Computers and Structures 235 (2020) 106268

Contents lists available at ScienceDirect

Computers and Structures

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

Interactive autodidactic school: A new metaheuristic optimization algorithm for solving mathematical and structural design optimization problems

Milad Jahangiri [a,*], Mohammad Ali Hadianfard [a], Mohammad Amir Najafgholipour [a], Mehdi Jahangiri [b], Mohammad Reza Gerami [c]

[a] Faculty of Civil and Environmental Engineering, Shiraz University of Technology, Shiraz, Iran
[b] Department of Mechanical Engineering, Shiraz University, Shiraz, Iran
[c] Deputy Director of the International Scientific Cooperation Unit, Farhangian Leading Student Education University, Tehran, Iran

ARTICLE INFO

Article history:
Received 1 September 2019
Accepted 16 April 2020
Available online 30 April 2020

Keywords:
Metaheuristic optimization
Interactive autodidactic school
Mathematical optimization
Structural optimization

ABSTRACT

A new efficient and robust metaheuristic algorithm called "Interactive Autodidactic School (IAS)" is proposed in this paper to solve numerical optimization and structural design optimization problems. IAS is a population-based algorithm on the basis of the interactions between students in an autodidactic school with the goal of increasing their knowledge through a combination of self-teaching/self-learning, interactive discussion, criticism, and the competition. IAS is tested in twenty mathematical optimization and seven structural optimization problems. Subsequently, its optimum solution is compared with other well-known optimization algorithms. The obtained results confirmed that the proposed IAS algorithm gives best optimal solution and has excellent performance compared with other optimization methods.

1. Introduction

Optimization algorithms in engineering problems are significant computational tools that have become widespread over the last decades. These optimization algorithms are classified into analytical and metaheuristic methods [1]. The analytical approaches, also called "gradient-based algorithms", are deterministic and always give the same optimal solution using an identical initial point. Some examples of analytical methods are: Conjugate Gradient (CG), Steepest Descent (SD), Davidson-Fletcher-Powell (DFP), Broyden-Fletcher-Goldfarb-Shanno (BFGS), and Augmented Lagrange Multiplier (ALM) [2]. Although these numerical methods are powerful in solving optimization problems, they have three prominent drawbacks compared with metaheuristic methods. First, they are not applicable when the objective function and the constraints are discrete since their gradients are not defined. Second, because of their dependence on the value of the initial point, they could be trapped in local minima. Finally, they are unstable and unreliable when the objective function and the constraints have multiple or sharp peaks [1,2]. Accordingly, to solve complex engineering optimization problems, the researchers turned to novel stochastic approaches with specific features instead of traditional analytical approaches.

The metaheuristic methods, also called "stochastic algorithms", either simulate the natural phenomena or are inspired by the social behavior of living creatures [3,4]. A list of the previously proposed metaheuristic optimization algorithms is presented in Table 1. A majority of metaheuristic optimization algorithms have similar features: they are stochastic and random walk algorithms, are independent of gradient information, are iterative methods, and are applicable for continuous and discrete problems [3]. The performance and efficiency of each metaheuristic algorithm depends on the complexity of cost function and the constraints that define the feasible search space [4].

In this paper, a new simple metaheuristic algorithm, named "Interactive Autodidactic School (IAS)", is proposed in order to obtain global optimum solutions for constrained structural engineering design optimization problems. This algorithm simulates the interactions within a group of students trying to educate themselves without the help of a teacher, thus forming an autodidactic school trailing. Similar to other population-based algorithms, the IAS algorithm iteratively uses a population within which, the forerunner member is called the leading student and the rest of the community, the trailing of following students with the aim of exploring the search space to reach the optimal solution.

* Corresponding author.
E-mail address: Mi.Jahangiri@sutech.ac.ir (M. Jahangiri).

https://doi.org/10.1016/j.compstruc.2020.106268
0045-7949/

Table 1
The comprehensive history of metaheuristic optimization algorithms.

No.	Year	Optimization Algorithm	No.	Year	Optimization Algorithm
1	1975	Genetic Algorithm (GA) [5]	26	2013	Cuttlefish Algorithm (CFA) [30]
2	1983	Simulated Annealing (SA) [6]	27	2013	Mine Blast Algorithm (MBA) [31]
3	1989	Tabu Search (TS) [7]	28	2014	Grey Wolf Optimizer (GWO) [32]
4	1995	Particle Swarm Optimization (PSO) [8]	29	2014	Symbiotic Organism Search (SOS) [33]
5	1997	Differential Evolution (DE) [9]	30	2015	Lightning Search Algorithm (LSA) [34]
6	1999	Ant Colony Optimization (ACO) [10]	31	2015	Ant Lion Optimizer (ALO) [35]
7	2001	Harmony Search (HS) [11]	32	2015	Passing Vehicle Search (PVS) [36]
8	2003	Shuffled Frog Leaping (SFL) [12]	33	2016	Whale Optimization Algorithm (WOA) [37]
9	2006	Bees Algorithm (BA) [13]	34	2016	Sine Cosine Algorithm (SCA) [38]
10	2006	Glowworm Swarm Optimization (GSO) [14]	35	2016	Crow Search Algorithm (CSA) [39]
11	2006	Big Bang–Big Crunch (BB-BC) [15]	36	2016	Lion Optimization Algorithm (LOA) [40]
12	2007	Artificial Bee Colony (ABC) [16]	37	2016	Water Evaporation Optimization (WEO) [41]
13	2007	Imperialist Competitive Algorithm (ICA) [17]	38	2016	Jaya Algorithm (JA) [42]
14	2007	River Formation Dynamics (RFD) [18]	39	2017	Electro-Search (ES) [43]
15	2007	Intelligent Water Drops (IWD) [19]	40	2017	Most Valuable Player Algorithm (MVPA) [44]
16	2009	Firefly Algorithm (FFA) [20]	41	2017	Grasshopper Optimization Algorithm (GOA) [45]
17	2009	Gravitational Search Algorithm (GSA) [21]	42	2017	Salp Swarm Algorithm (SSA) [46]
18	2009	Cuckoo Search (CS) [22]	43	2018	Artificial Flora (AF) [47]
19	2010	Charged System Search (CSS) [23]	44	2018	Multi-Verse Optimization (MVO) [48]
20	2010	Bat Algorithm (BA) [24]	45	2018	Butterfly Optimization Algorithm (BOA) [49]
21	2011	Teaching-Learning-Based Optimization (TLBO) [25]	46	2018	Interactive search Algorithm (ISA) [50]
22	2012	Water Cycle Algorithm (WCA) [26]	47	2018	Lion Pride Optimization Algorithm (LPOA) [51]
23	2012	Flower Pollination Algorithm (FPA) [27]	48	2019	Algorithm of the Innovative Gunner (AIG) [52]
24	2013	Ray Optimization (RO) [28]	49	2019	Flying Squirrel Optimizer (FSO) [53]
25	2013	Dolphin Echolocation (DE) [29]	50	2019	Sooty Tern Optimization Algorithm (STOA) [54]

The present work is organized as follows: The main concept and the flowchart of the IAS algorithm are described in Section 2 in detail. The implementation of IAS algorithm for optimization problems is explained in Section 3. The efficiency of the proposed method is verified by solving twenty classical benchmark mathematical optimization problems with side constraints in Section 4. Afterward, the performance of the IAS algorithm is assessed by solving seven structural engineering design optimization problems and its results are compared with other well-known algorithms. Finally, the concluding remarks of this research are summarized in Section 5.

2. The interactive autodidactic school (IAS) algorithm

Metaheuristic algorithms have been described under several classifications, mainly due to their diversity and versatility. Such classifications include but are not limited to nature-inspired against non-nature inspired, population-based against single point search, dynamic against static objective function, single neighborhood against various neighborhood structures, memory usage against memory-less methods, and numerical against analytical methods [55–57]. The aforementioned classes could overlap by some extent and population-based algorithms inspired by nature are very popular in optimization practice.

The core idea of the proposed IAS algorithm is adoptable to the knowledge development process in a community of self-learning students. The final level of knowledge of the students, which is represented through their marks, is increased in the course of an autodidactic process in interaction with other community members. The paradigm used in such an autodidactic school leads to gradual increase of the competence and skill of the students, not at the same pace for everybody, however. Some students, due to their personal or social advantages, could outstand the community and take the lead. Nevertheless, due to the progressive increase of the collective knowledge of the community, the leading role is under a continuous competition. The final goal of this process is the achievement of certain level of knowledge by one of the community members which is demonstrated through his didactic performance. The knowledge growth in the above processes is obtained through a combination of self-teaching/self-learning, interactive discussion and criticism, and the competition for leadership.

Similar to other population-based algorithms, the IAS generates randomly an initial population called the students. The eligibility of the students to attend IAS is controlled by the lower and upper bound values specified by a given problem. The student with the highest performance (minimum mark) at any stage, is granted the position of the "leading student", or simply the "leader". In the IAS optimization, the best performance is realized when the minimum value of the cost function is achieved. This position, however, could be passed to more competent students at any instant of the process. The procedure of generation and competence assessment of the students in the school could be described as follows:

$$For \;\; i = 1 : N\_student$$

$$S_i = LB + r_i(0,1) * (UB - LB);$$

$$M_i = f(S_i);$$

$$End \;\; For$$

$$f(LS) = \min\{M\}$$

where S_i is the ith student generated, LB and UB are the lower and the upper bounds of design variables, respectively,$r_i(0,1)$ is a random number between 0 and 1,$N\_student$ is the number of the students, M_i is the ith student's mark, and LS is the leading student.

The self-learning/self-training sessions in this interactive school are held in three different stages: "individual training", "collective training" and "challenge of the new student". In the individual training session, the leading student tries to discuss with and teach the trailing students, one by one. In the collective training session, which in fact consists of a pre-session discussion of the trailing students about the latest session and then collectively attending a trouble-shooting session with the leading student, the knowledge of the trailing students is improved in a social procedure; firstly, by discussion with their own and secondly, by the collective trouble-shooting session. Finally, the challenge of the new student denotes the event of joining of a new student with unknown level

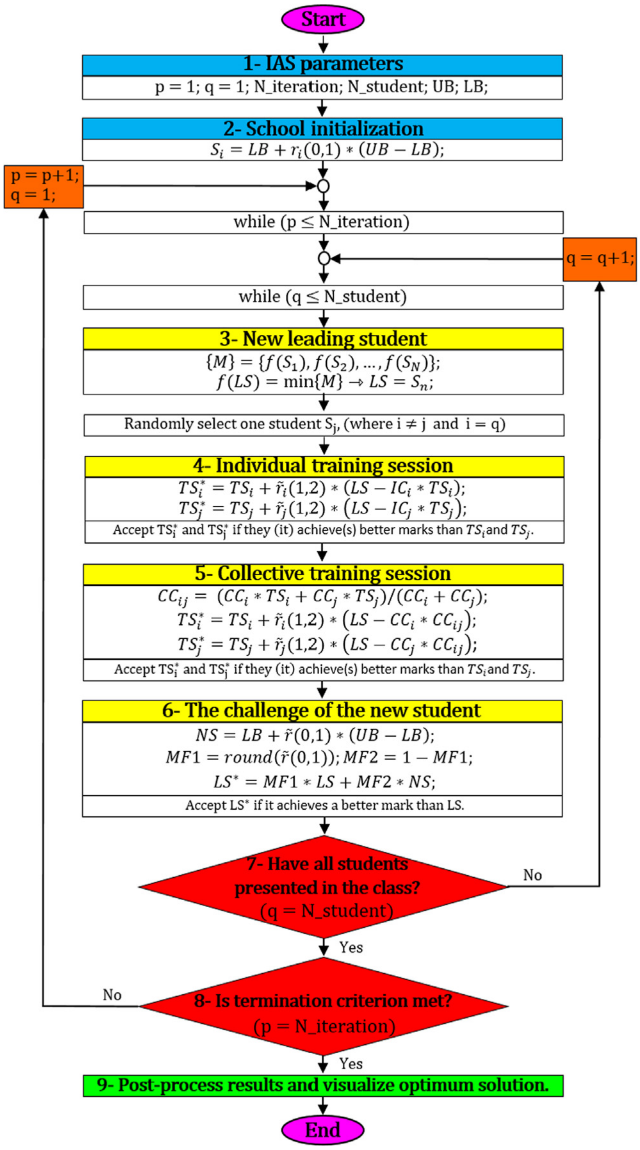

Fig. 1. Flowchart of the IAS algorithm.

of knowledge to the school who will then meet with the leading student. This meeting could be considered a challenge for the leader since in the meantime of evaluating the newcomer, he would need to preserve his leading position, or otherwise, to pass it to the freshman.

The entire process including all the three sessions is repeated until the termination criteria are met. It should be noted that by the termination of the procedure, each student has necessarily interacted at least once (without limitation for more though) with the leader. That is because in both individual and collective sessions, groups of two random students are chosen in the search space to undergo interactions with the leader and their own. One of the major advantages of the IAS compared with other population-based algorithms (GA, DE, PSO, BA, PBA, etc.) is that no additional tuning parameters are required. The proper choice of the tuning parameters including the number of students and the number of iterations can lead to a faster detection of the global optimum. The higher the number of students in the autodidactic school, the higher the probability of existence of elite students between them. Furthermore, the number of sessions held is equal to the number of students in the school. Hence, the population size in this interactive school has a drastic effect on the increase of the knowledge level of the students. Schematic illustration of IAS is shown in the flowchart of Fig. 1.

2.1. Individual training session

The individual training can be described as a learning directed interaction session between the leading student and another student. A random group of trailing students, consisting in particular of two members is first selected. Then, they discuss with the leading student, one at a time. In such a peer-to-peer discussion with the leader, the individual knowledge of the students grows. It should be noted, however, that the knowledge level of the trailing students after the individual learning session depends as well on their original competence, i.e., qualities such as their prior knowledge or intrinsic talent. Accordingly, the individual session could be formalized as follows:

$$For\ \ i = 1 : N_{student}$$

$$Randomly\ \ select\ \ one\ \ student\ \ S_j,\ \ where\ \ i \neq j$$

$$TS_i^* = TS_i + r_i(1,2) * (LS - IC_i * TS_i);$$

$$TS_j^* = TS_j + r_j(1,2) * \left(LS - IC_j * TS_j\right);$$

$$End\ \ For$$

$$Accept\ \ TS_i^*\ \ and\ \ TS_j^*\ \ if\ \ they\ \ (it)\ \ achive(s)\ \ the\ \ better\ \ marks$$
$$than\ \ TS_i\ \ and\ \ TS_j$$

where TS_i^* and TS_j^* are the first and the second trailing students, respectively, after the individual learning session and IC_i and IC_j are the intrinsic competence of the first and the second students, respectively,$r_i(1,2)$ and $r_j(1,2)$ are two different random vectors between 1 and 2. The individual competences (IC_i andIC_j) are randomly determined as either 1 or 2.

2.2. Collective training session

After the individual training session, each trailing student will have the opportunity to review the last session and also to interact with the other trailing student of the same group to realize the unclear points of the course. Afterwards, a collective session is held, wherein the leading student attempts to discuss with the group of students with the aim of troubleshooting. It is clear, therefore, that in addition to the knowledge status of the individually trained students, their social capabilities, such as communication skills, team working and cooperation, which we refer to as collective competences hereafter, can have an important impact on the learning efficiency of the group. Accordingly, the collective session could be formalized as follows:

$$For\ \ i = 1 : N\_student$$

$$CC_{ij} = (CC_i * TS_i + CC_j * TS_j)/(CC_i + CC_j);$$

$$TS_i^* = TS_i + r_i(1,2) * (LS - CC_i * CC_{ij});$$

$$TS_j^* = TS_j + r_j(1,2) * \left(LS - CC_j * CC_{ij}\right);$$

$$End\ \ For$$

$$Accept\ \ TS_i^*\ \ and\ \ TS_j^*\ \ if\ \ they\ \ (it)\ \ achive(s)\ \ the\ \ better\ \ marks$$
$$than\ \ TS_i\ \ and\ \ TS_j$$

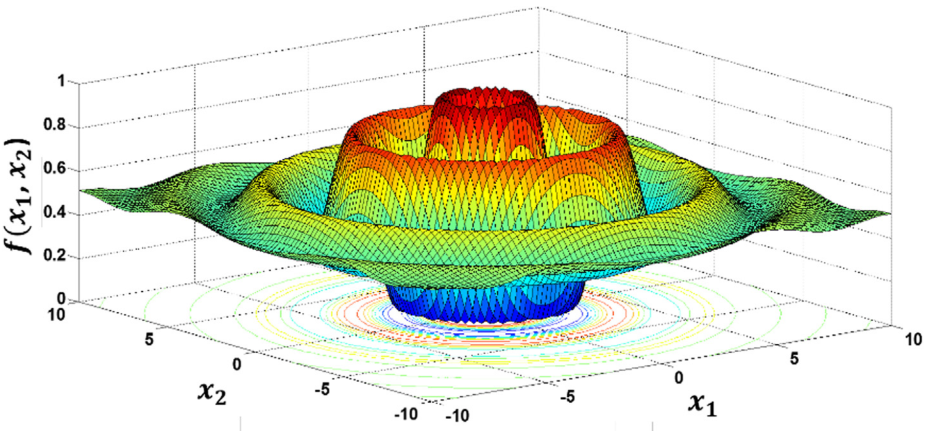

Fig. 2. Three dimensional and contour plots the Schaffer function.

where CC_{ij} is defined as the collective capability of the group as a team based on the weighted mean of the students competences. Also, $r_i(1,2)$ and $r_j(1,2)$ are two different random vectors between 1 and 2. The collective competences of individual students (CC_i andCC_j) are randomly determined as either 1 or 2.

2.3. The challenge of the new student

In some optimization problems, due to the complex nature of the cost function, the gradual improvement of the trailing students may be confined in a limited region of design space just around the leading student who is the temporary/local optimum of the time, though still far from the permanent/global optimum. Accordingly, a mal operating loop would form which impedes the smooth optimization process, and possibly, even fails in finding the global optimum. In order to provide IAS with a more dynamic and exploratory character, the "challenge of the new student" is designed to supplement the algorithm with continuous revolt against the current leader. If the new student is more competent than the leading student in this challenge, he will take over the lead. The challenge of the new student could be formalized as follows:

$$NS = LB + r * (UB - LB);$$

$$MF1 = round(r(0,1));$$

$$MF2 = 1 - MF1;$$

$$LS^* = MF1 * LS + MF2 * NS;$$

$$Achieves\ a\ better\ mark\ than\ LS$$

where NS is the new student, $MF1$ and $MF2$ are first and second modifier factors, respectively, $r(0,1)$ is a random vector between 0 and 1. In addition, LS^* represents the new leading student of the school.

Table 2
Detailed description of the twenty benchmark mathematical functions (D: Dimension).

No.	Function	Range	D	Formulation	Min
1	Beale	[−4.5, 4.5]	2	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	0
2	Easom	[−100, 100]	2	$f(x) = -cos(x_1)cos(x_2)exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	−1
3	Matyas	[−10, 10]	2	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	0
4	Bohachevsky	[−100, 100]	2	$f(x) = x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_2) + 0.7$	0
5	Booth	[−10, 10]	2	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0
6	Michalewicz	[0, π]	2	$f(x) = -\sum_{i=1}^{2} sin(x_i)(sin(ix_i^2/\pi))^{20}$	−1.8013
7	Schaffer	[−100, 100]	2	$f(x) = 0.5 + \frac{sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	0
8	Six Hump Camel Back	[−5, 5]	2	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	−1.0316
9	Shubert	[−10, 10]	2	$f(x) = (\sum_{i=1}^{5} icos(i+1)x_1 + i)(\sum_{i=1}^{5} icos(i+1)x_2 + i)$	−186.73
10	Zakharov	[−5, 10]	10	$f(x) = \sum_{i=1}^{10} x_i^2 + (\sum_{i=1}^{10} 0.5ix_i)^2 + (\sum_{i=1}^{10} 0.5ix_i)^4$	0
11	Step	[−5.12, 5.12]	30	$f(x) = \sum_{i=1}^{30} (x_i + 0.5)^2$	0
12	Sphere	[−100, 100]	30	$f(x) = \sum_{i=1}^{30} x_i^2$	0
13	SumSquares	[−10, 10]	30	$f(x) = \sum_{i=1}^{30} ix_i^2$	0
14	Schwefel2	[−10, 10]	30	$f(x) = \sum_{i=1}^{30} \lvert x_i \rvert + \prod_{i=1}^{30} \lvert x_i \rvert$	0
15	Rastrigin	[−5.12, 5.12]	30	$f(x) = \sum_{i=1}^{30} (x_i^2 - 10cos(2\pi x_i) + 10)$	0
16	Griewank	[−600, 600]	30	$f(x) = \frac{1}{4000}\left(\sum_{i=1}^{30} (x_i - 100)^2\right) - \left(\prod_{i=1}^{30} cos(\frac{x_i - 100}{\sqrt{i}})\right) + 1$	0
17	Ackley	[−32, 32]	30	$f(x) = -20exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2}\right) - exp\left(\frac{1}{30}\sum_{i=1}^{30} cos(2\pi x_i)\right) + 20 + e$	0
18	Dixon_Price	[−10, 10]	30	$f(x) = (x_1 - 1)^2 + \sum_{i=1}^{30} i(2x_i^2 - x_i - 1)^2$	0
19	Quartic	[−1.28, 1.28]	30	$f(x) = \sum_{i=1}^{30} ix_i^4 + Rand$	0
20	Rosenbrock	[−30, 30]	30	$f(x) = (x_i - 1)^2 + \sum_{i=1}^{29} 100(x_{i+1} - x_i^2)^2$	0

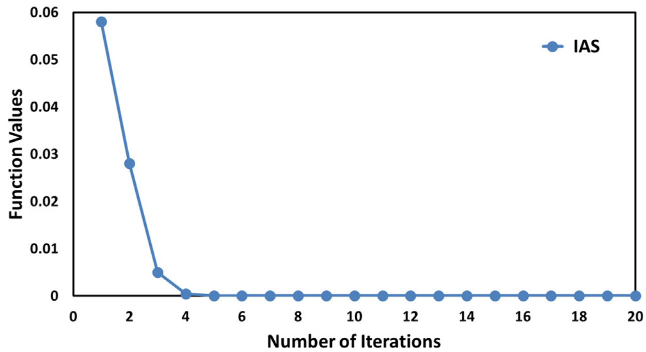

Fig. 3. Convergence history diagram for the Schaffer function optimization problem.

3. Implementation of IAS algorithm for an optimization problem

The step-wise procedure of the IAS algorithm for solving an optimization problem is described in this section. One mathematical function with two design variables, the Schaffer function (Eq. (1)) [55], was chosen to explain the step-by-step procedure of IAS to solve optimization problems.

$$f(x_1,x_2) = 0.5 + \frac{sin^2\left(\sqrt{x_1^2+x_2^2}\right) - 0.5}{\left(1+0.001(x_1^2+x_2^2)\right)^2} \tag{1}$$

Initialization range were set as [-10, 10] for the Schaffer function. The global minimum for this function is $f_{min} = 0 at \{x_1 = 0, x_2 = 0\}$. Fig. 2 illustrates the three dimensional and contour plots the Schaffer function.

Step 1: The optimization problem and IAS parameters.

- Minimize: $f(x_1,x_2) = 0.5 + \frac{sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{\left(1+0.001(x_1^2+x_2^2)\right)^2}$.
- Subjected to: $-10 \le x_1 \le 10 and -10 \le x_2 \le 10$.
- Number of students (N_student) = 10.
- Number of iterations (N_iteration) = 20.
- p = 1 and q = 1;

Step 2: School initialization.

Generate randomly the initial students of the school based on number of design variables and N_student as follows (Note that at least three students are required for constructing a school):

$$\text{School} = \begin{bmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \\ S_{3,1} & S_{3,2} \\ S_{4,1} & S_{4,2} \\ S_{5,1} & S_{5,2} \\ S_{6,1} & S_{6,2} \\ S_{7,1} & S_{7,2} \\ S_{8,1} & S_{8,2} \\ S_{9,1} & S_{9,2} \\ S_{10,1} & S_{10,2} \end{bmatrix} = \begin{bmatrix} -6.5461 & -1.0762 \\ -3.6833 & 5.6483 \\ 1.0420 & 4.1858 \\ -0.7812 & -2.7162 \\ -8.0933 & 4.7977 \\ -1.5625 & -1.1147 \\ 1.4688 & -7.2051 \\ 9.7183 & 1.4167 \\ -7.5508 & -1.9601 \\ -3.3718 & 9.7711 \end{bmatrix};$$

Step 3: New leading student.

Among the initial students, the position of the new leading student endowing to the best student with the optimum mark (Herein, the optimum mark corresponds to the lowest value of the cost function).

$$f(LS) = \min\left(function \begin{bmatrix} (S_{1,1},S_{1,2}) \\ (S_{2,1},S_{2,2}) \\ (S_{3,1},S_{3,2}) \\ (\boldsymbol{S}_{4,1},\boldsymbol{S}_{4,2}) \\ (S_{5,1},S_{5,2}) \\ (S_{6,1},S_{6,2}) \\ (S_{7,1},S_{7,2}) \\ (S_{8,1},S_{8,2}) \\ (S_{9,1},S_{9,2}) \\ (S_{10,1},S_{10,2}) \end{bmatrix} \right)$$

$$= \min\left\{ \begin{matrix} 0.3700 \\ 0.4012 \\ 0.7594 \\ 0.1204 \\ 0.4434 \\ 0.8782 \\ 0.5687 \\ 0.4659 \\ 0.6057 \\ 0.5101 \end{matrix} \right\} = 0.1204 \rightarrow LS = S_4 = (S_{4,1},S_{4,2});$$

The fourth student has the best mark among all students and is selected as the new leading student of the school.

Step 4: Individual training session.

In this session, the leading student tries to improve the individual knowledge of the students using the training process. With i set initially at 1, the first student is matched to the ith student. Moreover, the second student is selected randomly from the school members. In this case TS_i and TS_j are chosen as the first and second trailing students. The individual competences (IC_1 andIC_2) are randomly determined as either 1 or 2.

- $IC_1 = 1$ and $IC_2 = 1$.
- $TS_1^* = TS_1 + r_1 * (LS - IC_1 * TS_1); TS_1^* = (0.2864, -2.9934)$ and $f(TS_1^*) = 0.0544$.
- $TS_2^* = TS_2 + r_2 * (LS - IC_2 * TS_2); TS_2^* = (1.4368, -6.7259)$ and $f(TS_2^*) = 0.4424$.
- TS_1^* mark is better than TS_1. Thus, TS_1 is upgraded to $TS_1^* = (0.2864, -2.9934)$.
- TS_2^* mark is not better than TS_2. Thus, TS_2 is not upgraded.

Step 5: Collective training session.

In this session, the leading student tries to troubleshoot the problems of the students using collective training. The collective competences of the students (CC_1 andCC_2) are randomly determined as either 1 or 2. Also, CC_{12} is the collective capability based on the weighted mean of the student's competences.

- $CC_1 = 2$ and $CC_2 = 1$.
- $CC_{12} = \frac{CC_1*TS_1+CC_2*TS_2}{CC_1+CC_2} = (-1.0368, -0.1129)$.
- $TS_1^* = TS_1 + r_1 * (LS - CC_1 * CC_{12}); TS_1^* = (2.5404, -6.7927)$ and $f(TS_1^*) = 0.5477$.
- $TS_2^* = TS_2 + r_2 * (LS - CC_2 * CC_{12}); TS_2^* = (-3.3052, 1.0551)$ and $f(TS_2^*) = 0.1539$.
- TS_1^* mark is not better than TS_1. Thus, TS_1 is not upgraded.
- TS_2^* mark is better than TS_2. Thus, TS_2 is upgraded to $TS_2^* = (-3.3052, 1.0551)$.

Step 6: The challenge of the new student.

This session could improve the performance of the school and the knowledge of the leading student. The new student is generated in the allowable range between lower and upper bounds of design variables.

- $NS = LB + r * (UB - LB) = (-9.4861, 5.5318);$
- $MF1 = round(r) = (1, 1);$
- $MF2 = 1 - MF1 = (0, 0);$
- $LS^{*} = MF1 * LS + MF2 * NS = (-0.7812, -2.7162);$
- LS^{*} mark is same with LS. Thus, the position of the leading student is not changed.

Step 7: Presence and absence of the students.

If (q = N_student) => all students are present; otherwise go to step 3 (q = q + 1).

Step 8: Termination Criterion.

If (p = N_iteration) => the termination criterion is met; otherwise go to step 3 (p = p + 1).

Step 9: Post-process results and visualization.

Finding a global minimum of the Schaffer function is a complicated problem since it is wave-shaped and has many local minima in the search space. Fig. 3 shows the convergence history diagram of the IAS algorithm for the Schaffer function optimization problem. IAS found the global optimum value equal to zero after

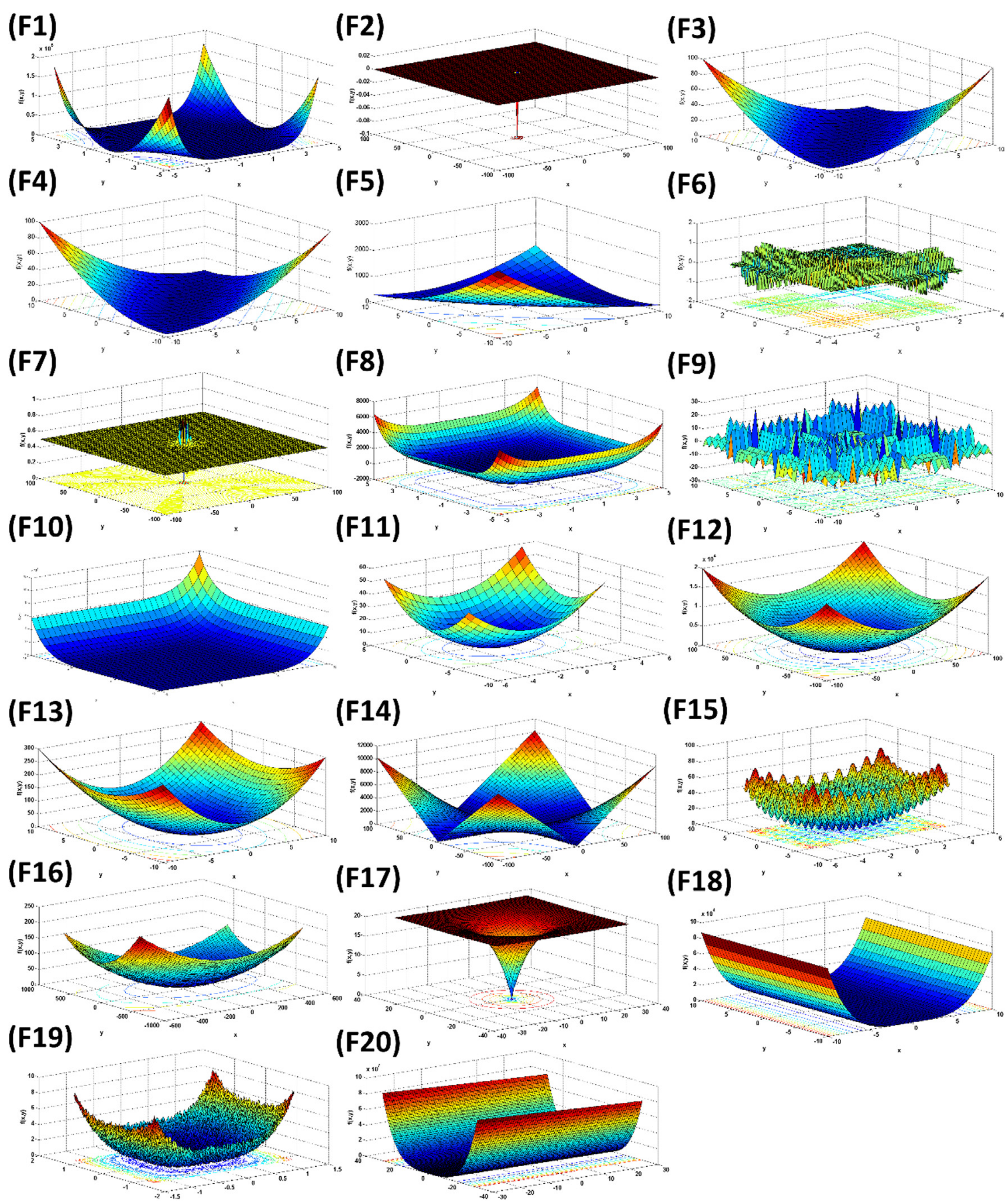

Fig. 4. 2D search space plots of F1-F20 benchmark mathematical functions.

Table 3
The tuning parameters of the IAS algorithm.

No.	Function	*N_student*	*N_iteration*
1	F 1–10	10	100
2	F 11–18	30	500
3	F 19–20	50	1000

5 iterations. It is observed that the Schaffer function values are converged to minimum value at the early iterations.

4. Investigation of IAS performance

In this research, twenty mathematical and seven structural optimization problems are solved to assess the performance of IAS. The proposed algorithm is implemented using MATLAB in a computer with an Intel Core i7 CPU (1.73 GHz) and 4 GB of RAM. In order to properly evaluate the robustness of the meta-heuristic optimization algorithm, 30 independent runs are performed for each numerical example. The average values of each objective function of the IAS algorithm are reported. In order to validate the proposed algorithm, optimization results have been compared with other well-known optimizers. In this paper, three significant criteria including the best optimum solution, total computational cost, and reliability index have been considered for the efficiency appraisal of evolutionary algorithms. The best optimal

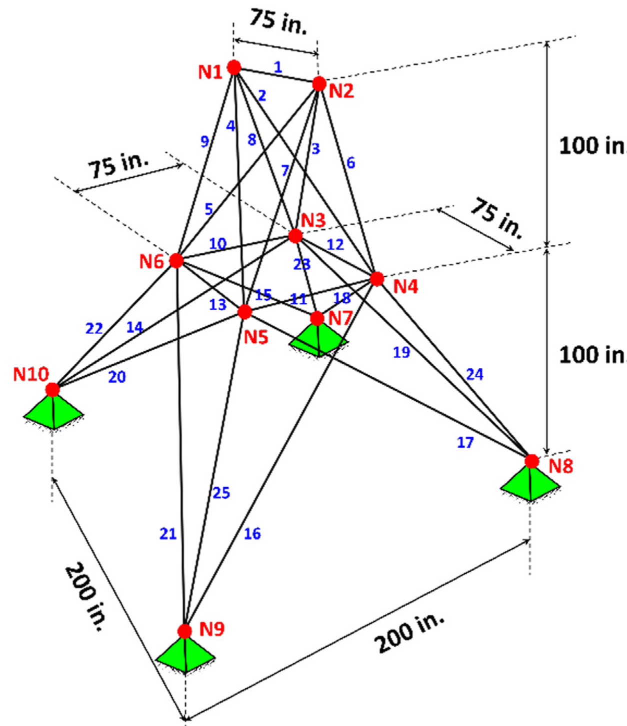

Fig. 5. Schematic of the spatial 25-bar truss structure.

Table 4
Comparative results of IAS with GA, DE, PSO, BA, PBA, and SOS (gray is average and white is *StdDev*).

No.	Functions	Min	GA [60]	DE [60]	PSO [60]	BA [60]	PBA [60]	SOS [33]	IAS
1	Beale	**0**	0	0	0	1.88e−05	0	0	0
			0	0	0	1.94e−05	0	0	0
2	Easom	**−1**	−1	−1	−1	−0.99994	−1	−1	−1
			0	0	0	4.50e−05	0	0	0
3	Matyas	**0**	0	0	0	0	0	0	0
			0	0	0	0	0	0	0
4	Bohachevsky	**0**	0	0	0	0	0	0	0
			0	0	0	0	0	0	0
5	Booth	**0**	0	0	0	0.00053	0	0.03382	0
			0	0	0	0.00074	0	0.12870	0
6	Michalewicz	**−1.8013**	−1.8013	−1.8013	−1.57287	−1.8013	−1.8013	−1.8013	−1.8013
			0	0	0.11986	0	0	0	0
7	Schaffer	**0**	0.00424	0	0	0	0	0	0
			0.00476	0	0	0	0	0	0
8	Six Hump Camel Back	**−1.0316**	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
			0	0	0	0	0	0	0
9	Shubert	**−186.73**	−186.73	−186.73	−186.73	−186.73	−186.73	−186.73	−186.73
			0	0	0	0	0	0	0
10	Zakharov	**0**	0.01336	0	0	0	0	0	0
			0.00453	0	0	0	0	0	0
11	Step	**0**	1.17e + 03	0	0	5.12370	0	0	0
			76.56145	0	0	0.39209	0	0	0
12	Sphere	**0**	1.11e + 03	0	0	0	0	0	0
			74.21447	0	0	0	0	0	0
13	SumSquares	**0**	1.48e + 02	0	0	0	0	0	0
			12.40929	0	0	0	0	0	0
14	Schwefel2	**0**	11.0214	0	0	0	7.59e−10	0	0
			1.38686	0	0	0	7.10e−10	0	0
15	Rastrigin	**0**	52.92259	11.7167	43.97714	0	0	0	0
			4.56486	2.53817	11.72868	0	0	0	0
16	Griewank	**0**	10.63346	0.00148	0.01739	0	0.00468	0	0
			1.16146	0.00296	0.02081	0	0.00672	0	0
17	Ackley	**0**	14.67178	0	0.16462	0	3.12e−08	0	0
			0.17814	0	0.49387	0	3.98e−08	0	0
18	Dixon_Price	**0**	1.22e + 03	0.66667	0.66667	0.66667	5.65e−10	0	0
			2.66e + 02	e−09	e−08	1.16e−09	0	0	0
19	Quartic	**0**	0.18070	0.00136	0.00116	1.72e−06	0.00678	9.13e−05	4.55e−05
			0.02712	0.00042	0.00028	1.85e−06	0.00133	3.71e−05	3.25e−05
20	Rosenbrock	**0**	1.96e + 05	18.2039	15.088617	28.834	4.2831	1.04e−07	0
			3.85e + 04	5.03619	24.170196	0.10597	5.7877	2.95e−07	0

Table 5
Loading conditions for the spatial 25-bar truss structure.

Node	Case 1			Case 2		
	F_x (kips)	F_y (kips)	F_z (kips)	F_x (kips)	F_y (kips)	F_z (kips)
1	0.0	20.0	−5.0	1.0	10.0	−5.0
2	0.0	−20.0	−5.0	0.0	10.0	−5.0
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0

Table 6
Discrete values of cross-sectional areas used in the 25-bar truss problem.

Case 1	D_1 = {0.01, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0, 4.4, 4.8, 5.2, 5.6, 6.0} (in^2)
Case 2	D_2 = {0.111, 0.141, 0.196, 0.250, 0.307, 0.391, 0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1.000, 1.130, … 1.228, 1.266, 1.457, 1.563, 1.620, 1.800, 1990, 2.130, 2.380, 2.620, 2.630, 2.880, 2.930, 3.090, … 3.380, 3.470, 3.550, 3.630, 3.840, 3.870, 3.880, 4.180, 4.220, 4.490, 4.590, 4.800, 4.970, 5.120, … 5.740, 7.220, 7.970, 8.530, 9.300, 10.850, 11.500, 13.500, 13.900, 14.200, 15.500, 16.000, … 16.900, 18.800, 19.900, 22.000, 22.900, 24.500, 26.500, 28.000, 30.000, 33.500} (in^2)

solution implies the best value of the objective function (f_{min}), the total computational cost signifies the number of function evaluations (*NFEs*), and reliability index denotes the standard deviation (*StdDev*) on optimized cost. In each iteration of the IAS algorithm, the cost function is evaluated for five times. Therefore, the overall computational cost of IAS equals five times the product of the two tuning parameters ($NFEs = 5 \times N\_student \times N\_iteration$).

4.1. Benchmark mathematical functions

Twenty benchmark mathematical optimization functions [58] are examined for the efficiency appraisal of IAS algorithm compared with other evolutionary algorithms including GA, DE, PSO, BA, PBA, and SOS. Test problem are described in detail in Table 2.

In order to compare the proposed algorithm with the above-mentioned evolutionary algorithms, the convergence criterion of less than 10^{-12} has been considered as zero. 2D search space of benchmark mathematical functions are shown in Fig. 4. The tuning parameters of the IAS algorithm are presented in Table 3. Table 4 compares optimization results of IAS, GA, DE, PSO, BA, PBA and SOS. The gray and white rows in Table 4 represent the average values and the standard deviations of the objective functions after 30 independent runs, respectively. Also, the bold numbers denote the absolute minimum values of the objective functions.

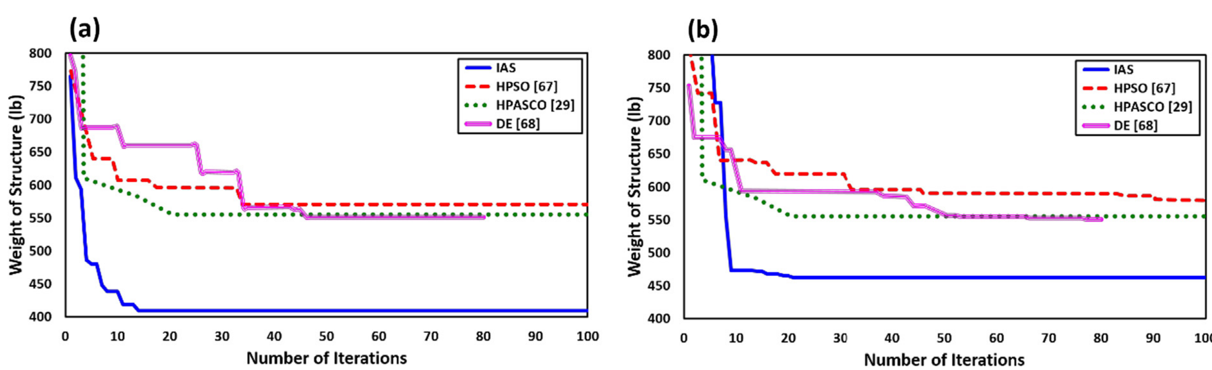

Fig. 6. Comparison of convergence curves of IAS for the 25-bar truss problem: (a) Case 1, (b) Case 2.

Table 7
Comparison of optimization results obtained for the 25-bar truss problem.

Case	Element group	Wu et al. [65]	Lee et al. [66]	Li et al. [67]	Kaveh et al. [29,68]		Degertekin et al. [70]	Present Work
		GA	HS	HPSO	HPASCO	DE	DAJA	IAS
1	A_1	0.40	0.01	0.01	0.01	0.01	0.01	0.01
	A_2–A_5	2.00	2.00	2.00	1.60	1.60	2.00	2.00
	A_6–A_9	3.60	3.60	3.60	3.20	3.20	3.60	3.60
	A_{10}–A_{11}	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	A_{12}–A_{13}	0.01	0.01	0.01	0.01	0.01	0.01	0.40
	A_{14}–A_{17}	0.80	0.80	0.80	0.80	0.80	0.80	0.40
	A_{18}–A_{21}	2.00	1.60	1.60	2.00	2.00	1.60	1.60
	A_{22}–A_{25}	2.40	2.40	2.40	2.40	2.40	2.40	0.01
	Weight (lb)	563.52	560.59	560.59	551.6	551.6	560.59	409.86
	NFEs	40,000	30,000	50,000	–	–	530	3500

Table 8
Comparison of optimization results obtained for the 25-bar truss problem.

Case	Element group	Wu et al. [65]	Li et al. [67]	Kaveh et al. [29,68]		Degertekin et al. [70]	Present Work
		GA	HPSO	HPASCO	DE	DAJA	IAS
2	A_1	0.31	0.11	0.11	0.11	0.11	0.11
	A_2–A_5	1.99	2.13	2.13	2.13	2.13	0.11
	A_6–A_9	3.13	2.88	2.88	2.88	2.88	3.55
	A_{10}–A_{11}	0.11	0.11	0.11	0.11	0.11	0.11
	A_{12}–A_{13}	0.14	0.11	0.11	0.11	0.11	2.13
	A_{14}–A_{17}	0.77	0.77	0.77	0.77	0.76	0.76
	A_{18}–A_{21}	1.62	1.62	1.62	1.62	1.62	0.11
	A_{22}–A_{25}	2.62	2.62	2.62	2.62	2.62	3.87
	Weight (lb)	556.43	551.14	551.1	551.1	551.14	462.04
	NFEs	40,000	30,000	–	–	946	5250

It can be seen that the IAS algorithm found the absolute minimum values of almost all of the mathematical functions in Table 4. This makes the IAS algorithm a superior tool compared to previously proposed algorithms whose results are listed in Table 3. It is worth noting that the proposed algorithm was the only optimizer that successfully solved problem 20 (Rosenbrock function) as a high complexity optimization problem. Based on the "no free lunch theorem", the IAS algorithm is the most reliable algorithm with the standard deviation of zero in almost all cases [59].

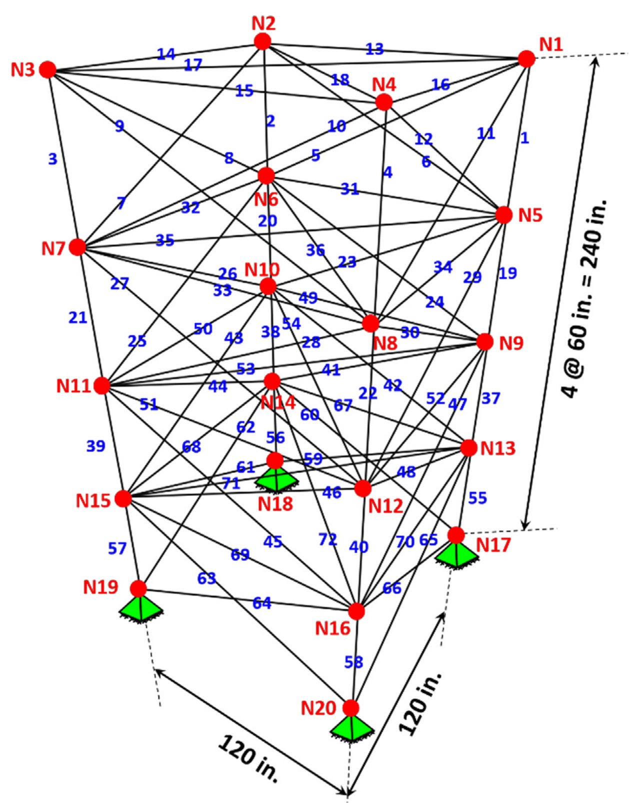

Fig. 7. Schematic of the spatial 72-bar truss structure.

4.2. Structural optimization problems

Seven structural engineering design optimization problems are solved to evaluate efficiency of the IAS algorithm. In the first three test problems, the objective is to minimize the weight of a spatial 25-bar, a spatial 72-bar truss and a planar 200-bar truss structures. The 4th test problem is about a stepped cantilever beam for minimizing the weight of the beam. The 5th one regards a reinforced concrete beam design problem to minimize the values of the bars and the cross-sectional area. The 6th one is a welded beam design problem to find the best value of the geometrical dimensions of weld and structural members. The 7th problem aims to minimize total cost including the cost of materials, forming, and welding of a cylindrical pressure vessel. IAS is a parameter-free algorithm, therefore, the constraint handling method is applied in order to solve these engineering design optimization problems [60–63]. To ensure fairness of the proposed optimization algorithm, 30 independent runs are performed for each engineering problem. The average values of objective functions corresponding to the IAS algorithm are plotted in the convergence history diagrams as well as the best results.

Table 10
Discrete values of cross-sectional areas used in the 72-bar truss problem.

Case 1	D_1 = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, … 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2} (in.^{2})
Case 2	D_2 = {0.111, 0.141, 0.196, 0.250, 0.307, 0.391, 0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1.000, 1.130, … 1.228, 1.266, 1.457, 1.563, 1.620, 1.800, 1990, 2.130, 2.380, 2.620, 2.630, 2.880, 2.930, 3.090, … 3.380, 3.470, 3.550, 3.630, 3.840, 3.870, 3.880, 4.180, 4.220, 4.490, 4.590, 4.800, 4.970, 5.120, … 5.740, 7.220, 7.970, 8.530, 9.300, 10.850, 11.500, 13.500, 13.900, 14.200, 15.500, 16.000, … 16.900,18.800, 19.900, 22.000, 22.900, 24.500, 26.500, 28.000, 30.000, 33.500} (in.^{2})

Table 9
Loading conditions for the spatial 72-bar truss structure.

Node	Case 1			Case 2		
	F_x (kips)	F_y (kips)	F_z (kips)	F_x (kips)	F_y (kips)	F_z (kips)
1	5.0	5.0	−5.0	0.0	0.0	−5.0
2	0.0	0.0	0.0	0.0	0.0	−5.0
3	0.0	0.0	0.0	0.0	0.0	−5.0
4	0.0	0.0	0.0	0.0	0.0	−5.0

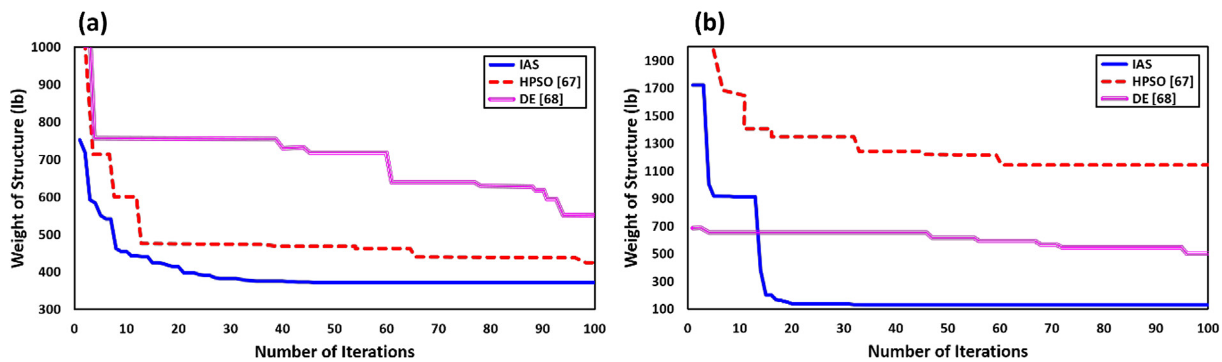

Fig. 8. Comparison of convergence curves of IAS for the 72-bar truss problem: (a) Case 1, (b) Case 2.

Table 11
Comparison of optimization results obtained for the 72-bar truss problem.

Case	Element group	Wu et al. [65]	Lee et al. [66]	Li et al. [67]	Kaveh et al. [29,68]		Degertekin et al. [70]	Present Work
		GA	HS	HPSO	HPASCO	DE	DAJA	IAS
1	A_1–A_4	0.2	0.2	0.2	0.2	0.2	1.9	0.1
	A_5–A_{12}	0.5	0.5	0.5	0.6	0.6	0.5	0.5
	A_{13}–A_{16}	0.5	0.4	0.3	0.4	0.4	0.1	0.3
	A_{17}–A_{18}	0.7	0.6	0.7	0.6	0.6	0.1	0.6
	A_{19}–A_{22}	0.5	0.6	0.5	0.6	0.5	1.4	0.5
	A_{23}–A_{30}	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	A_{31}–A_{34}	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	A_{35}–A_{36}	0.2	0.1	0.1	0.1	0.1	0.1	0.1
	A_{37}–A_{40}	1.3	1.4	1.4	1.3	1.3	0.5	1.5
	A_{41}–A_{48}	0.5	0.6	0.5	0.5	0.5	0.5	0.5
	A_{49}–A_{52}	0.2	0.1	0.1	0.1	0.1	0.1	0.1
	A_{53}–A_{54}	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	A_{55}–A_{58}	1.5	1.9	2.1	1.9	2.0	0.2	2.0
	A_{59}–A_{66}	0.7	0.5	0.6	0.5	0.5	0.6	0.5
	A_{67}–A_{70}	0.1	0.1	0.1	0.1	0.1	0.4	0.1
	A_{71}–A_{72}	0.1	0.1	0.1	0.1	0.1	0.6	0.1
	Weight (lb)	400.66	387.94	388.94	385.54	385.54	385.54	372.41
	NFEs	40,000	30,000	50,000	–	–	1873	10,750

Table 12
Comparison of optimization results obtained for the 72-bar truss problem.

Case	Element group	Wu et al. [65]	Li et al. [67]	Kaveh et al. [29,68]		Degertekin et al. [70]	Present Work
		GA	HPSO	HPASCO	DE	DAJA	IAS
2	A_1–A_4	1.563	0.391	0.196	0.196	1.990	0.442
	A_5–A_{12}	0.766	1.457	0.563	0.563	0.563	0.111
	A_{13}–A_{16}	0.141	0.766	0.442	0.307	0.111	0.111
	A_{17}–A_{18}	0.111	1.563	0.563	0.563	0.111	0.111
	A_{19}–A_{22}	1.800	1.563	0.563	0.442	1.228	0.442
	A_{23}–A_{30}	0.602	1.228	0.563	0.563	0.422	0.111
	A_{31}–A_{34}	0.141	0.111	0.111	0.111	0.111	0.111
	A_{35}–A_{36}	0.307	0.196	0.250	0.111	0.111	0.111
	A_{37}–A_{40}	0.391	2.88	1.228	1.457	0.563	0.442
	A_{41}–A_{48}	0.391	1.457	0.563	0.563	0.563	0.111
	A_{49}–A_{52}	0.141	0.141	0.111	0.111	0.111	0.111
	A_{53}–A_{54}	0.111	0.111	0.111	0.111	0.111	0.111
	A_{55}–A_{58}	0.196	4.97	1.800	2.130	0.196	0.602
	A_{59}–A_{66}	0.602	1.228	0.422	0.442	0.563	0.111
	A_{67}–A_{70}	0.307	0.111	0.141	0.111	0.391	0.111
	A_{71}–A_{72}	0.766	0.111	0.111	0.111	0.563	0.111
	Weight (lb)	427.203	933.09	393.380	391.329	389.334	130.309
	NFEs	40,000	3000	–	–	3376	8500

4.2.1. Spatial 25-bar truss problem

Discrete sizing optimization of truss structures is a challenging problem in structural design. Hence, a 25-bar spatial truss design problem is selected in order to assess the performance of the proposed algorithm [64]. The 25-bar spatial truss structure which shown in Fig. 5 has been studied by many scholars, such as, for

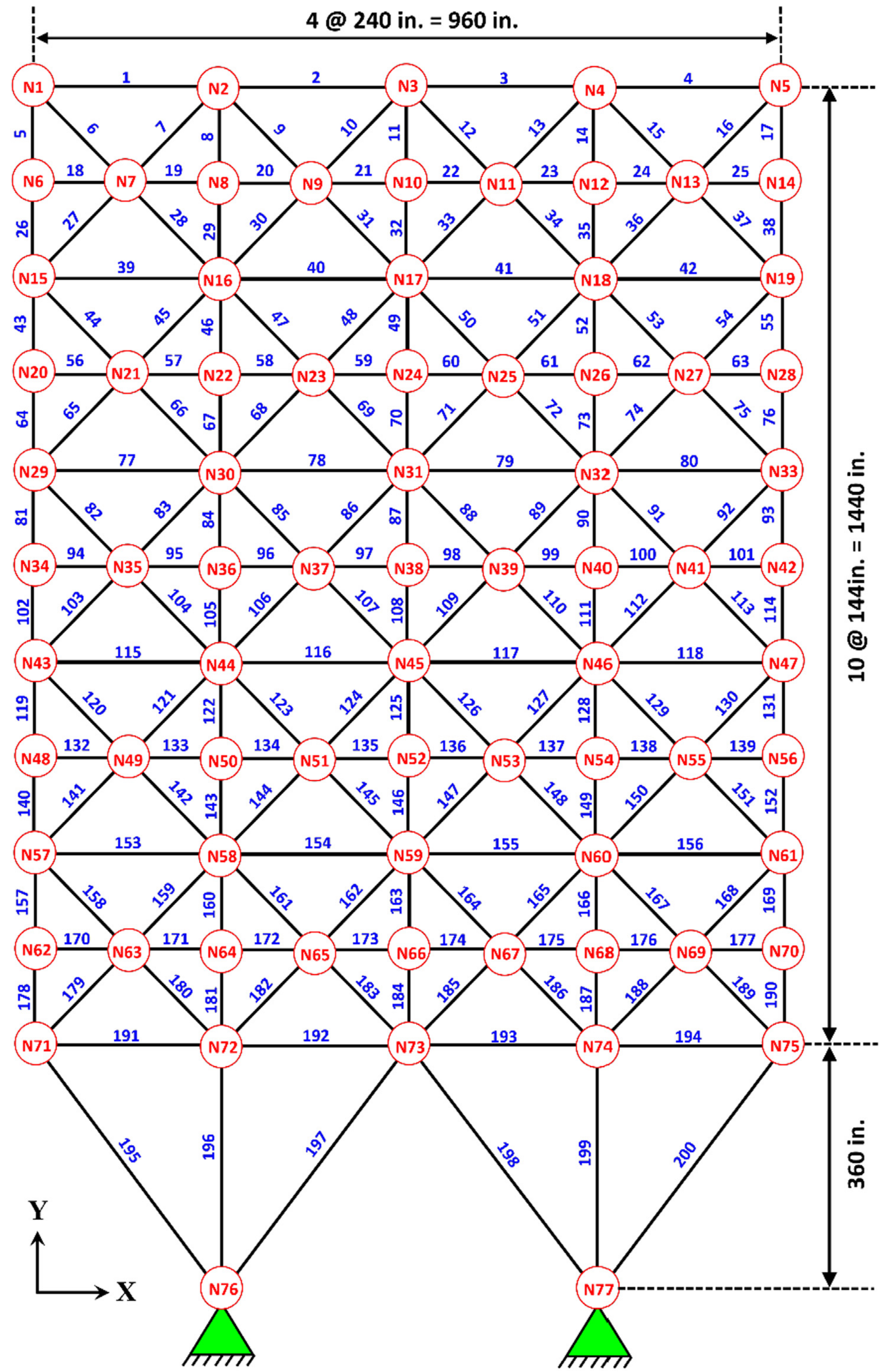

Fig. 9. Schematic of the planar 200-bar truss structure.

example, Wu and Chow [65], Lee et al. [66], Li et al. [67], Kaveh and Talatahari [68] and Degertekin et al. [69,70]. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10 Msi. The stress limitations of the members are ±40000 psi. All nodes in three coordinate directions are subjected to displacement limitations of ±0.35 in.

The structure includes 25 members, which are divided into eight groups, as follows: (1) A_1, (2) A_2-A_5, (3) A_6-A_9, (4) A_{10}-A_{11}, (5) A_{12}-A_{13}, (6) A_{14}-A_{17}, (7) A_{18}-A_{21}, and (8) A_{22}-A_{25}. Two different optimization cases are implemented. The loads and available discrete values of cross-sectional areas for both Cases are listed in Tables 5 and 6, respectively.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 6 compares the convergence curves of the IAS algorithm with HPSO [67], HPASCO [29] and DE [68] for both load Cases considered in the optimization. According to Tables 7, 8, and Fig. 6, IAS finds the optimum in approximately 20 iterations in Case 1 and 30 iterations in Case

Table 13
Design variables for the planar 200-bar truss structure.

Design variables	Element number	Design variables	Element number
1	1,2,3,4	16	82,83,85,86,88,89,91,92,103,104,106,107,109,110,112,113
2	5,8,11,14,17	17	115,116,117,118
3	19,20,21,22,23,24	18	119,122,125,128,131
4	18,25,56,63,94,101,132,139,170,177	19	133,134,135,136,137,138
5	26,29,32,35,38	20	140,143,146,149,152
6	6,7,9,10,12,13,15,16,27,28,30,31,33,34,36,37	21	120,121,123,124,126,127,129,130,141,142,144,145,147,148,150,151
7	39,40,41,42	22	153,154,155,156
8	43,46,49,52,55	23	157,160,163,166,169
9	57,58,59,60,61,62	24	171,172,173,174,175,176
10	64,67,70,73,76	25	178,181,184,187,190
11	44,45,47,48,50,51,53,54,65,66,68,69,71,72,74,75	26	158,159,161,162,164,165,167,168,179,180,182,183,185,186,188,189
12	77,78,79,80	27	191,192,193,194
13	81,84,87,90,93	28	195,197,198,200
14	95,96,97,98,99,100	29	196,199
15	102,105,108,111,114		

Table 14
Comparison of optimized designs for the 200-bar truss problem.

Design variable (in^2)	CMLPSA [71]	ABC-AP [72]	SAHS [73]	TLBO [74]	HPSSO [75]	FPA [76]	WEO [77]	JA [69]	IAS (This study)
1	0.1468	0.1039	0.154	0.146	0.1213	0.1425	0.1144	0.147258	0.1469138
2	0.94	0.9463	0.941	0.941	0.9426	0.9637	0.9443	0.940434	0.9401964
3	0.1	0.1037	0.1	0.1	0.122	0.1005	0.131	0.100109	0.1001083
4	0.1	0.1126	0.1	0.101	0.1	0.1	0.1016	0.100098	0.1000200
5	1.94	1.952	1.942	1.941	2.0143	1.9514	2.0353	1.941704	1.9414864
6	0.2962	0.293	0.301	0.296	0.28	0.2957	0.3126	0.296783	0.2964016
7	0.1	0.1064	0.1	0.1	0.1589	0.1156	0.1679	0.100096	0.1000000
8	3.1042	3.1249	3.108	3.121	3.0666	3.1133	3.1541	3.106749	3.1049954
9	0.1	0.1077	0.1	0.1	0.1002	0.1006	0.1003	0.100095	0.1000072
10	4.1042	4.1286	4.106	4.173	4.0418	4.11	4.1005	4.108109	4.1048547
11	0.4034	0.425	0.409	0.401	0.4142	0.4165	0.435	0.403975	0.4037530
12	0.1912	0.1046	0.191	0.181	0.4852	0.1843	0.1148	0.193079	0.1912158
13	5.4284	5.4803	5.428	5.423	5.4196	5.4567	5.3823	5.434236	5.4293638
14	0.1	0.106	0.1	0.1	0.1	0.1	0.1607	0.100095	0.1002672
15	6.4284	6.4853	6.427	6.422	6.3749	6.4559	6.4152	6.434203	6.4293582
16	0.5734	0.56	0.581	0.571	0.6813	0.58	0.5629	0.575306	0.5742185
17	0.1327	0.1825	0.151	0.156	0.1576	0.1547	0.401	0.135485	0.1358575
18	7.9717	8.0445	7.973	7.958	8.1447	8.0132	7.9735	7.9802	7.9747420
19	0.1	0.1026	0.1	0.1	0.1	0.1	0.1092	0.100157	0.1000102
20	8.9717	9.0334	8.974	8.958	9.092	9.0135	9.0155	8.980345	8.9746820
21	0.7049	0.7844	0.719	0.72	0.7462	0.7391	0.8628	0.709002	0.7083200
22	0.4196	0.7506	0.422	0.478	0.2114	0.787	0.222	0.437247	0.4355879
23	10.8636	11.3057	10.892	10.897	10.9587	11.1795	11.0254	10.89123	10.883503
24	0.1	0.2208	0.1	0.1	0.1	0.1462	0.1397	0.10015	0.1000000
25	11.8606	12.273	11.887	11.897	11.9832	12.1799	12.034	11.89141	11.883408
26	1.0339	1.4055	1.04	1.08	0.9241	1.3424	1.0043	1.049144	1.0473339
27	6.6818	5.16	6.646	6.462	6.7676	5.4844	6.5762	6.610648	6.6171873
28	10.8113	9.993	10.804	10.799	10.9639	10.1372	10.7265	10.77913	10.775835
29	13.8404	14.70144	13.87	13.922	13.8186	14.5262	13.9666	13.8783	13.875904
Weight (lb)	25445.63	25533.79	25491.9	25488.15	25698.85	25521.81	25674.83	25463.53	25450.02
NFEs	9650	1,450,000	19,670	28,059	14,406	10,685	19,410	31,580	43,500

2, while HPSACO requires about 100 iterations and DE about 50 iterations [29,67].

The obtained results by the IAS algorithm and other algorithms including GA, HS, HPSO, HPSACO, DE, and DAJA are compared in Table 7 for Case 1 and Table 8 for Case 2. It can be seen that IAS is very competitive with the other optimizers considered in this study [59].

4.2.2. Spatial 72-bar truss problem

The second truss-design problem is the discrete sizing optimization of the spatial 72-bar truss shown in Fig. 7 [64]. For this structure, the material density is 0.1 lb/in^3 and the modulus of elasticity is 10 Msi. The stress limitations of the members are ±25ksi. All nodes in three coordinate directions are subjected to displacement limitations of ±0.25 in.

The structure includes 72 members, which are divided into sixteen groups, as follows: (1) A_1-A_4, (2) A_5-A_{12}, (3) A_{13}-A_{16}, (4) A_{17}-A_{18}, (5) A_{19}-A_{22}, (6) A_{23}-A_{30}, (7) A_{31}-A_{34}, (8) A_{35}-A_{36}, (9) A_{37}-A_{40}, (10) A_{41}-A_{48}, (11) A_{49}-A_{52}, (12) A_{53}-A_{54}, (13) A_{55}-A_{58}, (14) A_{59}-A_{66}, (15) A_{67}-A_{70}, and (16) A_{71}-A_{72}. Two different optimization cases are implemented. The loads and available discrete values of cross-sectional areas for both Cases are listed in Table 9 and Table 10, respectively.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 8 compares the convergence curves of the IAS algorithm with HPSO [67] and DE [68] for both load Cases considered in the optimization. According to Table 11, Table 12, and Fig. 8, IAS finds the optimum in approximately 50 iterations in Case 1 and 40 iterations in Case 2, while DE requires about 200 iterations [29].

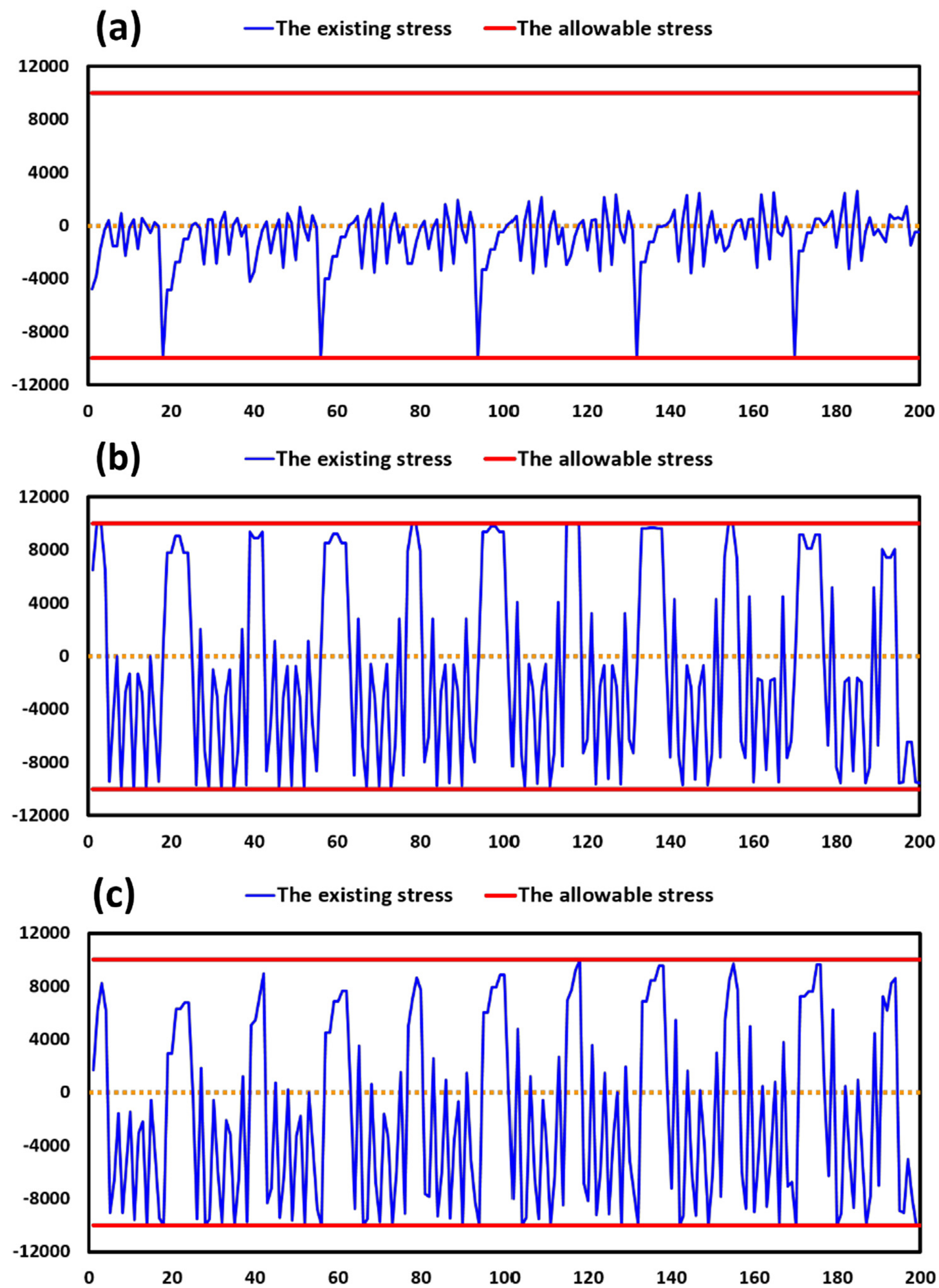

Fig. 10. Existing and allowable element stress value for the planar 200-bar truss: (a) Case 1; (b) Case 2 and (c) Case 3.

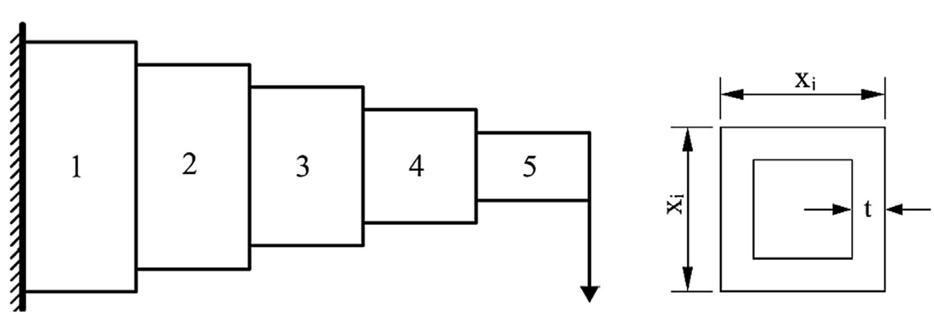

Fig. 11. Stepped cantilever beam design problem.

Table 15
Formulation of the stepped cantilever beam design problem.

Function	Mathematical formulation	Information
Objective function	$f(x_1, x_2, x_3, x_4, x_5) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5)$	*t = 2/3 in;*
Inequality constraints	$g_1(x_1, x_2, x_3, x_4, x_5) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$	*x_i = width or height cross section;*
Side constraints	$0.01 \leq x_i \leq 100; i = 1, 2, \cdots, 5$	$i = 1, 2, \cdots, 5$

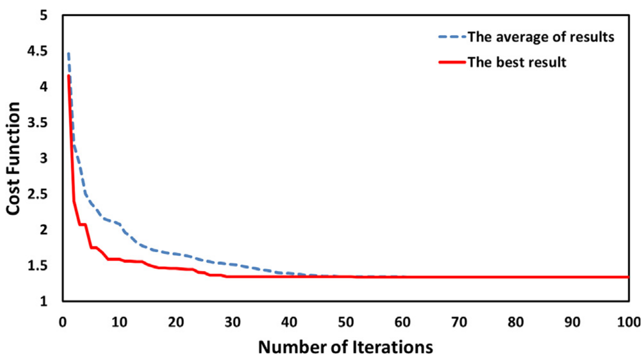

Fig. 12. Convergence curves for the stepped cantilever beam design problem.

The obtained results by the IAS algorithm and other algorithms including GA, HS, HPSO, HPSACO, DE, and DAJA are compared in Table 11 for Case 1 and Table 12 for Case 2. The present algorithm is very competitive with the other optimizers also in this test problem [59].

4.2.3. Planar 200-bar truss problem

The third example considered for size optimization is the planar 200-bar truss structure, schematized in Fig. 9. All members are made of steel: the material density is 0.283 lb/in3 and the modulus of elasticity is 30 Msi. This truss is subjected to constraints only on stress limitations of ± 10 ksi. There are three loading conditions: (1) 1.0 kip acting in the positive x-direction at nodes 1, 6, 15, 20, 29, 43, 48, 57, 62, and 71; (2) 10 kips acting in the negative y-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15,16, 17, 18, 19,

Table 16
Comparison of optimization results obtained for the stepped cantilever beam design problem.

Parameter	CONLIN [79]	MMA [79]	GCA(I) [79]	GCA(II) [79]	CS [80]	SOS [33]	IAS
x_1 (in)	6.0100	6.0100	6.0100	6.0100	6.0089	6.01878	5.99140
x_2 (in)	5.3000	5.3000	5.3000	5.3000	5.3049	5.30344	5.30850
x_3 (in)	4.4900	4.4900	4.4900	4.4900	4.5023	4.49587	4.51190
x_4 (in)	3.4900	3.4900	3.4900	3.4900	3.5077	3.49896	3.50210
x_5 (in)	2.1500	2.1500	2.1500	2.1500	2.1504	2.15564	2.16010
f_{min} (in)	N.C.	1.3400	1.3400	1.3400	1.33999	1.33996	1.34000
StdDev	N.A.	N.A.	N.A.	N.A.	N.A.	1.1e−5	5.27e−14
NFEs	N.A.	N.A.	N.A.	N.A.	N.A.	15,000	7250

Note: N.C. means not converged and N.A. means not available.

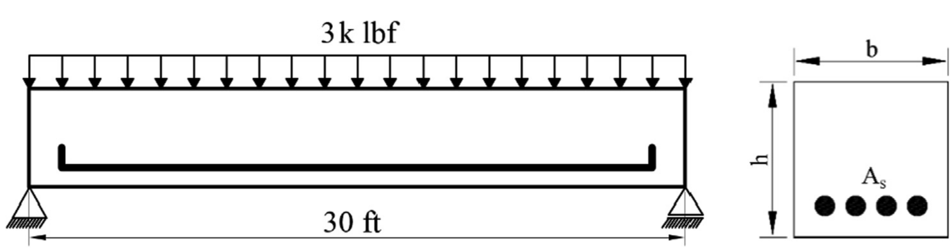

Fig. 13. Reinforced concrete beam design problem.

Table 17
Formulation of the reinforced concrete beam design problem.

Function	Mathematical formulation	Information
Objective function	$f(A_s, b, h) = 2.9A_s + 0.6bh$	*σ_y = 50 ksi;*
Inequality constraints	$g_1(A_s, b, h) = 7.375\frac{A_s^2}{b} - A_s h + 180 \leq 0 g_2(b, h) = \frac{h}{b} - 4 \leq 0$	*σ_c = 5 ksi;*
Side constraints and discrete values of variables	$5 \leq b \leq 10 A_s = [6, 6.16, 6.32, 6.6, 7.0, 7.11, 7.2, 7.8, 7.9, 8.0, 8.4]$;	*M_d = 1350 in kip;*
	$h = [28, 29, 30, \cdots, 40]$;	*M_l = 2700 in kip;*
		A_S = cross-sectional area of reinforcing bars;
		b = cross section width;
		h = cross section height;

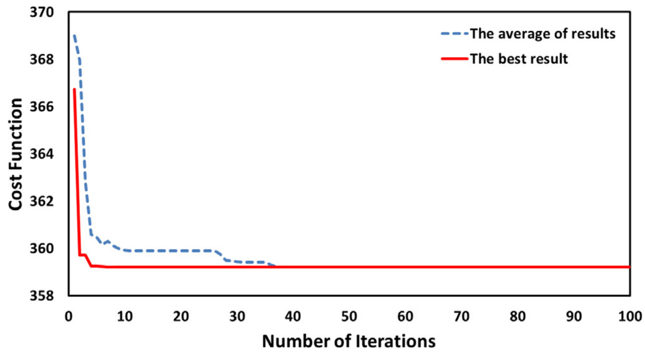

Fig. 14. Convergence curves for the reinforced concrete beam design problem.

20, 22, 24,. . ., 71, 72, 73, 74, and 75; and (3) Conditions (1) and (2) acting together. The 200 members of this truss are divided into 29 groups, as shown in Table 13. The minimum cross-sectional area of all members is 0.1 in^2.

To solve this problem using IAS, the population size and iterations number are set to 100 and 200, respectively. Table 14 compares the optimal results of the CMLPSA [71], ABC-AP [72], SAHS [73], TLBO [74], HPSSO [75], FPA [76], WEO [77], JA [69] and IAS. The best weight obtained by the IAS algorithm, namely, 25450.02 lb, was achieved after 87 iterations (number of structural analyses: 43500), with a standard deviation 543.31 lb. On this basis, the outperformance of the IAS and JA algorithms for the optimal weight calculation of the 200-bar truss structure can be concluded, in comparison with other algorithms.

It should be mentioned that the two best weights for this problem are 25156.5 and 25193.2 lb, obtained by the HPSACO [78] and ray optimization algorithms [28], respectively. The results of these two methods were not reported in the Table 14 since they violate the stress constraints by 9.97% and 12.7%, respectively. The third best weight is 25445.63 lb, obtained by the CMLPSA algorithm, which provides only minor violation of the constraints. Nevertheless, comparison between IAS and CMPLSA is not indicative, because, unlike the other metaheuristic optimizers of Table 14, the stated algorithm utilized gradient in formation that are explicitly available in sizing optimization of truss structures. Fig. 10 illustrates the existing and allowable element stress values of the IAS algorithm for Cases 1, 2 and 3. As can be seen, the design controlling factors (stress limitations) of this problem are satisfied in all the three Cases.

4.2.4. Stepped cantilever beam design problem

The fourth structural optimization problem regards the weight minimization of the stepped cantilever beam shown in Fig. 11. This problem, described in [64], was solved by Chickermane and Gea [79]. In this problem, a stepped cantilever beam comprised of five hollow square section elements with constant thicknesses, is subjected to a point load at the free end. The design variables of this problem are the cross-sectional dimensions of each beam segment (x_1 through x_5), see Fig. 11. The mathematical formulation of the stepped cantilever beam design problem is given in Table 15.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 12 shows the convergence curves of the IAS algorithm for the stepped cantilever beam design problem. It can be seen that IAS found the best optimal solution (1.34) at 44th iteration.

The comparison of the optimization results obtained by IAS and the other six methods listed in Table 16 indicates that the IAS algorithm has a satisfactory performance in terms of both the optimal solution and the reliability index. The StdDev value found by IAS reveals the robustness of this algorithm [59].

4.2.5. Reinforced concrete beam design problem

Minimizing the fabrication cost of structural elements such as reinforced concrete beams is one of the interesting problems in civil engineering. In this regard, the flexural design of a simply supported reinforced concrete beam, which was first solved by Amir and Hasegawa [81], has been selected to assess the performance of the IAS algorithm [64]. This problem has three design variables including steel reinforcement area ($x_1 = A_s$), beam width ($x_2 = b$), and beam depth ($x_3 = h$), as illustrated in Fig. 13.

The dimensions of the reinforced concrete beam should be proportioned to satisfy the following relation between design flexural strength (M_u) based upon the ACI building code 318–77 and the induced internal moments in the beam due to external dead load (M_d) and live load (M_l) as follows [80]:

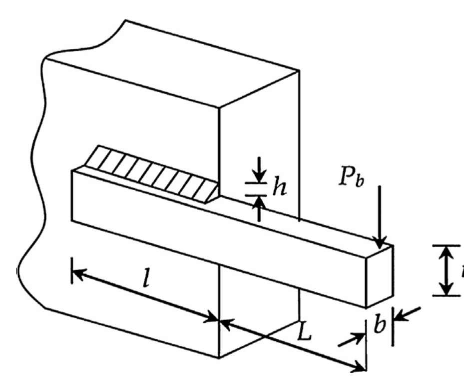

Fig. 15. The welded beam design problem.

Table 18
Comparison of optimization results obtained for the reinforced concrete beam design problem.

Parameter	Amir and Hasegawa [81]	Shih and Yang [82]		Yun [83]		Gandomi et al. [80]	Present study
	SDRC	GHNALM	GHNEP	GA	FLCAHGA	CS	IAS
A_s (in^2)	7.8	6.6	6.32	7.20	6.16	6.32	6.3200
b (in)	7.79	8.495227	8.637180	8.0451	8.7500	8.5000	8.5000
h (in)	31	33	34	32	35	34	34.0000
g_1	−0.0205	−0.1155	−0.0635	−0.0224	0	0	0.0000
g_2	−4.2012	0.0159	−0.7745	−2.8779	−3.6173	−0.2241	−0.22409
f_{min} (in kip)	374.2	362.2455	362.00648	366.1459	364.8541	359.2080	359.2080
StdDev	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	8.54e-18
NEFs	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	1250

Note: SDRC: Hybrid discrete steepest descent and rotating coordinate directions method; GHNALM: Generalized Hopfield network based augmented Lagrange multiplier approach; GHNEP: GHN based extended penalty approach; FLCAHGA Adaptive hybrid GA with fuzzy logic controller.

Table 19
Formulation of the welded beam design problem.

Function	Mathematical formulation	Information
Objective function	$f(h,l,t,b) = 1.10471h^2l + 0.04811tb(14+l)$	$P_b = 6000lb$;
Inequality constraints	$g_1(h,l,t) = \tau(h,l,t) - \tau_{max} \le 0 g_2(t,b) = \sigma(t,b) - \sigma_{max} \le 0 g_3(t,b) = \delta(t,b) - \delta_{max} \le 0 g_4(t,b) = P_b - P_c(t,b) \le 0$ *Herein:*	*G = 12E + 6 psi;*
	$\tau(h,l,t) = \sqrt{(\frac{P_b}{hl\sqrt{2}})^2 + \frac{MP_b l}{Jhl\sqrt{2}} + (\frac{MR}{J})^2}$;	*E = 3E + 7 psi;*
	$M = P_b(L + 0.5l)$; $R = \sqrt{(\frac{l}{2})^2 + (\frac{h+t}{2})^2}$; $J = 2hl\sqrt{2} \times ((\frac{l}{\sqrt{12}})^2 + (\frac{h+t}{2})^2)$;	*L = 14 in;* $\tau_{max} = 13600psi$;
	$\sigma(t,b) = \frac{6P_bL}{t^2b}$; $\delta(t,b) = \frac{4P_bL^3}{Et^3b}$; $P_c(t,b) = \frac{4.013E\sqrt{(\frac{t^2b^6}{36})}}{L^2} \times \left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}}\right)$;	$\sigma_{max} = 30000psi$; $\delta_{max} = 0.25in$;
Geometric constraints	$g_5(h,b) = h - b \le 0$	*h = weld thickness;*
	$g_6(h,l,t,b) = 0.10471x_1^2 + 0.04811tb(14+l) - 5 \le 0$	*l = weld length;*
	$g_7(h) = 0.125 - h \le 0$	*t = bar height;*
Side constraints on design variables	$0.1 \le h \le 2$ *and* $0.1 \le l \le 10$	*b = bar width;*
	$0.1 \le b \le 2$ *and* $0.1 \le t \le 10$	

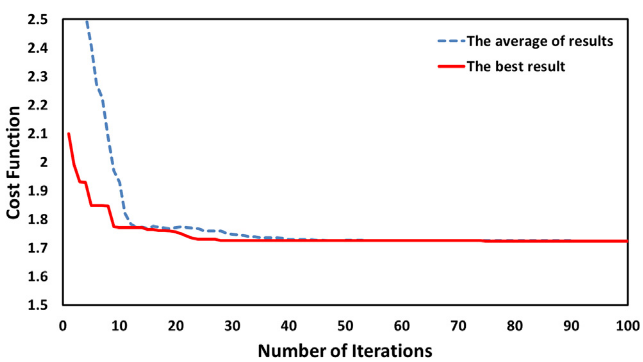

Fig. 16. Convergence curves for the welded beam design problem.

$$M_u = 0.72A_s\sigma_y h\left(1 - \frac{0.59A_s\sigma_y}{0.8bh\sigma_c}\right) \ge 1.4M_d + 1.7M_l \quad (2)$$

where σ_y and σ_c are the yield stress of the reinforcing steel bars and the concrete compressive strength, respectively. Also, the effective depth is assumed to be: (d = 0.8 h). The mathematical description of the reinforced concrete beam design problem is presented in Table 17.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 14 demonstrates the reduction of objective function values with respect to the optimization iterations. It is found that IAS converged to near optimum solution at early iterations. This advantage is seen in other engineering design optimization problems and may be considered as a definite strength point of the proposed method.

The optimization results obtained by IAS and other six algorithms are compared in Table 18. It can be seen that the present algorithm and the CS algorithm converged to the global optimum. Furthermore, IAS has a very low standard deviation thus confirming its robust convergence behavior [59].

4.2.6. Welded beam

Another important issue in designing structural members is finding the minimum value of the thicknesses and lengths of welds in connections. Hence, the welded beam engineering design problem, which was firstly proposed by Coello [84], has been solved in order to assess the performance of the proposed algorithm [64]. In this test case, the dimensions of the welds and the structural members in the connection are subjected to four inequality constraints and three geometric constraints. The four constraints are shear stress (τ), bending stress (σ) in the beam, buckling load on the bar (P_b), and the end deflection of the beam (δ). This problem has four design variables including weld thickness ($x_1 = h$), weld length ($x_2 = l$), bar height ($x_3 = t$), and bar width ($x_4 = b$), as shown in Fig. 15 [31]. The mathematical definition of the welded beam design problem is given in Table 19.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 16 shows the convergence curves of the IAS algorithm for the welded beam design problem. It can be seen that IAS found the best optimal solution (1.7249) at 40th iteration.

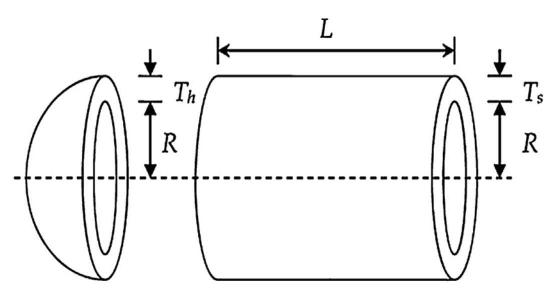

Fig. 17. Pressure vessel design problem.

Table 20
Comparison of optimization results obtained for the welded beam design problem.

Parameter	GA4 [31]	CPSO [31]	CAEP [31]	HPSO [31]	NMPSO [31]	TLBO [25]	MBA [31]	CSA [39]	IAS
h (in)	0.2059	0.2023	0.2057	0.2057	0.2058	N.A.	0.2057	0.2057	0.2057
l (in)	3.4713	3.5442	3.4705	3.4704	3.4683	N.A.	3.4704	3.4704	3.4705
t (in)	9.0202	9.0482	9.0366	9.0366	9.0366	N.A.	9.0366	9.0366	9.0366
b (in)	0.2064	0.2057	0.2057	0.2057	0.2057	N.A.	0.2057	0.2057	0.2057
f_{min} (in^3)	1.7282	1.7280	1.7248	1.7248	1.7247	1.7284	1.7248	1.7248	1.7249
StdDev	7.47e−02	1.29e−02	4.43e−01	4.01e−02	3.50e−03	N.A.	6.94e−19	1.19e−15	3.16e−19
NFEs	80,000	240,000	50,020	81,000	800,004	10,000	47,340	N.A.	6750

Table 21
Formulation of the cylindrical pressure vessel design problem.

Function	Mathematical formulation	Information
Objective function	$f(T_s,T_h,R,L) = 0.6224T_sRL + 1.7781T_hR^2 + 19.84T_s^2R + 3.1661T_S^2L$	T_s = *shell thickness;*
Inequality constraints	$g_1(T_s,R) = -T_s + 0.0193R \le 0$	T_h = *head thickness;*
	$g_2(T_h,R) = -T_h + 0.00954R \le 0$	R = *inner radius;*
	$g_3(R,L) = -\pi R^2L - (\frac{4}{3})\pi R^3 + 1,296,000 \le 0$	L = *cylindrical vessel length;*
	$g_4(L) = L - 240 \le 0$	
Side constraints	$0 \le T_s \le 100$ *and* $10 \le R \le 200$	
	$0 \le T_h \le 100$ *and* $10 \le L \le 200$	

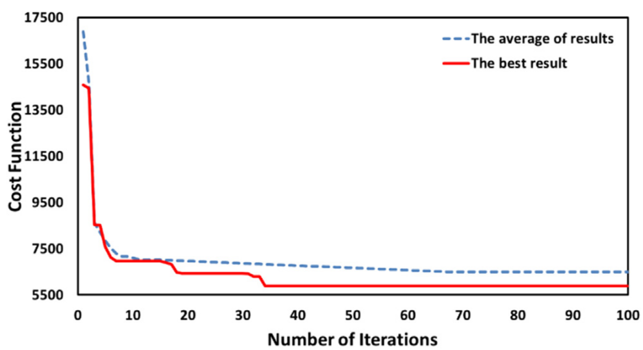

Fig. 18. Convergence curves for the cylindrical pressure vessel design problem.

The results obtained by the IAS algorithm and previous optimization methods such as GA4, CPSO, CAEP, HPSO, NM-PSO, TLBO, MBA, and CSA algorithms are compared in Table 20. It can be seen that the IAS algorithm has a desirable performance in terms of the number of function evaluations and optimized cost. It appears that MBA, CSA and IAS outperform GA4, CPSO, CAEP, HPSO, NMPSO, and TLBO algorithms in terms of the f_{min} and *StdDev*. Moreover, IAS and TLBO are the fastest optimizers overall [59].

4.2.7. Cylindrical pressure vessel design problem

The last structural optimization problem solved in this study regards the minimization of the total cost of a cylindrical pressure vessel including the costs of materials, forming, and welding. This problem was proposed by Kannan and Kramer [85]. This problem has four design variables including the thickness of the shell ($x_1 = T_s$), the thickness of the head ($x_2 = T_h$), inner radius ($x_3 = R$), and length of the cylindrical section of the vessel ($x_4 = L$), as illustrated in Fig. 17 [31]. The mathematical formulation of the cylindrical pressure vessel design problem is given in Table 21.

For solving this problem with IAS, population size and iterations number are set to 50 and 100, respectively. Fig. 18 shows the convergence curves of the IAS algorithm for the cylindrical pressure vessel design problem. It can be seen that the global optimum (5885.33) is obtained at the 33th iteration. The rapid convergence of the IAS algorithm results from the combination of search strategy and constraint handling approach. Concerning the constraint handling strategy, the static penalty function was adopted in this work, which is consistent with the school framework.

The comparison of optimization results obtained by the IAS algorithm and GA3, GA4, CPSO, HPSO, NM-PSO, GQPSO, CDE, TLBO, MBA, and CSA (see Table 22) confirms that the proposed algorithm has the best performance in both terms of optimal solution and number of function evaluations [59].

The excellent response in terms of best optimal solution, convergence speed and robustness exhibited by IAS in this test problem as well as in the previous six problems indicates that the present algorithms can be considered an efficient tool for structural optimization.

5. Conclusion

In this paper, a novel population-based optimization algorithm called "Interactive Autodidactic School" (IAS) is proposed. The main idea of the proposed algorithm is inspired by the interactions between students in an autodidactic school, with the goal of increasing their knowledge to a certain level, through a combination of self-teaching/self-learning, interactive discussion, criticism, and the competition for leadership. Unlike other metaheuristic algorithms, one of the prominent characteristic of the IAS is that it is a free-parameter algorithm. This feature can decrease diversi-

Table 22
Comparison of optimization results obtained for the cylindrical pressure vessel design problem.

Parameter	GA3 [31]	GA4 [31]	CPSO [31]	HPSO [31]	NMPSO [31]	GQPSO [31]	CDE [31]	TLBO [25]	MBA [31]	CSA [39]	IAS
T_s *(in)*	0.8125	0.8125	0.8125	0.8125	0.8036	0.8125	0.8125	N.A.	0.8125	0.8125	0.7781
T_h *(in)*	0.4375	0.4375	0.4375	0.4375	0.3972	0.4375	0.4375	N.A.	0.3856	0.4375	0.3846
R *(in)*	40.3239	42.00974	42.0913	42.0984	41.6392	42.0984	42.0984	N.A.	40.4292	42.09844	40.3196
L *(in)*	200.00	176.6540	176.7465	176.6366	182.412	176.6372	176.637	N.A.	198.4964	176.6365	200.00
g_1	−3.4e−02	−2.0e−03	−1.3e−06	−8.8e−07	**3.6e−05**	−8.7e−07	−6.6e−07	N.A.	0	−4.02e−09	0.00
g_2	−5.2e−02	−3.5e−02	−3.5e−04	−3.5e−02	**3.7e−05**	−3.5e−02	−3.5e−02	N.A.	0	−0.0358	0.00
g_3	−304.40	−24.7593	−118.768	**3.1226**	−1.5914	−0.2179	−3.7051	N.A.	−86.3645	−7.12e−04	−0.0011
g_4	−400.00	−63.3460	−63.2535	−63.3634	−57.587	−63.3628	−63.362	N.A.	−41.5035	−63.3634	−40.00
f_{min} *(in^3)*	6288.74	6059.94	6061.07	6059.71	5930.31	6059.72	6059.73	6059.71	5889.32	6059.7143	5885.33
StdDev	7.4133	130.9297	86.45	86.20	9.161	448.4711	43.0130	N.A.	160.34	384.9454	132.542
NFEs	900,000	80,000	240,000	81,000	80,000	8000	204,800	10,000	70,650	N.A.	8500

Note: values typed in bold denote violated constraints.

fication and uncertainty in solutions since the lack of tuning parameters can increase intensification and stability to the performance of the proposed algorithm.

To assess the performance and versatility of the IAS algorithm as well as its computational costs, twenty classical benchmark mathematical functions and seven structural engineering design optimization problems were solved. According to the optimization results of the mathematical functions, the superiority of the proposed algorithm was in its accuracy to yield the best optimal solution. Among a number of well-known metaheuristic optimization algorithms such as GA, DE, BA, PSO, PBA, and SOS, the proposed algorithm was the only one capable of successfully solving the optimization task in almost all the numerical functions in Table 4 (especially with regard to Dixon-Price, Quartic, and Rosenbrock functions). Furthermore, the IAS algorithm showed a satisfactory performance in solving seven structural engineering design optimization problems in terms of the best optimal solution as the best value of objective function (f_{min}), the computational cost as the number of function evaluations (*NFEs*), and the reliability index as the standard deviation (*StdDev*) on optimized cost. Remarkably, the proposed algorithm always converged to the best feasible design in all structural optimization test problems.

Declaration of Competing Interest

None.

References

[1] Venkataraman P. Applied Optimization with MATLAB Programming. John Wiley & Sons; 2009.
[2] Griva I, Nash SG, Sofer A. Linear and Nonlinear Optimization: Vol 108; Siam; 2009.
[3] Arora JS. Introduction to Optimum Design. Elsevier; 2004.
[4] Yang X-S. Nature-inspired Metaheuristic Algorithms. Luniver press; 2010.
[5] Holland JH. Adaptation in Natural and Artificial Systems. University of Michigan Press; 1975.
[6] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.
[7] Glover F. Tabu search-part I. ORSA J Comput 1989;1:190–206.
[8] Eberhart R, Kennedy J. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks: Citeseer; 1995. p. 1942–8.
[9] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11:341–59.
[10] Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat No 99TH8406): IEEE; 1999. p. 1470–7.
[11] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. J Simul 2001;76:60–8.
[12] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plann Manage 2003;129:210–25.
[13] Pham DT, Ghanbarzadeh A, Koç E, Otri S, Rahim S, Zaidi M. The bees algorithm-a novel tool for complex optimisation problems. Intelligent Production Machines and Systems: Elsevier; 2006. p. 454–9.
[14] Krishnanand K, Ghose D. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. Multiagent Grid Syst 2006;2:209–22.
[15] Erol OK, Eksin I. A new optimization method: big bang–big crunch. Adv Eng Softw 2006;37:106–11.
[16] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 2007;39:459–71.
[17] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. 2007 IEEE congress on evolutionary computation: IEEE; 2007. p. 4661–7.
[18] Rabanal P, Rodríguez I, Rubio F. Using river formation dynamics to design heuristic algorithms. International conference on unconventional computation: Springer; 2007. p. 163–77.
[19] Hosseini HS. Problem solving by intelligent water drops. 2007 IEEE congress on evolutionary computation: IEEE; 2007. p. 3226–31.
[20] Yang X-S. Firefly algorithms for multimodal optimization. International symposium on stochastic algorithms: Springer; 2009. p. 169–78.
[21] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[22] Yang X-S, Deb S. Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC): IEEE; 2009. p. 210–4.
[23] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–89.
[24] Yang X-S. A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010). Springer 2010:65–74.
[25] Rao RV, Savsani VJ, Vakharia D. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43:303–15.
[26] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 2012;110:151–66.
[27] Yang X-S. Flower pollination algorithm for global optimization. International conference on unconventional computing and natural computation: Springer; 2012. p. 240–9.
[28] Kaveh A, Khayatazad M. Ray optimization for size and shape optimization of truss structures. Comput Struct 2013;117:82–94.
[29] Kaveh A, Farhoudi N. A new optimization method: Dolphin echolocation. Adv Eng Softw 2013;59:53–70.
[30] Eesa AS, Brifcani AMA, Orman Z. Cuttlefish algorithm-a novel bio-inspired optimization algorithm. Int J Sci Eng Res 2013;4:1978–86.
[31] Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 2013;13:2592–612.
[32] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.
[33] Cheng M-Y, Prayogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 2014;139:98–112.
[34] Shareef H, Ibrahim AA, Mutlag AH. Lightning search algorithm. Appl Soft Comput 2015;36:315–33.
[35] Mirjalili S. The ant lion optimizer. Adv Eng Softw 2015;83:80–98.
[36] Savsani P, Savsani V. Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 2016;40:3951–78.
[37] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67.
[38] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 2016;96:120–33.
[39] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 2016;169:1–12.
[40] Yazdani M, Jolai F. Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 2016;3:24–36.
[41] Kaveh A, Bakhshpoori T. Water evaporation optimization: a novel physically inspired optimization algorithm. Comput Struct 2016;167:69–85.
[42] Rao R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Computat 2016;7:19–34.
[43] Tabari A, Ahmad A. A new optimization method: Electro-Search algorithm. Comput Chem Eng 2017;103:1–11.
[44] Bouchekara H. Most valuable player algorithm: a novel optimization algorithm inspired from sport. Oper Res Int J 2017:1–57.
[45] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 2017;105:30–47.
[46] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 2017;114:163–91.
[47] Cheng L, Wu X-h, Wang Y. Artificial flora (AF) optimization algorithm. Appl Sci 2018;8:329.
[48] Sayed GI, Darwish A, Hassanien AE. A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. J Exp Theor Artif Intell 2018;30:293–317.
[49] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 2019;23:715–34.
[50] Mortazavi A, Toğan V, Nuhoğlu A. Interactive search algorithm: a new hybrid metaheuristic optimization algorithm. Eng Appl Artif Intell 2018;71:275–92.
[51] Kaveh A, Mahjoubi S. Lion pride optimization algorithm: a meta-heuristic method for global optimization problems. Scientia Iranica 2018;25:3113–32.
[52] Pijarski P, Kacejko P. A new metaheuristic optimization method: the algorithm of the innovative gunner (AIG). Eng Optim 2019:1–20.
[53] Azizyan G, Miarnaeimi F, Rashki M, Shabakhty N. Flying Squirrel Optimizer (FSO): a novel SI-based optimization algorithm for engineering problems. Iran J Optim 2019.
[54] Dhiman G, Kaur A. STOA: a bio-inspired based optimization algorithm for industrial engineering problems. Eng Appl Artif Intell 2019;82:148–74.
[55] Beheshti Z, Shamsuddin M. A review of population-based meta-heuristic algorithm. Int J Adv Soft Comput Appl 2013;5:1.
[56] Ghasemi H, Park H, Rabczuk T. A level-set based IGA formulation for topology optimization of flexoelectric materials. Comput Methods Appl Mech Eng 2017;313:239–58.
[57] Ghasemi H, Park H, Rabczuk T. A multi-material level set-based topology optimization of flexoelectric composites. Comput Methods Appl Mech Eng 2018;332:47–62.
[58] Jamil M, Yang X-SA. literature survey of benchmark functions for global optimization problems. Int J Math Modell Num Optim 2013;4(2):150–94.
[59] Wolpert D, Macready W. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:1089–778.

[60] Cheng M-Y, Lien L-C. Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization. J Comput Civil Eng 2012;26:612–24.
[61] Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 2000;186:311–38.
[62] Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 2002;191:1245–87.
[63] Mezura-Montes E, Coello CAC. Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 2011;1:173–94.
[64] Koziel S, Yang X-S. Computational optimization, methods and algorithms. Springer; 2011.
[65] Wu S-J, Chow P-T. Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 1995;56:979–91.
[66] Lee KS, Geem ZW, Lee S-H, Bae K-W. The harmony search heuristic algorithm for discrete structural optimization. Eng Optim 2005;37:663–84.
[67] Li L, Huang Z, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 2009;87:435–43.
[68] Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. J Constr Steel Res 2009;65:1558–68.
[69] Degertekin SO, Lamberti L, Ugur IB. Sizing, layout and topology design optimization of truss structure susing the Jaya algorithms. J Appl Soft Comput 2017;70:903–28.
[70] Degertekin SO, Lamberti L, Ugur IB. Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm. J Appl Soft Comput 2019;79:363–90.
[71] Lamberti L. An efficient simulated annealing algorithm for design optimization of truss structures. Comput Struct 2008;86:1936–53.
[72] Sonmez M. Artificial bee colony algorithm for optimization of truss structures. J Appl Soft Comput 2011;11(2):2406–18.
[73] Degertekin SO. Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 2012;92:229–41.
[74] Degertekin SO, Hayalioglu MS. Sizing truss structures using teaching-learning-based optimization. Comput Struct 2013;119:177–88.
[75] Kaveh A, Bakhshpoori T, Afshari E. An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 2014;143:40–59.
[76] Bekdas G, Nigdeli SM, Yang XS. Sizing optimization of truss structures using flower pollination algorithm. J Appl Soft Comput 2015;37:322–31.
[77] Kaveh A, Bakhshpoori T. A new metaheuristic for continuous structural optimization: water evaporation optimization. Struct Multidiscipl Optim 2016;54(1):23–43.
[78] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;92:229–41.
[79] Chickermane H, Gea H. Structural optimization using a new local approximation method. Int J Numer Meth Eng 1996;39:829–46.
[80] Gandomi AH, Yang X-S, Alavi AH. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 2013;29:17–35.
[81] Amir HM, Hasegawa T. Nonlinear mixed-discrete structural optimization. J Struct Eng 1989;115:626–46.
[82] Shih C, Yang Y. Generalized Hopfield network based structural optimization using sequential unconstrained minimization technique with additional penalty strategy. Adv Eng Softw 2002;33:721–9.
[83] Yun Y. Study on adaptive hybrid genetic algorithm and its applications to engineering design problems; 2005.
[84] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 2000;41:113–27.
[85] Kannan B, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 1994;116:405–11.