ISA Transactions 53 (2014) 1168–1183

Contents lists available at ScienceDirect

ISA Transactions

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

Interior search algorithm (ISA): A novel approach for global optimization

Amir H. Gandomi *

The University of Akron, OH USA

ARTICLE INFO

Article history:
Received 8 March 2013
Received in revised form
23 October 2013
Accepted 31 March 2014
Available online 29 April 2014
This paper was recommended for publication by Prof. A.B. Rad

Keywords:
Interior search algorithm
Global optimization
Metaheuristic

ABSTRACT

This paper presents the interior search algorithm (ISA) as a novel method for solving optimization tasks. The proposed ISA is inspired by interior design and decoration. The algorithm is different from other metaheuristic algorithms and provides new insight for global optimization. The proposed method is verified using some benchmark mathematical and engineering problems commonly used in the area of optimization. ISA results are further compared with well-known optimization algorithms. The results show that the ISA is efficiently capable of solving optimization problems. The proposed algorithm can outperform the other well-known algorithms. Further, the proposed algorithm is very simple and it only has one parameter to tune.

1. Introduction

Metaheuristic optimization algorithms are used extensively for solving complex optimization problems. Compared to conventional methods based on formal logics or mathematical programming, these metaheuristic algorithms are generally more powerful [57]. Diversification and intensification are the main features of the metaheuristic algorithms [67]. The diversification phase guarantees that the algorithm explores the search space more efficiently. The intensification phase searches through the current best solutions and selects the best candidates. Modern metaheuristic algorithms are developed to solve problems faster, to solve large problems, and to obtain more robust methods [60]. The metaheuristic algorithms do not have limitations in using sources (e.g. physic-inspired charged system search [34]).

In this paper, a new metaheuristic algorithm, called interior search algorithm (ISA), is introduced for global optimization. There is another optimization algorithm in the literature called the interior point method. This method is a mathematical programming not a metaheuristic algorithm and, therefore, it is not related to the current algorithm. The ISA takes into account the aesthetic techniques commonly used for interior design and decoration to investigate global optimization problems, therefore, it can also be called aesthetic search algorithm. The performance and efficiency of the ISA are verified using some widely used benchmark problems. The results confirm the applicability of ISA for solving optimization tasks. The ISA can also outperform the existing metaheuristic algorithms. The paper is organized as follows: Section 2 provides a brief review of the metaheuristic algorithms. Section 3 presents the interior design and decoration metaphor and the characteristics of the proposed ISA, including the formulation of the algorithm. Numerical examples are presented in Section 4 to verify the efficiency of the ISA. In Section 5, the performance of the proposed algorithm is also tested using some well-known engineering design problems which have been previously employed to validate different algorithms. Finally, some concluding remarks and suggestions for future research are provided in Section 6.

2. Metaheuristic algorithms

Optimization techniques can be divided in two groups, mathematical programming and metaheuristic algorithms. In general, the existing metaheuristic algorithms may be divided into two main categories as follows:

- Evolutionary algorithms
- Swarm algorithms

2.1. Evolutionary algorithms

The evolutionary algorithms are generally inspired from biological evolution and use an iterative progress to solve optimization

* Tel.: +1 234 788 0619.
E-mail addresses: ag72@uakron.edu, a.h.gandomi@gmail.com
URL: http://gozips.uakron.edu/~ag72/

http://dx.doi.org/10.1016/j.isatra.2014.03.018
0019-0578/

problems. Evolutionary strategy (ES) [20], genetic algorithm (GA) [30], and differential evolution (DE) [59] are the most well-known paradigms of the evolutionary algorithms which use biological mechanisms such as crossover and mutation. These algorithms have been improved a lot and new variants of them have been introduced (e.g. cellular GA [9]). Genetic programming (GP) is known as an extension of GAs [36]. Several attempts have been made to improve the performance of GP (e.g. multi-stage GP [22]). The mentioned evolutionary algorithms have been widely used to solve engineering optimization problems (e.g. [69]).

Harmony search (HS) algorithm is another evolutionary algorithm proposed by Geem et al. [27]. This algorithm simulates the musicians' behavior when they are searching for better harmonies.

2.2. Swarm algorithms

The most well-known paradigm in the area of swarm intelligence is particle swarm optimization (PSO). The PSO algorithm is based on the simulation of the collective behavior of particles (or animals). The PSO algorithm was first proposed by Eberhart and Kennedy [19]. PSO is a population-based method inspired by the social behavior of bird flocking or fish schooling. Several new PSO algorithms have been proposed up to now.

Another well-known swarm algorithm is ant colony optimization (ACO) which was first proposed by Dorigo et al. [18]. ACO is inspired by the collective foraging behavior of ants. Lucic and Teodorovic [44] developed bee colony optimization algorithm (BCO) which is based on the simulation of the food foraging behavior of honey bees. The bacterial foraging behavior has also been a source for developing new optimization algorithms, called bacterial foraging optimization (BFO) algorithm [52].

Several extensions to the major categories of the swarm intelligence algorithms have been presented in the literature. Yang [67] proposed a novel optimization algorithm, called firefly algorithm (FA), inspired by the firefly's biochemical and social aspects. Yang and Deb [66] formulated a new metaheuristic algorithm via Lévy Flights, called cuckoo search (CS). Gandomi and Alavi [25] have recently proposed a new swarm intelligence algorithm namely Krill Herd (KH). They used the Lagrangian model of the krill herding for the function optimization. The swarm intelligence algorithms have been widely used to solve different kinds of optimization tasks (e.g. [63]).

3. Interior design and decoration metaphor

3.1. Composition design

The interior design procedure follows a coordinated and systematic methodology. This includes research, analysis, and integration of knowledge into the creative process. In order to produce an interior space that fulfills the project goals, the needs and resources of the client should be satisfied [49]. The interior design usually starts from bounds to center. That is to say, a designer commonly starts designing the composition of the elements from the wall and then the space will be limited to design other parts of it. In this process, a designer changes the composition of elements to find a more beautiful view and environment. Referring to the basics of the interior design, the design space should be limited at different stages of the process. Composition of each element is not changed except it produces a more beautiful view based on the satisfaction of the client(s). This procedure can be simply employed for optimization. In this process, an element place is changed if only it achieves a better fitness (more decorative view) and also satisfies the constraints (clients).

3.2. Mirror work

Mirror work is one of the latest innovations in fine art utilized by the Persian designer for decoration. A mirror worker uses different mirrors to produce a more decorative environment. An important part of this process is that mirrors are placed near the most beautiful element(s) to emphasize their prettiness. This aesthetic process can be used for optimization by placing some mirrors near the global best(s) or fittest element(s) to find some other beautiful views.

3.3. The proposed interior search algorithm

The proposed algorithm is inspired by the architectural processes mentioned in the previous sections. In this algorithm, the elements are divided into two groups with the exception of the

Initialization
while any stop criteria is not satisfied
  find the x_{gb}^{j}
  for i = 1 to n
    if x_{gb}
$$x_{gb}^{j} = x_{gb}^{j-1} + r_n \times \lambda$$
    else if $r_1 \le \alpha$
$$x_{m,i}^{j} = r_3 x_i^{j-1} + (1 - r_3) x_{gb}^{j}$$
$$x_i^{j} = 2x_{m,i}^{j} - x_i^{j-1}$$
    else
$$x_i^{j} = LB^{j} + (UB^{j} - LB^{j}) \times r_2$$
    end if
  check the boundaries except for decomposition elements.
  end for
  for i = 1 to n
    evaluate the $f(x_i^j)$
$$x_i^j = \begin{cases} x_i^j & f(x_i^j) < f(x_i^{j-1}) \\ x_i^{j-1} & \text{Else} \end{cases}$$
  end for
end while

Fig. 1. Pseudo code of the interior search algorithm.

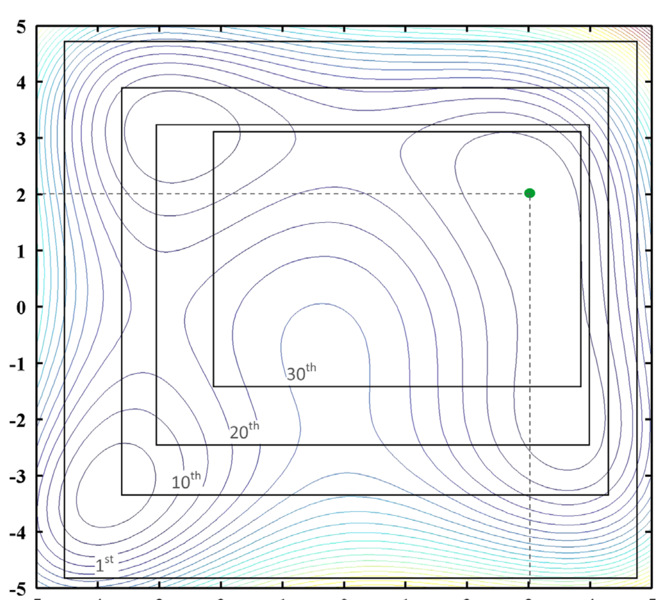

Fig. 2. Different search spaces on the 2D contours of the Himmelblau function for layout optimization part.

fittest element. In one of these groups, called the composition group, the composition of elements is changed to find a more beautiful view. In the other one, called the mirror group, mirrors are placed between these elements and the fittest element to find better views. The detailed algorithm is described below:

1. Randomly generate the locations of elements between upper bounds (***UB***) and lower bounds (***LB***), and find their fitness values.
2. Find the fittest element. If it is a minimum optimization problem, the fittest element has the minimum objective function and vice versa. This element at jth iteration, $\boldsymbol{x}_{gb}^{j}$, is the global best of it.
3. Randomly divide other elements into two groups, composition group and mirror group. For this action, a parameter, α, is defined. For each element, if $r_1 \le \alpha$ it is goes to the mirror group or else it goes to the composition group. Where r_1 is a random value between 0 and 1. It is clear that theoretically α is also a value in this range. However for a balance between intensification and diversification, it should be tuned carefully because it is the only parameter of the algorithm.
4. For the composition group, the composition of each element is randomly changed within a limited search space. This can be formulated as

$$\boldsymbol{x}_i^j = \boldsymbol{LB}^j + (\boldsymbol{UB}^j - \boldsymbol{LB}^j) \times r_2 \tag{1}$$

where $\boldsymbol{x}_i^j$ is the ith elements in the jth iteration, $\boldsymbol{LB}^j$ and $\boldsymbol{UB}^j$ are respectively lower and upper bounds of the composition group elements in jth iteration and they are, respectively, minimum and maximum values of all elements in $(j-1)$th iteration. r_2 is a random value between 0 and 1.
5. For the elements of the mirror group, the mirror work is utilized. First, a mirror is randomly placed between each element and the fittest element (global best). The location of a mirror for ith element at jth iteration is formulated as follows:

$$\boldsymbol{x}_{m,i}^j = r_3 \boldsymbol{x}_i^{j-1} + (1-r_3)\boldsymbol{x}_{gb}^j \tag{2}$$

where r_3 is a random value between 0 and 1. The location of the image or virtual location of the element depends on the mirror location, and it can be formulated as follows:

$$\boldsymbol{x}_i^j = 2\boldsymbol{x}_{m,i}^j - \boldsymbol{x}_i^{j-1} \tag{3}$$

6. For the global best, it is advantageous to slightly change its location using the random walk. It can be formulated as

$$\boldsymbol{x}_{gb}^j = \boldsymbol{x}_{gb}^{j-1} + \boldsymbol{r_n} \times \lambda \tag{4}$$

where $\boldsymbol{r_n}$ is a vector of normally distributed random numbers which has the same size of $\boldsymbol{x}$, and λ is a scale factor which depends on the size of the search space. Here, λ is set as $0.01 \times (\boldsymbol{UB} - \boldsymbol{LB})$. This random walk works as a local search because it searches around the global best.
7. Calculate the fitness values of the new locations of the elements and images (virtual elements). Then update each location if its fitness is improved for revival design. For a minimization problem, this can be expressed as

$$\boldsymbol{x}_i^j = \begin{cases} \boldsymbol{x}_i^j & f(\boldsymbol{x}_i^j) < f(\boldsymbol{x}_i^{j-1}) \\ \boldsymbol{x}_i^{j-1} & else \end{cases} \tag{5}$$

8. If any of the stop criteria is not satisfied, repeat from step 2.

The pseudo code of the proposed ISA is presented in Fig. 1.

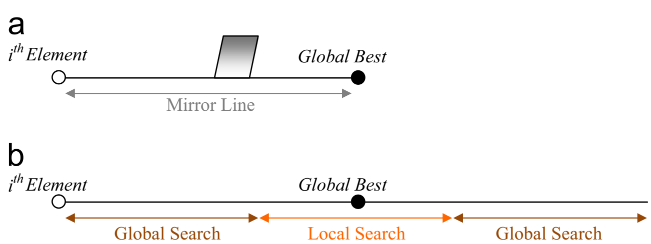

Fig. 3. Schematic mirror search part of the proposed algorithm.

Table 1
Classical benchmark problems.

ID	Function name	Dimension	Range	Optimum
Low dimension benchmarks				
F1	Dekkers and Aarts	2	$(-20\ 20)^2$	$-24{,}777$
F2	Easom	2	$(-10\ 10)^2$	-1
F3	Goldstein and Price	2	$(-2\ 2)^2$	3
F4	Hartman 3	3	$(0\ 1)^3$	-3.862782
F5	Hartman 6	6	$(0\ 1)^6$	-3.322368
F6	Kowalik	4	$(-5\ 5)^4$	3.0748×10^{-4}
F7	Wood	4	$(-10\ 10)^4$	0
High dimension benchmarks				
F8	Ackley	10	$(-32\ 32)^{10}$	0
F9	Griewank	10	$(-600\ 600)^{10}$	0
F10	Levy and Montalvo 1	20	$(-10\ 10)^{20}$	0
F11	Levy and Montalvo 2	20	$(-5\ 5)^{20}$	0
F12	Rastrigin	10	$(-5.12\ 5.12)^{10}$	0
F13	Rosenbrock	10	$(-30\ 30)^{10}$	0
F14	Sphere	20	$(-100\ 100)^{20}$	0

Table 2
Normalized mean optimum results for the benchmark functions with different α.

ID	α										
	0.0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
F1	10.6151	1.0642	1.0009	1	1	1	1	8.7576	8.7579	9.3289	100
F2	1.0809	1.0042	1.0001	1	1	1	1	1	1	1	100
F3	1.0178	1.0001	1.0001	7.1875	7.1875	13.375	1	1	38.125	31.937	100
F4	5.9594	1.0019	1.0001	1	1	1	1	1.1022	1.0681	64.587	100
F5	100	25.700	1	3.4879	4.7494	13.633	12.982	5.8491	1.9022	37.063	89.85
F6	20.639	1	11.7184	1.6806	36.461	41.345	43.969	52.407	71.170	84.320	100
F7	7.5785	2.5657	1	1.7388	1.6876	2.4646	3.2885	3.8904	8.7291	28.560	100
F8	18.488	1	1.3318	3.0811	7.5380	11.594	15.252	25.434	38.211	56.945	100
F9	52.832	9.6641	4.9102	1	2.7195	4.4719	6.4607	11.173	21.084	36.060	100
F10	57.104	1	1.2849	1.1541	1.8734	2.4475	6.1206	6.3156	9.0927	21.728	100
F11	2.1134	1	1.0158	1.1222	1.1660	1.2713	1.8494	2.3478	5.7990	20.061	100
F12	89.780	26.416	7.0897	13.498	1	14.706	25.902	25.255	42.339	61.404	100
F13	2.5036	1	1.1423	1.1776	1.2758	1.8428	1.8852	1.9310	5.5136	10.730	100
F14	2.1569	2.4274	1.3247	1.0166	1	1.0536	1.0899	1.2694	2.8384	9.9026	100

3.4. Evaluation of the proposed interior search algorithm

The proposed algorithm has different parts which are evaluated in this section. In the main part, called the composition optimization part, the ISA works with the composition group of the elements. This part is a random search with limiting the search space. The composition optimization part is a global search which is used for diversification phase. As an example, this part is evaluated using the modified 2D Himmelblau function [32]. In this problem, 20 agents are used to find the global minima shown by ● mark. The limited search spaces after 1st, 10th, 20th and 30th iterations are shown in Fig. 2. It should be noticed that the elements of this group never violate the boundaries because of the strategy that limits the boundaries used here. Therefore, there is no need to check the boundary for these groups of elements.

Another novelty of the proposed algorithm is the mirror search part. The mirror idea used herein is interesting, because it searches through a line when its middle point is the global best. In this process, a mirror is placed in a line between ith element and the fittest element. The mathematical process used for finding the mirror place is presented in Eq. (2) and a schematic of this step is visualized in Fig. 3(a). Because of the mirror position, an image can be placed in line starts from element and pass the global best; where the global best is in the middle of this line (see Fig. 3(b)). As it can be seen in Fig. 3(b), the image can be placed near the global best or not. Generally, if the image is placed near the global best, it works as a local search otherwise it works as a global search. Thus, both intensification and diversification are done in the mirror search part of the proposed algorithm.

As it is mentioned in the previous section, it is effective to change the global best position for local search. It is not included in the composition and mirror groups. This local search helps to improve the intensification of the algorithm. In addition, with

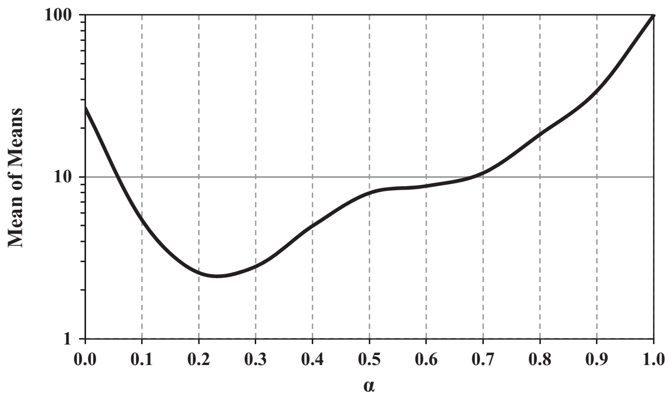

Fig. 4. Effect of α value on the results.

Table 3
Statistical results of the benchmarks using ISAs.

ID and optimum	α	Best	Mean	Median	Worst	Std. dev.	Ave. time
F1	i	−24,776.5	−24,776.5	−24,776.5	−24,776.4	3.3E−02	0.033
−24,777	ii	−24,776.5	−24,776.2	−24,776.5	−24,769.7	1.3E+00	0.021
	iii	−24,776.5	−24,776.5	−24,776.5	−24,775.5	1.9E−01	0.026
F2	i	−1	−1	−1	−1	3.0E−08	0.032
−1	ii	−1	−1	−1	−1	7.4E−08	0.020
	iii	−1	−1	−1	−1	1.7E−07	0.025
F3	i	3	3	3	3	4.3E−07	0.048
3	ii	3	3	3	3	1.2E−07	0.019
	iii	3	3	3	3	6.1E−07	0.025
F4	i	−3.862782	−3.862782	−3.862782	−3.862781	1.9E−07	0.055
−3.862782	ii	−3.862782	−3.862782	−3.862782	−3.862782	6.0E−08	0.047
	iii	−3.862782	−3.862782	−3.862782	−3.862779	5.4E−07	0.067
F5	i	−3.322359	−3.268738	−3.321786	−3.088750	7.3E−02	0.062
−3.322368	ii	−3.322357	−3.248997	−3.275782	−3.104213	7.8E−02	0.056
	iii	−3.322339	−3.273568	−3.319594	−3.132527	6.6E−02	0.060
F6	i	7.178E−4	2.989E−3	9.990E−4	1.849E−2	4.1E−03	0.036
3.0748E−4	ii	4.684E−4	1.748E−3	1.019E−3	1.029E−2	2.1E−03	0.026
	iii	4.086E−4	2.117E−3	8.257E−4	1.078E−2	2.6E−03	0.042
F7	i	7.93E−03	2.14E+00	1.49E+00	7.81E+00	2.2E+00	0.029
0	ii	6.22E−02	2.36E+00	1.36E+00	8.12E+00	2.3E+00	0.021
	iii	1.52E−01	3.27E+00	2.76E+00	7.88E+00	2.5E+00	0.026
F8	i	1.38E−07	4.12E−05	1.27E−06	1.12E−03	2.0E−04	0.558
0	ii	6.17E−08	7.67E−06	1.44E−06	1.59E−04	2.9E−05	0.614
	iii	2.80E−07	2.87E−06	1.80E−06	1.62E−05	3.3E−06	0.403
F9	i	1.72E−02	1.85E−01	1.76E−01	4.46E−01	1.2E−01	0.504
0	ii	9.86E−03	1.52E−01	1.11E−01	4.88E−01	1.2E−01	0.398
	iii	2.96E−02	1.52E−01	1.47E−01	3.14E−01	7.7E−02	0.432
F10	i	2.13E−07	5.86E−03	5.31E−05	1.56E−01	2.8E−02	0.692
0	ii	1.00E−07	1.05E−02	3.56E−05	1.56E−01	3.9E−02	0.670
	iii	2.78E−08	5.24E−03	3.57E−06	1.56E−01	2.8E−02	0.668
F11	i	3.08E−07	2.25E−03	4.10E−05	1.27E−02	4.3E−03	0.307
0	ii	5.50E−07	2.16E−03	7.31E−05	1.14E−02	4.1E−03	0.275
	iii	9.93E−08	3.96E−04	6.02E−06	1.10E−02	2.0E−03	0.282
F12	i	2.98E+00	1.62E+01	1.54E+01	2.97E+01	6.0E+00	0.263
0	ii	2.98E+00	1.63E+01	1.35E+01	3.27E+01	7.8E+00	0.287
	iii	6.04E+00	1.85E+01	1.75E+01	3.84E+01	6.9E+00	0.251
F13	i	2.96E+00	9.95E+00	5.76E+00	7.39E+01	1.5E+01	0.265
0	ii	7.16E−02	5.55E+00	5.67E+00	8.70E+00	1.6E+00	0.261
	iii	5.81E−03	5.70E+00	5.72E+00	9.60E+00	1.6E+00	0.258
F14	i	4.57E−04	5.06E−01	2.15E−02	1.17E+01	2.1E+00	0.277
0	ii	8.11E−05	1.52E−01	3.51E−02	7.94E−01	2.3E−01	0.279
	iii	5.68E−06	7.99E−02	4.15E−03	1.02E+00	2.1E−01	0.251

changing the global best position, mirror lines are also changed and it can improve the performance of the ISA.

It should be noticed that the virtual places of the elements and the global best can be placed out of the limited bounds in the composition optimization. There are some simple schemes in the literature for the boundary constraint handling. Here, evolutionary boundary constraint-handling scheme is used [24] as an efficient scheme to satisfy the simple bounds. In this efficient method, a violated component of an element is replaced with a random value between the violated bound and the related component of the global best.

4. Implementation and numerical experiments

The computational procedures described above have been implemented in a MATLAB™ computer program. In order to evaluate the ISA performance, it is validated using some classical benchmark problems. The benchmarks include both low and high dimension problems, and each class has seven well-known functions. The descriptions of the functions are presented in Table 1. Sphere function is the one of the simplest benchmarks and it is the sum of squares of variables' components. More details about other benchmark functions can be found in [2].

Owing to the random nature of the ISA and other metaheuristic algorithms, their performance cannot be judged by the result of a single run. More than one trial with independent population initializations should be made to evaluate the performance of the approach. Therefore, in this study the results are obtained in 30 trials. For solving the first seven benchmarks, the maximum number of function evaluations was 1000. The maximum number of search was increased for the high-dimensional benchmarks to 10,000. The population size of 25 and 50 were used for low- and high-dimensional problems, respectively.

4.1. Tuning

All optimization algorithms have some parameters which should be tuned before the simulation. For the metaheuristic algorithms, more than two parameters are usually tuned, but for the proposed algorithm, α is the only parameter to be tuned. It is one of the advantages of the ISA because it is potentially more adaptive to a wider class of optimization problems. The α is varied from 0 to 1 for the benchmark problems and the mean of optimum values over runs are presented in Table 2. It should be noted that all the values are normalized between 1 and 100 in this table to simply compare the results. At first, the results confirm that optimum α may vary for different problems. From Table 2, it is also clear that neither composition optimization nor mirror search can be successful alone and there should be a trade-off between these two strategies. Mean of mean values for different α values is shown in Fig. 4. As it is shown, the general optimum value of α for these functions is 0.2.

Because it is the first trial of the proposed algorithm, some strategies are also examined here for tuning α. Similar to other algorithms, the optimum α is changed with the problem. However, it generally is around 0.2 for unconstrained optimization problem. Therefore, three strategies are examined here to tune it:

i. A constant value (here it is a number equal to 0.2).
ii. Random numbers (here it is chosen between 0.1 and 0.3).

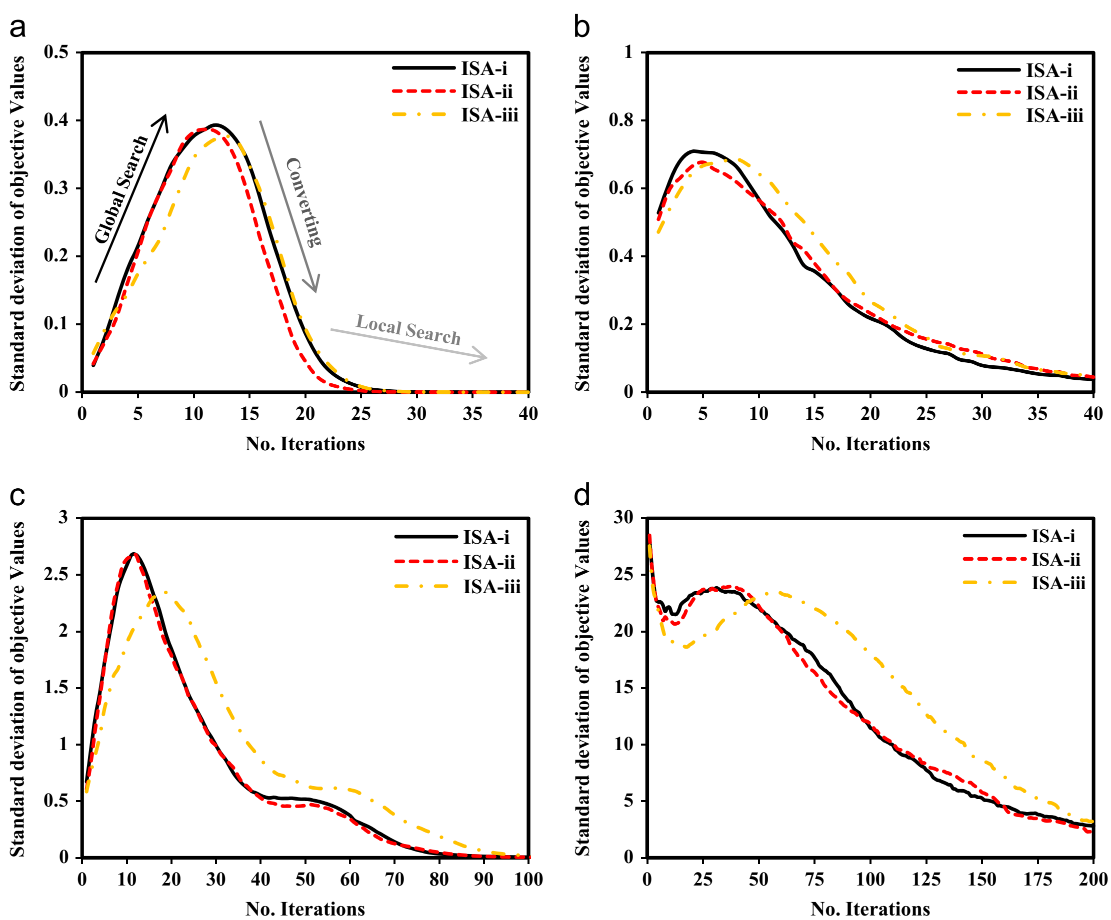

Fig. 5. Convergence of four benchmark functions.

iii. A variable number (here it is changed from 0.1 to 0.3 during the iterations).

In the third strategy, α is equal to 0.1 at the beginning of the search to emphasize exploration and then it is linearly increased to 0.3 at the end to encourage exploitation.

The simulation results of the strategies are presented in Table 3. At first glance, it is clear that all the algorithms have found the global optimums of the benchmarks with high degree of accuracy. From the statistical results, the simulation results of different α are close to each other and none of them clearly outperforms the others. Generally, it can be seen that the ISA-i and ii perform slightly better than ISA-iii for the low-dimensional problems. For the high-dimensional problems, ISA-iii slightly outperforms the other strategies.

4.2. Investigating the globality

Ideally, when a global algorithm is used to solve a global benchmark problem, the algorithm passes through three stages to find the global optimum:

1. Global search
2. Converting stage
3. Local search

In the first stage, the algorithm mainly tries to explore the whole search space. After that, its search strategy converts from the global search to a local search. This phase is called the converting stage. Finally, the algorithm carries out a local search near the global best(s). To clarify the above mentioned stages and also demonstrate the globality of the proposed algorithm, some of the benchmarks are investigated. Fig. 5(a–d) shows the plot of standard deviation of objectives versus iterations for four benchmarks (two low-dimensional and two high-dimensional problems). As it can be seen in this figure, each algorithm first reaches a peak which identifies the global search stage [35]. Comparing different ISAs, it is clear that the third one, which used the iii strategy, has longer global search stage. Therefore, this algorithm is more global than the others. After the peaks in the plots, the lines fall sharply and this can be considered as the converting stage. At the final stage, the slope of the lines decreases and the standard deviation converges to a constant and miniscule number which shows the algorithm performing the local search.

4.3. Comparative study

Herein, the new ISA is compared with some well-known evolutionary and swarm algorithms. A well-known evolutionary algorithm namely DE algorithm is selected for comparison. To implement this algorithm, it is very important to tune its scale factor (F) and crossover probability (Cr). They are previously tuned for some benchmark problems as $F=0.9$ and $Cr=0.7$ in [23] so these values are used in our simulations. The most widely used swarm algorithm is PSO algorithm. The PSO is tuned as acceleration constants equal to 2 (C_1, $C_2=2$) and inertia weight (ω) begins with 1.1 and linearly decreases to 0.1 during the iterations to do both exploration and exploitation.

The statistical results of the DE, PSO and ISAs on benchmark problems are presented in Table 4. For a simple comparison of the algorithms, the statistical results are normalized between 0 and 1. Therefore, the worst and best values of each statistical parameter are changed to 0 and 1, respectively. As it is shown, the proposed algorithm is far better than DE and PSO in all cases.

The convergence plots of the DE, PSO and ISAs on the benchmark problems are also presented in Fig. 6. It should be noted that the global optimums of the first six benchmarks are shifted to the origin to draw the semi-logarithmic convergence plots. At first, the plots clearly confirm that the performance of the proposed algorithms is better than the well-known DE and PSO. The differences between performances are obvious in the convergence plots of high-dimensional problems. The slope of the proposed algorithm is steep and it shows that the ISA converges quickly and rarely stops before finding the global optimum. It is shown that the convergence plots of different ISAs indicate that they perform similarly. However, the performance of the iii strategy is slightly better for the high-dimensional problems.

One of the effective strategies for comparison of metaheuristic algorithms is the oracle-based view of computation [62]. According to this strategy, the best solution should be found within a certain number of function evaluations. Herein, the best values can be used for comparison because of the equal number of function evaluation value for the methods. Referring to the result presented in Table 4 and based on the oracle-based view, the performances of ISAs are much better than the DE and PSO algorithms. This confirms the robustness of the proposed algorithm.

4.4. Convergence rate

In addition to performance evaluation using the statistical results of the best solutions, the performances could be evaluated using the number of function evaluations. Rahnamayan et al. [53]

Table 4
Normalized statistical results of DE, PSO and ISAs for the benchmark problems.

ID	Parameter	DE	PSO	IS-i	IS-ii	IS-iii
F1	Best	0	0.997002	1	1	0.999999
	Mean	0.895777	0	1	0.999582	0.999958
	Std. dev.	0.971378	0	1	0.999725	0.999965
F2	Best	0	0.999946	1	1	1
	Mean	0	0.999620	1	0.999999	0.999997
	Std. dev.	0	0.999142	1	0.999998	0.999994
F3	Best	0	0.999997	1	1	1
	Mean	0	0.999786	0.999999	1	0.999998
	Std. dev.	0	0.999668	0.999997	1	0.999995
F4	Best	0	0.999766	0.999994	0.999999	1
	Mean	0	0.187904	0.999992	1	0.999964
	Std. dev.	0.686917	0	0.999989	1	0.999960
F5	Best	0	0.993159	1	0.999975	0.999762
	Mean	0	0.626057	0.976234	0.879095	1
	Std. dev.	0.633362	0	0.884934	0.792024	1
F6	Best	0	0.907799	0.657453	0.933824	1
	Mean	0.248771	0	0.257184	1	0.779006
	Std. dev.	0.926355	0	0.297228	1	0.840991
F7	Best	0	0.951021	1	0.984216	0.957982
	Mean	0	0.570682	1	0.989653	0.946284
	Std. dev.	0.520294	0	1	0.997675	0.988386
F8	Best	0	0.986927	1	1	1
	Mean	0	0.524069	0.999995	0.999999	1
	Std. dev.	0	0.172545	0.999887	0.999986	1
F9	Best	0	0.864023	0.991441	1	0.977092
	Mean	0	0.643700	0.967673	1	0.999772
	Std. dev.	0.863273	0	0.839842	0.846065	1
F10	Best	0	0.510263	1	1	1
	Mean	0	0.423560	0.999477	0.995562	1
	Std. dev.	0	0.267543	0.999844	0.976505	1
F11	Best	0.474927	0	1	0.999999	1
	Mean	0.560875	0	0.998315	0.998400	1
	Std. dev.	0.771524	0	0.996161	0.996542	1
F12	Best	0	0.886207	1	1	0.886900
	Mean	0	0.924291	1	0.996630	0.925420
	Std. dev.	0	0.390799	1	0.320603	0.675492
F13	Best	0	0.956048	0.985525	0.999677	1
	Mean	0	0.103781	0.993795	1	0.999783
	Std. dev.	0.815689	0	0.991816	1	0.999993
F14	Best	0	0.809088	0.999999	1	1
	Mean	0	0.586485	0.999551	0.999924	1
	Std. dev.	0	0.075572	0.992406	0.999931	1

used a strategy to compare the convergence speed of an optimization algorithm to a benchmark optimization algorithm. In their proposed process, both algorithms need to reach a pre-defined value called value-to-reach. Finding a proper value may not be convenient as too small a value may not properly reflect the convergency and if the number is too large, an algorithm either may not converge to this value or not converge until after a huge number of searches.

Here, a general scheme is proposed to find the convergency rate (*CR*) for any number of optimization algorithms without defining the value-to-reach as follows:

1. Run all algorithms up to the maximum number of function evaluations ($\text{NFE}^{\max}$).
2. Set the maximum objective value found (in minimization problems) as the value-of-interest.
3. Find in which number of function evaluation, each algorithm has reached to this value and set it as NFE^{R}. For the algorithm that has the maximum objective value found, NFE^{R} is equal to $\text{NFE}^{\max}$.
4. Compute the *CR* for each algorithm as

$$CR = \left(1 - \frac{\text{NFE}^{R}}{\text{NFE}^{\max}}\right) \times 100\% \tag{6}$$

where $CR \in [0,1)$ and *CR* equal to zero for the algorithm with the worst convergency.

The average *CR* values of the low- and high-dimensional test functions of different algorithms are presented in Fig. 7. From the

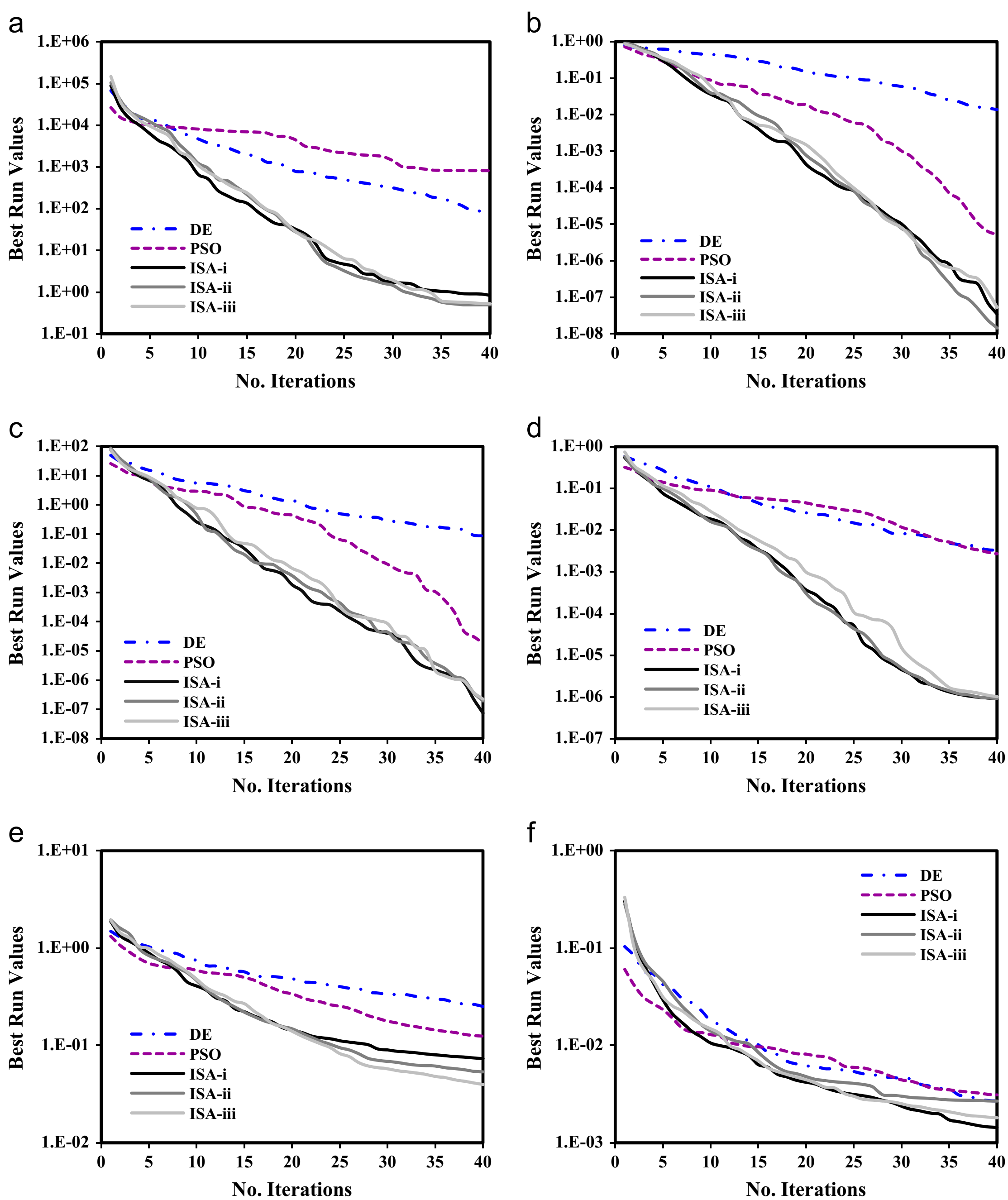

Fig. 6. Comparison of the convergence history for the benchmark problems. (a) Dekkers and Aarts Function, (b) Easom Function, (c) Goldstein and Price's Function, (d) Hartman 3 Function, (e) Hartman 6 Function, (f) Kowalik Function, (g) Wood Function, (h) Ackley Function, (i) Griewank Function, (j) Levy and Montalvo 1 Function, (k) Levy and Montalvo 2 Function, (l) Rastrigin Function, (m) Rosenbrock Function and (n) Sphere Function.

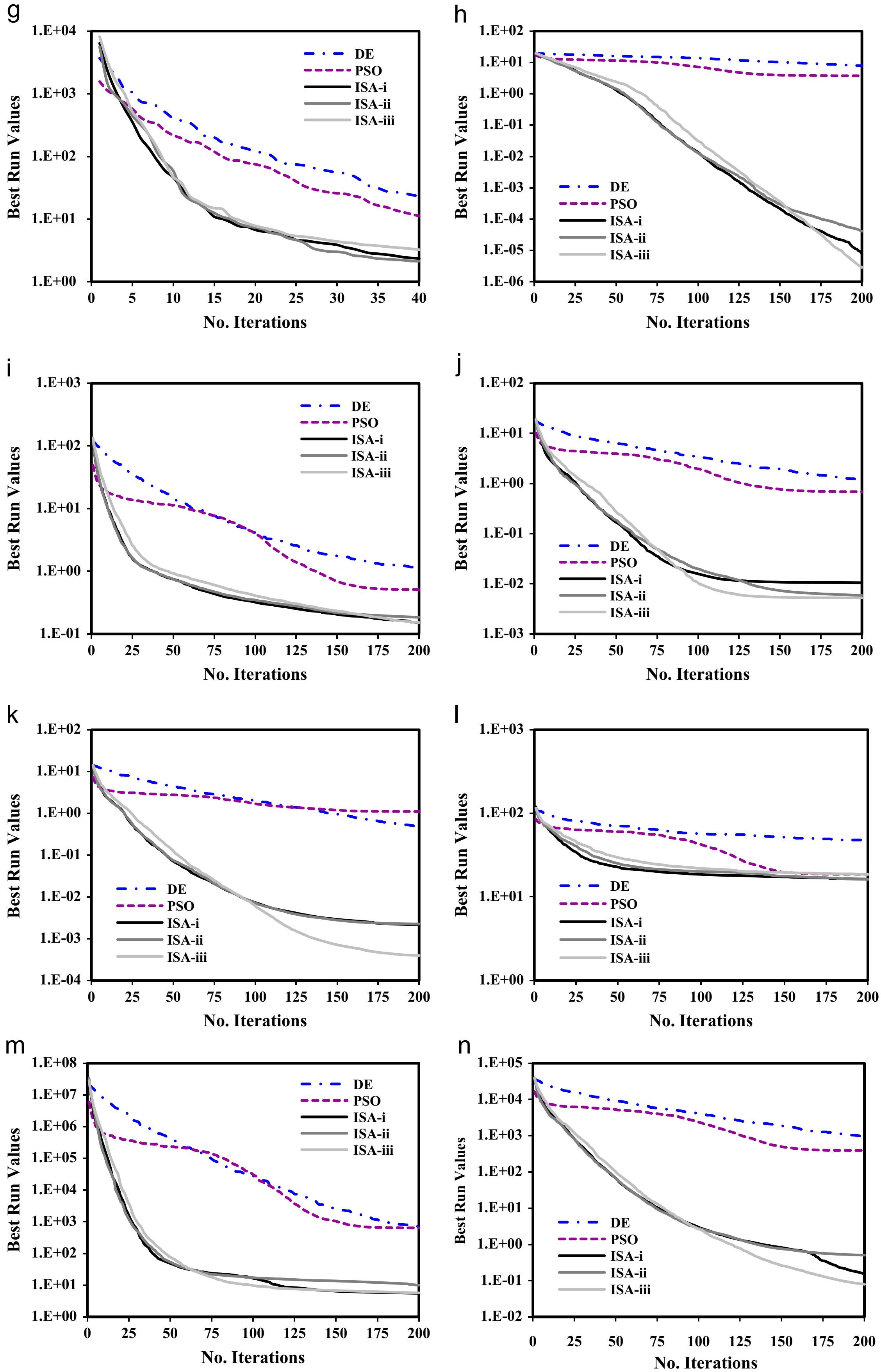

Fig. 6. (*continued*)

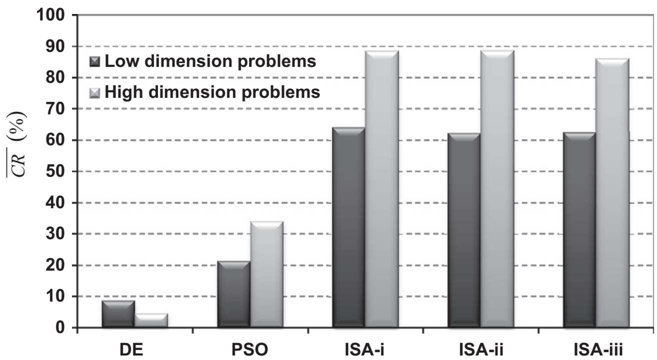

Fig. 7. Comparison of average *CR* for different algorithms.

figure, it is clear that the DE has the worst *CR* and the ISAs have the best *CR*.

5. Engineering design problems

Real-world engineering optimization problems are usually non-linear and involve with complex geometrical and mechanical constraints. To assess the performance of the ISA, it was tested using some constrained engineering design problems. The proposed algorithm was first applied to some well-known benchmark engineering problems including welded beam design, pressure vessel design and spring design. Then it applied to a real-world problem, design optimization of 72 bar space truss under multi loading. The detailed formulation of all these problems can be found in [37].

As discussed in the previous sections, the dynamic strategy (iii strategy) performs slightly better than the others for the unconstrained optimization. Therefore, an adaptive version of this strategy is used for the constrained optimization. For an effective constrained optimization, based on the experiences, α is set as a variable number changed linearly from 0.1 to 0.9 during the iterations. It can be observed that this strategy improves the efficiency of the algorithm for a more careful exploration. The constraint handling used here is similar to the parameter-less penalty scheme presented in [5]. The statistical results of the constrained engineering problems are obtained in 30 trials.

5.1. Engineering benchmark problems

The welded beam design, pressure vessel design and spring design problems have been previously employed to check the effectiveness and validity of different algorithms for the constrained optimization. The main features of these constrained engineering design problems are presented in Table 5.

5.1.1. Welded beam design

The proposed ISA was applied to the design of a well-known welded beam problem for minimum overall cost of fabrication (see Fig. 8). This steel beam 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 (x_1), the length of the welded joint, l (x_2), the width of the beam, t (x_3) and the thickness of the beam, b (x_4), are the design variables of this case study. The objective function of the problem is expressed as follows:

$$\text{minimize}: \quad f(X)=(1+C_1)x_1^2x_2+C_2x_3x_4(L+x_2) \tag{6}$$

This problem is subject to five constraints including shear stress (τ), bending stress (σ), buckling load (P_c), deflection (δ) and geometric

Table 5
Main features of the constrained engineering problems.

Design problem	No. var.	Var. type	ρ (%)[a]	No. LC[b]	No. NLC[c]
Welded beam	4	Continues	2.686	4	1
Pressure vessel	4	Mixed	39.676	3	1
Tension/compression spring	3	Continues	0.754	1	3

[a] ρ is the ratio of the size of the feasible region to the size of the entire search space [46].
[b] LC is the linear constraint.
[c] NLC is the non-linear constraint.

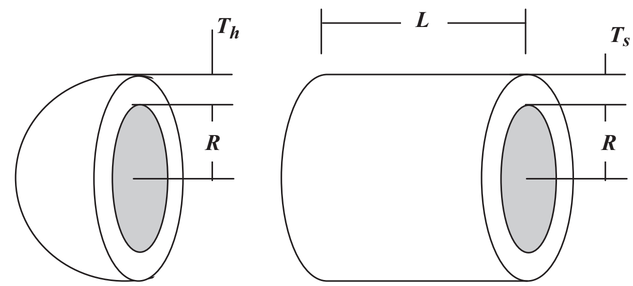

Fig. 8. Schematic of the welded beam design problem.

constraints as follows:

$$g_1(X)=\tau_d-\tau(X)\geq 0 \tag{7}$$

$$g_2(X)=\sigma_d-\sigma(X)\geq 0 \tag{8}$$

$$g_3(X)=x_4-x_1\geq 0 \tag{9}$$

$$g_4(X)=P_c(X)-P\geq 0 \tag{10}$$

$$g_5(X)=0.25-\delta(X)\geq 0 \tag{11}$$

where

$$\tau(X)=\sqrt{(\tau'(X))^2+(\tau''(X))^2+x_2\tau'(X)\tau''(X)/\sqrt{0.25(x_2^2+(x_1+x_3)^2)}} \tag{12}$$

$$\sigma(\boldsymbol{X})=\frac{504,000}{\boldsymbol{x}_3^2\boldsymbol{x}_4} \tag{13}$$

$$\boldsymbol{P_c}(\boldsymbol{X})=64,746.022(1-0.0282346x_3)\boldsymbol{x}_3\boldsymbol{x}_4^3 \tag{14}$$

$$\delta(\boldsymbol{X})=\frac{2.1952}{\boldsymbol{x}_3^3\boldsymbol{x}_4} \tag{15}$$

$$\tau'(\boldsymbol{X})=\frac{6000}{\sqrt{2}\boldsymbol{x}_1\boldsymbol{x}_2} \tag{16}$$

$$\tau''(\boldsymbol{X})=\frac{6000(14+0.5\boldsymbol{x}_2)\sqrt{0.25(\boldsymbol{x}_2^2+(\boldsymbol{x}_1+\boldsymbol{x}_3)^2)}}{2\{0.707\boldsymbol{x}_1\boldsymbol{x}_2(\boldsymbol{x}_2^2/12+0.25(\boldsymbol{x}_1+\boldsymbol{x}_3)^2)\}} \tag{17}$$

The simple bounds of the problem are: $0.125\leq \boldsymbol{x}_1\leq 5$, and $0.1\leq \boldsymbol{x}_2$, $\boldsymbol{x}_3$, $\boldsymbol{x}_4\leq 10$. The constant values for the formulation are also shown in Table 6.

The ISA was run to find the global optima of the above design problem. ISA had located the global optima at a maximum computational cost of 30,000 function evaluations per run. The proposed method obtained the best design overall of 2.3812 corresponding to X=[0.24433032 6.21993053 8.29152130 0.2443689]. Table 7 compares the optimization results found by ISA with those of other optimization algorithms reported in the literature. As it can be seen in Table 7, the best cost is the same as that obtained by

Bernardino et al. [6] and Deb [15], and it is slightly worse than the cost obtained by Aragon et al. [3]. However, the number of searches of the ISA method is far less than these studies.

5.1.2. Pressure vessel design

This benchmark engineering problem is shown in Fig. 9. It is a mixed type of optimization with two continuous variables while others two being an integer multiples of a basic dimension. The total cost, including a combination of welding cost, material and forming cost, is to be minimized in $. The welded beam is fixed and designed to support a load (P). The involved variables are the thickness, T_s (x_1), thickness of the head, T_h (x_2), the inner radius, R (x_3), and the length of the cylindrical section of the vessel, L (x_4). The thicknesses of the variables are discrete values which are integer multiples of 0.0625 in. The problem can be expressed as follows:

$$\text{minimize}: \quad f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_2^2x_4 \tag{18}$$

Subject to four constraints are set in accordance with the American Society of Mechanical Engineers design codes:

$$g_1 = -x_1 + 0.0193x_3 \le 0 \tag{19}$$

$$g_2 = -x_2 + 0.00954x_3 \le 0 \tag{20}$$

$$g_3 = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1,296,000 \le 0 \tag{21}$$

$$g_4 = x_4 - 240 \le 0 \tag{22}$$

where the simple bounds of the problem are: $1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625$ and $10 \le x_3, x_4 \le 200$.

The minimum cost and the statistical values of the best solutions obtained by ISA and results for several approaches are reported in Table 8. As it is seen, the results obtained by the ISA are better than the available solutions in all cases. The variable are $X=$ [0.8125, 0.4375, 42.09845, 176.6366] which satisfy all the constraints. It is worth pointing out that not only is the minimum cost obtained by the proposed algorithm (6059.7143 $) better than the all available solutions but also its number of function evaluations is very fewer than the other studies. This also confirms the excellent convergence of the proposed algorithm.

5.1.3. Tension/compression spring design

Another well-known constrained engineering optimization task is the design of a tension/compression spring for a minimum weight. Fig. 10 shows a tension/compression spring with three design variables. The weight of the spring is to be minimized, subject to four constraints on the minimum deflection, shear, and surge frequency, and limits on the outside diameter. The design variables involved are the wire diameter, d (x_1), the mean coil diameter, D (x_2), and the number of active coils, N (x_3). Then, the

Table 6
Constant values of specifications.

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	13,600 (psi)
σ_d	Design normal stress of the bar material	30,000 (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.)

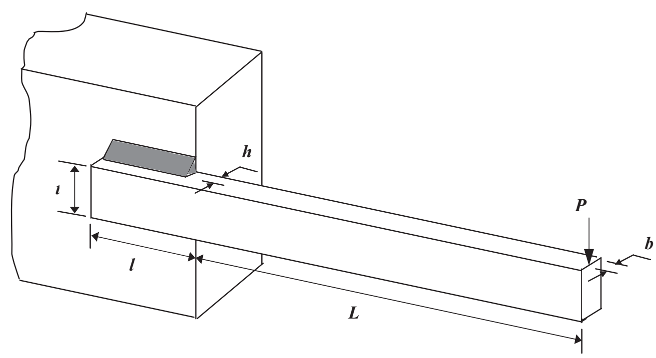

Fig. 9. Schematic of the pressure vessel design problem.

Table 7
Best results of the welded beam design example using different methods.

Reference	Method	Best	Mean	Worst	Std. dev.	No. eval.
Akhtar et al. [1]	SBS[a]	2.4426	N/A	N/A	N/A	20,000
Aragon et al. [3]	TCA[b]	2.3811	2.7104	2.4398	9.31E−2	320,000
Barbosa and Lemonge [4]	GA-AP[c]	2.3814	5.9480	3.4956	9.09E−1	320,000
Bernardino et al. [6]	GA-AIS[d]	2.3812	2.3899	2.4139	N/A	320,000
Bernardino et al. [7]	GA-AIS	2.3834	4.0560	2.9930	2.02E−1	320,000
Deb [16]	GA	2.4331	N/A	N/A	N/A	40,080
Deb [15]	GA-PLP[e]	2.3812	N/A	2.6458	N/A	320,080
Lemonge and Barbosa [39]	GA-AP	2.3816	2.4172	2.9553	N/A	320,000
Montes and Coello Coello [47]	MOEA[f]	2.3829	2.4209	N/A	2.56E−2	80,000
Montes and Ocaña [48]	BFO	2.3868	2.404	N/A	1.60E−2	48,000
Ray and Liew [54]	SCA[g]	2.3854	3.2550	6.3997	9.60E−1	33,095
Runarsson and Yao [56]	GA-SR[h]	2.5961	10.1833	4.3326	1.29E+0	320,000
Wang et al. [61]	AATM[i]	2.3823	2.3870	2.3916	2.20E−3	30,000
Present study	ISA	2.3812	2.4973	2.6700	1.02E−1	30,000

[a] SBS is the socio behavioral simulation.
[b] TCA is the T-Cell algorithm.
[c] AP is the adaptive penalty.
[d] AIS is the artificial immune system.
[e] PLP is the parameter-less penalty.
[f] MOEA is the multi-objective evolutionary algorithm.
[g] Society and civilization algorithm.
[h] SR is the stochastic ranking.
[i] AATM is the accelerating adaptive trade-off model.

Table 8
Statistical results of the pressure vessel design example by different models.

Reference	Method	Best	Mean	Worst	Std. dev.	No. eval.
Akhtar et al. [1]	SBS	6171.000	6335.050	6453.650	N/A	20,000
Aragon et al. [3]	TCA	6390.554	7694.067	6737.065	357	80,000
Barbosa and Lemonge [4]	GA-AP	6065.821	8248.003	6632.376	515	80,000
Lemonge and Barbosa [39]	GA-AP	6060.188	6311.766	6838.939	N/A	80,000
Bernardino et al. [6]	GA-AIS	6060.138	6845.496	6385.942	N/A	80,000
Bernardino et al. [7]	GA-AIS	6059.854	7388.160	6545.126	124	80,000
Coello Coello [13]	GA-CP[a]	6288.745	6293.840	6308.150	7.4133	900,000
Coello Coello [14]	GA-DP[b]	6127.414	6616.933	7572.659	358.85	2,500,000
Coello Coello and Cortes [11]	GA-AIS	6061.120	6734.090	7368.060	458	150,000
Runarsson and Yao [56]	GA-SR	6832.584	8012.615	7187.314	267	80,000
Coello Coello and Montes [12]	GA	6059.946	6177.250	6469.320	130.93	80,000
He and Wang [28]	PSO	6061.078	6147.130	6363.800	86.455	200,000
Parsopoulos and Vrahatis [51]	PSO	6544.270	9032.550	11,638.200	995.57	100,000
Renato and Dos Santos [55]	CPSO[c]	6363.804	6147.133	6061.078	86.450	240,000
Montes and Coello Coello [47]	MOEA	6059.926	6172.527	N/A	124	80,000
Montes and Ocaña [48]	BFO	6060.460	6074.625	N/A	15.6	48,000
Brajevic et al. [8]	BCO	6059.768	6060.210	N/A	0.0069	240,000
Huang et al. [31]	DE	6059.734	6085.230	6371.050	43.013	240,000
Wang et al. [61]	AATM	6059.726	6061.988	6090.802	4.70	30,000
Present study	ISA	6059.714	6410.087	7332.846	384.6	5,000

[a] CP is the co-evolutionary penalty.
[b] DP is the death penalty.
[c] CPSO is the co-evolutionary.

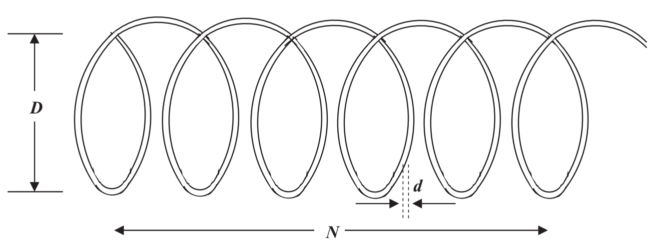

Fig. 10. Schematic of the tension/compression spring design problem.

optimization problem can be expressed as follows:

$$\text{minimize}: \quad f(X) = (x_3 + 2) \times x_2 x_1^2 \tag{23}$$

The weight of the spring is to be minimized, subject to constraints on the minimum deflection (g_1), shear (g_2), and surge frequency (g_3), and limits on the outside diameter (g_4) as follows:

$$g_1 = 1 - \frac{x_2^3 x_3}{71,785 x_1^4} \le 0 \tag{24}$$

$$g_2 = \frac{4x_2^2 - x_1 x_2}{12,566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0 \tag{25}$$

$$g_3 = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \tag{26}$$

$$g_4 = \frac{x_1 + x_2}{1.5} - 1 \le 0 \tag{27}$$

where the simple bounds of the problem are $0.05 \le x_1 \le 1$, $0.25 \le x_2 \le 1.3$ and $2 \le x_3 \le 15$.

Table 9 compares the optimization results obtained using ISA with those reported in the literature. From this table, the best weight is 0.012665 which is obtained by TCA, SA-DS and the proposed ISA. An interesting part of the results is that the ISA reaches this optimum only after 8900 function evaluations which is far less than the other studies presented here. Hence, according the oracle-based view of computation, it can be concluded that it clearly outperforms the other algorithms presented in Table 9.

5.2. Real-world problems

5.2.1. 72 bar space truss

Truss optimization is known to be a demanding area of structural optimization. Several optimization algorithms have been employed to minimize the weight (or volume) of truss structures [37]. To benchmark the proposed ISA against a complex real-world problem, a space truss optimization problem is solved. In practice, the structures are designed using defined sections, so this truss problem is optimized by discrete variables. Therefore, derivative free metaheuristics such as GA, PSO and ISA can be suitable for this design problem.

The 72 bar space truss with 20 nodes is shown in Fig. 11. The objective function of the weight minimization of the 72 bar space truss design problem can be expressed as

$$\text{minimize}: \quad W(A) = \sum_{i=1}^{72} L_i A_i \rho_i \tag{28}$$

where L is the length of member, A is the cross-sectional area of the member, and ρ is the mass density and it is equal to 2768 kg/m^3 (0.10 lb/in.^{3}) for all members in this problem. All members are assumed to be constructed from a material with an elastic modulus of $E = 68.971$ GPa (10,000 ksi).

The truss structure is subjected to two loading conditions given in Table 10. For the structural analysis of the space truss, the finite element method is utilized. The nodes are subjected to the displacement limits of ± 6.35 mm (± 0.25 in.) and the members are subjected to the stress limits of ± 172.375 MPa (± 25 ksi). Due to the symmetry of the truss, variable linking is adopted by grouping cross-sectional areas of the members into 16 different groups. The cross sections are discrete values which are integer multiples of 64.516 mm^2 (0.1 in.^{2}) and the simple bounds of the problem are: $1 \times 64.516 \le A_i \le 32 \times 64.516$. The number of possible solutions for the 72 bar space truss design problem is $32^{16} \approx 1.21 \times 10^{24}$.

The statistical results of the ISA for this problem are presented in Table 11. With 50 elements, the algorithm converged to the optimum after about 8200 structural analyses. The convergence history of the results is shown in Fig. 12.

Tolerances of the variables at the end of runs are presented in Fig. 13 where they are scaled between 0 and 1. From this figure, it

Table 9
Statistical results of the spring design example using different methods.

Reference	Method	Best	Mean	Worst	Std. dev.	No. eval.
Aragon et al. [3]	TCA	0.012665	0.012732	0.013309	9.40E−5	36,000
Barbosa and Lemonge [4]	GA-AP	0.012679	0.014022	0.017794	1.47E−3	36,000
Bernardino et al. [6]	GA-AIS	0.012666	0.013880	0.012974	N/A	36,000
Bernardino et al. [7]	GA-AIS	0.012666	0.013131	0.015318	6.28E−4	36,000
Runarsson and Yao [56]	GA-SR	0.012680	0.013993	0.017796	1.27E−3	36,000
Coello Coello [13]	GA-CP	0.012705	0.012769	0.012822	3.94E−5	900,000
Coello Coello and Montes [12]	GA	0.012681	0.012742	0.012973	5.90E−5	30,000
He and Wang [28]	PSO	0.012675	0.012730	0.012924	5.20E−5	200,000
Parsopoulos and Vrahatis [51]	PSO	0.013120	0.022948	0.050365	7.21E−3	100,000
Renato and Dos Santos [55]	CPSO	0.012924	0.012730	0.012674	5.20E−4	240,000
Hedar and Fukushima [29]	SA-DS[a]	0.012665	0.012665	0.012665	N/A	49,531
Jaberipour and Khorram [32]	HS	0.012666	N/A	N/A	N/A	200,000
Mahdavi et al. [45]	HS	0.012671	N/A	N/A	N/A	50,000
Montes and Coello Coello [47]	MOEA	0.01268	0.01296	N/A	3.63E−04	80,000
Montes and Ocaña [48]	BFO	0.012671	0.012759	N/A	1.36E−04	48,000
Brajevic et al. [8]	BCO	0.012667	0.012700	N/A	2.4E−7	240,000
Ray and Liew [54]	SCA	0.012669	0.012923	0.016717	5.92E−4	25,167
Wang et al. [61]	AATM	0.012668	0.012708	0.012861	4.50E−5	25,000
Present study	ISA	0.012665	0.013165	0.012799	1.59E−2	8000

[a] SA-SD is the simulated annealing and direct search.

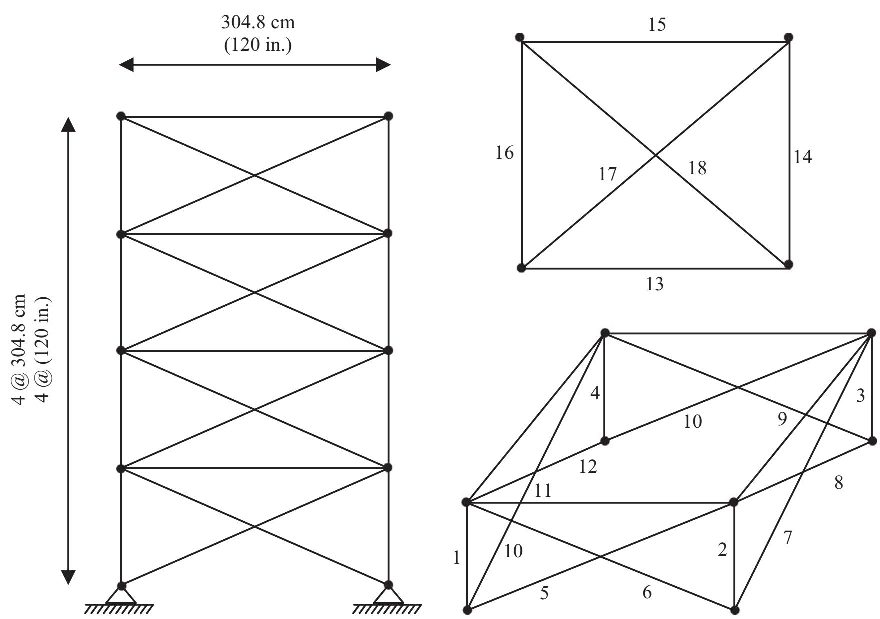

Fig. 11. Schematic representation of the 72 bar space truss design problem.

Table 10
Two loading conditions for the 72 bar space truss design problem.

Node no.	Case I, kN (kips)			Case II, kN (kips)		
	P_X	P_Y	P_Z	P_X	P_Y	P_Z
17	22.25 (5)	22.25 (5)	−22.25 (−5)	0	0	−22.25 (−5)
18	0	0	0	0	0	−22.25 (−5)
19	0	0	0	0	0	−22.25 (−5)
20	0	0	0	0	0	−22.25 (−5)

Table 11
Statistical results of ISA for the 72 bar space truss design problem.

Best	Mean	Median	Worst	Std. dev.	No. elem.	No. iter.
174.8792	178.6296	177.9208	186.5268	2.8885	50	200

is clear that all 30 runs found the variables near the variables obtained for the best solution.

This discrete design problem is first solved by Wu and Chow [65] using GA. Then it is solved by Lee et al. [38] using the HS method with five different parameter configurations. Recently, this problem is also solved by Li et al. [40,41] using a group search optimizer (GSO) and three different PSO algorithms including standard PSO, PSO with passive congregation (PSOPC) and heuristic/harmony PSO (HPSO) which is a hybrid of PSO and HS. The best results in the literature and the best results obtained in this study are presented in Table 12. The results obtained by the ISA are better than the results obtained by GA, HSs, GSO and PSOs. Comparing ISA and GA, the mean value of the proposed algorithm is also better than the results obtained by GA. For GSO and PSOs, only the HPSO could converge and the other algorithms get into local minima. The best result obtained by HPSO is worse than the results obtained by ISA. Most interesting part is the number of structural analysis required for the ISA which is far less than those of GA, HSs, GSO and PSOs.

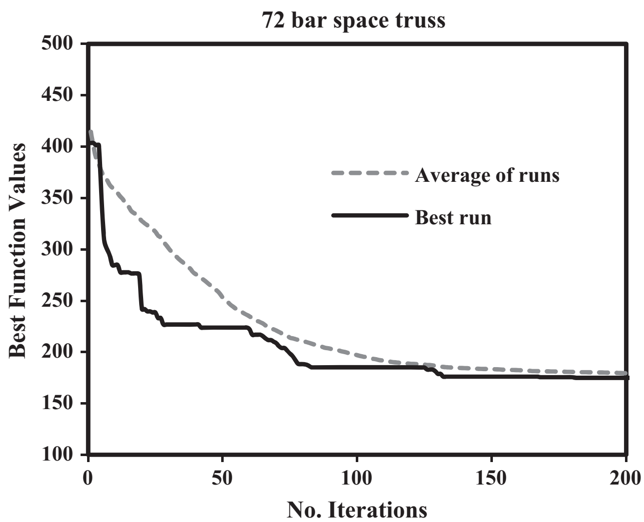

Fig. 12. Convergence history for the 72 bar space truss design problem.

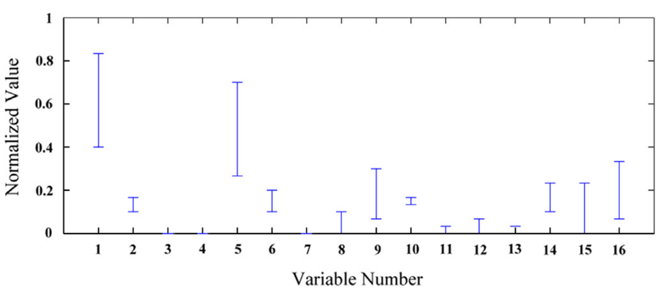

Fig. 13. Tolerances of the scaled variables.

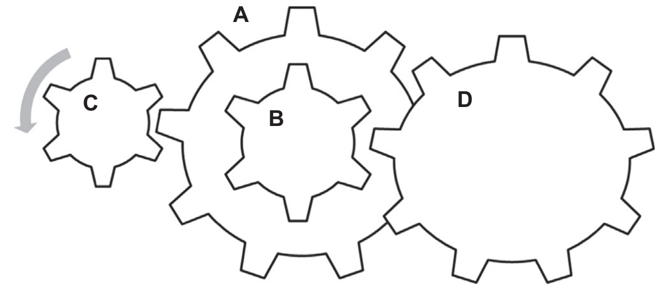

Fig. 14. Schematic representation of gear train design problem.

Table 13
Statistical results of isa for the gear train design problem.

Best	Mean	Median	Worst	Std. dev.	No. elem.	No. iter.
2.701×10^{-12}	8.50×10^{-8}	1.83×10^{-8}	8.95×10^{-7}	2.00×10^{-7}	10	20

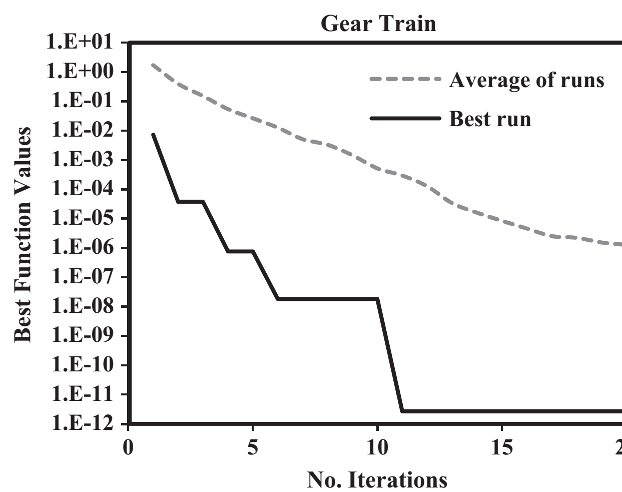

Fig. 15. Convergence history for the gear train design problem.

Table 12
Optimal design comparison for the 72 bar space truss design problem.

No.	Element group	Wu and Chow [65]	Lee et al. [38]					Li et al. [41]	Li et al. [40]			Present study
		GA	HS-I	HS-II	HS-III	HS-IV	HS-V	GSO	PSO	PSOPC	HPSO	ISA
1	A_1–A_4	96.774	103.226	129.032	122.580	116.129	109.677	193.548	167.742	193.548	135.484	122.58
2	A_5–A_{12}	45.161	32.258	32.258	32.258	32.258	32.258	96.774	96.774	90.322	38.710	32.26
3	A_{13}–A_{16}	6.452	6.452	6.452	6.452	6.452	6.452	6.4516	19.355	12.903	6.452	6.452
4	A_{17}–A_{18}	6.452	6.452	6.452	6.452	6.452	6.452	6.4516	6.452	6.452	6.452	6.452
5	A_{19}–A_{22}	83.871	83.871	83.871	90.322	83.871	83.871	167.742	135.484	174.193	90.322	90.32
6	A_{23}–A_{30}	32.258	38.710	32.258	38.710	38.710	38.710	96.774	96.774	122.580	32.258	32.26
7	A_{31}–A_{34}	12.903	6.452	6.452	6.452	6.452	6.452	6.452	38.710	45.161	6.452	6.452
8	A_{35}–A_{36}	6.452	6.452	6.452	6.452	6.452	6.452	6.452	19.355	51.613	6.452	6.452
9	A_{37}–A_{40}	32.258	38.710	32.258	38.710	25.806	32.258	103.226	141.935	90.322	32.258	32.26
10	A_{41}–A_{48}	32.258	38.710	38.710	32.258	38.710	38.710	90.322	122.580	77.419	32.258	32.26
11	A_{49}–A_{52}	6.452	6.452	6.452	6.452	6.452	6.452	6.452	12.903	51.613	6.452	6.452
12	A_{53}–A_{54}	12.903	6.452	6.452	6.452	6.452	6.452	25.806	58.064	6.452	6.452	6.452
13	A_{55}–A_{58}	12.903	12.903	12.903	12.903	12.903	12.903	25.806	25.806	25.806	12.903	12.903
14	A_{59}–A_{66}	32.258	32.258	32.258	32.258	32.258	32.258	103.226	122.580	122.580	32.258	38.71
15	A_{67}–A_{70}	32.258	25.806	25.806	25.806	25.806	25.806	83.871	45.161	58.064	19.355	25.81
16	A_{71}–A_{72}	45.161	38.710	45.161	38.710	38.710	38.710	83.871	103.226	83.871	45.161	38.71
Weight (kg)		181.72	176.47	176.40	175.95	176.47	176.47	438.89	494.32	485.21	176.41	174.879
Weight (lb)		400.66	389.08	388.94	387.94	389.08	389.08	967.68	1089.88	1069.79	388.94	385.581
No. eval.		N/A	16,347	27,846	16,044	13,779	27,113	N/A	N/A	N/A	N/A	8950
Max. no. eval.		60,000	30,000	30,000	30,000	30,000	30,000	50,000	50,000	50,000	50,000	10,000

Table 14
Optimal design comparison for the gear train design problem.

Ref.	Method	T_a	T_b	T_c	T_d	*Gear ratio*	f_{min}	Max. eval.
Sandgran [58]	MP[a]	45	22	18	60	0.1467	5.712×10^{-6}	N/A
Kannan and Kramer [33]	ALM[b]	33	15	13	41	0.1441	2.146×10^{-8}	N/A
Loh and Papalambros [43]	SL[c]	42	16	19	50	0.1447	0.23×10^{-6}	N/A
Fu et al. [21]	MP	47	29	14	59	0.1464	4.5×10^{-6}	N/A
Zhang and Wang [70]	SA	52	15	30	60	0.1442	2.36×10^{-9}	N/A
Wu and Chow [64]	GA	N.A	N.A	N.A	N.A	N.A	2.33×10^{-7}	10,000
Cao and Wu [10]	EP[d]	52	15	30	60	0.1442	2.36×10^{-9}	N/A
Deb and Goyal [17]	GA	33	14	17	50	0.1442	1.362×10^{-9}	N/A
Litinetski and Abramzon [42]	RS[e]	49	16	19	43	0.1442	2.701×10^{-12}	5100
Parsopoulos and Vrahatis [50]	PSO	43	16	19	49	0.1442	2.701×10^{-12}	100,000
Yun [68]	FLGA[f]	43	19	16	49	0.1442	2.701×10^{-12}	100,000
Gandomi et al. [26]	[g]CPSO	49	19	16	43	0.1442	2.701×10^{-12}	2000
Present study	ISA	43	16	19	49	0.1442	2.701×10^{-12}	200

[a] MP is the mathematical programming.
[b] ALM is the augmented Lagrange multiplier.
[c] SL is the sequential linearization.
[d] EP is the evolutionary programming.
[e] RS is the random search.
[f] FLGA is GA with the fuzzy logic controller.
[g] CPSO is the chaotic PSO.

5.2.2. *Gear train design*

Gear train design problem is a discrete optimization problem and initially introduced by Sandgran [58]. The gear train design problem is cost optimization of the gear ratio of a compound gear train and the ratio can be defined as

$$Gear\ \ ratio=\frac{\text{angular velocity of the output shaft}}{\text{angular velocity of the input shaft}} \tag{29}$$

As it is shown in Fig. 14, the current gear design problem has four gears and the error between a required gear ratio (1/6.931) and an obtained gear ratio shown to be minimized. Therefore it is the discrete problem with error reduction as the objective function. The problem variables are of teeth of for each gearwheel (T_i). The number teeth are integer values between 12 and 60. Therefore, the number of possible solutions for the gear train design problem is $49^4 \approx 5.8\times10^6$.

The statistical results of the ISA for this problem are presented in Table 13. The convergence history of the results is shown in Fig. 15. The best solution of ISA is obtained after only 100 searches with the objective value of 2.7009×10^{-12}. Table 14 shows the comparison of the best results of different algorithms for this problem. As it can be seen from the table, the best result obtained in this study is same as the results obtained by Litinetski and Abramzon [42], Parsopoulos and Vrahatis [51], Yun [68] and Gandomi et al. [26], and it is better than the other studies presented in Table 14. Although, these four studies obtained the same results, the numbers of searches they used (see Table 14) are far more than the number of evaluations used here.

6. Summary and conclusion

In the current study, the ISA is presented as a new paradigm for global optimization. The ISA is inspired by interior design and decoration. The derivative information is not necessary in the ISA since it uses a stochastic random search instead of a gradient search. The proposed algorithm is very simple and it has only one parameter, α, to tune which makes it adaptive to a wider class of optimization problems.

The validity of the ISA method is verified using several benchmark unconstrained and constrained problems. According to the results, the proposed algorithm can outperform the well-known DE and PSO algorithms for unconstrained benchmark problems. Although the convergency of the algorithm is evaluated here using convergency plots and convergency rate, it is advantageous to find a mathematical approach to demonstrate the convergency. Based on the results, the proposed algorithm is also very effective for constrained engineering problems. However, this new optimization algorithm could be improved by finding the most suitable constraint handling method.

An interesting feature of the proposed ISA is that it has an excellent convergence in comparison with other algorithms. This method is very encouraging for its future application to optimization tasks. Although the basic ISA is so effective, further improvements in designing of the ISA are possible such as improvements for multi-objective, non-convex and high-dimensional optimization problems. Future research in this context may focus on evaluating and tuning α, and its application into real-world and unsolved problems.

Appendix A

The evolutionary boundary constraint handling used here is formulated as [24]

$$f(z_i \to x_i)=\begin{cases}\alpha\times lb_i+(1-\alpha)x_i^b & \text{if} \quad z_i<lb_i\\ \beta\times ub_i+(1-\beta)x_i^b & \text{if} \quad z_i>ub_i\end{cases} \tag{A.1}$$

where α and β are random numbers between 0 and 1, and x_i^b is the related component of the global best solution.

Appendix B

The constraint handling is based on the following rules [5]:

- If both solutions are feasible, the individual with the better objective function value is better.
- A feasible solution is better than an infeasible one.
- If both solutions are infeasible, the individual with fewer amounts of constraint violation is better.

The amount of constraint violation is calculated using the normalized constraints as

$$Violation\ (x)=\sum_{i=1}^{\text{No. C}}\frac{g_i(x)}{g_{\max\ i}} \tag{A.2}$$

where No. C is the number of constraints, $g_i(x)$ are the ith constraint of the problem, and $g_{\max\ i}$ is the largest violation of the ith constraint so far.

References

[1] Akhtar S, Tai K, Ray T. A socio-behavioral simulation model for engineering design optimization. Eng Optim 2002;34(4):341–54.
[2] Ali MM, Khompatraporn C, Zabinsky ZB. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J Global Optim 2005;31:635–72.
[3] Aragon VS, Esquivel SC, Coello Coello CA. A modified version of a T-Cell algorithm for constrained optimization problems. Int J Numer Methods Eng 2010;84:351–78.
[4] Barbosa HJC, Lemonge ACC. An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In: Langdon WB, Cantu-Paz E, Mathias K, Ro R, Davis D, Poli R, et al., editors. Proceedings of the genetic and evolutionary computation conference (GECCO'2002). San Francisco, Los Altos, CA: Morgan Kaufmann; 2002. p. 287–94.
[5] Becerra RL, Coello Coello CA. Cultured differential evolution for constrained optimization. Comput Methods Appl Mech Eng 2006;195:4303–22.
[6] Bernardino, HS, Barbosa, HJC, Lemonge, ACC. A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In: Proceedings of the congress on evolutionary computation (CEC 2007); 2007. pp. 646–53.
[7] Bernardino HS, Barbosa HJC, Lemonge ACC, Fonseca LG. A new hybrid AIS-GA for constrained optimization problems in mechanical engineering. In: Proceedings of the congress on evolutionary computation (CEC'2008), Hong Kong. Piscataway, NJ, USA: IEEE Service Center; 2008. p. 1455–62.
[8] Brajevic I, Tuba M, Subotic M. Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. Int J Math Comput Simul 2011;5(2):135–43.
[9] Canyurt OE, Hajela P. Cellular genetic algorithm technique for the multicriterion design optimization. Struct Multidiscip Optim 2010;40(1–6):201–14.
[10] Cao YJ, Wu QH. Mechanical design optimization by mixed-variable evolutionary programming. IEEE conference on evoluionary computation. Indianapolis, IN: IEEE Press; p. 443–6.
[11] Coello Coello CA, Cortes NC. Hybridizing a genetic algorithm with an artificial immune system for global optimization. Eng Optim 2004;36(5):607–34.
[12] Coello Coello CA, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 2002;16:193–203.
[13] Coello Coello CA. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 2000;41(2):113–27.
[14] Coello Coello CA. 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.
[15] Deb K. Optimal design of a welded beam via genetic algorithms. AIAA J 1991;29(11):2013–5.
[16] Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 2000;186:311–38.
[17] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inf 1996:30–4526 1996:30–45.
[18] 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.
[19] 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.
[20] Fogel LJ, Owens AJ, Walsh MJ. Artificial intelligence thorough simulated evolution. Chichester, UK: John Wiley & Sons, Ltd; 1966.
[21] Fu JF, Fenton RG, Cleghorn WL. A mixed integer-discrete-continuous programming method and its application to engineering design optimization. Eng Optim 1991;17:263–80.
[22] Gandomi AH, Alavi AH. Multi-stage genetic programming: a new strategy to nonlinear system modeling. Inf Sci 2011;181:5227–39.
[23] Gandomi AH, Yang XS, Talatahari S, Deb S. Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. Comput Math Appl 2012;63(1):191–200.
[24] Gandomi AH, Yang XS. Evolutionary boundary constraint handling scheme. Neural Comput Appl 2012;21(6):1449–62.
[25] Gandomi AH, Alavi AH. Krill Herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 2012;17(12):4831–45.
[26] Gandomi AH, Yun GJ, Yang XS, Talatahari S. Chaos-enhanced accelerated particle swarm algorithm. Commun Nonlinear Sci Numer Simul 2013;18(2):327–40.
[27] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.
[28] He Q, Wang L. An effective co-evolutionary particle swarm optimization for engineering optimization problems. Eng Appl Artif Intel 2006;20:89–99.
[29] Hedar A, Fukushima M. Derivative-free filter simulated annealing method for constrained continuous global optimization. J Global Optim 2006;35:521–49.
[30] Holland J. Adaptation in natural and artificial systems. Ann Anbor: University of Michigan Press; 1975.
[31] Huang FZ, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 2007;186(1):340–56.
[32] Jaberipour M, Khorram E. Two improved harmony search algorithms for solving engineering optimization problems. Commun Nonlinear Sci Numer Simulat 2010;15(11):3316–31.
[33] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 1994;116:318–20.
[34] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–89.
[35] Kitayama S, Arakawa M, Yamazaki K. Differential evolution as the global optimization technique and its application to structural optimization. Appl Soft Comput 2011;11(386):3792–803.
[36] Koza JR. Genetic programming: on the programming of computers by natural selection. Cambridge, MA: MIT Press; 1992.
[37] Gandomi AH, Yang XS. Benchmark Problems in Structural Optimization. In: Koziel S, Yang XS, editors. Computational Optimization, Methods and Algorithms. Berlin, Germany: Springer; SCI, 356; 2011. p. 259–81 (Chapter 12).
[38] Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search heuristic algorithm for discrete structural optimization. Eng Optim 2005;37(7):663–84.
[39] Lemonge ACC, Barbosa HJC. An adaptive penalty scheme for genetic algorithms in structural optimization. Int J Numer Methods Eng 2004;59(5):703–36.
[40] Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 2009;87(7 and 8):435–43.
[41] Li LJ, Liu F, Xu XT, Liu F. The group search optimizer and its application to truss structure design. Adv Struct Eng 2010;13(1):43–51.
[42] Litinetski VV, Abramzon BM. Mars—a multistart adaptive random search method for global constrained optimization in engineering applications. Eng Optim 1998;30(2):125–54.
[43] Loh HT, Papalambros PY. A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems. J Mech Des 1991;113:325–34.
[44] Lucic, P, Teodorovic, D. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV triennial symposium on transportation analysis, Sao Miguel, Azores Islands; 2001. p. 441–5.
[45] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Appl Math Comput 2007;188:1567–79.
[46] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. Evolut Comput 1996;4(1):1–32.
[47] Montes EM, Coello Coello CA. Constrained optimization via multiobjective evolutionary algorithms. In: Knowles J, Corne D, Deb K, editors. Multiobjective problem solving from nature. Springer; 2008. p. 53–75.
[48] Montes, EM, Ocaña, BH. Bacterial foraging for engineering design problems: preliminary results. In: Proceedings of the fourth Mexican conference on evolutionary computation (COMCEV'2008), CIMAT, Guanajuato, México, October 2008. p. 33–8.
[49] NCIDQ 2011. ⟨http://www.ncidq.org⟩ Retrieved Jan. 20, 2011.
[50] Parsopoulos, KE, Vrahatis, MN. Unified particle swarm optimization for solving constrained engineering optimization problems. In: Lecture notes in computer science (LNCS), vol. 3612; 2005. p. 582–91.
[51] Parsopoulos KE, Vrahatis MN. Unified particle swarm optimization for solving constrained engineering optimization problems. In: Wang L, Chen K, Ong YS, editors. Advances in natural computation. Berlin, Heidelberg: Springer-Verlag; 2005. p. 582–91.
[52] Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 2002;22:53–67.
[53] Rahnamayan S, Tizhoosh HR, MMA Salama. A novel population initialization method for accelerating evolutionary algorithms. Comput Math Appl 2007;53:1605–14.
[54] Ray T, Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 2003;7(4):386–96.
[55] Renato AK, Dos Santos CL. Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 2006;36:1407–16.
[56] Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 2000;4(3):284–94.
[57] Sahab MG, Toropov VV, Gandomi AH. Traditional and modern structural optimization techniques—theory and application. In: Gandomi AH, Yang XY, Talatahari S, Alavi AH, editors. Metaheuristic applications in structures and infrastructures. Waltham, MA: Elsevier; 2013 (Chapter 2).
[58] Sandgran E. Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 1990;112(2):223–9.
[59] Storn R, Price KV. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59.
[60] Talbi EG. Metaheuristics: from design to implementation. Hoboken, NJ, USA: John Wiley & Sons; 2009.
[61] Wang Y, Cai Z, Zhou Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. Int J Numer Meth Eng 2009;77:1501–34.
[62] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[63] Wu P, Gao L, Zou D, Li S. An improved particle swarm optimization algorithm for reliability problems. ISA Trans 2011;50(1):71–81.

[64] Wu SJ, Chow PT. Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 1995;56(6):979–91.
[65] Wu SJ, Chow PT. Genetic algorithms for nonlinear mixed discrete integer optimization problems via meta-genetic parameter optimization. Eng Optim 1995;24:137–59.
[66] Yang, XS, Deb, S. Cuckoo search via Lévy flights. In: Proceedings of world congress on nature & biologically inspired computing (NaBIC 2009, India). USA: IEEE Publications; 2009. p. 210–4.
[67] Yang XS. Nature-inspired metaheuristic algorithms. 2nd ed. Frome, UK: Luniver Press; 2010.
[68] Yun YS. Study on adaptive hybrid genetic algorithm and its applications to engineering design problems [M.Sc. thesis]. Waseda University; 2005.
[69] Yüzgeç U, Türker M, Hocalar A. On-line evolutionary optimization of an industrial fed-batch yeast fermentation process. ISA Trans 2009;48(1):79–92.
[70] Zhang C, Wang HP. Mixed-discrete nonlinear optimization with simulated annealing. Eng Optim 1993;21:277–91.