Labanda S, Stolpe M (2015) Benchmarking optimization solvers for structural topology optimization. Struct Multidiscip Optim 52(3):527–547

Roque CMC, Martins PALS (2015) Differential evolution for optimization of functionally graded beams. Compos Struct 133:1191–1197

Saka MP, Dogan E (2012) Recent developments in metaheuristic algorithms: a review. Comput Technol Rev 5:31–78

Saka MP, Geem ZW (2013) Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review. Math Probl Eng 2013:33

Saka MP, Dogan E, Aydogdu I (2013) Review and analysis of swarm-intelligence based algorithms. Swarm intelligence and bio-inspired computation. Elsevier, London, pp 25–47

Sergeyev YD, Kvasov DE (2015) A deterministic global optimization using smooth diagonal auxiliary functions. Commun Nonlinear Sci Numer Simul 21(1–3):99–111

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. The IEEE International Conference on Evolutionary Computation, Piscataway, p 69–73

Solomon TH, Weeks ER, Swinney HL (1994) Chaotic advection in a two-dimensional flow: Lévy flights and anomalous diffusion. Physica D: Nonlinear Phenom 76(1–3):70–84

Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Tilahun SL, Ong HC (2012) Modified firefly algorithm. J Appl Math 2012:12

Triguero FH (2016) Statistical inference in computational intelligence and data mining. http://sci2s.ugr.es/sicidm

Wang GG (2003) Adaptive response surface method using inherited latin hypercube design points. J Mech Des 125(2):210–220

Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15(1):55–66

Wang Y, Li H-X, Huang T, Li L (2014) Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Appl Soft Comput 18:232–247

Weeks E, Solomon TH, Urbach J, Swinney H (1995) Observation of anomalous diffusion and Lévy flights. In: Shlesinger M, Zaslavsky G, Frisch U (eds) Lévy flights and related topics in physics. Springer Berlin Heidelberg, p 51–71

Yang X-S (2008) Firefly algorithm. Luniver Press, Bristol

Yang X-S (2010a) Firefly algorithm, Lévy flights and global optimization. In: Bramer M, Ellis R, Petridis M (eds) Research and development in intelligent systems XXVI. Springer, London, pp 209–218

Yang X-S (2010b) A new metaheuristic bat-inspired algorithm. In: González J, Pelta D, Cruz C, Terrazas G, Krasnogor N (eds) Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, pp 65–74

Yang X-S (2014a) Analysis of algorithms. Nature-inspired optimization algorithms. Elsevier, Oxford, pp 23–44

Yang X-S (2014b) Chapter 8 - firefly algorithms. In: Yang X-S (ed) Nature-inspired optimization algorithms. Elsevier, Oxford, pp 111–127

Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing, NaBIC 2009. p 210–214

Yang D, Liu Z, Zhou J (2014) Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. Commun Nonlinear Sci Numer Simul 19(4):1229–1246

Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178(15): 3043–3074

Zhang J, Liang C, Huang Y, Wu J, Yang S (2009) An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization. Appl Math Comput 211(2):392–416

Zhou G, Ma Z-D, Cheng A, Li G, Huang J (2015) Design optimization of a runflat structure based on multi-objective genetic algorithm. Struct Multidiscip Optim 51(6):1363–1371