Alexandria Engineering Journal (2022) 61, 11537–11563

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

Colonial competitive evolutionary Rao algorithm for optimal engineering design

Shahab S. Band [a,*], Sina Ardabili [b], Amir Seyed Danesh [c], Zulkefli Mansor [d], Ibrahim AlShourbaji [e], Amir Mosavi [f,g,h,*]

[a] *Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan*
[b] *Department of Informatics, J. Selye University, Komarom, Slovakia*
[c] *Faculty of Technology and Engineering, East of Guilan, University of Guilan, Rudsar-Vajargah, Iran*
[d] *Faculty of Information Science and Technology, Universiti Kebangsaan, Malaysia*
[e] *Department. of Computer and Network Engineering, Jazan University, Jazan 45142, Kingdom of Saudi Arabia*
[f] *John von Neumann Faculty of Informatics, Obuda University, Budapest, Hungary*
[g] *Institute of Information Society, University of Public Service, 1083 Budapest, Hungary*
[h] *Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in, Bratislava, Slovakia*

Received 17 August 2021; revised 23 March 2022; accepted 10 May 2022
Available online 25 May 2022

KEYWORDS

RAO algorithms;
Colonial Competitive RAO (CCRAO);
Engineering optimization;
Machine Learning;
Artificial Intelligence;
Evolutionary algorithm

Abstract Rao algorithms that include three algorithms are very simple and parameter-less algorithms with effective and desirable performance. This paper modifies these three algorithms, merges them, and establishes a powerful group algorithm. In the first optimization step, the suggested algorithm is tested on 30 standard CEC2014 functions with 50 dimensions to compare it with main algorithms, several well-known algorithms, and modified versions of RAO algorithm. It becomes evident in the first test that the suggested optimizer is effective, and reliable for optimization of real-parameter functions, and it has shown its superiority to original RAO algorithm and several modern and modified versions of RAO algorithms for most of the test functions and achieved more acceptable results than them. Moreover, the suggested algorithm benefits a faster convergence characteristic than original RAO algorithms. The proposed Colonial Competitive RAO (CCRAO) has been applied on five popular engineering problems and its results have been compared with those of recent papers. According to the results, CCRAO is an effective, robust, and reliable optimizer for engineering design problems and can contain all useful features of RAO algorithms altogether. CCRAO has succeeded to converge to the best solution for these engineering problems and surpasses most of the other algorithms.

* Corresponding authors at: Obuda University.
E-mail addresses: shamshirbands@yuntech.edu.tw (S.S. Band), amir.mosavi@kvk.uni-obuda.hu (A. Mosavi).
Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2022.05.018
1110-0168

1. Introduction

Optimization aims to discover the optimal acceptable solution subject to the limitations and requirements of any problem. To define a problem, different solutions are possible. A function called objective function is defined so that the solutions are compared, and the optimal solution is chosen. One of the critical steps in optimization process is to select an appropriate objective function. Occasionally, several objectives are optimized at the same time. Optimization problems that include several objective functions are known as multi-objective problems. Establishing a new objective function with linear combination of main objective functions is one of the straightforward approaches to address this type of problems, where the impact of each function is specified based on its dedicated weight. Every optimization problem consists of several independent variables called design variables, shown by a *D*-dimension vector, *x*. The optimization aims to specify these design variables so that the different limits of the objective function is established [1]. One of the important optimization approaches that have gained the attention of engineers and users during recent years is metaheuristic algorithms so that the acceptable solution is obtained at a shorter time thanks to their simple structure. Metaheuristic algorithms aim to present a solution to the problem during an acceptable time. Such algorithms may not be the best real solution to solve a problem, but they can find the closest possible solution. Metaheuristic algorithms are inspired by nature, physics, and humans and have widespread applications in solving optimization problems.

Metaheuristic algorithms are usually combined with other algorithms to reach an acceptable solution or escape from a locally optimal solution [1]. Such an example is a combined genetic algorithm (GA) and gravitational search algorithm (GSA), known as GA-GSA, that attempts to maximize reliability, availability, and maintainability (RAM) parameters at the same time so that performance and productivity of the system are improved [2]. In [3], GSA and GA algorithms are merged to form GSA-GA algorithm to be applied to constrained nonlinear optimization problems with mixed variables. To overcome this challenge and to discover the global searching capability of Harris Hawks Optimization (HHO), reference [4] presents improved teaching–learning HHO (ITLHHO) method to provide solution to most of the related engineering and numerical optimization problems. The authors in [5] introduce the particle swarm optimization (PSO)-GA (PSO-GA) method to optimize various constrained engineering and numerical optimization problems. The role of PSO is improving the vector, but the GA modifies decision vectors with the help of genetic original operators (i.e., crossover and mutation). These operators also assist to create an equilibrium between exploitation and exploration agents.

There are many novel algorithms introduced in the literature which their names and inspirations are briefly summarized in Table 1.

It is mentioned some of the previously mentioned optimizers face with an imbalance between global and local finding operators during the optimization procedure. This may conduce to hooking in local optima. Some other algorithms suffer from complicated overwhelming computation. Furthermore, selecting appropriate values for parameters become challenging in some algorithms; thus, degrading the optimization efficiency of the optimizer. Additionally, reaching the proper optimal solution needs more iteration steps and is time-demanding in some of the algorithms mentioned above. Based on the no free lunch (NFL) theorem [45], one cannot find a specific algorithm to optimize all styles of optimization test functions. As a result, we need more sophisticated hybrid algorithms to resolve the issue. Therefore, simple, and powerful RAO algorithms are adopted in this paper to present a novel strong algorithm called Colonial Competitive RAO (CCRAO), the description of which is provided in the rest of the study.

The structure of the paper is introduced here. Details of the CCRAO algorithm are described in Section 2. RAO optimizer is outlined in Section 3. Section 4 tests the performance of suggested method via real-world test functions. This section applies the algorithm to solve several well-known engineering optimization problems that suffer from complex search spaces. The related discussions and future trends for researchers are also given in this section. Eventually, conclusions have been briefly stated in Section 5 of the paper.

2. RAO algorithms

Rao introduced powerful and parameter-less optimization algorithms called RAO algorithms in 2020 [46]. RAO algorithms include three separate algorithms. They have become popular among scholars thanks to their simplicity and power such as the photovoltaic cell parameter estimation [47,48], the reinforced concrete cantilever weight optimization [49], mechanical design optimization [50], the cropping pattern under a constraint environment [51], and optimal design of mechanical system components [52].

The basic of RAO algorithms is that the worse member is always subtracted from the better member, i.e., the population moves towards the best solution. These algorithms include three separate optimization vectors as follows [46]:

Rao-1 algorithm:

$$Y^{new}_{i,j} = Y^{Iter}_{i,j} + r_{1,j}\left(Y^{Iter}_{best,j} - Y^{Iter}_{worst,j}\right) \tag{1}$$

Rao-2 algorithm:

$$\begin{cases} Y^{new}_{i,j} = Y^{Iter}_{i,j} + r_{1,j}\left(Y^{Iter}_{best,j} - Y^{Iter}_{worst,j}\right) \\ \qquad + r_{2,j}\left(\left|Y^{Iter}_{i,j}\right| - \left|Y^{Iter}_{k,j}\right|\right)\ iff\left(Y^{Iter}_{i}\right) < f\left(Y^{Iter}_{k}\right) \\ Y^{new}_{i,j} = Y^{Iter}_{i,j} + r_{1,j}\left(Y^{Iter}_{best,j} - Y^{Iter}_{worst,j}\right) + r_{2,j}\left(\left|Y^{Iter}_{k,j}\right| - \left|Y^{Iter}_{i,j}\right|\right)\ else \end{cases} \tag{2}$$

Rao-3 algorithm:

$$\begin{cases} Y^{new}_{i,j} = Y^{Iter}_{i,j} + r_{1,j}\left(Y^{Iter}_{best,j} - \left|Y^{Iter}_{worst,j}\right|\right) \\ \qquad + r_{2,j}\left(\left|Y^{Iter}_{i,j}\right| - Y^{Iter}_{k,j}\right)\ iff\left(Y^{Iter}_{i}\right) < f\left(Y^{Iter}_{k}\right) \\ Y^{new}_{i,j} = Y^{Iter}_{i,j} + r_{1,j}\left(Y^{Iter}_{best,j} - \left|Y^{Iter}_{worst,j}\right|\right) + r_{2,j}\left(\left|Y^{Iter}_{k,j}\right| - Y^{Iter}_{i,j}\right)\ else \end{cases} \tag{3}$$

In above equations, denotes the *i*th member position in algorithm population in the current iteration (*Iter*). $j = 1{:}D$ shows the *j*th dimension of each member in the population, *D* shows the dimension of the problem. Y^{Iter}_{best} and Y^{Iter}_{worst} are

Table 1 Some of metaheuristic optimization algorithms and their inspirations.

Algorithm	Ref.
Description	
Heat Transfer Search (HTS)	[6]
Based on the thermodynamics and heat transfer laws.	
Pity Beetle Algorithm (PBA)	[7]
Mimics the searching behavior for nest and food of a type of beetle.	
Sine-Cosine Algorithm (SCA)	[8]
Adopts a mathematical mode to oscillate the solutions outwards or towards the best solution using sine and cosine functions.	
Turbulent Flow of Water-Based Optimization (TFWO):	[9]
Inspiration by moving the water drops appearing in rebellious water circulation.	
Rainfall Optimization (RO):	[10]
Mimics the behavior of raindrops to optimize different problems.	
Collective Decision Optimization (CDO):	[11]
Imitates the social behavior of human in making decision.	
Biogeography-Based Optimization (BBO):	[12]
A different optimizer inspired on population and biogeography mathematics.	
Supernova Optimizer (SO):	[13]
Follows the movement of supernova phenomena in physics.	
Atom Search Optimization (ASO):	[14]
Follows the atomic movement model in physics.	
Water Evaporation Optimization (WEO):	[15]
Imitates evaporation of water particles on an illiquid area with various wettability.	
Thermal Exchange Optimization (TEO):	[16]
According to Newton's rules of chilling out.	
The CFA optimization framework	[17]
Based on interaction of electrical particles due to electrical attraction and repulsion forces.	
Artificial Bee Colony (ABC)	[18]
Inspired by the smart search of honeybee population.	
Grasshopper Optimization Algorithm (GOA)	[19]
Follows the smart search of grasshopper population to find solutions	
Squirrel Search Algorithm (SSA)	[20]
Mimics the impellent exploring treatment of migrating squirrels and their impressive movement.	
Colliding Bodies Optimization (CBO)	[21]
Inspired by the collisions between objects.	
Salp Swarm Algorithm (SSA)	[22]
Based on the movement of salps during foraging and navigating in open waters.	
Butterfly Optimization Algorithm (BOA)	[23]
Based on the search and moving behaviors of butterflies.	
Algorithm Of Innovative Gunner (AIG)	[24]
Mimics selecting artillery factors to send a shot exactly toward the target.	
Fitness-Distance Balance (FDB)	[25]
Aims to optimize the premature convergence function in the *meta*-heuristic search framework.	
Water Strider Algorithm (WSA)	[26]
Inspired by the lifespan of water strider bugs.	
Slime Mould Algorithm (SMA)	[27]
Inspired by the swing style of slime mould.	
Color Harmony Algorithm (CHA)	[28]
Inspired by the related situations circa the dye circle in the harmonic templates and Munsell dye system.	
Red Deer Algorithm (RDA)	[29]
Competing male red deers to reach a harem with more hinds.	
Symbiotic Organisms Search (SOS)	[30]
An imitation of symbiotic interaction methods seen in organisms to live and reproduce within the ecosystem.	
Bat-Inspired Algorithm (BA)	[31]
Inspired by the echolocation search of bats.	
Cuckoo Search (CS)	[32]
Inspiration by the obligate brood parasitic behavior of a special type of cuckoos.	
Fireworks algorithm (FWA)	[33]
Based on the explosion of firework.	
Social-Based Algorithm (SBA)	[34]
Seeks individuals who participate in community development characteristic.	
Soccer League Competition Algorithm (SLCA)	[35]
Based on soccer leagues and competitions among teams and players	
Sandpiper Optimization Algorithm (SOA)	[36]
According to the travelling and attacking movement of sandpipers.	

(*continued on next page*)

Table 1 (*continued*)

Algorithm	Ref.
Tunicate Swarm Algorithm (TSA)	[37]
Inspired by the population search of tunicates in foraging and navigation processes	
Seagull Optimization Algorithm (SOA)	[38]
Based on the different attacking and travelling methods of seagulls.	
Spotted hyena optimizer (SHO)	[39]
Emulates spotted hyenas' behavior.	
Spring Search Algorithm (SSA)	[40]
Specifying the governing laws.	
Wild Geese Algorithm (WGA)	[41]
Mimics wild geese and provides a model of different modes of their living including regular cooperative travelling, evolution, and fatality, etc.	
Nuclear Reaction Optimization (NRO)	[42]
Imitates the nuclear reaction process.	
Lightning Search Algorithm (LSA)	[43]
Inspired by the physical event of lightning.	
Human Mental Search (HMS)	[44]
Inspired by the finding methods of the bid place in online auctions.	

positions of the best and worst members at the current iteration. r_1 and r_2 represent two figures randomly selected in the range of 0 and 1. Y_k^{Iter} shows the position of the kth member that is randomly selected, and $f(Y_{Iter})$ illustrates the objective function value for considered member in the current iteration. In all three algorithms mentioned above, the situation of the ith member at the next generation is calculated in (4):

$$\begin{cases} Y_i^{Iter+1} = Y_i^{new} \; iff(Y_i^{new}) \leqslant f(Y_i^{Iter}) \\ Y_i^{Iter+1} = Y_i^{Iter} \quad else \end{cases} \tag{4}$$

2.1. Modeling abrupt changes in the population to correct RAO algorithms through adding a variety

One of the shortcomings of RAO algorithms is to trap the population in the local optima. The most known reason for that is losing the population variety of RAO algorithms as the number of iterations increases. To overcome this issue, we have adopted abrupt changes in the population. Such changes establish new features in the individuals. Abrupt changes in the proposed algorithm are modeled through the random displacement of one of the members of the group, except for the leader, to a new random position. These changes in the population, from an algorithm and optimization perspective, help the evolutionary movement to eschew trapping in the local optimum points, and sometimes improves the position of one member and displaces it to a better optimality range. Each member of the group is selected with a probability of 0.1 and one of its dimensions is changed, i.e.,

$$\begin{gathered} if \, rand < 0.1 \\ j = 1 + round(rand * (D-1)) \\ Y_{i,j}^{Iter+1} = Min_j + rand * (Max_j - Min_j) \\ end \end{gathered}$$

Min and *Max* show the down and up limits of control parameters.

3. Colonial competitive modeling among three RAO algorithms: The proposed Colonial competitive optimizer

With the formation of the first groups of the algorithm's population and determining the position of the group members and the best member of each group as the group leader, the imperialist competition between these groups begins with the aim of getting bigger and stronger. Any group whose leaders, i.e., the best group member in each iteration, if cannot reach a better position over time, has not been able to succeed in imperialistic competition and conflict with other groups and increase its power so that it will gradually lose its members, where it is even possible that each of the group will be given to a stronger group and that group will fall altogether.

Therefore, the survival of a group will depend on the power of the leader of that group and the domination of the members of the weaker groups as a colony. As a result, during competition between different groups, the power of larger groups with stronger leaders will gradually increase, and weaker groups will be eliminated, and their members will be dominated by other groups. To start the optimization process with the CCRAO algorithm, we create N_C initial groups from N_{pop} populations and divide the population between these groups in a row in the first iteration. That is, we select three of the strongest members and form three groups, and we give 4 members to the first group, 5 to the second group, and 6 to the third group. Then the cycle continues so that the whole population can be placed in the groups. Next, Fig. 1 illustrates the optimization cycle is carried out to reach the stage of competition between groups. Also, Fig. 2 illustrates the pseudo code of the suggested algorithm.

At this stage, the weakest member of the group with the weakest leader is removed and given to the group with the best function value (greatest power). The same optimization cycle continues until the last iteration specified by the user, and if in a particular iteration a group has only the leading member left, we transfer that leader to the strongest group.

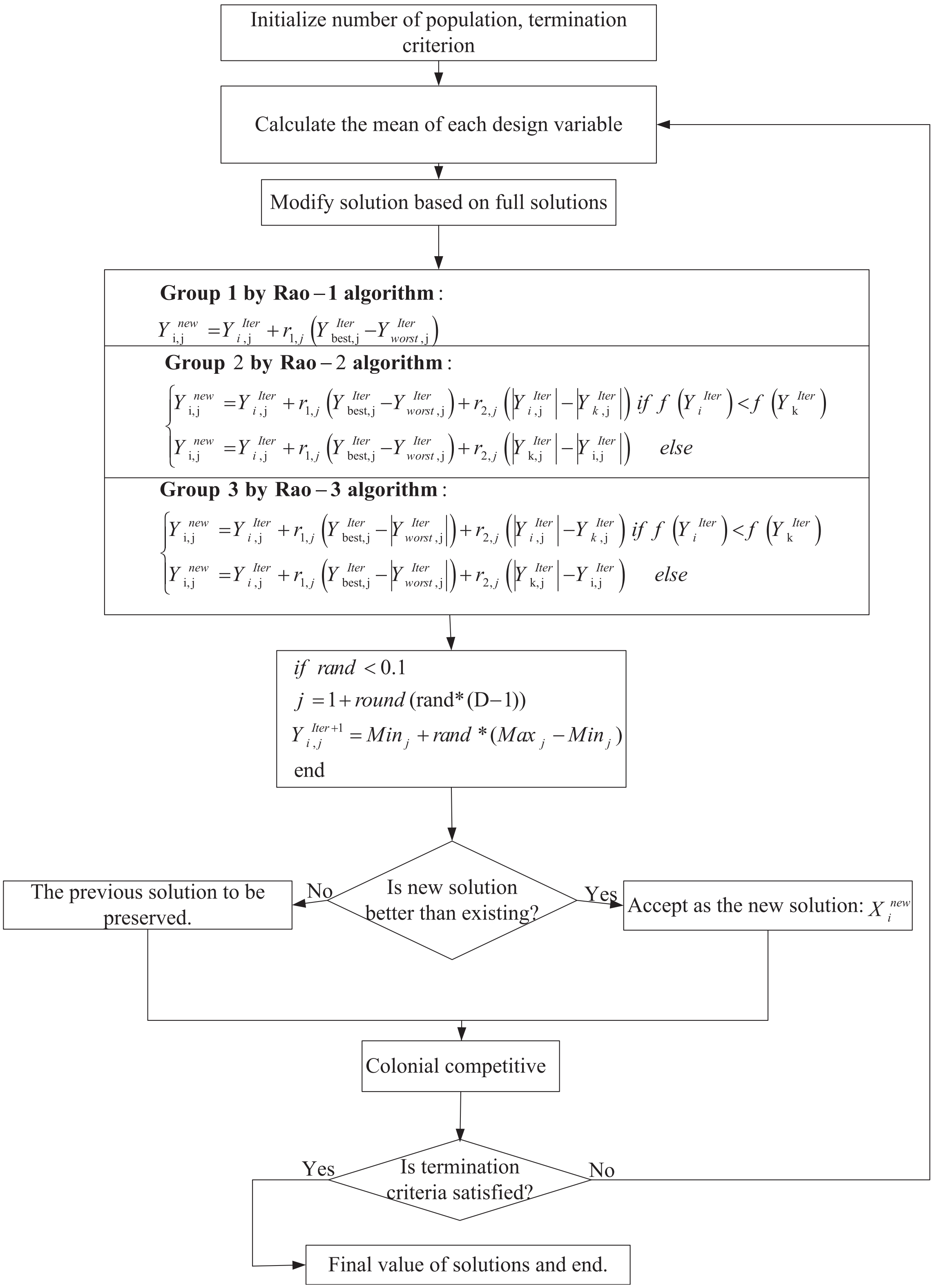

Fig. 1 The proposed Colonial Competitive Optimizer (CCRAO).

Algorithm 1: The CCRAO Optimizer based on the Rao algorithms.

1: Get control parameters of the CCRAO Optimizer, that are number of iterations $Iter_{max}$, the initial swarm size *Npop* and N_C individuals. Set *Iter* = 0 for individuals;

2: Generate the initial random swarm *Npop* (*i*=1, 2, …, *Npop*);

3: Evaluate the fitness of each population individual;

4: Rate initial groups N_C with N_C of strongest the initial swarm;

5: Divide the initial weak swarm ($NPw = Npop - N_C$). NPw^m of the initial weak swarm is randomly chosen and given to the *m*th group;

6: **while** The *Iter* **till** $Iter_{max}$ **do**

7: Set *Iter*= *Iter*+1;

8: **for** t = 1 to N_C **do**

9: **for** i = 1 to NPw^m **do**

10: Select the uniform randomly $k \neq i$;

11: **for** j = 1 to D do

12: **if** $t = 1$

13: Update using Eq. (1)

14: **end if**

15: **if** $t = 2$

16: Update using Eq. (2)

17: **end if**

18: **if** $t = 3$

19: Update using Eq. (3)

20: **end if**

21: **if** $f\left(Y_i^{new}\right) \leq f\left(Y_i^{Iter}\right)$

22: $Y_i^{Iter+1} = Y_i^{new}$

23: **end if**

Fig. 2 . The pseudo-code of the suggested algorithm.

```
            if rand < 0.1
                j = 1 + round(rand*(D−1))
                Y_{i,j}^{Iter+1} = Min_j + rand*(Max_j − Min_j);
                if f(Y_i^{new}) ≤ f(Y_i^{Iter})
                    Y_i^{Iter+1} = Y_i^{new}
                end if
            end if
        end for
    end for
  end for
  Evaluate the total power of each group;
  Remove the weakest swarm from the weakest group and give the strongest group;
  Remove the swarm group mth with no weak swarm (NPw^m = 0) and give the strongest group;
end while
```

Fig. 2 (*continued*)

4. Experimental study

4.1. RAO algorithms for optimization of CEC2014 test functions

To illustrate the capability of the CCRAO method suggested in this paper compared to main algorithms, i.e., Rao algorithms, 30 standard well-known test functions from CEC2014 test functions [53] with dimension 50, which have been widely used in recent years, is chosen. Table 2 provides a summary of these functions. For RAO algorithms and the proposed CCRAO algorithm, 32 populations with 15,625 iterations have been selected so that the number of calculations of the objective function per independent run time for all optimization algorithms is equal to 500,000 (15625 × 32 = 500000). Also, to achieve the simulation criteria, i.e., the mean, standard deviation, and the best optimal values, the algorithm for each function of CEC2014 has been performed by each algorithm 25 independent runs. We consider the optimal value for all optimization functions to be 0.

Table 2 CEC2014 test functions [53].

Unimodal test functions:
F1, F2, F3.
Simple Multimodal test functions:
F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16.
Hybrid test functions:
F17, F18, F19, F20, F21, F22.
Composition test functions:
F23, F24, F25, F26, F27, F28, F29, F30.

With a glance at the results given in Table 3 considering the criteria shown as well as the rank of each algorithm in each test function, one can see a remarkable superiority of the proposed CCRAO algorithm to RAO algorithms for optimization of a range of optimization functions. Furthermore, the last row in Table 3 provides the number of times the best solution, *Nb*, as well as the average rank value of thirty test functions, *Mr*, are achieved by the algorithm, where the proposed algorithm has the best statistical results. These numerical results well demonstrate the strength and potential of the approach suggested in this study as an equivalent algorithm and an alternative to the three Rao algorithms for optimization. To show the convergence efficiency of these optimizers, Fig. 3 show their convergence characteristic for all test functions.

4.2. Comparison with modern optimizers

To better understand the performance of CCRAO, several well-known algorithms as well as improved versions of RAO algorithms are examined. These algorithms are as follows, where the codes of these algorithms are downloaded from the authors' websites and the control of their parameters is

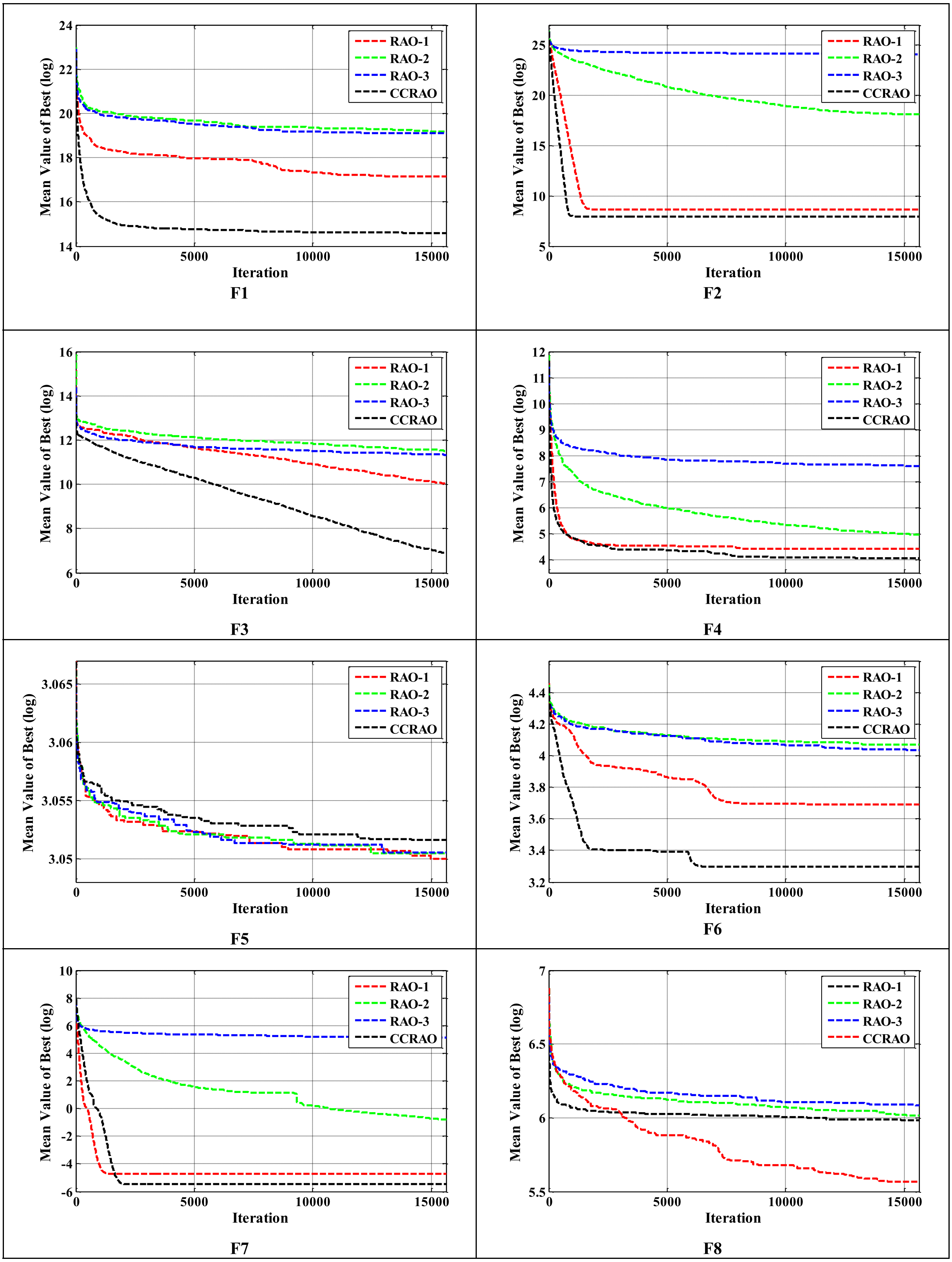

Fig. 3 The convergence characteristics of RAO algorithms and CCRAO.

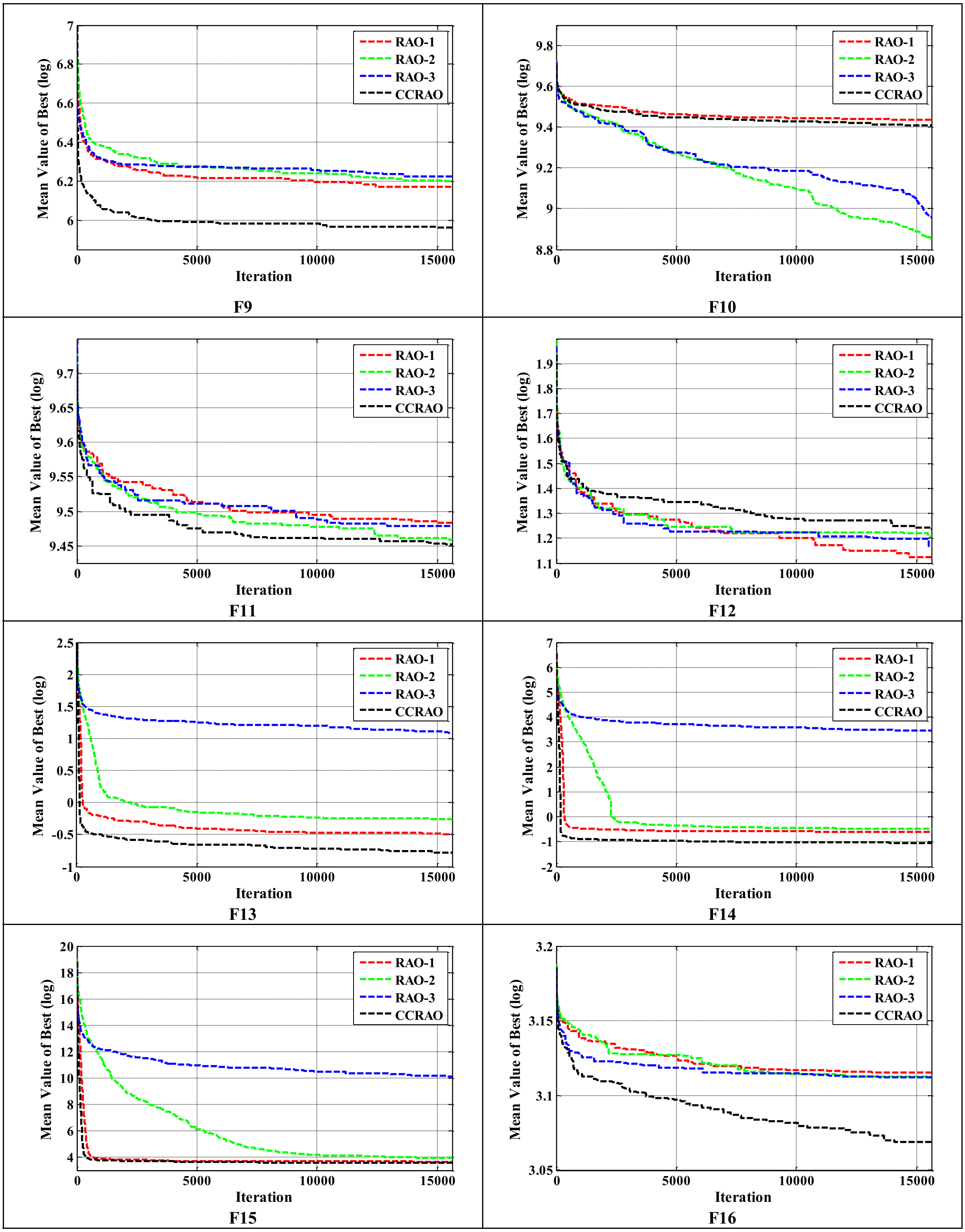

Fig. 3 (*continued*)

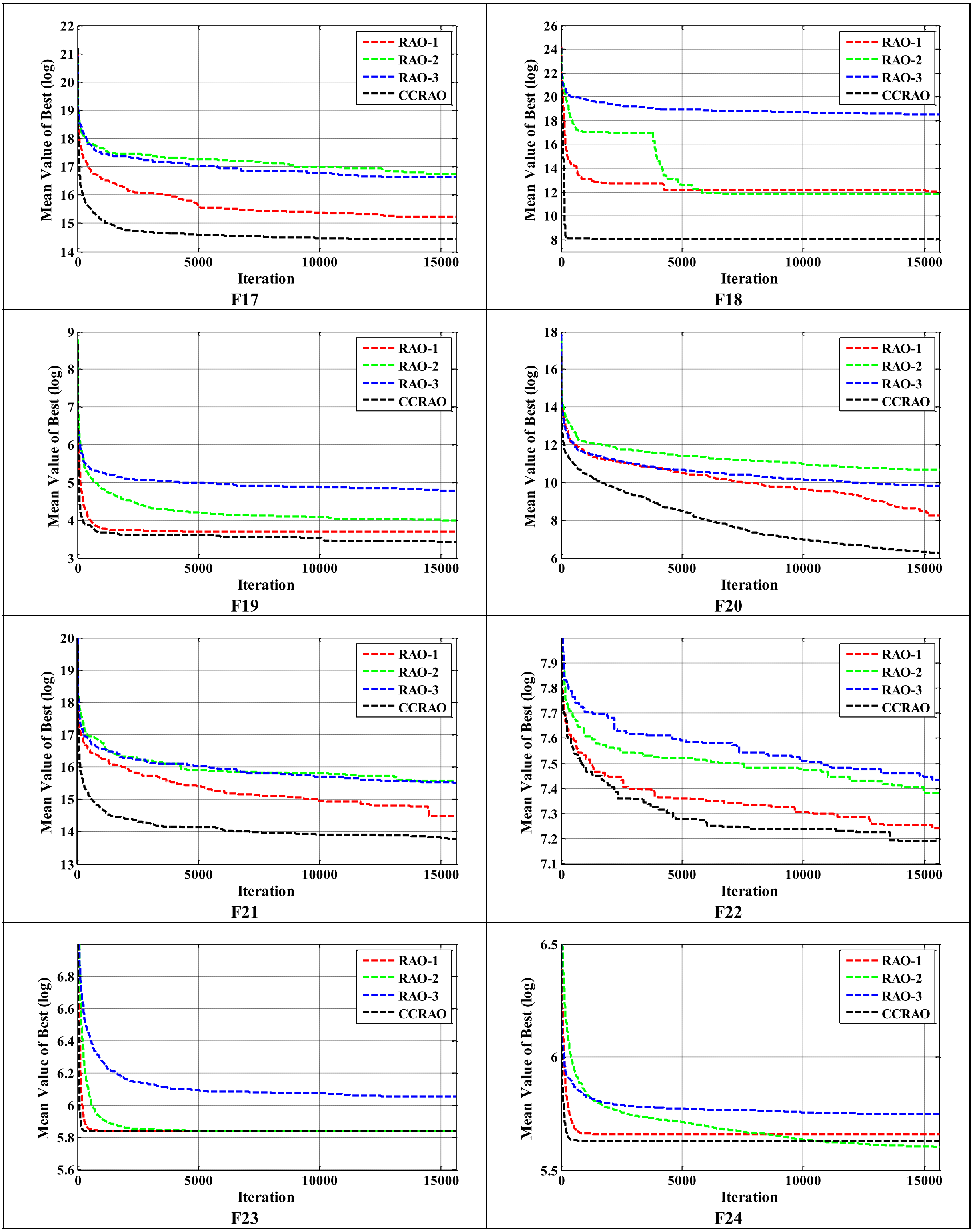

Fig. 3 (*continued*)

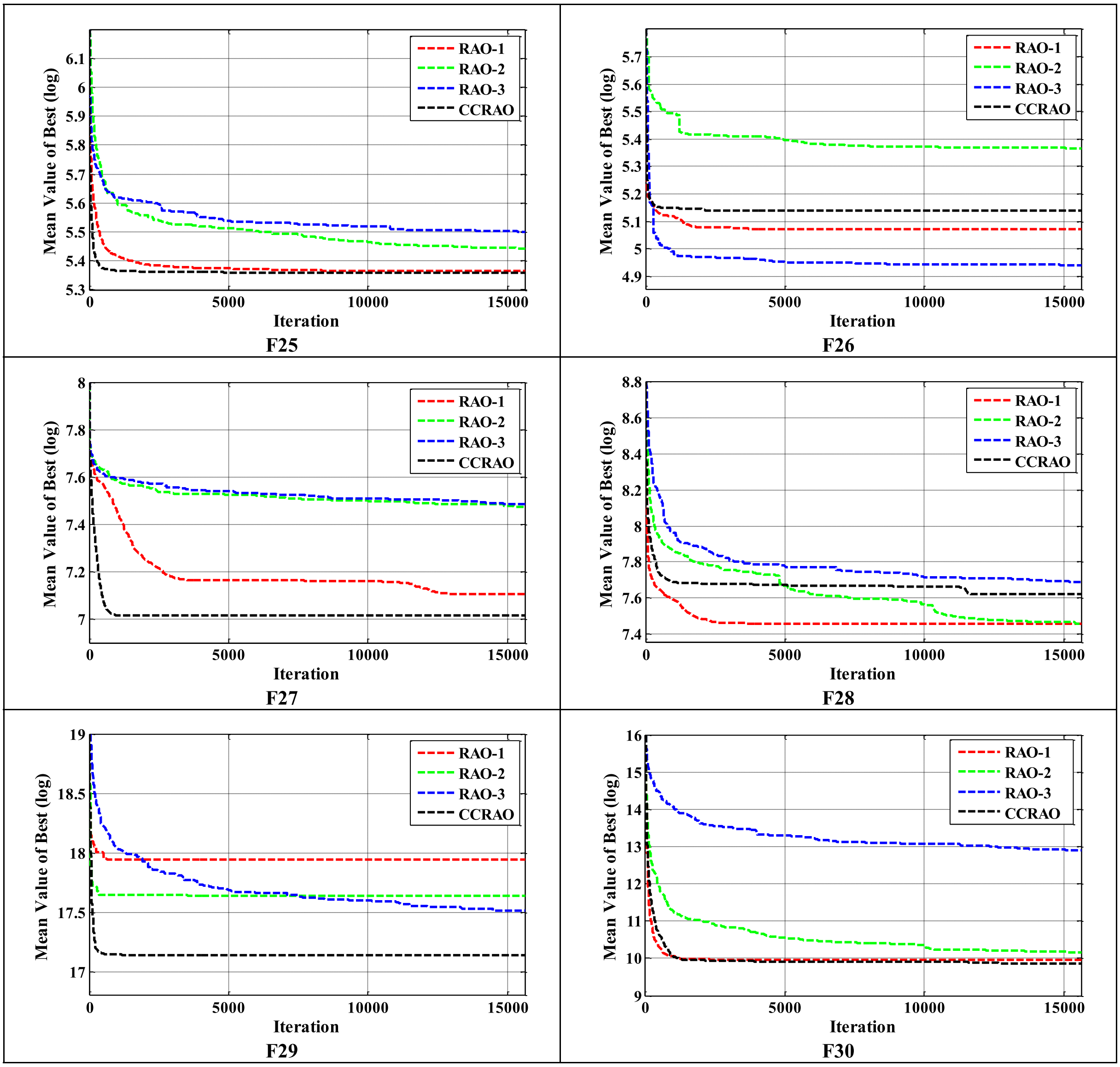

Fig. 3 (*continued*)

according to Table 4 and Table 5 for 30 and 50 dimensions, respectively:

- Imperialist competitive algorithm (ICA) [54]
- Modified imperialist competitive algorithm (MICA) [54]
- Harris hawks optimization (HHO) [55]
- The self-adaptive population Rao algorithm (SAP-Rao) [56]
- Quasi-oppositional-based Rao-2 algorithm (QO-Rao-2) [57]
- GWO [58]

Table 6 and Table 7 summarize the simulation results obtained for ICA, GWO, HHO, MICA, SAP-Rao, QO-Rao-2 and CCRAO (this study) algorithms for *D* equal to 30 and 50, respectively. Symbols "-", "+", and "=" show that the performance of corresponding algorithm is worse than, better than, and like that of CCRAO, respectively. According to the obtained result, CCRAO performs better than the rest of mentioned algorithms. As can be observed in Table 6 and Table 7, CCRAO with an average rank of 1.46 and 1.9333 is the decisive winner of this comparative study for 30- and 50-dimensions test functions, respectively. The second ranked is QO-Rao-2 with the average rank of 3.1667 and 3.2667 for these two dimensions, respectively. As can be observed, there is a significant difference between the proposed CCRAO and the second winner, i.e., QO-Rao-2.

The Wilcoxon's test is adopted here to see whether two algorithms show different behaviors [59]. Table 8 gives the *p*-values when using Wilcoxon's test on CCRAO and other algorithms for 30 and 50-dimensional test cases. The p-values smaller than 0.05 (the level of significance) are shown in bold. Regarding the given data, once can easily observe that CCRAO performance significantly better than the rest of algorithms. Even though the CCRAO does not surpass QO-Rao-1

Table 3 The obtained results of different optimization algorithms.

Number Functions	Best Rank				Mean Rank				Std. Rank			
	Rao-1	Rao-2	Rao-3	CCRAO	Rao-1	Rao-2	Rao-3	CCRAO	Rao-1	Rao-2	Rao-3	CCRAO
F1	2.20e + 6	7.49e + 7	1.41e + 8	**3.86e + 5**	2.77e + 7	2.13e + 8	1.97e + 8	**2.17e + 6**	2.03e + 7	1.15e + 8	4.51e + 7	**1.37e + 6**
	2	3	4	**1**	2	4	3	**1**	2	4	3	**1**
F2	7.18	1.31e + 6	2.13e + 10	**0.0436**	5810	7.29e + 7	2.84e + 10	**2885**	8508	2.15e + 8	5.76e + 9	**3231**
	2	3	4	**1**	2	3	4	**1**	2	3	4	**1**
F3	1.63e + 4	7.42e + 4	6.32e + 4	**725**	2.26e + 4	1.00e + 5	8.64e + 4	**995**	7012	1.59e + 4	1.43e + 4	**245**
	2	4	3	**1**	2	4	3	**1**	2	4	3	**1**
F4	17.3	105	1558	**14.2**	82.3	142	1976	**57.8**	24.2	52.8	455	**22.8**
	2	3	4	**1**	2	3	4	**1**	2	3	4	**1**
F5	21.04	21.05	21.07	**20.85**	**21.1**	**21.1**	**21.1**	**21.10**	0.0456	0.0397	0.037	**0.0175**
	2	3	4	**1**	**1**	**1**	**1**	**1**	4	3	2	**1**
F6	30.7	46.6	46.7	**19.98**	40.2	58.5	56.6	**27.1**	8.24	6.86	6.26	**4.56**
	2	3	4	**1**	2	4	3	**1**	4	3	2	**1**
F7	3.41e-13	0.0784	128	**2.68e-13**	0.00886	0.453	165	**0.00418**	**0.00734**	0.297	37.6	0.00913
	2	3	4	**1**	2	3	4	**1**	**1**	3	4	2
F8	129	356	360	**102**	397	409	440	**261**	123	33.0	35.2	**31.2**
	2	3	4	**1**	2	3	4	**1**	4	2	3	**1**
F9	421	465	457	**359**	478	493	505	**389**	35.3	**17.1**	31.2	22.9
	2	4	3	**1**	2	3	4	**1**	4	**1**	3	2
F10	1.12e + 4	2806	1150	**1122**	1.25e + 4	**6988**	7765	1.22e + 4	731	3488	4180	**518**
	4	3	2	**1**	4	**1**	2	3	2	3	4	**1**
F11	1.26e + 4	**1.21e + 4**	**1.21e + 4**	1.23e + 4	1.31e + 4	1.28e + 4	1.31e + 4	**1.27e + 4**	319	405	543	**286**
	3	**1**	**1**	2	3	2	3	**1**	2	3	4	**1**
F12	2.41	3.06	2.51	**2.02**	**3.07**	3.34	3.22	3.45	0.447	**0.187**	0.311	0.246
	2	4	3	**1**	**1**	3	2	4	4	**1**	3	2
F13	0.466	0.59	1.78	**0.345**	0.613	0.773	2.96	**0.458**	0.117	0.0935	0.500	**0.0853**
	2	3	4	**1**	2	3	4	**1**	3	2	4	**1**
F14	0.259	**0.219**	20.3	0.247	0.545	0.619	32.0	**0.35**	0.321	0.399	9.73	**0.146**
	3	**1**	4	2	2	3	4	**1**	2	3	4	**1**
F15	33.3	43.6	8117	**31.1**	38.2	50.4	2.51e + 4	**34.9**	**2.92**	5.42	2.12e + 4	3.03
	2	3	4	**1**	2	3	4	**1**	**1**	3	4	2
F16	22.0	22.1	22.1	**20.8**	22.5	22.5	22.5	**21.5**	0.248	**0.200**	0.211	0.460
	2	3	3	**1**	2	2	2	**1**	3	**1**	2	4
F17	1.44e + 6	9.75e + 6	9.75e + 6	**8.42e + 5**	4.11e + 6	1.86e + 7	1.65e + 7	**1.87e + 6**	2.93e + 6	9.91e + 6	4.90e + 6	**1.07e + 6**
	2	3	3	**1**	2	4	3	**1**	2	4	3	**1**
F18	4192	6430	7.40e + 7	**599**	1.71e + 5	1.40e + 5	1.10e + 8	**3153**	4.21e + 5	3.37e + 5	2.82e + 7	**2627**
	2	3	4	**1**	3	2	4	**1**	3	2	4	**1**
F19	**13.52**	18.5	98.4	13.6	40.3	54.2	120	**30.7**	32.5	35.6	17.4	**10.5**
	1	3	4	2	2	3	4	**1**	3	4	2	**1**
F20	1.70e + 4	1.90e + 4	1.35e + 4	**321**	3728	4.28e + 4	1.86e + 4	**520**	1504	21,174	5102	**83.8**
	3	4	2	**1**	2	4	3	**1**	2	4	3	**1**
F21	6.68e + 5	2.71e + 6	1.71e + 6	**2.42e + 5**	1.93e + 6	5.81e + 6	5.43e + 6	**9.64e + 5**	1.13e + 6	2.34e + 6	3.74e + 6	**4.42e + 5**
	2	4	3	**1**	2	4	3	**1**	2	3	4	**1**
F22	670	1411	989	**630**	1398	1609	1694	**1329**	293	198	334	**190**
	2	4	3	**1**	2	3	4	**1**	3	2	4	**1**

Table 3 *(continued)*

Number Functions	Best Rank				Mean Rank				Std. Rank			
	Rao-1	Rao-2	Rao-3	CCRAO	Rao-1	Rao-2	Rao-3	CCRAO	Rao-1	Rao-2	Rao-3	CCRAO
F23	**344.0**	**344**	403	**344.0**	**344**	**344**	426	**344**	1.4e-12	0.0948	24.3	**5.45e-13**
	1	**1**	2	**1**	**1**	**1**	2	**1**	2	3	4	**1**
F24	281.8	264	301	**254**	287	**271**	313	279	4.21	5.30	7.93	**2.82**
	3	2	4	**1**	3	**1**	4	2	2	3	4	**1**
F25	209	218	228	**206**	214	231	245	**213**	**5.56**	14.6	13.2	5.90
	2	3	4	**1**	2	3	4	**1**	**1**	4	3	2
F26	100.5	100.7	101	**100.4**	159	214	**140**	171	124	157	57.3	**48.4**
	2	3	4	**1**	2	4	**1**	3	3	4	2	**1**
F27	988	1559	1638	**927**	1219	1759	1780	**1112**	128	150	**73.5**	117
	2	3	4	**1**	2	3	4	**1**	3	4	**1**	2
F28	1380	1424	1664	**1330**	**1731**	1733	2182	2041	272	**258**	569	702
	2	3	4	**1**	**1**	2	4	3	2	**1**	3	4
F29	2.95e + 4	8275	8.48e + 6	**5316**	6.22e + 7	4.59e + 7	4.04e + 7	**2.78e + 7**	2.93e + 7	3.96e + 7	2.00e + 7	**1.91e + 7**
	3	2	4	**1**	4	3	2	**1**	3	4	2	**1**
F30	1.05e + 4	1.11e + 4	2.38e + 5	**1.03e + 4**	2.11e + 4	2.59e + 4	4.01e + 5	**1.94e + 4**	1.65e + 4	1.37e + 4	1.30e + 5	**9858**
	2	3	4	**1**	2	3	4	**1**	3	2	4	**1**
Nb/Mr	1/2.1667	3/2.8333	1/3.4667	27/1.10	4/2.10	4/2.8333	2/3.2333	25/1.3333	3/2.5333	4/2.8667	1/3.20	22/1.40

to a great level, it is better in terms of average ranking basis as listed in Table 8.

4.3. Applications of CCRAO in engineering

The proposed CCRAO is applied to some real optimization problems so that its effectiveness is verified. The size of swarm and value of iterations are respectively assumed to be 50 and 2000. Algorithm are executed thirty times to provide a fair comparison. Moreover, parameters are adjusted based on some original papers.

4.3.1. Optimal designing of a pressure vessel

The optimal design of a pressure vessel problem (problem 1) minimizes the total cost of a pressure vessel constrained by material, shaping, and welding costs. The problem consists of two discrete (y_1 and y_2) and two continuous (y_3 and y_4) decision variables as given in Fig. 4, where y_1 or T_s denotes the thickness of the shell, y_2 or T_h represents the thickness of the head, y_3 or R is the inner radius, and y_4 or L shows the length of the cylindrical section of the vessel. The nonlinear objective function of this problem with various inequality constraints are written by [60]:

Minimize:

$$f(Y) = 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 + 3.1661y_1^2y_4 + 19.84y_1^2y_3 \quad (5)$$

Subject to:

$$g_1(Y) = -y_1 + 0.0193y_3 \leqslant 0, \quad (6)$$

$$g_2(Y) = -y_2 + 0.00954y_3 \leqslant 0, \quad (7)$$

$$g_3(Y) = -\pi y_3^2 y_4 - \frac{4}{3}\pi y_3^3 + 1,296,000 \leqslant 0, \quad (8)$$

$$g_4(Y) = y_4 - 240 \leqslant 0, \quad (9)$$

$10 \leqslant y_i \leqslant 200,\ i = 3, 4$ and$0 \leqslant y_i \leqslant 100,\ i = 1, 2.$

Table 9 lists the obtained results by simulating the CCRAO for the pressure vessel design and compares it with some popular optimizers introduced in [60]-[82]. Based on Table 9, the fitness function value obtained through CCRAO is 6059.714335. Table 10 shows the best solutions to the pressure vessel optimal design.

4.3.2. Three-bar truss optimal design

The purpose of this optimal design (problem 2) is to reduce the volume of a statistically loaded three-bar truss constrained by constraints of bar's stress (σ), depicted in Fig. 5. The following equations express the problem subject to three constraints and two design parameters [60], i.e., the cross-sectional areas, $A_1(y_1)$and$A_2(y_2)$, which are shown in Fig. 5:

Minimize:

$$f(Y) = H * \left(2\sqrt{2}y_1 + y_2\right). \quad (10)$$

Subject to:

$$g_1(Y) = P * \frac{\sqrt{2}y_1 + y_2}{\sqrt{2}y_1^2 + 2y_1y_2} - \sigma \leqslant 0, \quad (11)$$

Table 4 Parameters of tested optimization algorithms for $D = 30$.

Algorithm	Parameters
ICA [54]	$\beta = 2, \gamma = \pi/5, \xi = 0.15, N_{country}$ (size of population of algorithm) = 30; N_{imp} (size initial population of empires) = 5.
MICA [54]	$\beta_{imp} = \beta = 2, \gamma_{imp} = \gamma = \pi/5, \xi = 0.15, N_{country}$ (size of population of algorithm) = 30; N_{imp} (size initial population of empires) = 5.
HHO [55]	The size of population of algorithm = 45
SAP-Rao [56]	The size of population of algorithm = 40, size of the groups of algorithm = 4.
QO-Rao-2 [57]	The size of population of algorithm = 40.
CCRAO	The size of population of algorithm = 32, $N_C = 3$.
GWO [58]	Wolves number $N = 45$, convergence constant $a = [2\ 0]$.

$$g_2(Y) = P * \frac{y_2}{\sqrt{2}y_1^2 + 2y_1y_2} - \sigma \leqslant 0, \tag{12}$$

$$g_3(Y) = P * \frac{1}{\sqrt{2}y_2 + y_1} - \sigma \leqslant 0, \tag{13}$$

$0 \leqslant y_i \leqslant 1,\ i = 1, 2.$

$\sigma = 2\text{kN/cm}^2,\ P = 2\text{kN/cm}^2,\ H = 100$ cm.

Table 11 tabulates the results obtained by simulating CCRAO for this optimal design and compares its performance with several standard algorithms like MBA [83], RL-BA [33], CSA [60].

According to Table 11, the proposed CCRAO is superior to RL-BA, DSS-MDE, PSO-DE, CSA, and HEAA. The standard deviation (Std.) of the solution is 3.6e-014, which provides the most reliable and robust solution for the design problem. Moreover, Table 12 provides the best solutions achieved in this problem.

4.3.3. Optimal designing of a welded beam

The optimal designing (problem 3) of a welded beam problem aims to discover the minimum cost of a welded beam. The problem contains four continuous decision variables, y_1 or h, y_2 or l, y_3 or t, and y_4 or b, illustrated in Fig. 6.

The following describes the nonlinear objective function of the problem, which consists of five nonlinear and two linear inequality constraints including bucking load on the bar (P_b), bending stress in the beam (σ), shear stress (τ), end deflection of the beam (δ) [60]:

Minimize:

$$f(Y) = 1.10471y_2y_1^2 + 0.04811y_3y_4(14 + y_2) \tag{14}$$

Subject to:

$$g_1(Y) = \tau(y) - \tau_{\max} \leqslant 0, \tag{15}$$

$$g_2(Y) = \sigma(y) - \sigma_{\max} \leqslant 0, \tag{16}$$

$$g_3(Y) = y_1 - y_4 \leqslant 0, \tag{17}$$

$$g_4(Y) = 0.10471y_1^2 + 0.04811y_3y_4(14 + y_2) - 5 \leqslant 0. \tag{18}$$

$$g_5(Y) = 0.125 - y_1 \leqslant 0, \tag{19}$$

$$g_6(Y) = \delta(y) - \delta_{\max} \leqslant 0, \tag{20}$$

$$g_7(Y) = P - P_c(y) \leqslant 0, \tag{21}$$

Subject to:

In this equation:

$$\tau(y) = \sqrt{(\tau)^2 + 2\tau\tau\frac{y_2}{2R} + (\tau)^2} \tag{22}$$

$$\tau = \frac{P}{\sqrt{2}y_1y_2},\ \tau = \frac{MR}{J} \tag{23}$$

$$M = P\left(L + \frac{y_2}{2}\right),\ R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2},\ \delta(y) = \frac{4PL^3}{Ey_3^3y_4} \tag{24}$$

$$J = 2\left[\sqrt{2}y_1y_2\left\{\frac{y_2^2}{12} + \left(\frac{y_1 + y_3}{2}\right)^2\right\}\right],\ \sigma(y) = \frac{6PL}{y_4y_3^2}, \tag{25}$$

$$P_c(y) = \frac{4.013E\sqrt{\frac{y_4^6y_3^2}{36}}}{L^2}\left(1 - \frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right), \tag{26}$$

$\delta_{\max} = 0.25$ in, $\tau_{\max} = 13{,}000$ psi, $\sigma_{\max} = 30{,}000$ psi, $G = 12e6$ psi, $E = 30e6$ psi, $L = 14$ in, $P = 6{,}000$ lb.

$0.1 \leqslant y_1, y_4 \leqslant 20.1 \leqslant y_2, y_3 \leqslant 10$

Table 13 lists the results provided by simulating the CCRAO for this problem and makes a comparison between it and standard algorithms such as those given in Table 13. This table illustrates the fitness function value found by the proposed CCRAO is 1.724852. The CCRAO is amongst the most reliable and robust approaches to optimizing a typical problem. Table 14 tabulates the optimal results found in problem 3.

4.3.4. Optimal designing of tension/compression spring

The problem 4 minimizes the weight of a tension/compression spring limited by material, shaping, and welding costs. The problem includes three continuous design variables, Fig. 7, where P or y_3 is the number of active coils, D or y_2 represents

Table 5 Parameters of tested optimization algorithms for $D = 50$.

Algorithm	Parameters
ICA [54]	$\beta = 2, \gamma = \pi/5, \xi = 0.15, N_{country}$ (size of population of algorithm) = 45; N_{imp} (size initial population of empires) = 5.
MICA [54]	$\beta_{imp} = \beta = 2, \gamma_{imp} = \gamma = \pi/5, \xi = 0.15, N_{country}$ (size of population of algorithm) = 45; N_{imp} (size initial population of empires) = 5.
HHO [55]	The size of population of algorithm = 60
SAP-Rao [56]	The size of population of algorithm = 60, size of the groups of algorithm = 4.
QO-Rao-2 [57]	The size of population of algorithm = 60.
CCRAO	The size of population of algorithm = 32, $N_C = 3$.
GWO [58]	Wolves number $N = 45$, convergence constant $a = [2\ 0]$.

Table 6 Brief results of CEC2014 test functions for various algorithms when $D = 30$.

Function		MICA	GWO	ICA	HHO	SAP-Rao	QO-Rao-2	RAO1	RAO2	RAO3	CCRAO
F1	Unimodal	2.86E+06	4.60E+07	1.28E+07	8.74E+06	6.49E+06	3.75E+06	9.34E+06	4.89E+07	8.35E+07	**6.75E+05**
		6.14E+05	3.13E+07	1.10E+07	3.54E+06	3.57E+06	3.00E+06	2.99E+06	3.42E+07	4.20E+07	**1.18E+05**
		2/−	8/−	7/−	6/−	4/−	3/−	5/−	9/−	10/−	**1**
F2		9.22E+02	1.04E+09	1.58E+03	7.26E+05	1.33E+02	9.83E+01	1.26E+02	7.01E+05	5.99E+09	**1.12E+01**
		5.28E+02	1.60E+09	3.24E+02	2.43E+05	2.96E+01	6.47E+01	9.72E+01	5.33E+05	3.04E+09	**9.24E+00**
		5/−	9/−	6/−	8/−	4/−	2/−	3/−	7/−	10/−	**1**
F3		3.19E+03	2.95E+04	4.49E+04	1.70E+03	1.90E+03	9.25E+02	5.00E+03	4.11E+04	3.45E+04	**5.48E+02**
		8.75E+02	1.08E+04	9.28E+03	1.36E+03	7.19E+02	8.70E+02	3.26E+03	1.35E+04	8.75E+03	**4.00E+02**
		5/−	7/−	10/−	3/−	4/−	2/−	6/−	9/−	8/−	**1**
F4	Simple	1.38E+02	1.86E+02	1.24E+02	6.58E+01	5.49E+01	4.81E+01	5.96E+01	1.28E+02	3.39E+02	**2.09E+01**
	Multimodal	6.14E+01	5.16E+01	7.42E+01	3.29E+01	3.64E+01	2.95E+01	4.53E+01	4.00E+01	7.03E+01	**1.49E+01**
		8/−	9/−	6/−	5/−	3/−	2/−	4/−	7/−	10/−	**1**
F5		2.09E+01	2.09E+01	2.10E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	**2.07E+01**
		9.49E-02	5.87E-02	2.14E-01	9.40E-02	3.47E-01	7.08E-02	7.35E-02	8.64E-02	6.20E-02	**4.25E-02**
		2/−	3/−	2/−	2/−	2/−	2/−	2/−	2/−	2/−	**1**
F6		1.16E+01	1.22E+01	2.32E+01	**4.32E-01**	1.34E+01	1.42E+01	1.60E+01	2.98E+01	3.12E+01	9.79E+00
		7.46E+00	2.72E+00	8.50E+00	**8.50E-02**	5.63E+00	3.20E+00	6.97E+00	6.39E+00	3.87E+00	7.18E+00
		3/−	4/−	8/−	**1/+**	5/−	6/−	7/−	9/−	10/−	2
F7		4.17E-03	1.14E+01	6.37E-02	2.67E-03	4.57E-03	**5.22E-13**	5.86E-03	1.00E-01	1.49E+01	3.69E-03
		2.34E-03	1.33E+01	7.94E-01	3.81E-02	1.67E-03	**8.69E-13**	6.72E-03	2.40E-01	6.31E+00	7.21E-03
		4/−	9/−	7/−	2/+	5/−	**1/+**	6/−	8/−	10/−	3
F8		1.92E+01	7.06E+01	1.35E+01	1.86E+01	1.46E+01	9.83E-13	1.24E+02	1.93E+02	1.96E+02	**7.11E-13**
		3.28E+00	1.58E+01	5.17E+00	4.59E+00	6.14E+00	4.12E-13	5.11E+01	3.00E+01	3.18E+01	**3.43E-13**
		6/−	7/−	3/−	5/−	4/−	2/−	8/−	9/−	10/−	**1**
F9		**9.59E+0**	8.87E+01	2.84E+02	3.76E+02	2.19E+02	1.98E+02	2.30E+02	2.53E+02	2.46E+02	1.45E+02
		3.18E+01	1.91E+01	9.35E+01	5.22E+01	1.74E+01	1.65E+01	2.00E+01	1.81E+01	3.62E+01	7.91E+00
		1/+	2/+	9/−	10/−	5/−	4/−	6/−	8/−	7/−	3
F10		5.97E+03	2.09E+03	5.21E+03	5.90E+03	4.12E+03	4.17E+03	5.87E+03	4.19E+03	4.49E+03	**1.14E+03**
		5.13E+02	5.21E+02	1.31E+03	6.70E+02	7.13E+02	7.20E+02	3.95E+02	8.79E+02	1.24E+03	**9.35E+02**
		10/−	2/−	7/−	9/−	3/−	4/−	8/−	5/−	6/−	**1**
F11		4.02E+03	**2.70E+03**	4.88E+03	5.17E+03	6.01E+03	5.82E+03	6.87E+03	6.67E+03	6.92E+03	3.83E+03
		3.42E+02	**6.16E+02**	6.00E+02	1.32E+03	9.34E+02	1.62E+03	5.02E+02	4.16E+02	3.97E+02	2.14E+02
		3/−	**1/+**	4/−	5/−	7/−	6/−	9/−	8/−	10/−	2
F12		1.02E+00	1.44E+00	5.13E-01	1.21E+00	2.42E+00	2.09E+00	2.40E+00	2.59E+00	2.56E+00	**1.08E-01**
		4.57E-01	1.18E+00	6.12E-01	4.50E-01	3.31E-01	9.15E-01	4.10E-01	2.82E-01	2.98E-01	**2.26E-01**
		3/−	5/−	2/−	4/−	8/−	6/−	7/−	10/−	9/−	**1**
F13		5.16E-01	**3.49E-01**	5.23E-01	3.60E-01	5.06E-01	4.59E-01	5.14E-01	5.49E-01	1.48E+00	3.57E-01
		6.17E-02	**7.67E-02**	3.40E-01	7.15E-02	2.00E-01	7.64E-02	2.34E-01	9.21E-02	3.96E-01	7.69E-02
		7/−	**1/+**	8/−	3/−	6/−	4/−	5/−	9/−	10/−	2
F14		4.76E-01	2.35E+00	6.85E-01	4.71E-01	5.24E-01	5.15E-01	5.41E-01	8.04E-01	8.01E+00	**2.51E-01**
		3.16E-01	4.22E+00	2.64E-02	8.52E-02	6.95E-02	9.32E-02	3.02E-01	2.98E-01	5.15E+00	**6.19E-02**
		3/−	9/−	7/−	2/−	5/−	4/−	6/−	8/−	10/−	**1**
F15		1.83E+01	1.97E+01	1.64E+01	2.35E+01	1.58E+01	1.60E+01	1.68E+01	2.28E+01	7.14E+01	**1.43E+01**
		7.16E+00	1.24E+01	1.91E+01	6.00E+00	5.84E+00	8.37E+00	2.36E+00	9.86E+00	4.56E+01	**3.17E+00**
		6/−	7/−	4/−	9/−	2/−	3/−	5/−	8/−	10/−	**1**

(*continued on next page*)

Table 6 (*continued*)

Function		MICA	GWO	ICA	HHO	SAP-Rao	QO-Rao-2	RAO1	RAO2	RAO3	CCRAO
F16		1.30E+01	1.07E+01	1.32E+01	1.37E+01	1.28E+01	1.28E+01	1.28E+01	1.28E+01	1.28E+01	**1.03E+01**
		3.20E-01	7.31E-01	6.91E-01	6.53E-01	5.24E-01	4.90E-01	9.21E-01	7.63E-01	8.43E-01	**4.52E-01**
		4/−	2/−	5/−	6/−	3/−	3/−	3/−	3/−	3/−	**1**
F17	Hybrid	6.24E+05	1.94E+06	3.82E+06	**1.10E+05**	8.42E+05	7.19E+05	9.70E+05	4.13E+06	2.63E+06	1.46E+05
		6.89E+04	1.74E+06	2.86E+06	**9.31E+04**	1.36E+06	1.94E+05	5.12E+05	2.62E+06	7.89E+05	9.66E+04
		3/−	7/−	9/−	**1/+**	5/−	4/−	6/−	10/−	8/−	2
F18		3.14E+02	1.16E+03	3.04E+03	2.38E+02	6.69E+02	5.43E+02	1.98E+05	2.97E+07	3.71E+07	**1.24E+02**
		9.19E+01	1.26E+03	8.27E+02	1.14E+06	5.30E+03	9.45E+03	5.35E+05	7.68E+07	2.90E+07	**8.15E+01**
		3/−	6/−	7/−	2/−	5/−	4/−	8/−	9/−	10/−	**1**
F19		3.66E+01	3.88E+01	3.74E+01	1.75E+01	1.12E+01	1.18E+01	1.26E+01	2.53E+01	2.60E+01	**1.03E+01**
		1.95E+01	2.14E+01	1.13E+01	1.83E+01	1.23E+00	1.76E+00	1.72E+00	3.53E+01	2.21E+00	**3.41E+01**
		8/−	10/−	9/−	5/−	2/−	3/−	4/−	6/−	7/−	**1**
F20		**7.75E+02**	1.40E+03	1.45E+03	2.60E+03	1.34E+03	7.95E+02	1.51E+03	4.19E+03	3.82E+03	8.87E+02
		5.11E+01	7.48E+02	2.53E+03	1.21E+03	6.64E+02	2.42E+02	7.68E+02		2.63E+03	8.56E+01
		1/+	5/−	6/−	8/−	4/−	2/−	7/−	1.89E+03	9/−	3
									10/−		
F21		4.93E+05	2.32E+05	4.77E+05	9.62E+05	1.47E+05	1.38E+05	2.91E+05	1.26E+06	7.24E+05	**1.20E+05**
		3.83E+05	2.30E+05	2.56E+05	5.65E+05	6.94E+04	1.49E+05	1.90E+05	1.36E+06	4.02E+05	**6.93E+04**
		7/−	4/−	6/−	9/−	3/−	2/−	5/−	10/−	8/−	**1**
F22		5.02E+02	**2.60E+02**	7.14E+02	2.74E+02	4.40E+02	3.97E+02	5.90E+02	4.62E+02	6.71E+02	2.65E+02
		4.62E+02	**1.26E+02**	3.70E+02	1.19E+02	1.99E+02	1.81E+02	1.71E+02	1.54E+02	1.39E+02	1.06E+02
		7/−	**1/+**	10/−	3/−	5/−	4/−	8/−	6/−	9/−	2
F23	Composition	3.18E+02	3.28E+02	3.30E+02	3.32E+02	**3.15E+02**	**3.15E+02**	**3.15E+02**	**3.15E+02**	3.34E+02	**3.15E+02**
		7.24E-03	4.13E+00	4.55E+00	6.16E+00	**6.89E-13**	**6.10E-13**	**9.54E-13**	**8.90E-03**	8.15E+00	**4.21E-13**
		2/−	3/−	4/−	5/−	**1/=**	**1/=**	**1/=**	**1/=**	6/−	**1**
F24		2.11E+02	**2.00E+02**	2.15E+02	2.08E+02	**2.00E+02**	**2.00E+02**	2.39E+02	2.23E+02	**2.00E+02**	**2.00E+02**
		7.74E+00	**5.91E-04**	8.98E+00	5.67E+00	**7.41E-02**	**6.24E-02**	8.56E+00	1.19E+01	**5.60E-02**	**5.85E-02**
		3/−	**1/=**	4/−	2/−	**1/=**	**1/=**	6/−	5/−	**1/=**	**1**
F25		2.04E+02	2.11E+02	2.12E+02	2.10E+02	2.05E+02	2.05E+02	2.07E+02	2.14E+02	2.16E+02	**2.02E+02**
		3.90E+00	1.88E+00	7.32E+01	1.89E+01	6.47E+00	3.34E+00	5.21E+00	5.10E+00	4.42E+00	**8.01E-01**
		2/−	6/−	7/−	5/−	3/−	3/−	4/−	8/−	9/−	**1**
F26		1.03E+02	1.10E+02	1.09E+02	1.06E+02	1.10E+02	1.08E+02	1.24E+02	1.63E+02	1.14E+02	**1.00E+02**
		9.12E-02	3.15E+01	5.14E-01	1.42E-01	2.25E-01	2.78E-01	8.10E+01	1.08E+02	2.45E+01	**1.98E-01**
		2/−	6/−	5/−	3/−	6/−	4/−	8/−	9/−	7/−	**1**
F27		7.05E+02	6.02E+02	1.34E+03	1.95E+03	6.57E+02	**5.41E+02**	7.71E+02	9.69E+02	1.01E+03	5.48E+02
		9.16E+01	9.77E+01	2.78E+02	1.90E+02	2.14E+02	**2.32E+02**	1.86E+02	2.00E+02	8.68E+01	9.20E+01
		5/−	3/−	9/−	10/−	4/−	**1/+**	6/−	7/−	8/−	2
F28		1.30E+03	8.89E+02	2.62E+03	7.68E+02	9.40E+02	9.32E+02	1.26E+03	9.90E+02	1.35E+03	**6.56E+02**
		8.02E+02	5.31E+01	5.05E+02	9.40E+02	7.31E+02	6.45E+02	3.42E+02	8.25E+01	3.47E+02	**3.46E+01**
		8/−	3/−	10/−	2/−	5/−	4/−	7/−	6/−	9/−	**1**
F29		2.28E+06	**9.92E+04**	2.49E+06	5.61E+05	2.45E+06	2.70E+06	9.49E+06	3.61E+06	3.12E+06	4.00E+05
		1.14E+06	**1.61E+05**	2.33E+06	2.13E+05	1.28E+06	1.14E+06	8.00E+06	3.37E+06	3.49E+06	5.69E+05
		4/−	**1/+**	6/−	3/−	5/−	7/−	10/−	9/−	8/−	2
F30		7.24E+03	3.72E+04	1.20E+04	5.95E+04	3.59E+03	**3.16E+03**	2.19E+04	4.39E+03	9.81E+03	3.40E+03
		4.51E+03	3.51E+04	1.13E+04	3.06E+04	1.42E+03	**1.30E+04**	2.35E+04	1.38E+03	6.79E+03	1.86E+03

Table 6 (*continued*)

Function	MICA	GWO	ICA	HHO	SAP-Rao	QO-Rao-2	RAO1	RAO2	RAO3	CCRAO
	5/−	9/−	7/−	10/−	3/−	**1/+**	8/−	4/−	6/−	2
+/−/=	2/28/0	5/24/1	0/30/0	3/27/0	0/28/2	3/25/2	0/29/1	0/29/1	0/29/1	—
Nb/*Mr*	2/4.5333	5/4.6667	0/6.4667	2/4.9333	2/4.0667	5/3.1667	1/5.9333	1/7.30	1/8.0	**19/1.4667**
Final rank	4	5	8	6	3	2	7	9	10	**1**

Table 7 Brief results of CEC2014 test functions for various algorithms when $D = 50$.

Function		MICA	GWO	HHO	ICA	SAP-Rao	QO-Rao-2	CCRAO
F1	Unimodal	9.32E+06	8.14E+08	1.05E+07	3.17E+07	1.75E+07	3.65E+06	**2.17E+06**
		3.61E+06	7.29E+06	1.14E+07	4.82E+06	2.64E+06	2.16E+06	**1.37E+06**
		3/−	7/−	4/−	6/−	5/−	2/−	**1**
F2		3.75E+03	3.27E+09	2.62E+06	4.91E+03	8.19E+04	5.50E+03	**2.89E+03**
		5.81E+03	1.68E+09	2.47E+05	2.57E+03	4.25E+03	6.95E+03	**3.23E+03**
		2/−	7/−	6/−	3/−	5/−	4/−	**1**
F3		8.45E+03	8.34E+04	4.58E+03	6.29E+04	9.48E+03	1.80E+03	**9.95E+02**
		5.24E+03	5.14E+03	5.93E+02	1.12E+04	7.53E+02	1.46E+03	**2.45E+02**
		4/−	7/−	3/−	6/−	5/−	2/−	**1**
F4	Simple	2.04E+02	6.78E+02	8.75E+01	1.35E+02	9.20E+01	8.32E+01	**5.78E+01**
	Multimodal	6.73E+01	5.13E+01	3.29E+01	7.28E+01	6.43E+01	4.91E+01	**2.28E+01**
		6/−	7/−	3/−	5/−	4/−	2/−	**1**
F5		2.10E+01	2.11E+01	2.10E+01	2.08E+01	2.13E+01	**2.07E+01**	2.11E+01
		4.92E−02	5.25E−02	3.82E−02	4.19E−03	5.69E−02	**5.25E−02**	1.75E−02
		3/+	4/=	3/+	2/+	5/−	**1/+**	4
F6		2.79E+01	3.65E+01	**8.37E−01**	3.15E+01	4.33E+01	2.82E+01	2.71E+01
		5.29E+00	7.24E+00	**7.59E−01**	9.73E+00	6.51E+00	5.12E+00	4.56E+00
		3/−	6/−	**1/+**	5/−	7/−	4/−	2

(*continued on next page*)

Table 7 (*continued*)

Function		MICA	GWO	HHO	ICA	SAP-Rao	QO-Rao-2	CCRAO
F7		8.49E−03	3.21E+01	3.66E−03	9.48E−02	1.80E−01	**7.99E−13**	4.18E−03
		8.53E−03	9.18E+00	9.72E−04	6.64E−03	5.36E−02	**2.64E−13**	9.13E−03
		4/−	7/−	2/+	5/−	6/−	**1/+**	3
F8		3.56E+01	1.42E+02	4.03E+01	2.70E+01	5.74E+01	**8.20E−13**	2.61E+01
		2.00E+01	3.55E+01	1.78E+01	2.18E+01	6.77E+01	**5.39E−14**	3.12E+01
		4/−	7/−	5/−	3/−	6/−	**1/+**	2
F9		**1.95E+02**	2.27E+02	5.94E+02	8.93E+02	5.39E+02	5.63E+02	3.89E+02
		3.82E+01	3.69E+01	8.64E+01	5.34E+01	1.45E+02	3.47E+01	2.29E+01
		1/+	2/−	6/+	7/−	4/−	5/−	3
F10		1.39E+04	**8.27E+03**	3.60E+04	7.19E+04	3.27E+05	5.66E+04	1.22E+04
		8.24E+02	**6.96E+02**	6.73E+02	5.16E+03	9.50E+04	5.35E+02	5.18E+02
		3/−	**1/+**	4/−	6/−	7/−	5/−	2
F11		4.74E+04	6.42E+04	2.75E+04	5.35E+04	2.48E+04	3.90E+04	**1.27E+04**
		5.89E+03	8.14E+02	1.15E+03	1.98E+03	1.35E+03	3.81E+02	**2.86E+02**
		5/−	7/−	3/−	6/−	2/−	4/−	**1**
F12		3.72E+00	3.40E+00	4.01E+00	8.35E−01	3.86E+00	**3.00E+00**	3.45E+00
		2.86E−01	4.00E−01	8.49E−01	9.47E−02	5.34E−01	**5.14E−01**	2.46E−01
		4/−	2/+	6/−	7/−	5/−	**1/+**	3
F13		5.45E−01	5.12E−01	4.85E−01	6.31E−01	4.90E−01	5.68E−01	**4.58E−01**
		2.95E−01	4.62E−02	6.98E−02	1.96E−01	5.93E−02	1.10E−01	**8.53E−02**
		5/−	4/−	2/−	7/−	3/−	6/−	**1**
F14		5.00E−01	5.18E−01	3.72E−01	5.80E−01	3.75E−01	4.54E−01	**3.50E−01**
		1.87E−01	9.02E−02	9.61E−02	3.14E−01	9.43E−02	8.47E−02	**1.46E−01**
		5/−	6/−	2/−	7/−	3/−	4/−	**1**
F15		5.74E+01	1.83E+02	6.35E+01	9.81E+01	**2.26E+01**	5.59E+01	3.49E+01
		6.11E+00	5.37E+01	7.20E+00	2.20E+01	**1.87E+01**	6.63E+00	3.03E+00
		4/−	7/−	5/−	6/−	**1/+**	3/−	2
F16		2.18E+01	2.18E+01	3.09E+01	**2.15E+01**	2.16E+01	**2.15E+01**	**2.15E+01**
		7.21E−01	6.92E−01	4.41E−01	**7.13E−01**	5.82E−01	**8.75E−01**	**4.60E−01**
		3/−	3/−	4/−	**1/=**	2/−	**1/=**	**1**
F17	Hybrid	6.59E+06	5.24E+07	5.43E+06	8.20E+06	**1.57E+06**	4.73E+06	1.87E+06
		9.01E+05	8.75E+06	4.95E+06	3.45E+06	**1.93E+06**	2.36E+06	1.07E+06
		5/−	7/−	4/−	6/−	**1/+**	3/−	2
F18		4.58E+02	3.99E+03	3.52E+02	5.97E+03	3.72E+02	5.35E+03	**2.80E+02**
		3.26E+02	6.74E+02	2.84E+02	1.63E+03	7.40E+02	1.90E+03	**2.91E+02**
		4/−	5/−	2/−	7/−	3/−	6/−	**1**
F19		5.28E+01	9.26E+01	2.85E+01	7.86E+01	6.25E+01	1.82E+01	**1.21E+01**
		3.44E+01	4.35E+01	6.55E+00	2.32E+00	4.79E+00	2.73E+00	**1.60E+00**
		4/−	7/−	3/−	6/−	5/−	2/−	**1**
F20		2.40E+03	**2.19E+03**	1.29E+04	8.44E+04	5.38E+04	2.94E+03	3.15E+03
		9.32E+02	**1.05E+03**	5.63E+03	6.29E+04	4.42E+03	2.56E+03	2.63E+03
		2/+	**1/+**	5/−	7/−	6/−	3/+	4
F21		1.83E+06	2.79E+07	2.46E+06	1.87E+06	**3.20E+05**	3.97E+06	9.64E+05
		6.26E+05	3.51E+05	4.80E+06	6.10E+05	**8.35E+05**	9.62E+05	4.42E+05
		3/−	7/−	5/−	4/−	**1/+**	6/−	2

Table 7 (*continued*)

Function		MICA	GWO	HHO	ICA	SAP-Rao	QO-Rao-2	CCRAO
F22		**9.47E + 02**	1.00E + 03	2.04E + 03	1.75E + 03	9.18E + 03	6.26E + 03	1.33E + 03
		3.95E + 02	3.13E + 02	7.49E + 02	5.58E + 02	9.42E + 02	7.41E + 02	1.90E + 02
		1/ +	2/ +	5/ +	4/–	7/–	6/–	3
F23	Composition	3.46E + 02	**3.44E + 02**	3.50E + 02	3.50E + 02	3.52E + 02	**3.44E + 02**	**3.44E + 02**
		8.16E–01	**3.18E–10**	6.54E–10	5.21E–04	3.47E–02	**7.49E–13**	**5.45E–13**
		2/–	**1/ =**	3/–	3/–	4/–	**1/ =**	**1**
F24		2.95E + 02	2.84E + 02	3.16E + 02	2.90E + 02	2.80E + 02	**2.62E + 02**	2.79E + 02
		4.22E + 00	5.11E + 00	5.98E + 00	6.47E + 00	3.11E + 00	**1.30E + 01**	2.82E + 00
		6/–	4/–	7/–	5/–	3/–	**1/ +**	2
F25		2.20E + 02	2.25E + 02	2.20E + 02	2.30E + 02	2.18E + 02	2.19E + 02	**2.13E + 02**
		1.13E + 01	6.92E + 00	8.93E–01	9.19E + 00	6.72E + 00	7.30E + 00	**5.90E + 00**
		4/–	5/–	4/–	6/–	2/–	3/–	**1**
F26		1.66E + 02	**1.56E + 02**	1.86E + 02	1.68E + 02	2.32E + 02	1.93E + 02	1.71E + 02
		5.02E + 01	**6.87E + 01**	6.78E + 01	5.84E + 01	2.40E + 01	7.02E–02	4.84E + 01
		2/ +	**1/ +**	5/–	3/ +	7/–	6/–	4
F27		1.18E + 03	**6.99E + 02**	3.75E + 03	1.92E + 03	5.78E + 03	6.18E + 03	1.11E + 03
		4.57E + 02	**1.80E + 02**	1.09E + 02	8.67E + 01	9.35E + 03	4.91E + 02	1.17E + 02
		3/–	**1/ +**	5/–	4/–	6/–	7/–	2
F28		2.30E + 03	9.25E + 03	5.82E + 03	6.34E + 03	4.26E + 03	5.17E + 03	**2.04E + 03**
		5.49E + 02	3.65E + 02	5.45E + 02	1.07E + 03	9.47E + 02	9.86E + 01	**7.02E + 02**
		2/–	7/–	5/–	6/–	3–	4/–	**1**
F29		3.58E + 07	3.72E + 08	9.26E + 06	**4.55E + 06**	5.86E + 07	8.54E + 06	2.78E + 07
		1.05E + 07	2.83E + 07	8.64E + 05	**7.86E + 05**	3.34E + 07	3.62E + 06	1.91E + 07
		5/–	7/–	3/ +	**1/ +**	6/–	2/ +	4
F30		3.87E + 04	6.98E + 04	9.53E + 04	2.85E + 04	8.62E + 04	2.48E + 04	**1.94E + 04**
		2.10E + 04	2.36E + 04	5.41E + 03	1.26E + 04	1.53E + 04	1.30E + 04	**9.86E + 03**
		4/–	5/–	7/–	3/–	6/–	2/–	**1**
+ /–/ =		5/25/0	6/22/2	6/24/0	3/26/1	3/27/0	7/21/2	-
Nb/Mr		2/3.5333	5/4.8000	1/4.0667	2/4.9000	3/4.4333	7/3.2667	**14/1.9333**
Final rank		3	6	4	7	5	2	**1**

Table 8 The competitive results of Wilcoxon's test and performance of CCRAO versus other algorithms.

Corresponding algorithm	CCRAO versus	
	$D = 50$	$D = 30$
	p-values	
MICA	**1.7800E-05**	**3.5021E-06**
GWO	**1.3856E-05**	**5.8009E-06**
HHO	**8.8801E-07**	**2.9014E-09**
ICA	**9.2327E-08**	**6.0528E-10**
SAP-Rao	**3.3985E-06**	**5.9755E-08**
QO-Rao-1	**0.0053**	**9.1614E-04**

the mean coil diameter, and d or y_1 shows the wire diameter. The nonlinear objective function of the problem with the several various inequality constraints are provided by following equations [60]:

Minimize:

$$f(Y) = (y_3 + 2)y_2 y_1^2 \tag{26}$$

Subject to:

$$g_1(Y) = 1 - \frac{y_2^3 y_3}{71,785 y_1^4} \leqslant 0, \tag{27}$$

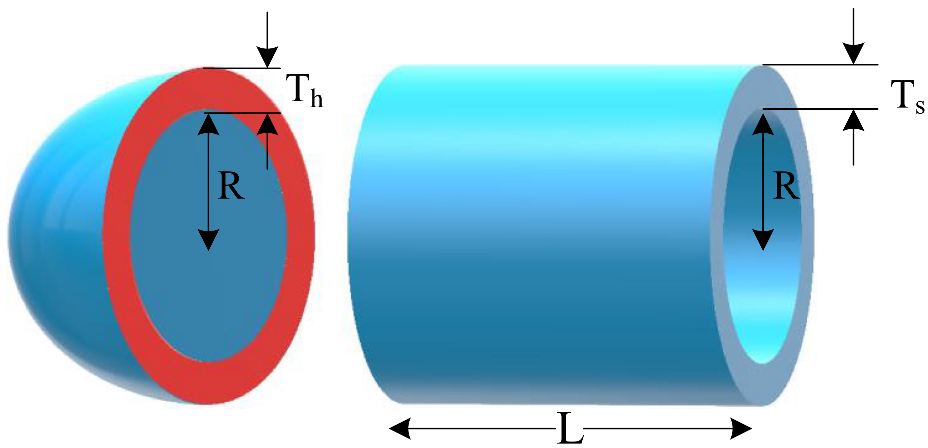

Fig. 4 The pressure vessel design.

Table 9 Optimal results of various optimizers for the problem 1.

Methods	Best	Mean	Worst	Std.
T-Cell [62]	6390.554	6737.065	7694.066	357
Sinusodial FFA [63]	6059.71441707	6061.25542415	6090.52625202	6.88963799
CPSO [64]	6061.0777	6147.1332	6363.8041	86.4545
DHOA [65]	6103.842	N.A.	N.A.	N.A.
NHAIS-GA [66]	6061.1229	6743.0848	7368.0602	457.99
G-QPSO [61]	6059.7208	6440.3786	7544.4925	448.4711
QS [67]	**6059.714**	6060.947	6090.526	N.A.
CSA [60]	6059.71436343	6342.49910551	7332.8416211	384.94541634
CB-ABC [68]	**6059.714335**	6126.623676	N.A.	1.14E + 02
HAIS-GA [69]	6832.584	7187.314	8012.615	276
QPSO [61]	6059.7209	6440.3786	8017.2816	479.2671
BFOA [70]	6060.460	6074.625	N.A.	156
ES [71]	6059.746	6850.00	7332.87	426
CDE [72]	6059.7340	6085.2303	6371.0455	43.013
GA3 [73]	6288.7445	6293.8432	6308.4970	7.4133
EO [74]	**6059.7143**	6668.114	7544.4925	566.24
CVI-PSO [75]	**6059.7143**	6292.1231	6820.4101	288.4550
PVS [76]	**6059.714**	6065.877	6090.526	N.A.
BA [77]	**6059.7143348**	6179.13	6318.95	137.223
BIANCA [78]	6059.9384	6182.0022	6447.3251	122.3256
UPSO [79]	6154.70	8016.37	9387.77	745.869
DEC-PSO [80]	**6059.714335**	6060.33057699	6090.52620169	4.35745530
GA4 [81]	6059.9463	6177.2533	6469.3220	130.9297
ABC [82]	6059.714339	6245.308144	N.A.	2.05E + 02
Rao-1 [50]	**6059.714334**	6069.230694	6093.903548	10.451664
Rao-2 [50]	**6059.714334**	6062.055668	6090.526202	7.171409
Rao-3 [50]	**6059.714334**	6061.883052	6090.526202	7.810982
CCRAO	**6059.71433**	**6060.28032**	**6075.93125**	**2.8927**

Table 10 The best solutions for the problem 1.

Design variables	**CCRAO**
y_1	0.8125
y_2	0.4375
y_3	42.098445590
y_4	176.63659592
$g_1(Y)$	−1.130000537585829e-10
$g_2(Y)$	−0.035880829071400
$g_3(Y)$	−2.788752317428589e-05
$g_4(Y)$	−63.363404080000009
Best	**6059.71433**

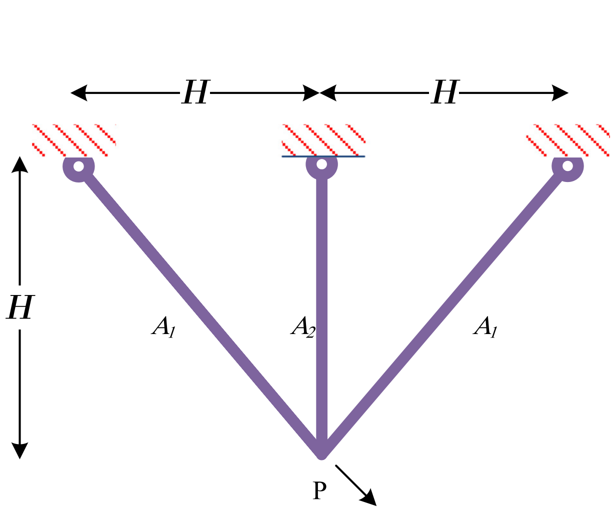

Fig. 5 Schematic of the three-bar truss design.

$$g_2(Y) = \frac{4y_2^2 - y_1y_2}{12,566(y_1^3y_2 - y_1^4)} + \frac{1}{5,108y_1^2} - 1 \leqslant 0, \tag{28}$$

$$g_3(Y) = 1 - \frac{140.45y_1}{y_2^2y_3} \leqslant 0, \tag{29}$$

Table 12 The best solutions for the optimal design problem 2.

Design variables	**CCRAO**
y_1	0.7886740
y_2	0.40825150
$g_1(Y)$	−3.091606970428984e-10
$g_2(Y)$	−1.464097966658180
$g_3(Y)$	−0.535902033650981
Best	**263.89584**

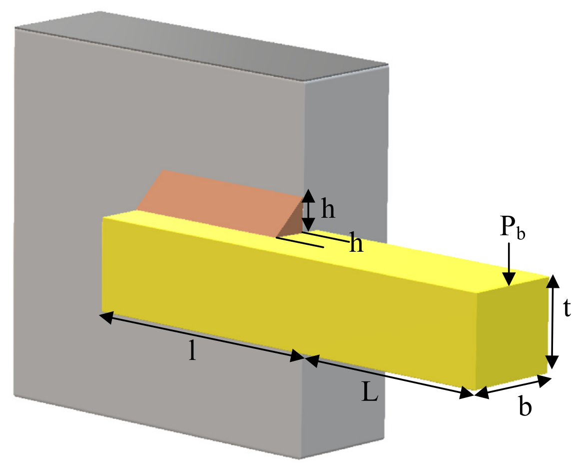

Fig. 6 The welded beam optimal design.

$$g_4(Y) = \frac{y_1 + y_2}{1.5} - 1 \leqslant 0. \tag{30}$$

$$0.05 \leqslant y_1 \leqslant 22 \leqslant y_3 \leqslant 150.25 \leqslant y_2 \leqslant 1.3 \tag{31}$$

In Table 15 the simulation results of the CCRAO for the problem 4 are given when a comparison is made between them and those of numerous standard algorithms mentioned in

Table 11 Optimal results of various optimizers applied to the problem 2.

Methods	Best	Mean	Worst	Std.
CS [85]	263.97156	264.0669	N.A.	9.0e-05
SFO [86]	263.89592128	N.A.	N.A.	N.A.
SBO [87]	263.8958466	263.90335672	263.96975638	1.3E-02
MBA [83]	263.895852	263.897996	263.915983	3.93E − 03
CSA [60]	263.8958433765	263.8958433765	263.8958433770	**1.0123E-10**
RL-BA [84]	263.89584	263.9003	263.924700	6.06E − 03
SDO [88]	263.895843	263.895845	263.895847	1.12e-06
PSO-DE [88]	263.89584338	263.89584338	263.89584338	1.2E-10
DSS-MDE [89]	263.8958434	263.8981518	263.95226	9.19E-03
HEAA [90]	263.895843	263.895865	263.896099	4.9e-05
Rao-1 [50]	263.895841	263.896207	263.897166	3.7109E-04
Rao-2 [50]	263.895846	263.897469	263.899734	1.0976E-03
Rao-3 [50]	263.895843	263.897243	263.899650	1.0390E-03
CCRAO	**263.89584**	**263.895842**	**263.895843**	3.9286E-07

Table 13 Optimal results of various optimizers for the problem 3.

Methods	Best	Mean	Worst	Std.
HGSO [91]	1.7260	1.7265	1.7325	7.66E-03
MRFO [92]	**1.7248523**	1.7248547	1.7248648	3.832E-06
BA [77]	1.7312065	1.8786560	2.3455793	2.677989E-01
EO [74]	1.724853	1.726482	1.736725	3.257E-03
IPSO [93]	2.3810	2.3819	N.A.	5.23E-03
BFOA [70]	2.3868	2.4040	N.A.	1.6E-02
GA4 [81]	1.728226	1.792654	1.993408	7.47E-02
CPSO [81]	1.728024	1.748831	1.782143	1.2926E-02
LFD [94]	1.77	2.30	3.04	3.16E-01
HSA-GA [95]	2.2500	2.26	2.28	7.8E-03
T-Cell [62]	2.3811	2.4398	2.7104	9.314E-02
RAER [96]	2.3816	N.A.	2.38297	3.4E-04
CDE [72]	1.73346	1.768158	1.824105	2.2194E-02
(l + λ)-ES [97]	**1.724852**	1.777692	N.A.	8.8E-02
HPSO [98]	**1.724852**	1.749040	1.814295	4.01E-02
SBO [89]	2.3854347	3.0025883	6.3996785	9.59E-01
SBM [99]	2.4426	2.5215	2.6315	N.A.
UPSO [64]	1.92199	2.83721	N.A.	6.83E-01
FSA [100]	2.3811	2.4041	2.4889	N.A.
SFO [88]	1.73231	N.A.	N.A.	N.A.
EPSO [101]	1.7248530	1.7282190	1.7472200	5.62E-03
Sinusodial FFA [63]	1.724868	1.724953	1.724868	4.6E-05
WCA [102]	1.724856	1.726427	1.744697	4.29E-03
CVI-PSO [75]	**1.724852**	1.725124	1.727665	6.12E-04
PFA [103]	1.7248530	N.A.	N.A.	N.A.
BIANCA [78]	1.725436	1.752201	1.793233	2.3001E-02
GWO [58]	1.72624	N.A.	N.A.	N.A.
Rao-1 [50]	**1.724852**	**1.724852**	**1.724852**	1.1387E-08
Rao-2 [50]	**1.724852**	**1.724852**	**1.724852**	1.0219E-08
Rao-3 [50]	**1.724852**	**1.724852**	**1.724852**	5.1434E-08
CCRAO	**1.724852**	**1.724852**	**1.724852**	**9.7241E-09**

Table 15. As per Table 15, the value of the fitness function obtained using CCRAO is 0.012665. Furthermore, CCRAO is recommended as one of the most reliable and robust approaches so that the typical problem can be solved. Table 16 tabulates the best solutions found for the problem 4.

4.3.5. Optimal designing of a gear train

Fig. 8 illustrates the schematic of a gear train optimal design (problem 5). The problem tries to give the best cost of the gear ratio of this system and includes only boundary limits of parameters. Parameters n_A (y_1), n_D (y_4), n_C (y_3), and n_B (y_2) are discrete decision variables as each gear should consist of an integral number of teeth. The objective function has been shown as [60]:

Minimize:

$$f(Y) = \left(\left(\frac{1}{6.931}\right) - \left(\frac{y_2 y_3}{y_1 y_4}\right)\right)^2 \tag{32}$$

$12 \leqslant y_i \leqslant 60, i = 1, 2, 3, 4.$

Table 17 lists the CCRAO simulation results of the problem 5. The table also compares the results with those of standard algorithms such as UPSO [79], MBA [83], CS [85]. Based on Table 17, the fitness function value found by CCRAO is 2.700857e-12. The suggested CCRAO provides the most reliable and robust approach to solve the optimization problem. Moreover, Table 18 finds the best solutions for the problem 5.

Table 14 The best solutions in the optimization problem 3.

Design variables	CCRAO
y_1	0.2057296398
y_2	3.47048866550
y_3	9.0366239101
y_4	0.20572963980
$g_1(Y)$	−2.265333023387939e-07
$g_2(Y)$	−3.193272277712822e-07
$g_3(Y)$	0.0
$g_4(Y)$	−3.432983785311915
$g_5(Y)$	−0.08072963980
$g_6(Y)$	−0.235540322584496
$g_7(Y)$	−1.105492628994398e-06
Best	**1.724852**

4.4. Discussions and prospect of the future

CCROA, several ROA algorithms, and some other algorithms are applied to five well-known engineering problems using CEC2014 standard benchmarks to make a comparison between them. According to this comparison one can observe that the proposed CCROA performs more acceptable and provides outstanding results besides showing great potential from

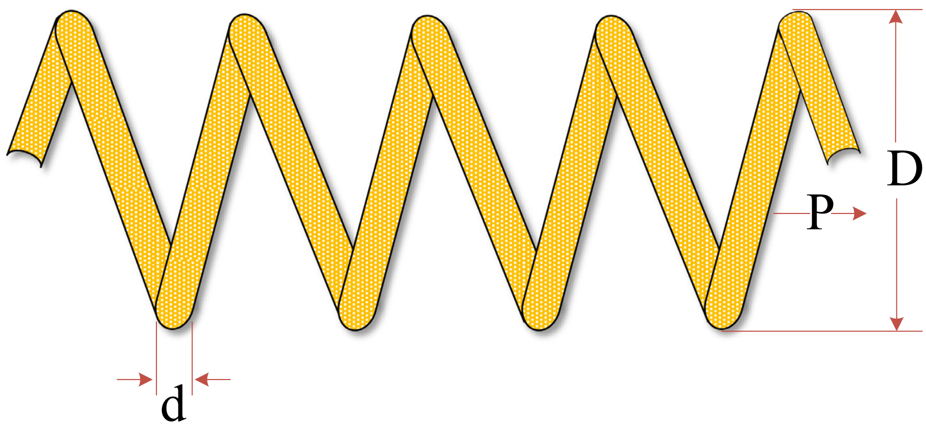

Fig. 7 The tension/compression spring optimal design.

Table 15 Optimal results of various optimizers for the problem 4.

Methods	Best	Mean	Worst	Std.
EO [74]	0.012666	0.013017	0.013997	3.91E-04
BFOA [72]	0.012671	0.012759	N.A.	1.36E-04
T-Cell [62]	**0.012665**	0.012732	0.013309	9.4E-05
CDE [72]	0.012670	0.012703	0.012790	2.07E-05
CPSO [64]	0.0126747	0.012730	0.012924	5.19E-05
SI [104]	0.013060	0.015526	0.018992	N.A.
CA [105]	0.012721	0.013568	0.0151156	8.4E-04
(1 + λ)-ES [97]	0.012689	0.013165	N.A.	3.9E-04
GA4 [81]	0.012681	0.012742	0.012973	9.5E-05
GA3 [73]	0.0127048	0.012769	0.012822	3.93E-05
HPSO [98]	**0.0126652**	0.0127072	0.0127190	1.58E-05
LFD [94]	0.0127	0.0138	0.0156	9.82E − 04
G-QPSO [61]	**0.012665**	0.013524	0.017759	1.268E-03
SBO [89]	0.012669249	0.012922669	0.016717272	5.92E-04
DSS-MDE [54]	**0.012665233**	0.012669366	0.012738262	1.25E-05
HGA [106]	0.012668	0.013481	0.016155	N.A.
UPSO [79]	0.01312	0.02294	N.A.	7.2E-03
GWO [58]	0.0126660	N.A.	N.A.	N.A.
SDO [86]	0.0126663	0.0126724	0.0126828	6.1899E-06
PFA [103]	**0.01266528**	N.A.	N.A.	N.A.
CVI-PSO [75]	0.0126655	0.012731	0.0128426	5.58E-05
Sinusodial FFA [63]	0.0126660517	0.0127072216	0.0127680353	2.57941E-05
DDAO [107]	0.0129065	0.0151829	0.0173199	1.26E-03
BA [77]	**0.01266522**	0.01350052	0.0168954	1.42027E-03
QS [67]	**0.012665**	**0.012666**	0.012669	N.A.
WCA [102]	**0.012665**	0.012746	0.012952	8.06E-05
BIANCA [78]	0.012671	0.012681	0.012913	5.1232E-05
Rao-1 [50]	0.012666	0.012712	0.012846	3.6195E-05
Rao-2 [50]	0.012669	0.013232	0.030455	2.5886E-03
Rao-3 [50]	0.012672	0.013086	0.017773	1.2062E-03
CCRAO	**0.012665**	**0.012666**	**0.012668**	**2.1639e-06**

exploration and exploitation aspects when applied to real-parameter (shifted) multimodal and enlarged multimodal functions, as well as in real-parameter unimodal functions.

Concerning the obtained results, it is seen on real-parameter composite and hybrid functions demonstrate that the CCRAO algorithm balances exploration and exploitation phases. CCRAO's average optimization results and standard deviation are like those generated by the modern optimizers. Convergence speed of CCROA is also much better than other algorithms. Therefore, CCRAO algorithm can be adopted for to complex objective functions of science and engineering fields in the upcoming years. In future works, researchers can focus on the impact of spirals on CCRAO's performance. Additionally, the authors recommend utilizing certain operators when solving multi-objective algorithms using the CCRAO algorithm.

Table 16 The best solutions for the design problem 4.

Design variables	**CCRAO**
y_1	0.0516233836
y_2	0.35513835890
y_3	11.382241176
$g_1(Y)$	−2.217884273347792e-06
$g_2(Y)$	−3.187804137771977e-06
$g_3(Y)$	−4.050626575050973
$g_4(Y)$	−0.728825505000000
Best	**0.012665**

Table 18 The best solutions in the problem 5.

Design variables	**CCRAO**
y_1	43
y_2	19
y_3	16
y_4	49
Best	**2.700857e-12**

5. Conclusion

The present study introduces a new colonial competitive optimizer using three modified RAO metaphor-less algorithms, called CCRAO. The CCRAO algorithm is investigated and modeled using mathematical equations. The standard CEC2014 functions with 30 and 50 dimensions are adopted to validate the algorithm and its convergence rate. To delve deeper into the applications of the suggested algorithm, several engineering optimization problems were incorporated, including optimal designs of a three-bar truss, gear train welded beam, pressure vessel, and tension/compression spring to show the optimization power of CCRAO. To prove the superiority of the CCRAO algorithm, a comprehensive analysis is performed to compare it with other popular modern algorithms. According to a comparative statistical analysis, the CCRAO algorithm is highly effective in finding optimal solutions with higher rates of convergence. The high performance of the CCRAO algorithm from convergence aspect, local optima avoidance, exploration and exploitation was shown in this study. The algorithm presented acceptable performance when adopting for solving practical engineering problems. According to simulations, the suggested CCRAO algorithm shows

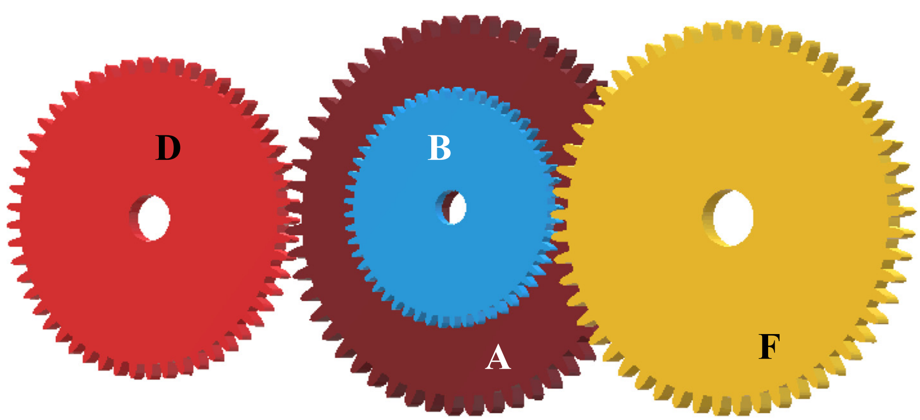

Fig. 8 Schematic of the gear train optimal design.

Table 17 Optimal results of various optimizers for the problem 5.

Methods	Best	Mean	Worst	Std.
CS [85]	2.7009E-12	1.9841E-9	2.3576E-9	3.5546E-9
UPSO [79]	**2.700857E-12**	3.80562E-8	N.A.	1.09E-07
MBA [83]	**2.700857E-12**	2.471635E-9	2.06290E-8	3.94E-09
Rao-1 [50]	**2.7009E-12**	9.6288E-08	1.2045E-06	2.6504E-07
Rao-2 [50]	**2.7009E-12**	5.7864E-08	8.9490E-07	1.6169E-07
Rao-[50]	**2.7009E-12**	1.2022E-07	8.9490E-07	2.5475E-07
CCRAO	**2.700857E-12**	**3.5216E-10**	**1.0063E-09**	**3.5216E-10**

robust and powerful performance when dealing with optimization problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R.V. Rao, V.J. Savsani, Mechanical design optimization using advanced optimization techniques, (2012).

[2] H. Garg, A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data, in: Handbook of Research on Artificial Intelligence Techniques and Algorithms, IGI Global, 2015: pp. 620–654.

[3] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Inf. Sci. 478 (2019) 499–523.

[4] T. Kundu, H. Garg, A hybrid ITLHHO algorithm for numerical and engineering optimization problems, Int. J. Intell. Syst. (2021).

[5] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305.

[6] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inf. Sci. 324 (2015) 217–246.

[7] N.A. Kallioras, N.D. Lagaros, D.N. Avtzis, Pity beetle algorithm–A new metaheuristic inspired by the behavior of bark beetles, Adv. Eng. Softw. 121 (2018) 147–166.

[8] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.

[9] M. Ghasemi, I.F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, L.i. Li, A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO), Eng. Appl. Artif. Intell. 92 (2020) 103666.

[10] S.H.A. Kaboli, J. Selvaraj, N.A. Rahim, Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems, J. Computat. Sci. 19 (2017) 31–42.

[11] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: a new heuristic optimization method, Neurocomputing. 221 (2017) 123–137.

[12] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.

[13] A.A. Hudaib, H.N. Fakhouri, Supernova optimizer: a novel natural inspired meta-heuristic, Mod. Appl Sci. 12 (2018) 32–50.

[14] W. Zhao, L. Wang, Z. Zhang, A novel atom search optimization for dispersion coefficient estimation in groundwater, Fut. Generat. Comput. Syst. 91 (2019) 601–610.

[15] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: a novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85.

[16] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.

[17] M. Ghasemi, S. Ghavidel, J. Aghaei, E. Akbari, L. Li, CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems, Int. Trans. Electr. Energy Syst. 28 (2018) e2536.

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

[19] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.

[20] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.

[21] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, Comput. Struct. 139 (2014) 18–27.

[22] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[23] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft. Comput. 23 (2019) 715–734.

[24] P. Pijarski, P. Kacejko, A new metaheuristic optimization method: the algorithm of the innovative gunner (AIG), Eng. Optim. 51 (12) (2019) 2049–2068.

[25] H.T. Kahraman, S. Aras, E. Gedikli, Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms, Knowl.-Based Syst. 190 (2020) 105169.

[26] A. Kaveh, A.D. Eslamlou, Water strider algorithm: A new metaheuristic and applications, in: Structures, Elsevier (2020) 520–541.

[27] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Fut. Generat. Comput. Syst. 111 (2020) 300–323.

[28] M. Zaeimi, A. Ghoddosian, Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization, Soft. Comput. 24 (2020) 12027–12066.

[29] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, Red deer algorithm (RDA): a new nature-inspired meta-heuristic, Soft. Comput. 24 (2020) 14637–14665.

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

[31] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010: pp. 65–74.

[32] Y. Xin-She, S. Deb, Cuckoo Search via Levy flights. in World Congress on Nature & Biologically Inspired Computing, 2009, NaBIC 2009. (2009).

[33] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: International Conference in Swarm Intelligence, Springer, 2010: pp. 355–364.

[34] F. Ramezani, S. Lotfi, Social-based algorithm (SBA), Appl. Soft. Comput. 13 (5) (2013) 2837–2856.

[35] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm Evol. Comput. 17 (2014) 14–24.

[36] A. Kaur, S. Jain, S. Goel, Sandpiper optimization algorithm: a novel approach for solving real-life engineering problems, Appl. Intellig. 50 (2020) 582–619.

[37] S. Kaur, L.K. Awasthi, A.L. Sangal, G. Dhiman, Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.

[38] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196.

[39] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.

[40] M. Dehghani, Z. Montazeri, G. Dhiman, O.P. Malik, R. Morales-Menendez, R.A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero, L. Parra-Arroyo, A spring search algorithm

applied to engineering optimization problems, Appl. Sci. 10 (2020) 6173.
[41] M. Ghasemi, A. Rahimnejad, R. Hemmati, E. Akbari, S.A. Gadsden, Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese, Array 11 (2021) 100074.
[42] Z. Wei, C. Huang, X. Wang, T. Han, Y. Li, Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization, IEEE Access 7 (2019) 66084–66109.
[43] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333, https://doi.org/10.1016/j.asoc.2015.07.028.
[44] S.J. Mousavirad, H. Ebrahimpour-Komleh, Human mental search: a new population-based metaheuristic optimization algorithm, Appl. Intellig. 47 (2017) 850–887.
[45] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[46] R. Rao, Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems, Int. J. Indust. Eng. Computat. 11 (2020) 107–130.
[47] L. Wang, Z. Wang, H. Liang, C. Huang, Parameter estimation of photovoltaic cell model with Rao-1 algorithm, Optik. 210 (2020) 163846.
[48] M. Premkumar, T.S. Babu, S. Umashankar, R. Sowmya, A new metaphor-less algorithms for the photovoltaic cell parameter estimation, Optik. 208 (2020) 164559.
[49] E.N. Kalemci, Rao-3 algorithm for the weight optimization of reinforced concrete cantilever retaining wall, Geomechan. Eng. 20 (2020) 527–536.
[50] R.V. Rao, R.B. Pawar, Self-adaptive multi-population Rao algorithms for engineering design optimization, Appl. Artif. Intellig. 34 (3) (2020) 187–250.
[51] V. Kumar, S.M. Yadav, Self-adaptive multi-population-based Jaya algorithm to optimize the cropping pattern under a constraint environment, J. Hydroinf. 22 (2020) 368–384.
[52] R.V. Rao, R.B. Pawar, Constrained design optimization of selected mechanical system components using Rao algorithms, Appl. Soft Comput. 89 (2020) 106141.
[53] S. Das, P.N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Jadavpur University, Nanyang Technological University, Kolkata. (2010) 341–359.
[54] M. Ghasemi, S. Ghavidel, M.M. Ghanbarian, M. Gharibzadeh, A.A. Vahed, Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm, Energy. 78 (2014) 276–289.
[55] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Fut. Generat. Comput. Syst. 97 (2019) 849–872.
[56] R.V. Rao, H.S. Keesari, A self-adaptive population Rao algorithm for optimization of selected bio-energy systems, J. Comput. Des. Eng. 8 (2021) 69–96.
[57] R.V. Rao, R.B. Pawar, Quasi-oppositional-based Rao algorithms for multi-objective design optimization of selected heat sinks, J. Comput. Des. Eng. 7 (6) (2020) 830–863.
[58] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[59] A. Ghosh, A. Chowdhury, S. Sinha, A. v Vasilakos, S. Das, A genetic Lbest particle swarm optimizer with dynamically varying subswarm topology, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012: pp. 1–7.
[60] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[61] L.D.S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Syst. Appl. 37 (2) (2010) 1676–1683.
[62] V.S. Aragón, S.C. Esquivel, C.A.C. Coello, A modified version of a T-Cell Algorithm for constrained optimization problems, Int. J. Numer. Meth. Eng. 84 (2010) 351–378.
[63] A. Baykasoğlu, F.B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Appl. Soft Comput. 36 (2015) 152–164.
[64] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[65] G. Brammya, S. Praveena, N.S. Ninu Preetha, R. Ramya, B.R. Rajakumar, D. Binu, Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm, Comput. J. (2019).
[66] H.S. Bernardino, H.J.C. Barbosa, A.C.C. Lemonge, L.G. Fonseca, A new hybrid AIS-GA for constrained optimization problems in mechanical engineering, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008: pp. 1455–1462.
[67] J. Zhang, M. Xiao, L. Gao, Q. Pan, Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems, Appl. Math. Model. 63 (2018) 464–490.
[68] I. Brajevic, Crossover-based artificial bee colony algorithm for constrained optimization problems, Neural Comput. Appl. 26 (7) (2015) 1587–1601.
[69] C.A.C. Coello, N.C. Cortés, Hybridizing a genetic algorithm with an artificial immune system for global optimization, Eng. Optim. 36 (5) (2004) 607–634.
[70] E. Mezura-Montes, B. Hernández-Ocana, Bacterial foraging for engineering design problems: preliminary results, in: Memorias Del 4o Congreso Nacional de Computación Evolutiva (COMCEV'2008), Centro de Investigación en Matemáticas Guanajuato, México, 2008.
[71] E. Mezura-Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (4) (2008) 443–473.
[72] F.-Z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356.
[73] A. Huong et al, A hierarchical optimisation framework for pigmented lesion diagnosis, CAAI Trans. Intell. Technol. 7 (1) (2022) 113–127, https://doi.org/10.1049/cit2.12073.
[74] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[75] I. Mazhoud, K. Hadj-Hamou, J. Bigeon, P. Joyeux, Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism, Eng. Appl. Artif. Intell. 26 (4) (2013) 1263–1273.
[76] P. Savsani, V. Savsani, Passing vehicle search (PVS): A novel metaheuristic algorithm, Appl. Math. Model. 40 (5-6) (2016) 3951–3978.
[77] A.H. Gandomi, X.-S. Yang, A.H. Alavi, S. Talatahari, Bat algorithm for constrained optimization tasks, Neural Comput. Appl. 22 (6) (2013) 1239–1255.
[78] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms, Comput. Methods Appl. Mech. Eng. 256 (2013) 70–87.
[79] K.E. Parsopoulos, M.N. Vrahatis, Unified particle swarm optimization for solving constrained engineering optimization problems, Int. Conf. Nat. Comput., Springer (2005) 582–591.
[80] S. Chun, Y.-T. Kim, T.-H. Kim, A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems, Adv. Mechan. Eng. 5 (2013) 130750.

[81] C.A. Coello Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Inf. 16 (3) (2002) 193–203.
[82] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, J. Intell. Manuf. 23 (4) (2012) 1001–1014.
[83] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (2013) 2592–2612.
[84] X.-B. Meng, H.-X. Li, X.-Z. Gao, An adaptive reinforcement learning-based bat algorithm for structural design problems, Int. J. Bio-Inspired Comput. 14 (2019) 114–124.
[85] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (2013) 17–35.
[86] S. Shadravan, H.R. Naji, V.K. Bardsiri, The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Eng. Appl. Artif. Intell. 80 (2019) 20–34.
[87] A. Ansari et al, A Hybrid Metaheuristic Method in Training Artificial Neural Network for Bankruptcy Prediction, IEEE Access 8 (2020) 176640–176650, https://doi.org/10.1109/ACCESS.2020.3026529.
[88] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft Comput. 10 (2) (2010) 629–640.
[89] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Inf. Sci. 178 (15) (2008) 3043–3074.
[90] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Struct. Multidiscip. Optim. 37 (4) (2009) 395–413.
[91] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, Fut. Generat. Comput. Syst. 101 (2019) 646–667.
[92] W. Zhao, Z. Zhang, L. Wang, Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications, Eng. Appl. Artif. Intell. 87 (2020) 103300.
[93] S. He, E. Prempain, Q.H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optim. 36 (5) (2004) 585–605.
[94] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020), https://doi.org/10.1016/j.engappai.2020.103731 103731.
[95] S.-F. Hwang, R.-S. He, A hybrid real-parameter genetic algorithm for function optimization, Adv. Eng. Inf. 20 (1) (2006) 7–21.
[96] J. Zhang, C. Liang, Y. Huang, J. Wu, S. Yang, An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization, Appl. Math. Comput. 211 (2) (2009) 392–416.
[97] L. Li et al, Improved tunicate swarm algorithm: Solving the dynamic economic emission dispatch problems, Appl. Soft Comput. 108 (2021) 107504–107510, https://doi.org/10.1016/j.asoc.2021.107504.
[98] Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Appl. Math. Comput. 186 (2) (2007) 1407–1422.
[99] S. Akhtar, K. Tai, T. Ray, A socio-behavioural simulation model for engineering design optimization, Eng. Optim. 34 (4) (2002) 341–354.
[100] A.-R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, J. Global Optim. 35 (4) (2006) 521–549.
[101] T.T. Ngo, A. Sadollah, J.H. Kim, A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems, J. Computat. Sci. 13 (2016) 68–82.
[102] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[103] H. Yapici, N. Cetinkaya, A new meta-heuristic optimizer: Pathfinder algorithm, Appl. Soft Comput. 78 (2019) 545–568.
[104] T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, Eng. Optim. 33 (6) (2001) 735–748.
[105] C.A. Coello Coello, R.L. Becerra, Efficient evolutionary optimization through the use of a cultural algorithm, Eng. Optim. 36 (2) (2004) 219–236.
[106] H.S. Bernardino, H.J.C. Barbosa, A.C.C. Lemonge, A hybrid genetic algorithm for constrained optimization problems in mechanical engineering, in2007., IEEE Congress on Evolutionary Computation, IEEE (2007) 646–653.
[107] H.N. Ghafil, K. Jármai, Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications, Appl. Soft Comput. 93 (2020) 106392.