Computers and Structures 89 (2011) 2325–2336

Contents lists available at SciVerse ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Mixed variable structural optimization using Firefly Algorithm

Amir Hossein Gandomi [a,*], Xin-She Yang [b], Amir Hossein Alavi [c]

[a] Department of Civil Engineering, University of Akron, Akron, OH 44325-3905, USA
[b] Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK
[c] School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 25 August 2010
Accepted 11 August 2011
Available online 13 September 2011

Keywords:
Metaheuristic optimization
Firefly Algorithm
Mixed continuous-discrete variable problems
Structural optimization

ABSTRACT

In this study, a recently developed metaheuristic optimization algorithm, the Firefly Algorithm (FA), is used for solving mixed continuous/discrete structural optimization problems. FA mimics the social behavior of fireflies based on their flashing characteristics. The results of a trade study carried out on six classical structural optimization problems taken from literature confirm the validity of the proposed algorithm. The unique search features implemented in FA are analyzed, and their implications for future research work are discussed in detail in the paper.

1. Introduction

Most design optimization problems in structural engineering are highly non-linear, include many different design variables and complicated constraints on stresses, displacements, load carrying capability, and geometrical configuration. Since non-linearity often results in multiple local optima only global algorithms should be used [1].

Another complication is that not all design variables are continuous, and some variables can only take certain discrete values. Mixed continuous/discrete optimization problems usually require search techniques that are problem-specific. However, there is no guarantee to find the global optimum. In large-scale problems, the existence of global optimum is not even ensured a priori.

Metaheuristic optimization algorithms can efficiently deal with this kind of optimization problems [2,3]. Metaheuristic techniques are global optimization methods that attempt to reproduce natural phenomena or social behavior: for example, biological evolution, stellar evolution, thermal annealing, animal behavior, music improvisation, etc. [2]. Two important characteristics of metaheuristic optimization methods are intensification and diversification [3]. Intensification serves to search around the current best solutions and to select the best candidate designs. Diversification allows the optimizer to explore the search space more efficiently, often by randomization.

Modern metaheuristic algorithms were developed to carry out global search trying to increase computational efficiency, solve larger problems, and implement robust optimization codes [2]. The metaheuristic algorithms do not have limitations in using sources (e.g. music-inspired harmony search [4]). However, nature is a principal source of inspiration to propose new metaheuristic optimization methods. Biologically-inspired algorithms are one of the main categories of the nature-inspired metaheuristic algorithms. More specifically, these algorithms are based on the selection of the fittest in biological systems which have evolved by natural selection over millions of years. Various bio-inspired optimization algorithms have been presented in literature. The most popular methods are genetic algorithm (GA) [5], particle swarm optimization (PSO) [6] and ant colony optimization (ACO) [7]. One of the new bio-inspired algorithms is cuckoo search (CS) [8] which is successfully applied to engineering problems [9]. More recently, Oftadeh et al. [10] conceptualized a new hunting search (HuS) metaheuristic algorithm inspired by group hunting of animals such as lions, wolves, and dolphins. Inspiring from the echolocation behavior of microbats. Yang [11] developed a new metaheuristic search algorithm, namely bat algorithm (BA). The BA has been successfully employed to optimize some benchmark engineering optimization problems [12]. Besides bio-inspired algorithms, there are the nature-inspired algorithms that mimic physical phenomena. Simulated annealing (SA) [13], big bang–big crunch (BB–BC) [14] and charged system search (CSS) [15].

A very promising recent development in the field of metaheuristic algorithms is the Firefly Algorithm (FA) proposed by Yang [16]. The FA algorithm is based on the idealized behavior of the flashing characteristics of fireflies. Preliminary studies indicate that FA is superior over GA and PSO [16].

* Corresponding author.
E-mail addresses: a.h.gandomi@gmail.com (A.H. Gandomi), xy227@cam.ac.uk (X.-S. Yang), ah_alavi@hotmail.com (A.H. Alavi).

0045-7949/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2011.08.002

This paper illustrates the use of FA in the solution of mixed variable structural optimization problems. Section 2 describes the proposed Firefly Algorithm. Section 3 explains how to handle nonlinear constraints. Test cases are described and optimization results are discussed in Section 4. Section 5 discusses the unique features of FA in comparison with other algorithms, and outlines directions for further research.

2. Firefly Algorithm

2.1. The proposed algorithm

The proposed Firefly Algorithm mimics the social behavior of fireflies flying in the tropical summer sky. Fireflies communicate, search for pray and find mates using bioluminescence with varied flashing patterns. By mimicking nature, various metaheuristic algorithms can be designed. In this paper, some of the flashing characteristics of fireflies were idealized so as to develop a firefly-inspired algorithm. For simplicity, only three rules were followed:

(1) All fireflies are unisex so that one firefly will be attracted at other fireflies regardless of their sex.
(2) Attractiveness is proportional to firefly brightness. For any couple of flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to the brightness which decreases with increasing distance between fireflies. If there are no brighter fireflies than a particular firefly, this individual will move randomly in the space.
(3) The brightness of a firefly is somehow related with the analytical form of the cost function. For a maximization problem, brightness can simply be proportional to the value of the cost function. Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms.

The basic steps of the FA are summarized by the pseudo code shown in Fig. 1 which consists of the three rules discussed above.

It should be noted that there is some conceptual similarity between the Firefly Algorithms and the bacterial foraging algorithm (BFA) [17]. However, there are some fundamental differences. First, in BFA, attraction among bacteria depends partly on their fitness and partly on their distance. Conversely, in FA, the attractiveness depends on cost function and decays monotonically with distance between fireflies. Second, individuals of FA have adjustable visibility and more versatility with respect to varying attractiveness: this usually leads to higher mobility and allows the search space to be explored more efficiently. Third, FA includes two important limit cases and it is possible to fine-tune the algorithm so to combine the advantages of both limit cases for exploring the search space more efficiently.

```
Firefly Algorithm
begin
    Objective function f (x), x = (x1, ..., xd)T
    Generate initial population of fireflies xi (i = 1,2, ...,n)
    Light intensity Ii at xi is determined by f (xi)
    Define light absorption coefficient γ
    while (t < MaxGeneration)
    for i = 1 : n all n fireflies
        for j = 1 : i all n fireflies
            if (Ij > Ii)
            Move firefly i towards j in d-dimension via L´evy flights
            end if
            Attractiveness varies with distance r via exp[-γr²]
            Evaluate new solutions and update light intensity
        end for j
    end for i
    Rank the fireflies and find the current best
    end while
    Postprocess results and visualization
end
```

Fig. 1. Pseudo-code of the proposed Firefly optimization algorithm.

2.2. Distance, attractiveness and limit cases

In the Firefly Algorithm, there are two important issues: the variation of light intensity and formulation of the attractiveness. For simplicity, we can always assume that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function.

As light intensity and thus attractiveness decreases as the distance from the source increases, the variations of light intensity and attractiveness should be monotonically decreasing functions. In most applications, the combined effect of both the inverse square law and absorption can be approximated using the following Gaussian form

$$I(r) = I_0 e^{-\gamma r^2} \tag{1}$$

where the light absorption coefficient γ can be taken as a constant. As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by

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

where β_0 is the attractiveness at $r = 0$. Eq. (2) defines a characteristic distance $\Gamma = 1/\sqrt{\gamma}$ over which the attractiveness changes significantly from β_0 to $\beta_0 e^{-1}$.

The distance between any two fireflies i and j at x_i and x_j, respectively, can be defined as the Cartesian distance $r_{ij} = \|x_i - x_j\|$.

The movement of a firefly i is attracted at another more attractive (brighter) firefly j is determined by

$$\Delta x_i = \beta_0 e^{-\gamma r^2}(x_j^t - x_i^t) + \alpha\varepsilon_i, \quad x_i^{t+1} = x_i^t + \Delta x_i \tag{3}$$

where the first term is due to the attraction, while the second term is randomization with α being the randomization parameter. Here ε_i is a vector of random numbers which are drawn from a Gaussian distribution. It is worth pointing out that this second term can be further improved by drawing from a Levy distribution [3]. That is, the step size is a random number drawn from

$$L(s) = A_s^{-1-q}, \quad A = q\Gamma(q)\sin(\pi q/2)/\pi, \tag{4}$$

where $\Gamma(q)$ is a Gamma function, and q is the exponent of the distribution. In the present case, we used $q = 3/2$.

From the implementation point of view, to generate the solution using (3), we should replace the last term by $\alpha\, L(s)$. For most problems, we can use a fixed value of $\alpha = 0.01$, while the $q = 1.5$ is used for all simulations. To generate a good, random step size $L(s)$ using Eq. (4), we can use the standard transformation method and other common techniques for generating pseudo-random numbers. Since (4) is only valid for large steps, we also set the minimum step size as 0.1, below which a standard Gaussian distribution $N(0,0.1)$ is used for simulations. Due to symmetry, we have generated symmetrical step sizes, so that an $L(s)$ takes both positive and negative values.

From Eq. (3), it is easy to see that there exist two limit cases when γ is small or large, respectively. When γ tends to zero, the attractiveness and brightness are constant: therefore, a firefly can be seen by all other fireflies; this is a special case. On the other hand, when γ is very large, then the attractiveness (and thus brightness) decreases dramatically, and all fireflies are short-sighted or equivalently fly in a deep foggy sky. This means that all fireflies move almost randomly, which corresponds to a random search technique. In general, the Firefly Algorithm corresponds to the situation between these two limit cases, and it is thus possible to fine-tune these parameters so that FA can outperform both PSO and random search. In fact, FA can find the global optima as well as all the local optima simultaneously in a very effective manner. This advantage will be demonstrated in detail later in the implementation. A further advantage of FA is that different fireflies will work almost independently. Therefore, FA is particularly suited for parallel implementation. It is even better than genetic algorithms and PSO because fireflies aggregate more closely around each optimum (without jumping around as in the case of genetic algorithms). The interactions between different subregions are minimal in parallel implementation.

2.3. Effect of internal parameters on the convergence behavior of Firefly Algorithm

The Firefly Algorithm requires the user to specify several internal parameters that may affect convergence behavior at different extents.

In the first place, the choice of initial locations or solutions is crucial for almost all stochastic population-based algorithms. The initial guess/solution should be generated as most diversely as possible to ensure that optimized design will be insensitive to initial population.

For that purpose, two strategies were followed in the present study: (a) to distribute initial solutions sampling rather uniformly the whole design space, thus not biasing any regions of the search space; (b) to generate each firefly as far as possible from other firefly, so to explore the search space more efficiently.

One hundred optimization runs were carried out for each initial population. It was found that optimization results are almost independent of the initial guess. This is confirmed by statistical measures such as mean value of cost function and corresponding standard deviation. This approach is far more general than simply relying on a few optimization runs.

Extensive sensitivity studies were carried out on the effect of population size and attractiveness.

It was found that the population size n = 10–25 is in general sufficient for most applications although a slightly higher population size may be used for more complex problems. A population of 50 fireflies should be adequate for almost all problems. Higher values are not recommended, as this will increase significantly computation time.

In addition, 100 different optimization runs were carried out for each combination of internal parameters, so as to produce statistically significant results. It was found that the best solutions are insensitive to the initial setting of parameters. However, the worst solutions and standard deviation slightly depend on the initial conditions: this can be due to the random nature of metaheuristic algorithms. Performing multiple optimization runs certainly contributes to make such dependence the least significant as possible. The γ parameter, which characterizes the variation of attractiveness, influences the convergence speed and the overall behavior of the FA algorithm. In theory, γ can range in the interval $(0, \infty)$.

However, its value depends on the characteristic length of γ of the system to be optimized: in most applications, it typically varies from 0.01 to 100. Preliminary investigation indicated that the simplest and yet efficient strategy is to set $\gamma = 1/\sqrt{L}$ where L is the typical length of design variables.

Since the initial value of attractiveness was found not to affect significantly optimization results, the fixed value $\beta_0 = 1$ was utilized.

Whilst the basic formulation of FA is very efficient, oscillatory behavior can however occur as the optimum design is approached. The solution quality can be improved by reducing the randomization parameter α with a geometric progression reduction scheme similar to the cooling schedule of simulated annealing. That is:

$$\alpha = \alpha_0 \theta^t \tag{5}$$

where $0 < \theta < 1$ is the reduction factor of randomization. This strategy was followed also in the present study by reducing α from 0.5 to 0.01.

However, the randomization parameter α should ideally be related to the actual scale of each design variable, as scales may vary significantly for different problems and even different variables within the same problem. In that case, it is usually a good idea to replace α by α_{sk} where the scaling parameters S_k $(k = 1, \ldots, d)$ in the d dimensions should be determined by the actual scales of the problem of interest.

3. Constraint handling

In the penalty function approach, nonlinear constraints can be collapsed with the cost function into a response functional. By doing this, the constrained optimization problem is transformed in an unconstrained optimization problem simpler to solve. For example, if there are some nonlinear equality constraints ϕ_i and some inequality constraints ψ_j, the response functional PI can be defined as follows:

$$\prod(x, \mu_i, v_j) = f(x) + \sum_{i=1}^{M} \mu_i \phi_i^2(x) + \sum_{j=1}^{N} v_j \psi_j^2(x) \tag{6}$$

where $1 \leqslant \mu_i$ and $0 \leqslant v_i$. The coefficients of penalty terms should be large enough: their values may depend on the specific optimization problem. The contribution of any equality constraints function to the response functional $\prod$ is null but increases significantly as soon as the constraint is violated. The same applies to inequality constraints when they become critical.

In the case of optimization problems including integer/discrete design variables, special cares should be taken to ensure that each new trial design can satisfy nonlinear constraint functions. Since some equality constraint function may be hard to satisfy because of the limited choice in terms of available discrete values of design variables, one can replace the critical constraint with two inequality constraint functions of the type $\leqslant 0$ and of the type $\geqslant 0$, respectively. This strategy facilitates convergence to the optimum design.

4. Implementation in structural optimization problems

The optimization algorithm was implemented in MATLAB™ 7.0. Optimization runs were executed on a PC with a 2.2 GHz Intel Dual Core processor and 1 GB of RAM memory. Optimization results were compared with data recently published in literature. The maximum number of optimization function evaluations varied from 25,000 to 75,000, depending on problem size.

4.1. Case I. Welded beam design

The welded beam shown in Fig. 2 must be designed for minimum cost of fabrication [18]. The beam is made of low carbon steel (C-1010), is welded to a rigid support and is loaded by the shear load P acting at the free tip. 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) were included as design variables. The values of h and l must be integer multiples of 0.0065 in. The objective function of the problem is expressed as follows:

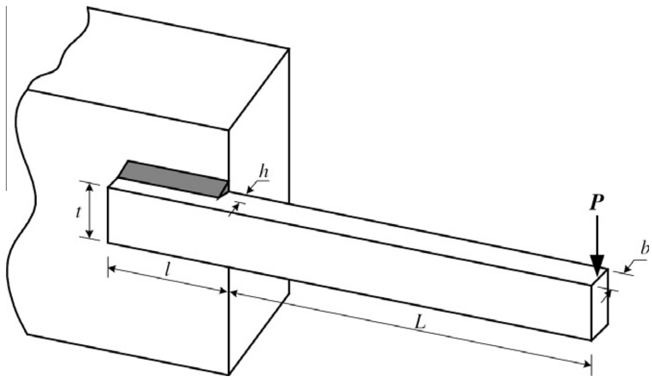

Fig. 2. Schematic of the welded beam design problem.

$$\text{Minimize}: \quad (f(h,L,t,b) = (1+C_1)h^2l + C_2tb(L+l) \tag{7}$$

which is subject to five constraints as follows:

- shear stress (τ)

$$g_1(x) = \tau_d - \tau(x) \geqslant 0 \tag{8}$$

- bending stress in the beam (σ)

$$g_2(x) = \sigma_d - \sigma(x) \geqslant 0 \tag{9}$$

- geometric constraints

$$g_3(x) = b - h \geqslant 0 \tag{10}$$

- buckling load on the bar (P_c)

$$g_4(x) = P_c(x) - P \geqslant 0 \tag{11}$$

- deflection of the beam (δ)

$$g_5(x) = 0.25 - \delta(x) \geqslant 0 \tag{12}$$

where

$$\tau(x) = \sqrt{(\tau'(x))^2 + (\tau''(x))^2 + l\tau'(x)\tau''(x)/\sqrt{0.25(l^2+(h+t)^2)}} \tag{13}$$

$$\sigma(x) = \frac{504000}{t^2b} \tag{14}$$

$$P_c(x) = 64746(1-0.0282346t)tb^3 \tag{15}$$

$$\delta(x) = \frac{2.1952}{t^3b} \tag{16}$$

$$\tau' = \frac{6000}{\sqrt{2}hl} \tag{17}$$

$$\tau'' = \frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+t)^2)}}{2\{0.707hl(l^2/12+0.25(h+t)^2)\}} \tag{18}$$

The simple bounds of the problem are: $0.125 \leqslant h \leqslant 5$, $0.1 \leqslant l$, $t \leqslant 10$ and $0.1 \leqslant b \leqslant 5$. The values of parameters involved in the formulation of the welded beam problem are also shown in Table 1.

With 25 fireflies, FA found the global optimum requiring 2000 iterations per optimization run. The results obtained by FA are presented in Table 2. The ratio between the optimized costs corresponding to worst and best designs is 1.355. Table 3 compares the optimization results found by FA with similar data reported in literature. Remarkably, FA obtained the best design overall of 1.7312. Mahdavi et al. [33] and Fesanghary et al. [34] found a better design but for the continuous optimization problem equal to 1.7248. In addition, FA requires only 50,000 function evaluations to complete the optimization process, hence much less than literature.

4.2. Case II. Pressure vessel design

The cylindrical pressure vessel capped at both ends by hemispherical heads (Fig. 3) must be designed for minimum cost. This optimization problem was originally formulated by Sandgren [40]. The compressed air tank has a working pressure of 3000 psi and a minimum volume of 750 ft^3, and must be designed according to the ASME code on boilers and pressure vessels. The total cost results from a combination of welding, material and forming costs. The thickness of the cylindrical skin (T_s), the thickness of the spherical head (T_h), the inner radius (R), and the length of the cylindrical segment of the vessel (L) were included as optimization variables. Thicknesses can only take discrete values which are integer multiples of 0.0625 in. The optimization problem can be stated as follows:

$$\text{Minimize}: \quad f(T_s,T_h,R,L) = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_h^2R \tag{19}$$

Constraints are set in accordance with the ASME design codes; g_3 represents the constraint on the minimum volume of 750 ft^3. The constraints are stated as follows:

$$g_1 = -T_s + 0.0193R \leqslant 0 \tag{20}$$

$$g_2 = -T_h + 0.00954R \leqslant 0 \tag{21}$$

$$g_3 = -\pi R^2L - \frac{4}{3}\pi R^3 + 750 \times 1728 \leqslant 0 \tag{22}$$

$$g_4 = L - 240 \leqslant 0 \tag{23}$$

where $1 \times 0.0625 \leqslant T_s$, $T_h \leqslant 99 \times 0.0625$, $10 \leqslant R \leqslant 200$ and $10 \leqslant L \leqslant 240$. Unlike the usual limit of 200 in considered in

Table 1
Values of parameters involved in the formulation of the welded beam problem.

Constant item	Description	Values
C_1	Cost per volume of the welded material	0.10471(\$/in^3)
C_2	Cost per volume of the bar stock	0.04811(\$/in^3)
τ_d	Design shear stress of the welded material	13600 (psi)
σ_d	Design normal stress of the bar material	30000 (psi)
δ_d	Design bar end deflection	0.25 (in)
E	Young's modulus of bar stock	30×10^6 (psi)
G	Shear modulus of bar stock	12×10^6 (psi)
P	Loading condition	6000 (lb)
L	Overhang length of the beam	14 (in)

Table 2
Statistical results of the FA optimization runs executed for the welded beam problem.

Best	Mean	Worst	S.D.	No. fireflies	No. iteration	Average time (s)
1.7312065	1.8786560	2.3455793	0.2677989	25	2000	23.2

Table 3
Welded beam problem: comparison of FA results with literature.

Researcher(s)	Method	h	l	t	b	Cost	No. evaluation
Coello [19]	GA	0.2088	3.4205	8.9975	0.2100	1.7483	N.A.
Leite and Topping [20]	GA	0.2489	6.1097	8.2484	0.2485	2.4000	6273
Deb [21]	GA	0.2489	6.1730	8.1789	0.2533	2.4331	320,080
Lemonge and Barbosa [22]	GA	0.2443	6.2117	8.3015	0.2443	2.3816	320,000
Bernardino et al. [23]	AIS[a]-GA	0.2444	6.2183	8.2912	0.2444	2.3812	320,000
Atiqullah and Rao [24]	SA[b]	0.2471	6.1451	8.2721	0.2495	2.4148	N.A.
Liu [25]	SA	0.2444	6.2175	8.2915	0.2444	2.3810	N.A.
Hwang and He [26]	SA-GA	0.2231	1.5815	12.8468	0.2245	2.2500	26,466
Hedar and Fukushima [27]	SA-DS[c]	0.2444	6.2158	8.2939	0.2444	2.3811	56,243
Parsopoulos and Vrahatis [28]	PSO	N.A.	N.A.	N.A.	N.A.	1.9220	100,000
He et al. [29]	PSO	0.2444	6.2175	8.2915	0.2444	2.3810	30,000
Zhang et al. [30]	EA[d]	0.2443	6.2201	8.2940	0.2444	2.3816	28,897
Coello [31]	EA	N.A.	N.A.	N.A.	N.A.	1.8245	N.A.
Lee and Geem [32]	HS[e]	0.2442	6.2231	8.2915	0.2443	2.381	110,000
Mahdavi et al. [33]	HS	0.2057	3.4705	9.0366	0.2057	1.7248	200,000
Fesanghary et al. [34]	HS-SQP[f]	0.2057	3.4706	9.0368	0.2057	1.7248	90,000
Siddall [35]	RA[g]	0.2444	6.2819	8.2915	0.2444	2.3815	N.A.
Akhtar et al. [36]	SBM[h]	0.2407	6.4851	8.2399	0.2497	2.4426	19,259
Ray and Liew [37]	SCA[i]	0.2444	6.2380	8.2886	0.2446	2.3854	33,095
Montes and Ocaña [38]	BFO[j]	0.2536	7.1410	7.1044	0.2536	2.3398	N.A.
Zhang et al. [39]	DE[k]	0.2444	6.2175	8.2915	0.2444	2.3810	24,000
Present study	FA	0.2015	3.562	9.0414	0.2057	1.73121	50,000

[a] Artificial immune system.
[b] Simulated annealing.
[c] Direct search.
[d] Evolutionary algorithms.
[e] Harmony search.
[F] Sequential quadratic programming.
[g] Random search.
[h] Socio-behavioral model.
[i] Society and civilization algorithm.
[j] Bacterial foraging optimization.
[k] Differential evolution.

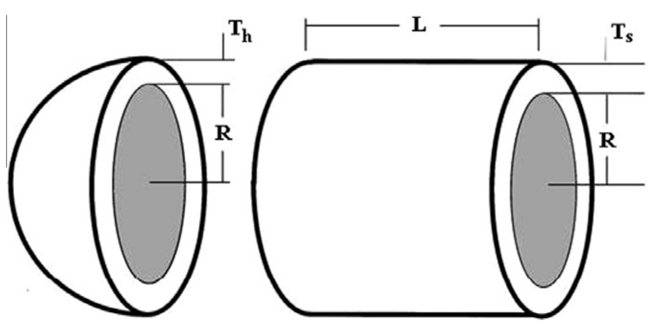

Fig. 3. Schematic of the pressure vessel design problem.

literature, the upper bound of design variable L was increased to 240 in to expand the search space.

Optimization results are presented in Table 4. With 25 fireflies, FA found the global optimum of 5850.383 within 25,000 function evaluations (i.e. 1000 optimization iterations). Table 5 compares the results obtained by FA with those reported in the literature [25,33,41]. It can be seen that designs optimized by PSO-GA and HS methods are not feasible because the third constraint (g_3) is violated. The optimum design found by SA-DS satisfies all constraints but corresponds to the highest cost overall. Therefore, the FA algorithm found the best results. The ratio between the optimized costs corresponding to worst and best designs is 1.069. The average execution time of the FA for this test case is 24.79 s.

It should be noted that the first and third inequalities become equalities. Therefore, once the first two variables are found, the third and fourth variables can be calculated from the first and third constraint functions.

4.3. Case III. Helical compression spring design

The helical compression spring shown in Fig. 4 is subject to an axially guided constant compression load. The spring must be designed for minimum volume. Spring ends are ground and squared. The winding coil diameter (D), wire diameter (d), and the number of spring coils (n) were included as optimization variables: D is continuous, n is an integer, and d can take one of the 42 discrete values listed in Table 6. The cost function to be minimized is the spring volume, expressed as:

$$\text{Minimize}: \quad f(D,d,n) = \frac{\pi D d^2 (n+2)}{4} \tag{24}$$

The following eight constraints specify the design limitations:

- The design shear stress caused by the compression load should be lower than the allowable maximum shear stress (S) of the material

Table 4
Statistical results of the FA optimization runs executed for the pressure vessel problem.

Best	Mean	Worst	S.D.	No. fireflies	No. iteration	Average time (s)
5850.38306	5937.33790	6258.96825	164.54747	25	1000	24.78667

Table 5
Pressure vessel problem: comparison of FA results with literature.

Reference Method	Dimopoulos [41] PSO-GA	Mahdavi et al. [33] HS	Hedar and Fukushima [27] SA-DS	Present study FA
$f_{\min}$	5850.38306	5849.76169	5868.76484	5850.38306
T_s	0.75	0.75	0.7683	0.75
T_h	0.375	0.375	0.3797	0.375
R	38.86010	38.86010	39.80962	38.86010
L	221.36549	221.36553	207.22555	221.36547
g_1	−0.0000	−0.0000	−0.0000	−0.0000
g_2	−0.0043	−0.0043	−0.0000	−0.0043
g_3	**0.0446**[a]	**0.2713**	−10.7065	−0.0134
g_4	−18.6345	−18.6345	−32.7744	−18.6345

[a] Bold sets are violated sets.

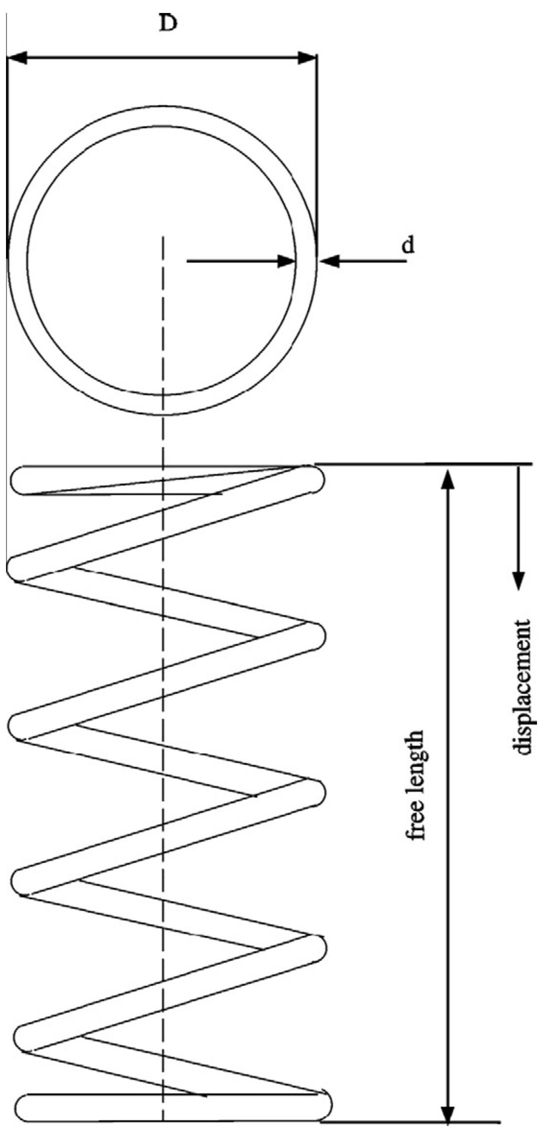

Fig. 4. Schematic of the helical compression spring design problem.

$$g_1 = \frac{8C_f P_{\max} D}{3.14156 d^3} - S \leqslant 0 \tag{25}$$

- The free length of the spring should be shorter than the maximum specified value L_{free}

$$g_2 = \frac{8KP_{\max} Dn}{C_f d^4} + 1.05(n+2)d - L_{free} \leqslant 0 \tag{26}$$

Table 6
Discrete values of the spring wire diameters.

d: Wire diameters (in)						
0.009	0.0095	0.0104	0.0118	0.0128	0.0132	0.014
0.015	0.0162	0.0173	0.018	0.020	0.023	0.025
0.028	0.032	0.035	0.041	0.047	0.054	0.063
0.072	0.080	0.092	0.105	0.120	0.135	0.148
0.162	0.177	0.192	0.207	0.225	0.244	0.263
0.283	0.307	0.331	0.362	0.394	0.4375	0.500

- The wire diameter must not be less than the specified minimum diameter $d_{\min}$

$$g_3 = d_{\min} - d \leqslant 0 \tag{27}$$

- The outer diameter of the coil should be smaller than the specified maximum diameter $D_{\max}$

$$g_4 = (d + D) - D_{\max} \leqslant 0 \tag{28}$$

- The inner coil diameter must be at least three times less than the wire diameter to avoid a lightly wound spring

$$g_5 = 3 - \frac{D-d}{d} \leqslant 0 \tag{29}$$

- The deflection under the given load δ must be less than the specified maximum deflection under preload δ_{pm}

$$g_6 = \delta - \delta_{pm} \leqslant 0 \tag{30}$$

- The combined deflection must be consistent with the coil free length L_{free}

$$g_7 = \frac{8KP_{\max} D^3 n}{C_f d^4} + \frac{P_{\max} - P_{load}}{K} + 1.05(n+2)d - L_{free} \leqslant 0 \tag{31}$$

- The deflection from preload to maximum load must be greater than the specified working deflection δ_w

$$g_8 = \delta_W - \frac{P_{\max} - P_{load}}{K} \leqslant 0 \tag{32}$$

where

$$C_f = \frac{4(S_i) - 1}{4(S_i) - 4} + \frac{0.615}{S_i} \tag{33}$$

$$S_i = \frac{D}{d} \tag{34}$$

$$K = \frac{Gd^4}{8nD^3} \tag{35}$$

Table 7
Values of parameters involved in the formulation of the helical spring problem.

Constant item	Description	Values
P_{max}	Maximum work load	1000.0 (lb)
S	Maximum shear stress	189×10^3 (psi)
E	Elastic modulus of the material	30×10^6 (psi)
G	Shear modulus of the material	11.5×10^6 (psi)
L_{free}	Maximum coil free length	14 (in)
d_{min}	Minimum wire diameter	0.2 (in)
D_{max}	Maximum outside diameter of the spring	3.0 (in)
P_{load}	Preload compression force	300.0 (lb)
δ_{pm}	Maximum deflection under preload	6.0 (in)
δ_w	Deflection from preload position to maximum load position	1.25 (in)

$$\delta_p = \frac{F_p}{K} \tag{36}$$

The values assigned to constant terms involved in the problem statement are listed in Table 7. The optimization results obtained by FA are presented in Table 8. With 25 fireflies, the optimization process was completed within 75,000 function evaluations. The ratio between the optimized costs corresponding to worst and best designs is 2.939. Table 9 compares the optimization results found by FA with similar data reported in literature. Whilst FA converged to the best design overall of 2.6586, some of the studies taken as reference found infeasible designs.

4.4. Case IV. A reinforced concrete beam design

The reinforced concrete beam shown in Fig. 5 must be designed for minimum cost. This problem was originally presented by Amir and Hasegawa [46]. The beam, simply supported at two points spaced by 30 ft, is subject to a live load of 2000 lbf and a dead load of 1000 lbf accounting for the beam weight. The concrete compressive strength (σ_c) is 5 ksi while the yield stress of the reinforcing steel (σ_y) is 50 ksi. The unit cost of concrete and steel are respectively 0.02 and 1.0 \$/in^2 per linear ft. The cross sectional area of the reinforcing bar (A_s), the width of the concrete beam (b) and the depth of the concrete beam (h) were included as optimization variables. The cross-sectional area of the reinforcing bar (A_s) is a discrete variable that must be chosen from the standardized dimensions listed in Table 10 and Ref. [46]. The width of the concrete beam (b) must be an integer variable. The effective depth is assumed to be 0.8 h.

The optimization problem can be expressed as:

$$\text{Minimize}: \quad f(A_s, b, h) = 29.4A_s + 0.6bh \tag{37}$$

Since the depth to width ratio of the beam should not exceed 4, the first optimization constraint can be written as:

$$g_1 = \frac{h}{b} - 4 \leqslant 0 \tag{38}$$

The structure must be designed so to satisfy the following safety requirement indicated in the ACI 318-77 code:

$$Mu = 0.9A_s\sigma_y(0.8h)\left(1.0 - 0.59\frac{A_s\sigma_y}{0.8bh\sigma_c}\right) \geqslant 1.4M_d + 1.7M_l \tag{39}$$

where M_u, M_d and M_l are, respectively, the flexural strength, dead load and live load moments of the beam. In this case, M_d = 1350 kip.in and M_l = 2700 kip.in. This constraint is simplified as [47]:

$$g_2 = 180 + 7.375\frac{A_s^2}{b} - A_sh \leqslant 0 \tag{40}$$

Design variables can vary as follows: b: $\{28,29,\ldots,40\}$ in, $5 \leqslant h \leqslant 10$ in, and the cross section of reinforcing bar can be chosen from discrete values listed in Table 10.

The optimization results obtained by FA are presented in Table 11. With 25 fireflies, the optimization process required 25,000 function evaluations. The ratio between the optimized costs corresponding to worst and best designs is 1.863.

Table 12 compares the optimization results found by FA with similar data reported in literature. FA converged to the best design

Table 8
Statistical results of the FA optimization runs executed for the helical spring problem.

Best	Mean	Worst	S.D.	No. fireflies	No. iteration	Average time (s)
2.658575665	4.3835958	7.8162919	4.6076313	25	3000	17.3

Table 9
Helical spring problem: comparison of FA results with literature.

Researcher(s) Method	Sandgren [40] N.A.	Guo et al. [42] PSO	Wu and Chow [43] GA	Deb and Goyal [44] GA	Yun [45] GA-FL[a]	Present study FA
d	0.283	0.283	0.283	0.283	0.263	0.283
D	1.180701	1.223	1.227411	1.226	1.1096	1.223049
N	10	9	9	9	9	9
g_1	−5430.9	−1008.81	−550.993	−713.51	**25154.82**	−1008.02
g_2	−8.8187	−8.946	−8.9264	−8.933	−9.1745	−8.946
g_3	−0.08298	−0.083	−0.0830	−0.083	−0.063	−0.083
g_4	−1.8193	−1.77696	−1.7726	−1.491	−1.890	−1.777
g_5	−1.1723	−1.3217	−1.3371	−1.337	−1.219	−1.322
g_6	−5.4643	−5.4643	−5.4485	−5.461	−5.464	−5.464
g_7	0	0	0	0	0	0
g_8	0.0000	**0.0001**	−0.0134	−0.0090	−0.0014	0.0000
f_{min}	2.7995	2.659	2.6681	2.665	2.0283	2.6586
No. Eval.	N.A.	N.A.	N.A.	N.A.	100,000	50,000

Artificial neural networks.
Bold sets are violated sets.
[a] Hybrid GA with fuzzy logic.

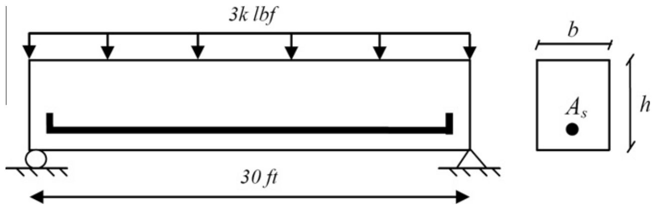

Fig. 5. Schematic of the reinforced concrete beam design problem.

overall of 359.208, about 1% less than the best optimized cost of 362 quoted in literature [48].

Interestingly, the second best solution with X = [6358.85] and $f_{\min}$ = 362.25 was obtained by considering 15 fireflies. The optimization process was completed within only 15,000 function evaluations thus saving 40% of CPU time with respect to the previous case. By increasing the number of fireflies to 25, the global optimum of 359.208 was found. This suggests carrying out more investigations on how FA deals with mixed integer optimization problems. It could be that the optimizer misses some portions of design space corresponding to variable bounds. This is confirmed by the fact that the cost function built for the reinforced concrete beam problem increases as all design variables increase: therefore, the global minimum should correspond to the smallest values of optimization variables forming a trial point contained in a feasible region of design space. Extreme points can be determined by setting one of the constraints equal to 0. By doing this, two points can be obtained: one design vector ([6.32,34,8.5]) corresponds to the global optimum with $f_{\min}$ = 359.208 while the other corresponds to the local optimum with cost equal to 362. Since most random number generators such as the "rand" function available in the MATLAB™ software cannot generate exactly 0, there is the possibility that some edge points are not visited, and thus the fireflies fly away in this case. A similar behavior can be observed also in particle swarm optimization as the boundaries of search domain are visited less frequently.

4.5. Case V: Stepped cantilever beam design

The stepped cantilever beam shown in Fig. 6 must be designed for minimum volume. This problem was originally presented by Thanedar and Vanderplaats [49]. The cantilever beam is comprised of five segments of variable cross section: since the height and width of the cross sectional area of each segment are chosen as design variables, this test case included 10 optimization variables. The cost function of the problem was stated as:

$$\text{Minimize}: \quad V = D(b_1h_1l_1 + b_2h_2l_2 + b_3h_3l_3 + b_4h_4l_4 + b_5h_5l_5) \qquad (41)$$

Subject to the following constraints [49]:

- Bending stresses in each beam segment must be lower than the allowable limit (σ_d):

$$g_1 = \frac{6Pl_s}{b_5h_5^2} - \sigma_d \leqslant 0 \qquad (42)$$

$$g_2 = \frac{6P(l_s + l_4)}{b_4h_4^2} - \sigma_d \leqslant 0 \qquad (43)$$

$$g_3 = \frac{6P(l_s + l_4 + l_3)}{b_3h_3^2} - \sigma_d \leqslant 0 \qquad (44)$$

$$g_4 = \frac{6P(l_s + l_4 + l_3 + l_2)}{b_2h_2^2} - \sigma_d \leqslant 0 \qquad (45)$$

$$g_5 = \frac{6P(l_s + l_4 + l_3 + l_2 + l_1)}{b_1h_1^2} - \sigma_d \leqslant 0 \qquad (46)$$

Table 10
Discrete values of the reinforcing bars.

Bar type	A_s (in^2)	Bar type	A_s (in^2)	Bar type	A_s (in^2)	Bar type	A_s (in^2)
1#4	0.2	6#5	1.86	9#6	3.95	9#8	7.11
1#5	0.31	10#4, 2#9	2	4#9	3.96	12#7	7.2
2#4	0.4	7#5	2.17	13#5	4	13#7	7.8
1#6	0.44	11#4, 5#6	2.2	7#7	4.03	10#8	7.9
3#4, 1#7	0.6	3#8	2.37	14#5	4.2	8#9	8
2#5	0.62	12#4, 4#7	2.4	10#6	4.34	14#7	8.4
1#8	0.79	8#5	2.48	15#5	4.4	11#8	8.69
4#4	0.8	13#4	2.6	6#8	4.65	15#7	9
2#6	0.88	6#6	2.64	8#7	4.74	12#8	9.48
3#5	0.93	9#5	2.79	11#6	4.8	13#8	10.27
5#4, 1#9	1	14#4	2.8	5#9	4.84	11#9	11
6#4, 2#7	1.2	15#4, 5#7, 3#9	3	12#6	5	14#8	11.06
4#5	1.24	7#6	3.08	9#7	5.28	15#8	11.85
3#6	1.32	10#5	3.10	7#8	5.4	12#9	12
7#4	1.4	4#8	3.16	13#8	5.53	13#9	13
5#5	1.55	11#5	3.41	10 7, 6#9	5.72	14#9	14
2#8	1.58	8#6	3.52	14#6	6	15#9	15
8#4	1.6	6#7	3.6	8#8	6.16		
4#6	1.76	12#5	3.72	15#6, 11#7	6.32		
9#4, 3#7	1.8	5#8	–6	7#9	6.6		

Table 11
Statistical results of the FA optimization runs executed for the reinforced concrete beam problem.

Best	Mean	Worst	S.D.	No. Cuckoos	No. iteration	Average time (s)
359.2080	460.706	669.150	80.73870	25	1000	7.47

Table 12
Reinforced concrete beam problem: comparison of FA results with literature.

Reference	Amir and Hasegawa [46]	Shih and Yang [48]		Yun [45]		Montes and Ocaña [38]	Present study
Method	SD-RC[a]	GHN-ALM[b]	GHN-EP[c]	GA	GA-FL	BFO	FA
$f_{\min}$	374.2	362.2455	362.00648	366.1459	364.8541	376.2977	359.2080
A_s	7.8	6.6	6.32	7.20	6.16	N/A	6.32
b	31	33	34	32	35	N/A	34
h	7.79	8.495227	8.637180	8.0451	8.7500	N/A	8.5000
g_1	−4.2012	0.0159[d]	−0.7745	−2.8779	−3.6173	N/A	−0.2241
g_2	−0.0205	−0.1155	−0.0635	−0.0224	0	N/A	0
No. evaluation	396	N.A.	N.A.	100,000	100,000	30,000	25,000

[a] Hybrid discrete steepest descent and rotating coordinate directions methods.
[b] Generalized Hopfield network based augmented Lagrange multiplier approach.
[c] GHN based extended penalty approach.
[d] Violated set.

- The deflection of the cantilever beam tip must be smaller than the limit deflection ($D_{\max}$):

$$g_6 = \frac{Pl^3}{3E}\left(\frac{1}{I_s} + \frac{7}{I_4} + \frac{19}{I_3} + \frac{37}{I_2} + \frac{61}{I_1}\right) - \Delta_{\max} \leqslant 0 \tag{47}$$

- The aspect ratio between height and width of the cross section of each segment must be less than 20:

$$g_7 = \frac{h_5}{b_5} - 20 \leqslant 0 \tag{48}$$

$$g_8 = \frac{h_4}{b_4} - 20 \leqslant 0 \tag{49}$$

$$g_9 = \frac{h_3}{b_3} - 20 \leqslant 0 \tag{50}$$

$$g_{10} = \frac{h_2}{b_2} - 20 \leqslant 0 \tag{51}$$

$$g_{11} = \frac{h_1}{b_1} - 20 \leqslant 0 \tag{52}$$

The design space includes six discrete variables (the cross-section dimensions of the first three segments of the cantilever beam) and is defined as follows: b_1: {1,2,3,4,5}, b_2, b_3: {2.4,2.6,2.8,3.1}, h_1, h_2: {45.0,50.0,55.0,60.0}, h_3: {30,31,...,65}, $1 \leqslant b_4,\ b_5 \leqslant 5$, and $30 \leqslant h_4,\ h_5 \leqslant 65$. All dimensions are expressed in cm. The values of constant parameters involved in the formulation of this optimization problem are listed in Table 13.

Table 14 presents the results obtained by FA. With 25 fireflies, the optimization process was completed within 50,000 function evaluations. The ratio between the optimized costs corresponding to the worst and best designs is 1.006. It can be seen from Table 15 that FA is superior over the other nonlinear optimization methods presented in literature. Remarkably, even the worst design of 64262.994 found by FA is better than all designs listed in Table 15.

4.6. Case VI: Car side impact design

Design of car side impact is also used as a benchmark problem of the proposed FA. The FEM model of this problem is illustrated in Fig. 7. In this case, the finite element dummy model is used which

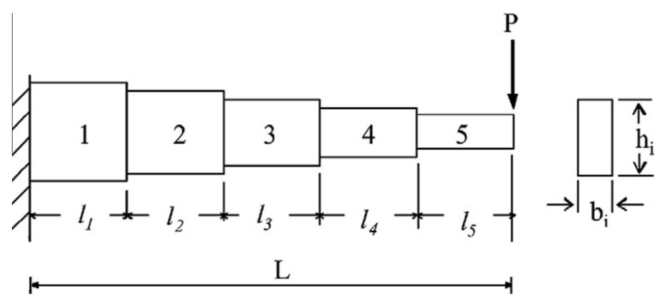

Fig. 6. Schematic of the stepped cantilever beam design problem.

contains a simplified regression model. On the foundation of European Enhanced Vehicle-Safety Committee (EEVC) procedures, a car is exposed to a side-impact. Here we want to minimize the weight using nine influence parameters including, thicknesses of B-Pillar inner, B-Pillar reinforcement, floor side inner, cross members, door beam, door beltline reinforcement and roof rail (x_1–x_7), materials of B-Pillar inner and floor side inner (x_8 and x_9) and barrier height and hitting position (x_{10} and x_{11}). The car side problem is stated by Gu et al. [53] and as an optimization problem it can be formulated as follows:

$$\text{Minimize} \quad f(x) = \text{Weight} \tag{53}$$

Subject to

$$g_1(x) = F_a \text{ (load in abdomen)} \leqslant 1 \text{ kN} \tag{54}$$

$$g_2(x) = \mathrm{VC}_u \text{ (dummy upper chest)} \leqslant 0:32 \text{ m/s} \tag{55}$$

$$g_3(x) = \mathrm{VC}_m \text{ (dummy middle chest)} \leqslant 0:32 \text{ m/s} \tag{56}$$

$$g_4(x) = \mathrm{VC}_l \text{ (dummy lower chest)} \leqslant 0:32 \text{ m/s} \tag{57}$$

$$g_5(x) = \Delta_{\mathrm{ur}} \text{ (upper rib deflection)} \leqslant 32 \text{ mm} \tag{58}$$

$$g_6(x) = \Delta_{\mathrm{mr}} \text{ (middle rib deflection} \leqslant 32 \text{ mm} \tag{59}$$

$$g_7(x) = \Delta_{\mathrm{lr}} \text{ (lower rib deflection)} \leqslant 32 \text{ mm} \tag{60}$$

$$g_8(x) = F_p \text{ (Pubic force)} \leqslant 4 \text{ kN} \tag{61}$$

$$g_9(x) = V_{\mathrm{MBP}} \text{ (Velocity of V-Pillar at middle point)} \leqslant 9:9 \text{ mm/ms} \tag{62}$$

$$g_{10}(x) = \mathrm{V}_{\mathrm{FD}} \text{ (Velocity of front door at V-Pillar)} \leqslant 15:7 \text{ mm/ms} \tag{63}$$

Table 13
Values of parameters involved in the formulation of the stepped cantilever beam problem.

Constant item	Description	Values
P	Concentrated load	50,000 (N)
σ_d	Design bending stress	14,000 (N/cm^2)
E	Elastic modulus of the material	2×10^7 (N/cm^2)
$D_{\max}$	Allowable deflection	2.7 (cm)
L	Total length of the five-stepped cantilever beam	500 (cm)

Table 14
Statistical results of the FA optimization runs executed for the stepped cantilever beam problem.

Best	Mean	Worst	S.D.	No. fireflies	No. iteration	Average time (s)
63893.52578	64144.75312	64262.99420	175.91879	25	2,000	12.4

Table 15
Stepped cantilever beam problem: comparison of FA results with literature.

Ref.	Method	No. evaluation	b1	h1	b2	h2	b3	h3	b4	h4	b5	h5	Objective
[51]	RNES[a] 1	12,000	3	60	3.1	55	2.6	50	2.311	43.108	1.822	34.307	64269.59
	RNES 2	12,000	3	60	3.1	55	2.6	50	2.267	43.797	1.849	34.282	64322.43
	RNES 3	12,000	3	60	3.1	55	2.6	50	2.348	42.804	1.783	34.753	64299.11
	RNES 4	12,000	3	60	3.1	55	2.6	50	2.491	41.51	2.113	33.231	65416.90
[52]	DOT	N.A.							N.A				65391.59
	SLP[b]	N.A.							N.A				65451.50
	MLD[c]-SLP	N.A.							N.A				65352.20
[49]	C/RU[d]	N.A.	4	62	3.1	60	2.6	55	2.205	44.09	1.751	35.03	73555
	PD[e]	N.A.	3	60	3.1	55	2.6	50	2.276	45.528	1.75	34.995	64537
	LAD[f]	N.A.	3	60	3.1	55	2.6	50	2.262	45.233	1.75	34.995	64403
	CAD[g]	N.A.	3	60	3.1	55	2.6	50	2.279	45.553	1.75	35.004	64403
[50]	GA 1	10,000	3	60	3.1	55	2.6	50	2.3	45.5	1.8	35	64558
	GA 2	10,000	3	60	3.1	55	2.6	50	2.27	45.25	1.75	35	64447
[22]	GA-APM[h]	35,000	3	60	3.1	55	2.6	50	2.289	45.626	1.793	34.593	64698.56
[23]	AIS-GA	35,000	3	60	3.1	55	2.6	50	2.235	44.395	2.004	32.879	65559.6
	AIS-GA-C[i]	35,000	3	60	3.1	60	2.6	50	2.311	43.186	2.225	31.250	66533.47
Present study		50,000	3	60	3.1	55	2.6	50	2.205	44.091	1.750	34.995	63893.52

[a] Rank-niche evolution strategy.
[b] Sequential linear programming.
[c] Move limit definition.
[d] Continuous/round up.
[e] Precise discrete.
[f] Linear approximate discrete.
[g] Conservative approximate discrete.
[h] Adaptive penalty method.
[i] Clearing.

Structural weight and response to impact can be approximated using global response surface methodology in order to simplify the analytical formulation of the optimization problem and speed up computations. The simplified models are defined as follows [54]:

$$\text{Weight} = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \tag{64}$$

$$F_a = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} \tag{65}$$

$$\text{VC}_u = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_10 + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} \tag{66}$$

$$\text{VC}_m = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_10 + 0.00121x_8x_{11} \tag{67}$$

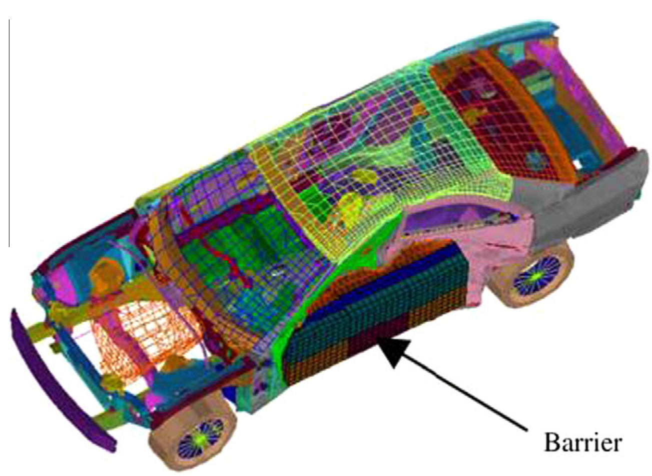

Fig. 7. Finite element model utilized in the car side impact problem [54].

$$VC_l = 0.74 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 \quad (68)$$

$$\Delta_{ur} = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} \quad (69)$$

$$\Delta_{mr} = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 \quad (70)$$

$$\Delta_{lr} = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} \quad (71)$$

$$F_p = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 \quad (72)$$

$$V_{MBP} = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} \quad (73)$$

$$V_{FD} = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 \quad (74)$$

The simple bounds of this problem are: $0.5 \leqslant x_1, x_3, x_4 \leqslant 1.5$; $0.45 \leqslant x_2 \leqslant 1.35$; $0.875 \leqslant x_5 \leqslant 2.625$; $0.4 \leqslant x_6, x_7 \leqslant 1.2$; $x_8, x_9 \in \{0.192, 0.345\}$; $0.5 \leqslant x_{10}, x_{11} \leqslant 1.5$.

In this test case, FA was run with 20 fireflies and 1000 iterations. The optimization problem was also solved using other meta-heuristic algorithms such as PSO, GA and differential evolution (DE) in order to have a valid basis of comparison with the proposed FA algorithm. Since implementation details may affect algorithm performance in terms of computation time, in the present study, the style of implementation was maintained the most uniform as possible so to minimize such differences. Furthermore, the relative performance of FA, PSO, GA and DE was evaluated over the same number of searches: this approach allowed the difference in behavior due to the inherent nature of algorithms to emerge clearly.

Table 16 summarizes the statistical results obtained by the different optimization algorithms for the car side impact design problem after 20,000 searches. It can be seen that the FA presented in this research outperformed GA and was slightly better than PSO and DE with the same number of evaluations and optimization runs. However, PSO presents a smaller standard deviation on optimized weight.

5. Discussions and conclusions

This paper introduced a new metaheuristic optimization algorithm, FA, to solve mixed variable structural optimization problems. The Firefly Algorithm mimics the social behavior of fireflies. The FA code was tested in six structural optimization problems taken from literature including welded beam design, pressure vessel design, helical compression spring design, reinforced concrete beam designs, stepped cantilever beam design, and car side impact design. The optimization results indicate that FA is more efficient than other metaheuristic algorithms such as PSO, GA, SA and HS.

Although FA is very efficient, oscillatory behavior was observed as the search process approaches the optimum design. The overall behavior of the FA algorithm can be improved by reducing gradually the randomization parameter as the optimization progresses. This can be an interesting topic of investigation.

As a relatively straightforward extension, FA can be modified to make it suited for multi-objective optimization problems. Hybridization of FA with other metaheuristic algorithms also may be an interesting direction for further research.

Table 16
Optimization results for the car side impact design problem.

Method	PSO	DE	GA	FA
Best objective	22.84474	22.84298	22.85653	22.84298
x_1	0.50000	0.50000	0.50005	0.50000
x_2	1.11670	1.11670	1.28017	1.36000
x_3	0.50000	0.5000	0.50001	0.50000
x_4	1.30208	1.30208	1.03302	1.20200
x_5	0.50000	0.50000	0.50001	0.50000
x_6	1.50000	1.50000	0.50000	1.12000
x_7	0.50000	0.50000	0.50000	0.50000
x_8	0.34500	0.34500	0.34994	0.34500
x_9	0.19200	0.19200	0.19200	0.19200
x_{10}	−19.54935	−19.54935	10.3119	8.87307
x_{11}	−0.00431	−0.00431	0.00167	−18.99808
Mean objective	22.89429	23.22828	23.51585	22.89376
Worst objective	23.21354	24.12606	26.240578	24.06623
S.D.	0.15017	0.34451	0.66555	0.16667

It should be noted that various studies attempted to set performance indicators for comparing different optimization algorithms. For example, Shilane et al. [55] suggested a framework for evaluating performance of evolutionary algorithms on a statistical basis. Because of the random nature of metaheuristic algorithms, the computational cost of the optimization process may be slightly sensitive to numerical implementation. For example, vectorized implementation in MATLAB™ may reduce the CPU time with respect to the case where 'FOR' loops are utilized. Relative merits of different optimization algorithms should hence be assessed independently of their numerical implementation. For example, comparing best designs obtained after a given number of structural analyses may be an effective strategy to capture differences caused by the inherent characteristics of each optimization algorithm. This issue is certainly an important research topic for algorithm development and software engineering.

More comprehensive trade studies comparing different optimization algorithms should be carried out in order to identify the strength points and weakness of metaheuristic algorithms currently available. Ultimately, even better optimization algorithms may emerge.

Acknowledgements

The authors gratefully acknowledge the work and help of Engineer Parvin Arjmandi.

References

[1] Yang X-S. Nature-inspired metaheuristic algorithms. Luniver Press; 2008.
[2] Talbi E. Metaheuristics: from design to implementation. Hoboken, New Jersey, USA: John Wiley & Sons; 2009.
[3] Yang X-S. Firefly algorithm, Lévy flights and global optimization. In: Bramer M et al., editors. Research and development in intelligent systems XXVI. London: Springer-Verlag; 2010. p. 209–18.
[4] Geem ZW, Kim J-H, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.
[5] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[6] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory, in: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan, 1995.
[7] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996;26(1):29–41.
[8] Yang X-S, Deb S. Cuckoo search via Lévy flights. Proceedings of the world congress on nature & biologically inspired computing (NaBIC 2009). IEEE Publications; 2009. p. 210–4.
[9] Gandomi AH, Yang X-S, Alavi AH. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput, in press. doi:10.1007/s00366-011-0241-y.
[10] Oftadeh R, Mahjoob MJ, Shariatpanahi M. A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. Comput Math Appl 2010;60:2087–98.

[11] Yang X-S. A new metaheuristic bat-inspired algorithm. In: Gonzalez JR et al., editors. Nature inspired cooperative strategies for optimization NISCO 2010, studies in computational intelligence, vol. 284. Berlin: Springer; 2010. p. 65–74.
[12] Yang X-S, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. Memetic Comput. Eng Comput, in press.
[13] Kirtrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220(4598):671–680,.
[14] Osman K. Erol, Ibrahim Eksin, A new optimization method: Big Bang–Big Crunch. Adv Eng Software 2006;37(2):106–11.
[15] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–89.
[16] Yang X-S. Firefly algorithms for multimodal optimization. in: Watanabe O, Zeugmann T, editors. Stochastic algorithms: foundations and applications, SAGA 2009, Lecture notes in computer science, vol. 5792. Berlin: Springer-Verlag; 2009. p. 169–78.
[17] Passino KM. Biomimicry of bacterial foraging for distributed optimization. Princeton: University Press; 2001.
[18] Rao SS. Engineering optimization: theory and practice. 3rd ed. Chichester, UK: John Wiley & Son; 1996.
[19] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Indus 2000;41(2):113–27.
[20] Leite JPB, Topping BHV. Improved genetic operators for structural optimization. Adv Eng Software 1998;29(7–9):529–62.
[21] Deb K. Optimal design of a welded beam via genetic algorithms. AIAA J 1991;29(11):2013–5.
[22] Lemonge ACC, Barbosa HJC. An adaptive penalty scheme for genetic algorithms in structural optimization. Int J Numer Methods Eng 2004;59:703–36.
[23] Bernardino HS, Barbosa HJC, Lemonge ACC. A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. IEEE Congr Evolution Comput 2007:646–53.
[24] Atiqullah Mir M, Rao SS. Simulated annealing and parallel processing: an implementation for constrained global design optimization. Eng Optimiz 2000;32(5):659–85.
[25] Hedar AR, Fukushima M. Derivative-free filter simulated annealing method for constrained continuous global optimization. J Global Optimiz 2006;35(4):521–649.
[26] Liu J-L. Novel orthogonal simulated annealing with fractional factorial analysis to solve global optimization problems. Eng Optimiz 2005;37(5):499–519.
[27] Hwang S-F, He R-S. A hybrid real-parameter genetic algorithm for function optimization. Adv Eng Inform 2006;20:7–21.
[28] Parsopoulos KE, Vrahatis MN. Unified particle swarm optimization for solving constrained engineering optimization problems. Lecture Notes in Computer Science 3612 (LNFA), 2005. p. 582–91.
[29] He S, Prempain E, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems. Eng Optimiz 2004;36(5):585–605.
[30] Zhang J, Liang C, Huang Y, Wu J, Yang S. An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization. Appl Math Comput 2009;211:392–416.
[31] Coello CAC. Constraint-handling using an evolutionary multiobjective optimization technique. Civil Eng Environ Syst 2000;17:319–46.
[32] Lee KS, Geem ZW. A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 2004;194:3902–33.
[33] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Appl Math Comput 2007;188:1567–79.
[34] Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y. Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Comput Methods Appl Mech Eng 2008;197:3080–91.
[35] Siddall JN. Analytical decision-making in engineering design. Englewood Cliffs: Prentice-Hall; 1972.
[36] Akhtar S, Tai K, Ray T. A socio-behavioural simulation model for engineering design optimization. Eng Optimiz 2002;34(4):341–54.
[37] Ray T, Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evolution Comput 2003;7(4):386–96.
[38] Montes EM, Ocaña BH. Modified bacterial foraging optimization for engineering design. In: Dagli Cihan H, et al., editors. Proceedings of the Artificial neural networks in engineering conference (ANNIE'2009), ASME Press series, Intelligent engineering systems through artificial neural networks, vol. 19, 2009. p. 357–64.
[39] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. Inform Sci 2008;178(15):3043–74.
[40] Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 1990;112(2):223–9.
[41] Dimopoulos GG. Mixed-variable engineering optimization based on evolutionary and social metaphors. Comput Methods Appl Mech Eng 2007;196:803–17.
[42] Guo CX, Hu JS, Ye B, Cao YJ. Swarm intelligence for mixed-variable design optimization. J Zhejiang Univ Sci 2004;5(7):851–60.
[43] Wu SJ, Chow PE. Genetic algorithms for nonlinear mixed discrete-integer optimization problems via metagenetic parameter optimizations. Eng Optimiz 1995;24:137–59.
[44] Deb K, Goyal M. Optimizing engineering designs using a combined genetic search. In: Back IT, editors. 7th international conference on genetic algorithms, 1997. p. 512–28.
[45] Yun YS. Study on adaptive hybrid genetic algorithm and its applications to engineering design problems. Waseda University: MSc thesis, 2005.
[46] Amir HM, Hasegawa T. Nonlinear mixed-discrete structural optimization. J Struct Eng 1989;115(3):626–45.
[47] Liebman JS, Khachaturian N, Chanaratna V. Discrete structural optimization. J Struct Eng, ASCE 1981;107(ST11):2177–97.
[48] Shih CJ, Yang YC. Generalized Hopfield network based structural optimization using sequential unconstrained minimization technique with additional penalty strategy. Adv Eng Software 2002;33:721–9.
[49] Thanedar PB, Vanderplaats GN. Survey of discrete variable optimization for structural design. J Struct Eng, ASCE 1995;2(121):301–6.
[50] Erbatur F, Hasancebi O, Tutuncu I, Kilic H. Optimal design of planar and space structures with genetic algorithms. Comput Struct 2000;75:209–24.
[51] Chen TY, Chen HC. Mixed-discrete structural optimization using a rank-niche evolution strategy. Eng Optimiz 2009;41(1):39–58.
[52] Lamberti L, Pappalettere C. Move limits definition in structural optimization with sequential linear programming. Part II: Numerical examples. Comput Struct 2003;81:215–38.
[53] Gu L, Yang RJ, Cho CH, Makowski M, Faruque M, Li Y. Optimization and robustness for crashworthiness. Int J Vehicle Des 2001;26(4):348–60.
[54] Youn BD, Choi KK. A new response surface methodology for reliability-based design optimization. Comput Struct 2004;82:241–56.
[55] Shilane D, Martikainen J, Dudoit S, Ovaska SJ. A general framework for statistical performance comparison of evolutionary computation algorithms. Inform Sci: Int J 2008;178:2870–9.