INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2020; 10(1):155-180

IMPERIALIST COMPETITIVE LEARNER-BASED OPTIMIZATION: A HYBRID METHOD TO SOLVE ENGINEERING PROBLEMS

M. Shahrouzi[*, †] and A. Salehi
Civil Engineering Department, Faculty of Engineering, Kharazmi University, Tehran, Iran

ABSTRACT

Imperialist Competitive Algorithm, ICA is a meta-heuristic which simulates collapse of weak empires by more powerful ones that take possession of their colonies. In order to enhance performance, ICA is hybridized with proper features of *Teaching-Learning-Based Optimization*, TLBO. In addition, ICA walks are modified with an extra term to intensify looking for the global best solution. The number of control parameters and consequent tuning effort has been reduced in the proposed *Imperialist Competitive Learner-Based Optimization,* ICLBO with respect to ICA and several other methods. Efficiency and effectiveness of ICLBO is further evaluated treating a number of test functions in addition to continuous and discrete engineering problems. It is discussed and traced that balancing between exploration and exploitation is enhanced due to the proposed hybridization. Numerical results exhibit superior performance of ICLBO vs. ICA and a variety of other well-known meta-heuristics.

Keywords: hybrid optimization method; imperialist competitive algorithm; teaching-learning-based optimization; parameter reduction.

Received: 2 October 2019; Accepted: 29 December 2019

1. INTRODUCTION

Optimization is the process of searching for a vector in a design domain that makes the best solution among a large number of possible feasible alternatives. Emerging with several fields of science and engineering, it is a challenging task to choose and tune the right algorithm for a specific problem. As indicated in the no-free-lunch theorem, a specifically

[*]Corresponding author: Civil Engineering Department, Faculty of Engineering, Kharazmi University, Tehran, Iran
[†]E-mail address: shahruzi@khu.ac.ir (M. Shahrouzi)

tuned algorithm is not the best for all the problems [1]. The matter declares why new methods are still being developed after a century of research in optimization. The choice of an algorithm may depend on the type of design variables, objective function and its constraints.

Meta-heuristic algorithms constitute a well-known branch of stochastic search; that mostly apply zero-order function sampling. They can be classified into different categories based on the source of inspiration. The main category is the biology-inspired algorithms, which generally use biological evolution and/or collective behavior of animals as their models [2]. Natural lows in science form another source of inspiration; they include physics and chemistry based algorithms. Moreover, art-inspired algorithms have been successful for global optimization. These are generally inspired by the creative behavior of artists such as musicians and architects. Social behavior is another source of inspiration that a category of meta-heuristic methods apply in order to solve optimization problems.

Meta-heuristic algorithms are well capable to extract information from a set of solutions and approach the global optimum of practical problems in reasonable time. During the 1960s, a pioneering category of optimization methods were highlighted within the Genetic Algorithm (GA) [3] by idealizing the evolution theory. Since then, many other meta-heuristics have emerged, such as Differential Evolution (DE) [4], Particle Swarm Optimization (PSO) [5], Harmony Search (HS) [6–8], Biogeography-Based Optimization (BBO) [9], Colliding Bodies Optimization (CBO) [10], Teaching–Learning-Based Optimization (TLBO) [11], Stochastic Directional Search[12], Interior Search Algorithm (ISA) [13], Symbiotic Organisms Search (SOS) [14], Water Evaporation Optimization [15], Opposition-Switching Search [16], Vibrating Particles Search[17] and Dragonfly Algorithm [18].

It has been of research interest to develop new methods with improvements in terms of computational and time complexity [19–21]. A well-experienced approach to achieve such a goal is to hybridize powerful features of distinct algorithms for a set of problems in hand within a new framework [22,23]. The present work concerns improvement of ICA as a widely used meta-heuristic in a variety of problems [24,25]. ICA is further enhanced via hybridization by some features of DE and TLBO to develop a new algorithm called *Imperialist Competitive Learning-Based Optimization;* ICLBO.

Remainder of this article is organized as follows. Section 2 describes ICA and TLBO in brief. Theoretical basis and algorithm of ICLBO are presented in Section 3. Consequently, performance of the proposed algorithm is evaluated by several benchmarks in Section 4 via comparison with the other methods. Finally, the present study is concluded in Section 5.

2. PRILIMINARY

Imperialist Competition or Colonial Competitive Algorithm was introduced by Atashpaz-Gargari and Lucas [24], and it has been successfully applied to a variety of engineering problems [26–31]. However, due to considerable number of control parameters, ICA requires computational effort to tune them for each specific problem. In contrary, reduced number of parameters is of practical interest to avoid burdensome tuning efforts [11,32]. ICA and TLBO are briefly reviewed in this section prior to hybridization within framework of the newly developed ICLBO.

2.1 Imperialist competitive algorithm

Imperialism is the policy of extending the power and rule of a government beyond its own boundaries. A country may attempt to dominate others by direct rule or by less obvious means such as a control of markets for good or raw material. In its initial forms, imperialism was just a political control over other countries in order to use their resources. In some cases, the reason for controlling another country is just preventing the opponent imperialist form taking possession of it. Such a social process is numerically simulated via ICA as introduced by [24].

According to ICA, any candidate solution vector in the design space is analogous to a country in the world. Countries are distinguished in two types based on the power score of them; that is colonies and imperialists. An empire consists of an imperialist together with its possessed colonies. In another word, some of the best countries in the population are selected as imperialists and the rest form colonies of such imperialists.

ICA starts with an initial population of countries forming a prescribed number of empires. All colonies of initial population are divided among the aforementioned imperialists based on their power. After assigning each colony to an imperialist, it starts moving toward its relevant imperialist. The total power of an empire depends on both the power of the imperialist and that of its associated colonies.

Afterwards, imperialistic competition takes places resulting in a gradual improvement in total score of more powerful empires meanwhile decreasing score of the weaker ones. The weakest empire iteratively loses its power until it finally collapse. Such a process is repeated with the remaining empires. The movement of colonies toward their relevant imperialists along with competition among empires and also the collapse mechanism will hopefully cause all the countries to converge into one empire possessing all the other countries as its colonies. The imperialist country in such a final empire is announced as the optimal solution. Procedure of the aforementioned ICA in a fitness maximization form can be summarized as follows:

Step 1: Initialize N_{imp} number of empires over randomly positioned N_{pop} countries after evaluating their fitness or cost function.

Step 2: In each empire, move every colony toward its relevant imperialist by:

$$X_{New} = X_{Col} + (\beta \times rand - 1) \times X_{imp} \tag{1}$$

where β is a control parameter greater than 1, N_{imp} stands for the empire position. X_{col} and X_{new} are the current and new positions of the colony, respectively. The function *rand* generates uniform distributed random numbers between 0 and 1.

Step 3: If there is a colony with lower cost (fitter) than the imperialist, exchange their position within the corresponding empire.

Step 4: Compute the total cost and power for each empire by the following relations:

$$T_{Imp} = Fitness(X_{imp}) + k\frac{\sum_{i=1}^{N_{imp}} Fitness(X_{imp})}{N_{imp}} \tag{2}$$

$$NT_{Imp} = T_{Imp} - \min(T_{Imp}) \tag{3}$$

$$P_{Imp} = \left| \frac{NT_{Imp}}{\sum_{i=1}^{N_{imp}} NT_{Imp}} \right| \tag{4}$$

where k is a positive number less than 1, N_{imp} denotes the number of empires, P_{imp} is the power of each empires, T_{imp} and NT_{imp} are the total fitness and normalized total fitness of each empires, respectively.

Step 5: Transfer the worst colony from the weakest empire it into the empire that has the most likelihood based on P_{imp} to possess it.

Step 6: Eliminate the powerless empires.

Step 7: Loop from Step 2 until termination criterion is satisfied; that is completing a prescribed number of function evaluations NFE_{max} .

According to the above relations, ICA has control parameters of NFE_{max} , β , N_{imp} , k and N_{pop}.

2.2 Teaching learning based optimization

Teaching–Learning-Based Optimization (TLBO) is of research interest among other parameter-less algorithms due to its simplicity and efficiency [33–36]. It has two distinct phases of position updating for any search agent known as a classmate. They are the teacher phase and the learner phase. The teacher phase simulates upgrading the mean grades of the classmates by means of their best; called the teacher. That is while the classmates interact each other in the learner phase.

Like many other meta-heuristics TLBO provides some diversification and intensification operators. Not only the initial population but also every pair of classmates in the learner phase are selected randomly to provide the diversification. In contrary, averaging the already found information of classmates and moving toward their best constitute intensification in the teacher phase. It is further improved by selection of the fittest among each pair of solutions in the learner phase. A standard algorithm of TLBO can be concerned within the following steps:

Step 1: Initialization; randomly generate a prescribed number of classmates within the design space and identify their best as the teacher after evaluation of such a population.

Step 2: Repeat for the teacher phase and the learner phase for a prescribed number of iterations $Iter_{max}$ and each of the classmates as follows:

Step 3: Teacher phase:

- For every classmate, X, generate a candidate new position; X_{new} by:

$$X_{New} = X + rand \times (X_{Teacher} - Tf.X_{Mean}) \quad (5)$$

where Tf is an integer scale which randomly switches between 1 and 2.
- Evaluate X_{new}
- Replace X with X_{new} if X_{new} is better than it.

Step 4: Learner phase:
- Randomly select two distinct classmates; X_i and X_j
- If X_j is better than X_i then s=+1 otherwise s=-1 . Then generate X_{new} by:

$$X_{New} = X + rand \times s \times (X_j - X_i) \quad (6)$$

- Evaluate X_{new}
- Replace X with X_{new} if X_{new} is better than it.

Step 5: Termination; as soon as NFE_{max} function calls is completed exit the loop and announce the updated teacher as the optimal solution.

3. THE PROPOSED HYBRID ALGORITHM

ICA can be distinguished from several other meta-heuristics due to its special way of employing dynamic-size subpopulations and then collapsing them into one empire. In another word, it provides delayed transfer from distributed search of the design space via these subpopulations toward the global search within the final empire. Having several control parameters enables fine tuning of the algorithm, however, in charge of extra computational effort that is a practical challenge.

In contrary, TLBO has no parameters rather than population size and NFE_{max} as the least critical ones. Such a feature is of practical interest, however, may avoid TLBO from achieving the best search refinement in various problems.

Here, some operators of ICA and TLBO are properly hybridized to take advantageous of both the methods in developing a more powerful algorithm; that is called Imperialist Competitive Learner-Based Optimization, ICLBO. This algorithm is introduced via the following steps:

Step 1: Initialization; generate N_{pop} randomly positioned countries and distribute them via $N_{imp}= N_{pop}/5$ empires.

For each empire do:

Step 2: Walking Phase:
- evaluate fitness or cost of countries and identify their best as imperialist of that empire
- move any colony except the imperialist by the following relation:

$$X_{New} = X + rand \times (X_{imp} - X) + rand \times C \times (X_{Teacher} - X) \quad (7)$$

where $X_{Teacher}$ stands for the best country among all the empires while X_{imp} denotes the best of the current empire. The scaling factor, C, linearly varies from 1 to 0 via iterations as

$$C = 1 - (Iter - 1)\big/(Iter_{max} - 1) \tag{8}$$

- Evaluate X_{new}
- Replace X with X_{new} if X_{new} is better than it.

Step 3: If there is a colony with lower cost (fitter) than the imperialist, exchange their position within the corresponding empire.

Step 4: Compute the total cost and power for each empire by Equations (9) to (11).

$$T_{Imp} = Fitness(X_{imp}) + \frac{1}{2}\frac{\sum_{i=1}^{N_{imp}} Fitness(X_{imp})}{N_{imp}} \tag{9}$$

$$NT_{Imp} = T_{Imp} - \max(T_{Imp}) \tag{10}$$

$$P_{Imp} = \left| \frac{NT_{Imp}}{\sum_{i=1}^{N_{imp}} NT_{Imp}} \right| \tag{11}$$

Step 5: Learner phase:

- Randomly select two distinct classmates; X_i and X_j
- If X_j is better than X_i then s=+1 otherwise s=-1 . Then generate X_{new} by:

$$X_{New} = X + rand \times s \times (X_j - X_i) \tag{12}$$

- Evaluate X_{new}
- Replace X with X_{new} if X_{new} is better than it.

End for

Step 6: Transfer the worst colony from the weakest empire it into the empire that has the most likelihood based on P_{imp} to possess it.

Step 7: Eliminate the powerless empires.

Step 8: Loop from Step 2 until termination criterion is satisfied; that is reaching a prescribed number of iterations; $Iter_{max}$ or a prescribed number of function evaluations NFE_{max}.

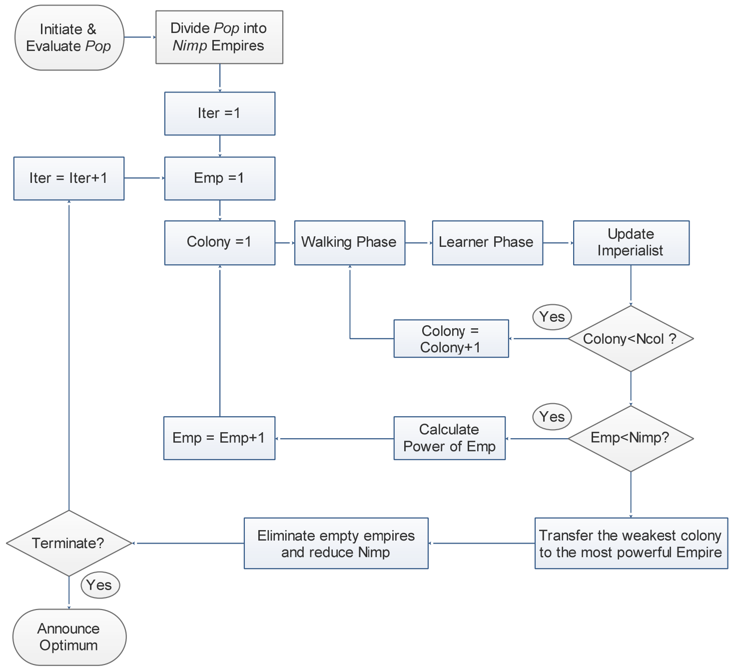


Figure 1. Flowchart of the proposed ICLBO

As can be realized, ICLBO hybridizes a local search (walking toward the imperialist in each empire) with a global search (moving toward the global best among all of the empires). Besides, a self-tuning strategy is utilized by variation of scaling factor in Eq.8 to control balance of such local and global search walks. Embedding the learner phase as another hybridization strategy is expected to provide additional search refinement; it is further evaluated via numerical tests. As an interesting feature in the proposed ICLBO, the control parameters have practically been reduced to NFE_{max} and N_{pop} ; just as TLBO. Fig.1 reveals flowchart of ICLBO.

4. NUMERICAL SIMULATION

In this section, the proposed method was tested through an array of experiments. Numerical simulation is performed via two parts. The first part, deals with benchmark test functions while in the second part, constrained engineering optimization problems are treated.

4.1 Benchmark mathematical functions

The employed test functions to be minimized are concerned within three general categories; i.e. unimodal, multimodal and fixed-dimension multimodal functions. These benchmark functions are described in Tables 1, 2 and 3.

Table 1: Unimodal benchmark functions

Function	Dim	Range
$f_1(x)=(x_1-1)^2+\sum_{i=2}^{d} i(2x_1^2-x_{i-1})^2$	5	[-10,10]
$f_2(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	5	[-5,10]
$f_3(x)=(\sum_{i=1}^{d} x_i^2)^2$	5	[-100,100]
$f_4(x)=\sum_{i=1}^{d}\left\|x_i^5-3x_i^4+4x_i^3+2x_i^2-10x_i-4\right\|$	5	[-10,10]
$f_5(x)=\sum_{i=1}^{d} x_i^4$	5	[-10,10]

In all experiments, N_{pop} and *NFE* are set to 50 and 1000, respectively. Extra control parameters of each algorithm are given in Table 4. To achieve more reliable results, 30 independent runs are performed by every method for each benchmark problem. Comparison of performance is made between PSO, GA, ICA, DE, CBO, TLBO, SOS and the proposed ICLBO; whereas the last five are treated as parameter-less methods.

Table 2: Multimodal benchmark functions

Function	Dim	Range
$f_6(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)-20+\mathbf{e}$	30	[-32.7,32.7]
$f_7(x)=\sum_{i=1}^{d}\frac{x_i^2}{4000}-\prod_{i=1}^{d}\cos(x_i/\sqrt{i})+1$	30	[-600,600]
$f_8(x)=-\sum_{i=1}^{d}\sin(x_i)\sin\left(\frac{ix_i^2}{\pi}\right)^{20}$	30	[0,π]
$f_9(x)=\sum_{i=1}^{d} x_i^6(2+\sin\frac{1}{x_i})$	30	[-1,1]
$f_{10}(x)=1-\cos\left(2\pi\sum_{i=1}^{d} x_i^2\right)+0.1\sqrt{\sum_{i=1}^{d} x_i^2}$	30	[-100,100]

In the test functions, constraints are limited to bounds on the design variables. They are called side-constraints [37]. Consequently, for such mathematical test functions, the problem is formulated as:

$$\begin{aligned} &Minimize \;\; f(X) \\ &\text{Subject to} \;\; L \le X \le U \end{aligned} \tag{13}$$

- During optimization, any design vector Evaluate X is forced to fall within the range described by L and U vectors as the lower and the upper bounds, respectively.

It is worth notifying that the employed variant of DE, in every iteration selects the best of every current search agent X and its moved state; X_{new} given by:

$$X_{New} = X + rand \times (X_j - X_i) + rand \times (X_{gb} - X\ ) \tag{14}$$

in which X_{gb} denotes the global best already-found solution while X_i and X_j stand for two randomly chosen members out of the current population of search agents.

Intrinsic parameters of PSO; i.e. c_1, c_2 and c_3 denote the inertial, cognitive and social factors, respectively. They arise in the following velocity updating relation according to standard PSO:

$$V_{New} = c_1 \times V + rand \times c_2 \times (X_{pb} - X\ ) + rand \times c_3 \times (X_{gb} - X\ ) \tag{15}$$

where X_{pb} introduces the previous best position of each particle with the current position X and velocity V. The new position for any such particle is thus calculated as.

$$X_{new} = X + V_{new} \tag{16}$$

A key feature in a meta-heuristic algorithm is how it controls diversity of the population during the search. To declare it, a Diversity Index, DI, is traced as defined by the following relation:

$$DI = \underset{j}{mean}\left(\frac{SD_j}{U_j - L_j}\right) \tag{17}$$

where stands for the standard deviation of the population members in the j[th] design variable. The vectors L and U denote the upper and lower bounds, respectively.

Functions F1~F5 are unimodal and they have just one global optimum. Therefore, they are employed to compare exploitation capability of the treated algorithms. In contrary, multimodal functions include several local optima whose number increases exponentially with the problem size (number of design variables). These functions can thus be used to evaluate exploration capability of the optimization methods.

Tables 5 and 6 reveal comparisons over the best and average of final results, respectively. The values are normalized to the optimal result of each function and the most desired ones for each test function are highlighted.

Table 3: Fixed-dimension multimodal benchmark functions

Function	Dim	Range
$f_{11}(x) = -\cos(x_1)\cos(x_2)\mathbf{exp}\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$	2	[-100,100]
$f_{12}(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1+i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2+i)\right)$	2	[-5.12,5.12]
$f_{13}(x) = \sin(x_1)\exp(1-\cos(x_2))^2+\cos(x_2)\exp(1-\sin(x_1))^2+(x_1+x_2)^2$	2	[-2π, 2π]
$f_{14}(x) = \sin(x_1+x_2)+(x_1-x_2)^2-\left(\frac{3}{2}\right)x_1+\left(\frac{5}{2}\right)x_2+1$	2	$-1.5 \le x_1 \le 4.0$ $-3.0 \le x_2 \le 3.0$
$f_{15}(x) = (\mathbf{exp}(-x_1)-x_2)^4+100(x_2-x_3)^6+(\tan(x_3-x_4))^4+x_1^8$	4	[-1,1]

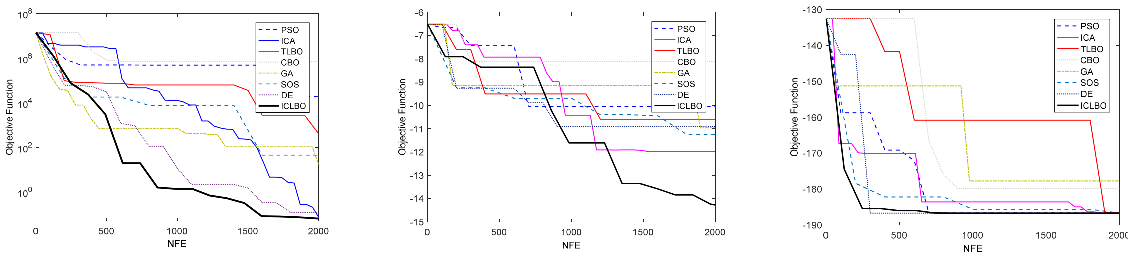
Table 4: Specific control parameters of the treated algorithms

Algorithm	Parameters	Algorithm	Parameters
PSO	$C_1=1, C_2=2, C_3=2$	**CBO**	--
GA	$P_m=0.1, b=2$	**TLBO**	--
ICA	$\beta=4, k=0.5$	**SOS**	--
DE	--	**ICBLO**	--

Table 5: The best results for the test functions by various algorithms

	PSO	GA	ICA	DE	CBO	TLBO	SOS	ICLBO
F01	9.37	3.46	3.33	1.50	604.47	17.46	4.52	**1.00**
F02	2.34	1.98	1.15	1.04	1.28	1.55	1.23	**1.00**
F03	51934.90	119.42	56.48	14.74	218488.18	3004.77	54.20	**1.00**
F04	11.56	5.14	1.71	1.37	844.66	2.76	1.46	**1.00**
F05	5406.60	30.13	40.71	6.38	238718.13	248.87	78.69	**1.00**
F06	3.04	**1.00**	1.84	2.11	2.55	1.86	2.12	1.90
F07	3.81	**1.00**	2.26	2.80	3.99	2.12	2.77	2.33
F08	1.30	1.15	1.16	1.19	1.48	1.12	1.06	**1.00**
F09	183855	148.42	9206	19272.00	**1.00**	15546.1	28444.00	13783.66
F10	2.99	1.34	2.23	2.66	**1.00**	1.87	1.33	2.00
F11	1.03	2.62	**1.00**	1.02	2.62	2.62	1.01	**1.00**
F12	1.10	**1.00**	**1.00**	**1.00**	1.13	1.02	**1.00**	**1.00**
F13	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F14	**1.00**	**1.00**	**1.00**	**1.00**	1.01	**1.00**	**1.00**	**1.00**
F15	3.40	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

According to Table 5, in majority of test functions ICLBO has achieved the first rank in capturing global optima with considerable superiority over some other treated methods. The matter is confirmed not only for the final results but also during the search by Fig. 2; that reveals the best results for F3, F8 and F12 as samples of unimodal, multimodal and fixed-dimension classes of test functions, respectively.

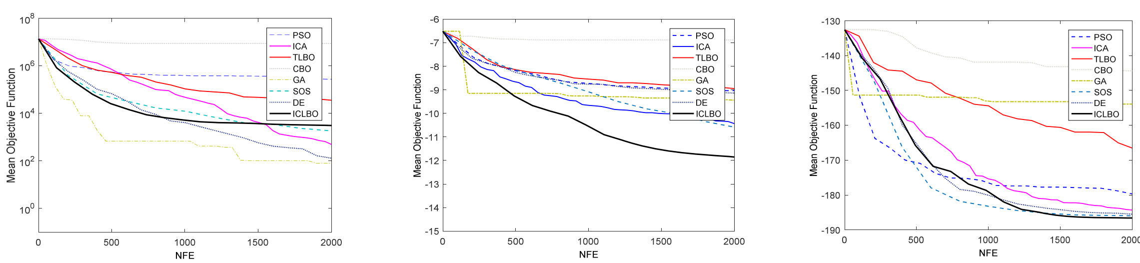


(a) (b) (c)

Figure 2. The best results of various algorithms for the functions (a) F3, (b) F8 and (c) F12

Table 6: Mean optimization results in the treated test functions

	PSO	GA	ICA	DE	CBO	TLBO	SOS	ICLBO
F01	265.66	1.01	9.00	8.58	222.18	53.07	14.06	**1.00**
F02	5.37	1.01	1.37	1.24	9.65	2.20	1.14	**1.00**
F03	1420.	**1.00**	42.51	23.01	1903.00	359.53	26.25	57.67
F04	15.11	1.58	1.44	1.26	418.50	2.31	1.40	**1.00**
F05	1140.9	**1.00**	280.23	23.73	28243.00	413.57	31.52	42.42
F06	3.22	**1.00**	2.17	2.47	3.90	2.31	2.42	2.29
F07	3.96	**1.00**	3.11	3.30	4.07	3.19	3.44	3.12
F08	1.17	1.12	1.05	1.16	1.34	1.10	1.01	**1.00**
F09	1286.4	**1.00**	17.71	68.80	1362.80	48.94	89.84	53.06
F10	2.01	**1.00**	1.49	1.67	2.33	1.52	1.37	1.40
F11	**1.00**	2.00	1.27	1.14	2.00	2.00	1.25	1.21
F12	1.34	1.16	1.11	1.04	1.61	1.37	1.01	**1.00**
F13	1.01	1.01	**1.00**	1.01	1.14	1.01	**1.00**	**1.00**
F14	**1.00**	**1.00**	**1.00**	**1.00**	1.09	**1.00**	**1.00**	**1.00**
F15	1909.2	**1.00**	**1.00**	177.98	73.60	**1.00**	188.10	355.77

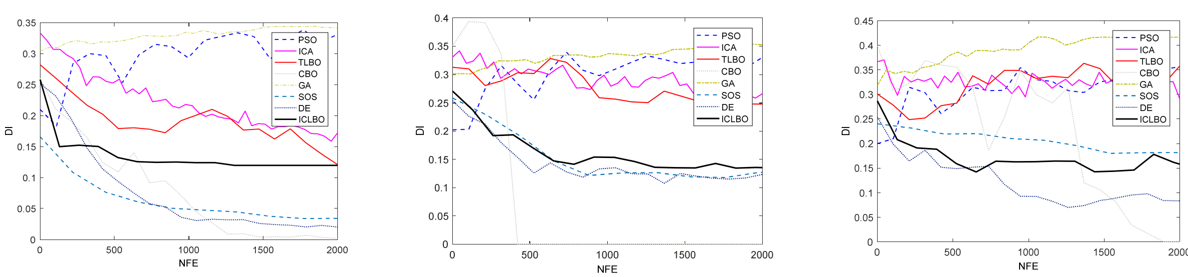


(a) (b) (c)

Figure 3. Mean results of various algorithms for the functions (a) F3, (b) F8 and (c) F12

Comparing the mean results in Table 6, it is realized that ICLBO has still the first rank treating F1, F2, F4, F8, F12~F14; i.e. 7 out of 15 functions. Although, ICLBO is not the best for the other functions; it has revealed competitive mean results among the treated methods. Fig. 3 shows that ICLBO has achieved lowest mean results for F8 and F12. GA has stood on the first rank for F3 but at the 4^{th} and 7^{th} for F8 and F12, respectively. Such an observation confirms that no single algorithm is the best for all the problems; as predicted by no-free-lunch theory.

Tracing DI history vs. NFE provides reasoning for difference in behavior of the algorithms so that some got trapped in local optima but the others had better search refinement. As evident from Fig. 4, DI of algorithms like PSO fluctuate or even increase during the search; while some others generally exhibit a decreasing trend. Comparing such DI variation with the results of Fig. 2 and Fig. 3 for the same functions; it is found that the best performance belongs to the methods which have proper balance between diversification (high DI) and intensification (vanishing DI). For example, too rapid decrease in DI has resulted to standard CBO get trapped in local optima. In contrary, fluctuating about a relatively high DI by PSO has avoided it from proper search refinement. It is while some other methods like ICLBO has maintained DI in a lower level after an early decrease in this value. The matter provides reasoning for observations in Fig. 2 and Fig. 3; that ICLBO has a high convergence rate at early iterations and continued search refinement while it progresses toward final results. In another word, it has more exploration (diversity) in the early stages of the search to widely access the design space. However, when the valley of global optimum is found, excessive divergence from it should be avoided to have better search refinement.



(a) (b) (c)

Figure 4. DI history of various algorithms for the functions (a) F3, (b) F8 and (c) F12

4.2 Constrained engineering problems

In this section, performance of ICLBO is evaluated by treating a number of constrained optimization problems that are widely used as benchmarks in literature. Definitions for the first set of engineering problems are provided in the Appendix. For the sake of true comparison, N_{pop} and NFE_{max} for majority of the treated examples are fixed to 50 and 5000, respectively. In addition, effectiveness of the proposed ICLBO is also shown via comparison with literature results for each example. Formulation of the constraint problem is converted to the following fitness maximization form:

$$\begin{aligned} &Max\ \ F(X) = -\varphi(\mathrm{X}) \\ &\text{where}\ \ \varphi(\mathrm{X}) = f(X) \times (1 + \eta \sum_k \max(0, g_k(X))) \end{aligned} \tag{18}$$

$\phi(\ )$ is the penalized objective function and $F(\ )$ stands for the fitness to be maximized. Any kth constraint is denoted by $g(\ )$and X stands for the design vector. The penalty coefficient η is set to 100 in the present study.

4.2.1 Tension/compression spring design

The tension/compression spring design problem is described by Arora for which it is aimed to minimize the weight *f(x)* of a spring subject to constraints on deflection, shear stress, surge frequency, outside diameter and geometrical design variables [37]. As depicted in Fig. 5, the design variables <x_1, x_2, x_3> are the wire diameter: d, mean coil diameter: D and the number of active coils: n, respectively.

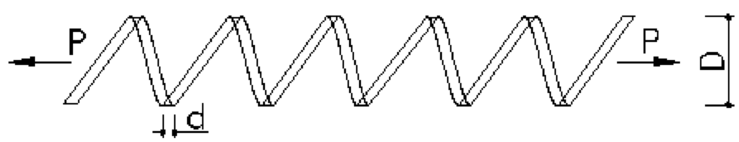


Figure 5. Tension/compression spring design problem

According to Table 7, it is evident that ICLBO has outperformed the other algorithms in the best results. Regarding the mean results, ICLBO has stood at the second rank after TLBO.

Table 7: Results of the present work for tension/compression spring design problem

	PSO	GA	ICA	DE	CBO	TLBO	SOS	ICLBO
X_1	0.0580	0.0722	0.0500	0.0708	0.0648	0.0538	0.0636	0.0515
X_2	0.5264	0.677	0.3147	0.8519	0.7557	0.4083	0.6429	0.3528
X_3	6.2919	10.553	15.000	3.7893	3.0194	9.0042	5.3580	11.5214
Best	0.0146	0.0444	0.0133	0.0247	0.0159	0.0123	0.0191	**0.0126**
Mean	0.0549	0.044	0.0690	0.1938	0.0205	0.0139	0.1486	**0.0129**

Table 8 compares the results obtained by ICLBO with those reported in literature. It is worth notifying that the proposed ICLBO has achieved the optimum *f*=0.01266* with just 5000 function evaluations while such *NFE* is 7650 for MBA to obtain the same result and is more for the other methods. HPSO has found *f*=0.01266* in charge of spending *NFE*= 81000. ICLBO has captured such a global optimum by *NFE* =11000; that shows its superior efficiency over the others.

Table 8: Comparison of ICLBO results with literature works for tension/compression spring

	MBA [38]	**HPSO** [39]	**WOA** [40]	**GSA** [41]	**ICLBO**
X_1	0.0516	0.0517	0.0512	0.0502	0.0515
X_2	0.3559	0.3571	0.3452	0.3236	0.3528
X_3	11.3446	11.2650	12.0040	13.5254	11.5214
Best	0.01266	0.01267	0.01267	0.01270	0.0126

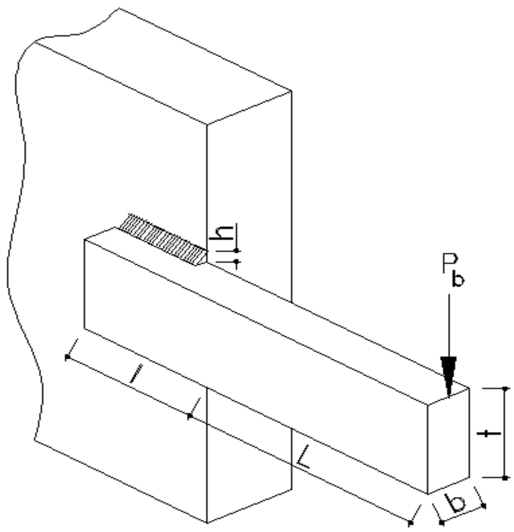


Figure 6. Welded beam design problem

4.2.2 Welded-beam design

This problem is an engineering benchmark introduced by Coello [42]; in which, a welded beam is designed for minimum cost subjected to constraints on maximal shear stress (τ), bending stress (σ) in the beam, buckling load on the bar (P_b), deflection of the beam end (δ) and side constraints. Four continuous design variables are considered for this problem as shown in Fig.6; including $x_1=h$, $x_2=l$, $x_3=t$ and $x_4=b$.

Table 9: Results of the present work for welded beam design problem

	PSO	**GA**	**ICA**	**DE**	**CBO**	**TLBO**	**SOS**	**ICLBO**
X_1	0.2220	0.2590	0.1736	0.1939	0.2052	0.2042	0.1912	0.2049
X_2	3.3738	4.8213	4.8411	3.6672	5.1358	3.3172	4.6770	3.4600
X_3	9.1788	7.0814	9.9110	9.2225	9.1542	10.00	8.5366	9.1125
X_4	0.2343	0.3688	0.2026	0.2121	0.2052	0.2045	0.2398	0.2057
Best	1.9810	2.7224	1.9814	1.81504	1.9677	1.8566	2.0280	**1.7315**
Mean	2.1833	3.5824	3.1754	1.9738	2.5698	1.9777	2.1735	**1.9648**

Table 10: Comparison of ICLBO results with literature works for welded beam design problem

	MBA [38]	**HPSO** [39]	**WOA** [40]	**GSA** [41]	**ICLBO**
X_1	0.2057	0.2057	0.2057	0.1821	0.2049
X_2	3.4705	3.4705	3.4842	3.8569	3.4600
X_3	9.0366	9.0366	9.0374	10.0000	9.1125
X_4	0.2057	0.2057	0.2062	0.2023	0.2057
Best	1.7248	1.7248	1.7304	1.8799	**1.7315**

Table 9 indicates that in the present work by spending *NFE*=5000 ICLBO has been competitive to the others both in the best and mean results. However, such *NFE* is not sufficient to capture the global optimum of this example as reported in literature. Increasing *NFE*, ICLBO could capture the same global optimum as MBA and HPSO, reported in Table 10.

4.2.3 Tubular-column design

Fig. 7 illustrates an example for designing a thin-wall tubular column to carry a compressive load P at minimum cost [43]. The wall thickness and mean diameter forms two design variables of this problem while behavior constraints are applied at both buckling and yielding stress. In the present work, various methods are compared to solve the problem by *NFE* of 5000. Table 11 gives the corresponding results declaring that ICLBO has the first rank not only for the best but also the mean result. It is also evident from Table 12 that the result of ICLBO for this example has been better than the other literature works.

Table 1: Results of the present work for tubular column design

	PSO	**GA**	**IC**	**DE**	**CBO**	**TLBO**	**SOS**	**ICLBO**
X_1	5.4743	5.6988	5.4298	5.4781	5.4813	5.4608	5.4540	5.4511
X_2	0.2909	0.3193	0.2955	0.2905	0.2904	0.2918	0.2927	0.2921
Best	26.5522	29.2302	26.5846	26.5539	26.5604	26.5374	26.5528	**26.5039**
Mean	26.8275	29.2302	26.9666	26.8731	27.4417	26.7295	26.7614	**26.5300**

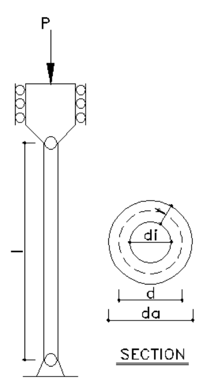


Figure 7. Tubular column design

Table 2: Comparison of ICLBO results with literature works for tubular column design

	CS [44]	**NLP** [43]	**ICLBO**
X_1	5.4513	5.44	5.4511
X_2	0.2919	0.293	0.2921
Best	26.5321	26.5321	26.5039

4.2.4 Gear-train design

Gear-train design represents an unconstrained integer optimization. This problem has four integer variables as introduced by Sandgren [45]. The objective is to minimize the gear ratio cost in the gear train of Fig. 8 to transfer rotation from the driver, D, to the follower gear, F.

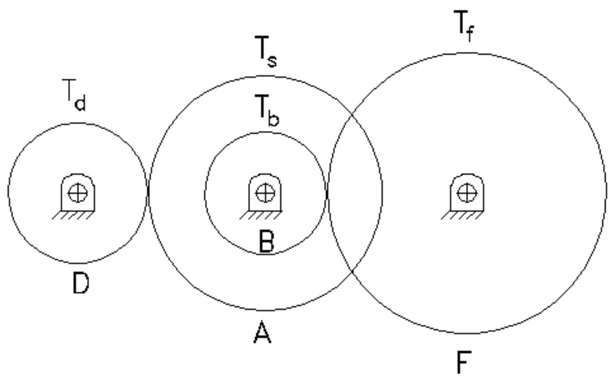


Figure 8. Gear train design

According to Table 13, the present algorithm is one of the two methods among 8 that captured global optimum of this example. Table 14 reveals extra results by other literature works. It is worth mentioning that such global optimum is found by ICLBO with of NFE=4300 while the best computational effort among the other methods is NFE =5000 by CS [44].

Table 13: Result for gear train design

	PSO	**GA**	**ICA**	**DE**	**CBO**	**TLBO**	**SOS**	**ICLBO**
X_1	53	56	60	54	51	51	49	49
X_2	20	18	37	22	26	30	19	19
X_3	13	16	14	17	15	13	16	16
X_4	34	34	60	48	53	53	43	43
Best	2.3E-11	4.E-05	1.5E-07	1.1E-10	2.3E-11	2.3E-11	**2.7E-12**	**2.7E-12**
Mean	3.1E-08	4.8E-05	1.6E-06	1.5E-08	7.9E-08	**1.5E-09**	2.2E-09	1.8E-09

Table 14: Comparison of ICLBO results with literature works for gear train design

	ABC [46]	**ALO** [47]	**MBA** [38]	**CS** [44]	**ICLBO**
$\mathbf{X_1}$	19	43	43	49	49
$\mathbf{X_2}$	16	19	19	19	19
$\mathbf{X_3}$	44	16	16	16	16
$\mathbf{X_4}$	49	49	49	43	43
Best	28.7E-12	2.70E-12	2.70E-12	2.70E-12	2.70E-12

4.2.5 120-bar truss design

As a well-known real-size problem, 120-bar dome of Fig. 9 is considered for sizing design. Material properties include density: $\rho = 0.288 lb/in^3$, elasticity modulus: $E = 30450ksi$ and yield stress: $F_y = 58ksi$. The structural members are divided into 7 groups where the design variables are confined within 0.775 to 20.000 in^2. Gyration radii are taken intermediate variables depending on member areas by $r = 0.4993A^{0.6777}$. Structural loading consists of $13.489 kips(60kN)$ at node 1, $6.744 kips(30kN)$ at nodes 2 to 13 and $2.248 kips(10kN)$ at the other free nodes. A number of investigators [48, 49] has addressed optimal design of this structure. Here, it is treated for weight minimization under stress constraints due to allowable stress design provisions of Iranian code of steel design [50] as follows:

$$\sigma_{tension}^{allowable} = 0.6F_y \tag{19}$$

$$\sigma_{compression}^{allowable} = \begin{cases} \dfrac{12\pi^2 E}{23\lambda^2} & \text{for } \dfrac{\lambda}{C_c} \geq 1 \\ (1 - \dfrac{\lambda^2}{2{C_c}^2})F_y \Big/ (\dfrac{5}{3} + \dfrac{3}{8}\dfrac{\lambda}{C_c} - \dfrac{\lambda^3}{8{C_c}^3}) & \text{for } \dfrac{\lambda}{C_c} < 1 \end{cases} \tag{20}$$

where $C_c = \sqrt{2\pi^2 E / F_y}$ and $\lambda = k\, l/r$ stands for the member slenderness ratio. The effective length factor is denoted by *k*, where *l* and *r* denote the member length and section gyration radius, respectively. A penalty approach is applied to avoid constraint violation.

Performance of the proposed ICLBO in optimal design of 120-bar truss is compared with other literature works in Table 15. The best result of ICLBO via 30000 analyses, is superior to HS and competitive with the others; however, the mean result of CBO is better. Fig. 10 shows that ICLBO has successfully activated stress constraint for this three-dimensional example.

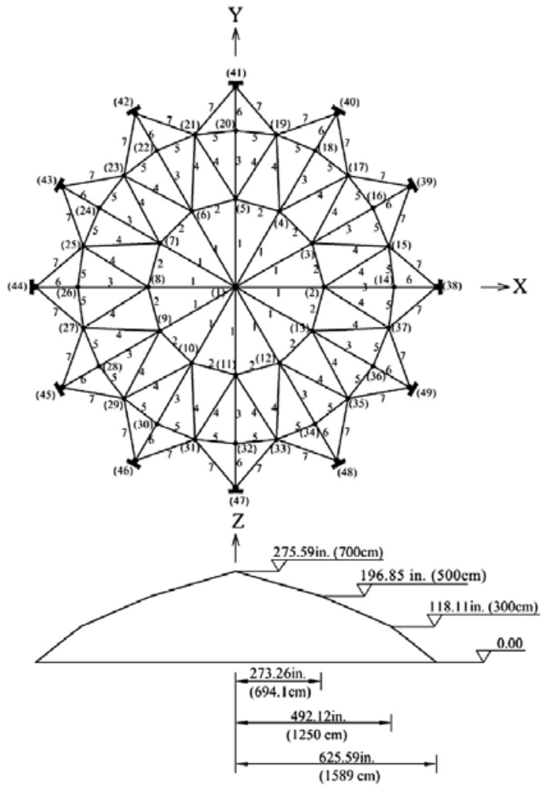


Figure 9. 120-bar dome [49]

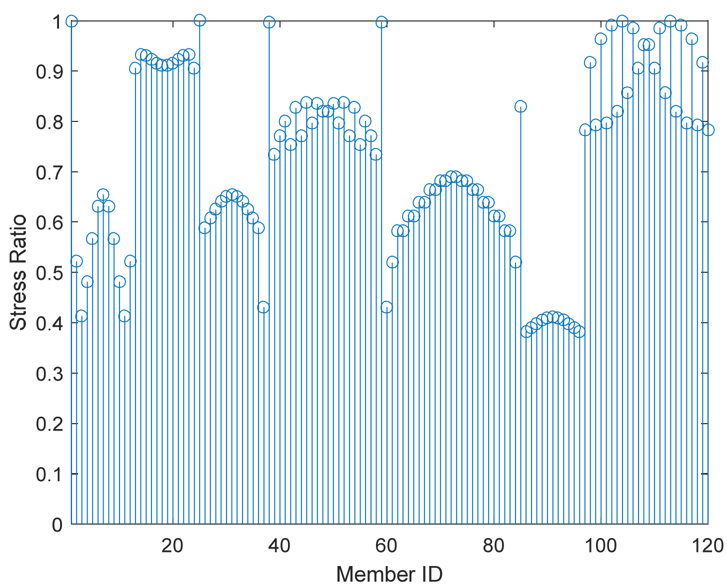
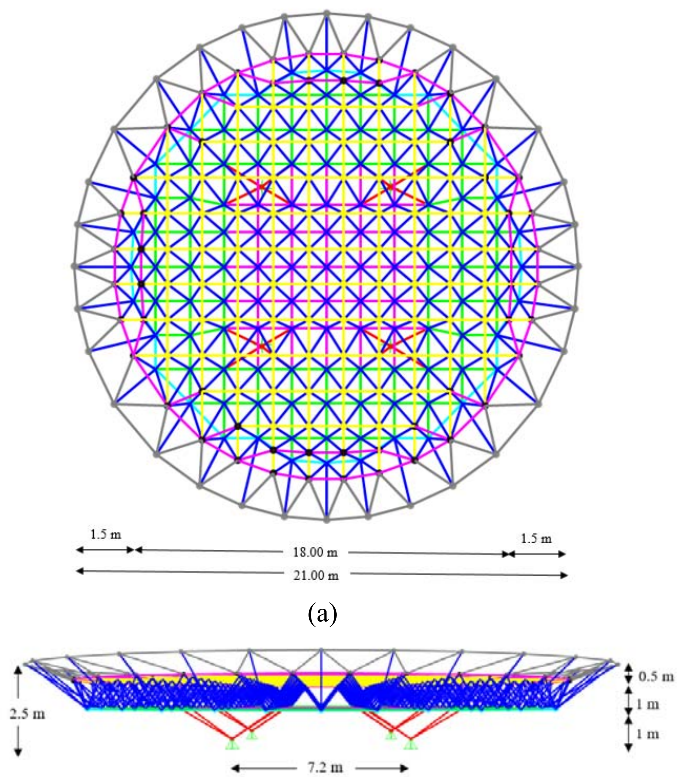


Figure 10. Satisfied stress constrained in the best result of ICLBO for 120-bar dome

Table 15: Comparison of the results for 120-bar dome design

	HS [48]	**HPSACO [21]**	**RO** [51]	**CBO** [49]	**ICLBO**
A1	3.295	3.311	3.128	3.123	3.124
A2	2.396	3.438	3.357	3.354	3.454
A3	3.874	4.147	3.874	4.112	4.113
A4	2.571	2.831	4.114	2.782	2.786
A5	1.150	0.775	0.775	0.775	0.775
A6	3.331	3.474	3.302	3.300	3.573
A7	2.784	2.551	2.453	2.446	2.446
Best (*lb*)	19707.8	19491.3	19476.2	19454.7	19680.6
Mean *(lb)*	-	-	-	19466.0	23661.0



(a)

(b)

Figure 11. 1104-bar helipad structure: a) top view, b) side view

4.2.6 1104-bar truss design

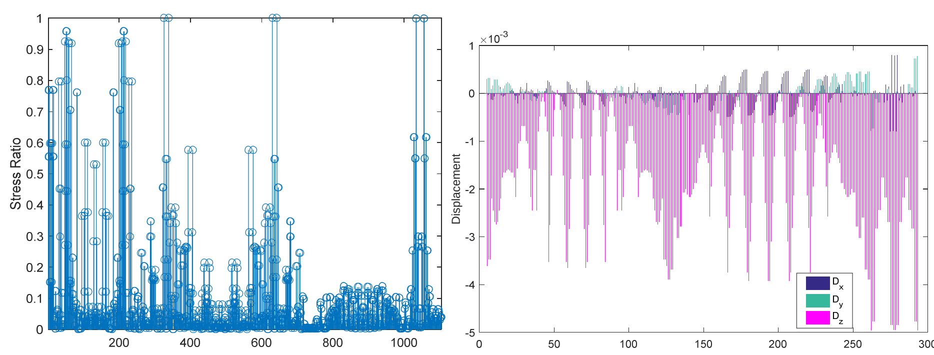
A practical large-scale example is introduced here; with continuous variables. The helipad geometry is shown in Fig. 11; whereas its weight is to be minimized under both stress and displacement constraints. Material properties include density of $\rho = 7850\, kg / m^3$, elasticity modulus of $E = 203.9GPa$ and yield stress of $Fy = 253.1MPa$.

Uniform gravitational load of $300\,kgf/m^2$ is exerted on the top level of helipad. In addition, concentrated load of $350\,kgf$ is applied at each of four central nodes where the helicopter has to land. Stress constraints are applied due to Iranian design code regulations [50]; the same as previous example. Nodal displacements are limited to $5\,cm$ in each orthogonal direction. The structural weigh is penalized to avoid violation of both displacement and stress constraints. Both the structure and loading are symmetric and 9 group of member areas constitute the design variables confined between $10\,cm^2$ and $100\,cm^2$.

This example is solved by ICLBO vs. ICA as well as *Lightening Attachment Procedure Optimization* [52] and *Bat Algorithm* [53]. The former is applied using extra control parameters; including 2.0 for maximum loudness and frequency, 0.9 for the geometric decay factor. The other three are parameter-less algorithms that utilize population size of 50 and 5000 analyses in each run.

Table 16: Comparison of the results for 1104-bar helipad design

	BA	**LAPO**	**ICA**	**ICLBO**
A1(cm^2)	15.09	34.88	18.35	40.03
A2	10.13	16.36	15.48	19.27
A3	19.57	10.00	24.67	10.00
A4	23.75	22.87	23.58	23.21
A5	35.56	42.68	34.65	29.15
A6	45.93	22.60	46.36	20.45
A7	49.31	41.51	49.97	16.94
A8	14.49	46.19	19.85	27.43
A9	52.54	66.17	56.36	82.82
Best (*kg*)	35520.62	36440.05	37191.50	32999.80
Mean	47511.46	38020.01	43975.59	36781.07



(a) (b)

Figure 12. Satisfied constraints in the best result of ICLBO for 1104-bar truss: a) member stress ratios and b) nodal displacements (cm)

Table 16, gives the statistic results of the treated methods in this example. It is evident that ICLBO has superior performance over ICA both in the best and mean results. The results of ICLBO are also competitve with the other treated methods; exhibiting performance of the proposed hybrid method in optimal design of such a constrained large-scale problem. Fig. 12 shows that ICLBO has sucessfully satisfied and activated the problem constraints.

5. CONCLUSION

In this study, proper features of TLBO as a popular parameter-less algorithm was hybridized with the framework of ICA; that acts like a multi-population meta-heuristic search. As a result, ICLBO was proposed as a more powerful stochastic search than either ICA or TLBO. It was declared that ICLBO is capable of capturing global optima in all treated unconstrained test functions. It could achieve the first rank compared with PSO, GA, ICA, DE, CBO, TLBO and SOS in the best and comparative performance in the mean results. The matter is supported by several independent runs.

The conclusion stood reliable in constrained engineering problems for which the proposed method was better than the aforementioned algorithms; in either the best or mean results or even both. ICLBO could also find global optima as reported in the literature works with relatively lower computational effort in majority of the treated engineering benchmarks.

A diversity index was introduced and traced in this study. As the first result, different behavior of meta-heuristic algorithms were distinguished. As the second, the reasoning for proper performance of ICLBO was provided. It is observed that ICA has a moderate decreasing trend of DI as the search progresses. In the other hand, TLBO may first increase and then decrease DI in a more accelerated manner. Hybridization of these two, via the novel ICLBO algorithm shows a DI drop in the early stages of the search followed by deserving a lower DI level up to the end. In another word, ICLBO first provides higher diversity to distribute its representatives among the entire search space, then it reduces diversity to allow better search refinement about already found solutions. Besides, it does not exhibit too rapid drops into zero diversity; as a key point to avoid premature convergence.

Capability of ICLBO was also tested in two real-size problems; one with just stress constraints and the other with both stress and displacement limits. The proposed method was sucessful in activating the constraints of each problem. Meanwhile, the constraints were satisifed in such structural problems. The optimal design of ICLBO in such conditions were obtained competitve with the other methods. It is concluded that the proposed ICLBO has superior performance with respect to ICA and many other metaheuristics. In addition, it is implemented with reduced number of parameters that is interesting from practical point of view.

APPENDIX

A.1. Tension/compression spring design problem

Minimize

$f(x) = (x_3 + 2) x_2 x_1^2$

Subject to

$g_1(x) = 1 - \left(\frac{x_2^3 x_3}{71.785 x_1^4} \right) \le 0$

$g_2(x) = 4x_2^2 - \frac{x_1 x_2}{12.566(x_1^3 x_2 - x_1^4)} + \left(\frac{1}{5108 x_1^2} \right) - 1 \le 0$

$g_3(x) = 1 - \left(\frac{140.45 x_1}{x_2^2 x_3} \right) \le 0$

$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0$

$0.05 \le x_1 \le 2.00$

$0.25 \le x_2 \le 1.30$

$2.00 \le x_3 \le 15.00$

A.2. Welded beam design problem

Minimize

$f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2)$

Subject to

$g_1(x) = \tau(x) - \tau_{max} \le 0$

$g_2(x) = \sigma(x) - \sigma_{max} \le 0$

$g_3(x) = x_1 - x_4 \le 0$

$g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \le 0$

$g_5(x) = 0.125 - x_1 \le 0$

$g_6(x) = \delta(x) - \delta_{max} \le 0$

$g_7(x) = P - P_C(x) \le 0$

$0.1 \le x_i \le 2.0 \qquad i = 1,4$

$0.1 \le x_i \le 10.0 \qquad i = 2\ 3$

where

$$\tau(x)=\sqrt{(\tau')^2+\frac{2\tau'\tau''x_2}{2R}+(\tau'')^2}\ ,\quad \tau'=\frac{P}{\sqrt{2}x_1x_2}\ ,\quad \tau''=\frac{MR}{J}$$

$$M=P\left(L+\frac{x_2}{2}\right),\quad R=\sqrt{\frac{x_2{}^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2}\ ,\quad J=2\left\{\sqrt{2}x_1x_2\left[\frac{x_2{}^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$$

$$\sigma(x)=\frac{6PL}{x_4x_3{}^2}\ ,\quad \delta(x)=\frac{4PL^3}{Ex_4x_3{}^3}\ ,\quad P_C(x)=\frac{4.013E\sqrt{\frac{x_3{}^2x_4{}^6}{36}}}{L^2}\times\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P=6000\,lb\ ,\ L=14\,in\ ,\quad E=30\times10^6\ psi\ ,\ G=12\times10^6\ psi$$

$$\tau_{max}=13.600\,psi\ ,\ \sigma_{max}=30.000\,psi\ ,\ \delta_{max}=0.25\,in$$

A.3. Tubular column design problem

Minimize

$$f(d.t)=9.8dt+2d$$

Subject to

$$g_1=\frac{P}{\pi dt\sigma_y}-1\le 0$$

$$g_2=\frac{8PL^2}{\pi^3Edt\left(d^2+t^2\right)}-1\le 0$$

$$g_3=\frac{2.0}{d}-1\le 0$$

$$g_4=\frac{d}{14}-1\le 0$$

$$g_5=\frac{0.2}{t}-1\le 0$$

$$g_6=\frac{t}{0.8}-1\le 0$$

where

$$P=2500\,kgf\ ,\quad \sigma_y=500\frac{kgf}{cm^2}\ ,\ E=0.85\times10^6\ \frac{kgf}{cm^2}$$

$$\rho=0.0025\frac{kgf}{cm^3}\ ,\quad L=250\,cm$$

A.4. Gear train design problem

Minimize

$$f\left(x\right)=(\left(\frac{1}{6.931}\right)-\left(\frac{x_3x_2}{x_1x_4}\right))^2$$

Subject to

$$x_i \in \{12,13,...,60\}$$

REFERENCES

1. Wolpert DH, Macready WG. No free lunch theorems for optimization, *IEEE Trans Evol Comput* 1997; **1**: 67–82.
2. Bäck T, Eiben AE, Van der Vaart NAL, Binitha S, Sathya SS. A survey of bio inspired optimization algorithms, *Int J Soft Comput Eng* 2012; **2**: 137–51.
3. Holland JH. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1998.
4. Storn R, Price K. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces, *J Glob Optim* 1997; **11**: 341–59.
5. Kennedy J, Eberhart R. Particle swarm optimization, *IEEE Int Conf Part Swarm Optim* 1995; **4**: 1942–8.
6. Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice, *Comput Methods Appl Mech Eng* 2005; **194**: 3902-33.
7. Saka MP, Hasançebi O, Geem ZW. Metaheuristics in structural optimization and discussions on harmony search algorithm, *Swarm Evol Comput* 2016; **28**: 88–97.
8. Geem ZW. *Music-inspired harmony search algorithm: theory and applications*, Berlin Heidelberg, Springer-Verlag, 2009.
9. Simon D, Member S. Biogeography-Based Optimization, *IEEE Transact Evolut Comput* 2008; **12**:702–13.
10. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Comput Struct* 2014; **139**: 18–27.
11. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput Des* 2011; **43**: 303–15.
12. Shahrouzi M, Pashaei M. Stochastic directional search: an efficient heuristic for structural optimization of building frames, *Sci Iran* 2013; **20**: 1124-32.
13. Gandomi AH. Interior search algorithm (ISA): A novel approach for global optimization, *ISA Trans* 2014; **53**: 1168–83
14. Cheng MY, Prayogo D. Symbiotic organisms search: A new metaheuristic optimization algorithm, *Comput Struct* 2014; **139**: 98-112.
15. Kaveh A, Bakhshpoori T. A new metaheuristic for continuous structural optimization: water evaporation optimization, *Struct Multidiscip Optim* 2016; **54**: 23–43.

16. Shahrouzi M. Optimal spectral matching of strong ground motion by opposition-switching search, *Eng Optim 2018 Proceedings of the 6th International Conference on Engineering Optimization*, Lisbon: Springer International Publishing, 2019, pp. 713–724.
17. Kaveh A, Ilchi Ghazaan M. A new meta-heuristic algorithm: vibrating particles system, *Sci Iran* 2017; **24**: 1–32.
18. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput Appl* 2016; **27**: 1053–73.
19. Wang GG, Gandomi AH, Zhao X, Chu HCE. Hybridizing harmony search algorithm with cuckoo search for global numerical optimization, *Soft Comput* 2016; **20**: 273-85.
20. Mahi M, Baykan ÖK, Kodaz H. A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem, *Appl Soft Comput* 2015; **30**: 484–90.
21. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Comput Struct* 2009; **87**: 267–83.
22. Raidl R. A Unified View on Hybrid Metaheuristics, *Hybrid Metaheuristics* (LNCS 4030), 2006: 1–12.
23. Blum C, Puchinger J, Raidl GR, Roli A. A brief survey on hybrid metaheuristics, *Proceedigns of the 4th International Conference Bioinspired Optimization Methods and Their Applications* BIOMA 2010: pp. 3–16.
24. Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, *2007 IEEE Congress Evolutionary Computation* CEC 2007: pp. 4661–4667.
25. Hosseini S, Al Khaled A. A survey on the Imperialist Competitive Algorithm metaheuristic: Implementation in engineering domain and directions for future research, *Appl Soft Comput* 2014; **24**: 1078–94.
26. Nazari-Shirkouhi S, Eivazy H, Ghodsi R, Rezaie K, Atashpaz-Gargari E. Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm, *Expert Syst Appl* 2010; **37**: 7615–26.
27. Ghaffarzadeh H, Raeisi F. Damage Identification in Truss Structures Using Finite Element Model Updating and Imperialist Competitive Algorithm, *Jordan J Civil Eng,* 2016; **10**: 266–77.
28. Eskandar H, Salehi P, Sabour MH. Imperialist competitive ant colony algorithm for truss structures, *Appl Sci* 2011; **12**: 94–105.
29. Kaveh A, Talatahari S, Imperialist competitive algorithm for engineering design problems, *Asian J Civ Eng,* = 2010; **11**: 675–97.
30. Talatahari S, Kaveh A, Sheikholeslami R. Chaotic imperialist competitive algorithm for optimum design of truss structures, *Struct Multidisc Optim* 2012; **46**: 355–67.
31. Mohammadi-ivatloo B, Rabiee A, Soroudi A, Ehsan M. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch, *Ener* 2012; **44**: 228–40.
32. Geem ZW, Sim KB. Parameter-setting-free harmony search algorithm, *Appl Math Comput* 2010; **217**: 3881–9.
33. Rao RV. *Teaching Learning Based Optimization Algorithm*, Cham, Springer International Publishing, 2016.

34. Shahrouzi M, Aghabagloua M, Rafiee F. Observer-teacher-learner-based optimization: An enhanced meta-heuristic for structural sizing design, *Struct Eng Mech* 2017; **62**: 537–50.
35. Rajasekhar A, Rani R, Ramya K, Abraham A. Elitist teaching learning opposition based algorithm for global optimization, *2012 IEEE International Conference on Systems, Man, and Cybernetics* 2012: pp. 1124–1129.
36. Shahrouzi M, Sabzi AH. Damage detection of truss structures by hybrid immune system and teaching–learning-based optimization, *Asian J Civil Eng* 2018; **19**: 811–25.
37. Arora JS. *Introduction to Optimum Design*, 2nd Ed, London, Elsevier acadenic press, 2004.
38. 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.
39. He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, *Appl Math Comput* 2007; **186**: 1407–22.
40. Mirjalili S, Lewis A. The Whale Optimization Algorithm, *Adv Eng Softw* 2016; **95**: 51–67.
41. Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: A Gravitational Search Algorithm, *Inf Sci* 2009; **179**: 2232–48.
42. Coello Coello CA. Use of a self-adaptive penalty approach for engineering optimization problems, *Comput Ind* 2000; **41**: 113–27.
43. Rao SS. *Engineering Optimization: Theory and Practice*, 3rd Ed, John Wiley & Sons, 2009.
44. Gandomi AH, Yang XS, Alavi AH. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, *Eng Comput* 2013; **29**: 17–35.
45. Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization, *J Mech Des* 1990; **112**: 223–9.
46. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J Glob Optim* 2007; **39**: 459–71.
47. Mirjalili S. The ant lion optimizer, *Adv Eng Softw* 2015; **83**: 80–98.
48. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm, *Comput Struct* 2004; **82**: 781–98.
49. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Comput Struct* 2014; **139**: 18–27.
50. INBC: Part-10. *Iranian National Building Code, Part-10: Design of Steel Structures*, 4th ed, Tehran: Roads, Housing and Urban Development of Iran, 2013.
51. Kaveh A, Khayatazad M. A new meta-heuristic method: Ray Optimization, *Comput Struct* 2012; **112–113**: 283–94.
52. Nematollahi AF, Rahiminejad A, Vahidi B. A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization, *Appl Soft Comput* 2017; **59**: 596–621.
53. Yang XS. A new metaheuristic: bat-inspired algorithm, *Stud Comput Intell* 2010; **284**: 65-74.