Journal of Ambient Intelligence and Humanized Computing (2020) 11:267–290
https://doi.org/10.1007/s12652-019-01265-7

ORIGINAL RESEARCH

Artificial bee colony with enhanced food locations for solving mechanical engineering design problems

Tarun K. Sharma[1] · **Ajith Abraham**[2]

Received: 26 September 2018 / Accepted: 6 March 2019 / Published online: 16 March 2019

Abstract

Artificial Bee colony (ABC) simulates the intelligent foraging behavior of bees. ABC consists of three kinds of bees: employed, onlooker and scout. Employed bees perform exploration and onlooker bees perform exploitation whereas scout bees are responsible for randomly searching the food source in the feasible region. Being simple and having fewer control parameters ABC has been widely used to solve complex multifaceted optimization problems. ABC performs well at exploration than exploitation. The success of any nontraditional algorithm depends on these two antagonist factors. Focusing on this limitation of ABC, in this study the food locations in basic ABC are enhanced using Opposition based learning (OBL) concept. This variant is improved by incorporating greediness in searching behavior and named as I-ABC *greedy*. The modifications help in maintaining population diversity as well as enhance exploitation. The proposal is validated on seven mechanical engineering design problems. The simulated results have been noticed competent with that of state-of-art algorithms.

Keywords Artificial bee colony · Engineering design problems · Constrained Optimization · Convergence · Opposition based learning

1 Introduction

Optimization problems exist in every sphere of human life; may it be Science, Social Science, Engineering and Management (D'Apice et al. 2014; Ha and Gao 2017; Chen and Chuang 2018; Safarzadeh et al. 2018). These optimization problems are generally solved by traditional or non traditional methods, based on the complexity of the problem. A traditional method such as gradient search requires a problem to be continuous and differentiable where as non-traditional method hardly requires any domain knowledge of the problem (Tang et al. 2014; Salomon 1998). These methods generally simulates the behavior of natural species such as flock of birds, school of fishes, ants, bees etc and are inspired by the Darwin theory of 'survival of the fittest' (Holland 1975). The brief overview of non traditional methods can be referred from (Karaboga 2005; Rajpurohit et al. 2017). Among these methods Artificial Bee Colony (ABC) is recently introduced by Karaboga (2005). ABC mimics the foraging behavior of natural bees. Due to its simple structure and fewer number of control parameters, ABC has been widely applied to solve many applications which are presented in Second Section. However ABC gained the popularity in the minimum span of time, but there is 'no free lunch' algorithm available that can equally be applied to solve all type of optimization problems. Every nontraditional algorithm has certain limitations. In ABC single search equation is responsible for performing both exploration as well as exploitation. This limits ABC in exploitation capability. Focusing on this limitation of ABC, in this study the food locations in basic ABC are enhanced using Opposition based learning (OBL) concept which helps in maintaining population diversity. Also the variant is modified by embedding greedy search in the search equation and named as I-ABC *greedy*. This study is based on Intermediate food locations in ABC (I-ABC) (Sharma and Pant 2011, 2013) that uses Opposition Based Learning (OBL) concept. I-ABC, further modified to accelerate the convergence rate and the enhanced variant is named as I-ABC *greedy*. In this work

✉ Tarun K. Sharma
taruniitr1@gmail.com

Ajith Abraham
ajith.abraham@ieee.org

1 Amity University Rajasthan, Jaipur, India

2 Machine Intelligence Research (MIR) Labs, Washington, USA

I-ABC *greedy* algorithm is implemented to solve seven constrained optimization problems of mechanical engineering.

The structure of the present study is as follows: Sect. 2 presents a brief literature review of the applications and modified variants of ABC. Introduction to ABC is presented in Sect. 3. Section 4 discusses the proposed I-ABC *greedy* algorithm. Parameter settings are given in Sect. 5. Seven constrained optimization problems of mechanical engineering with simulated results are presented and analyzed in Sect. 6. Finally paper concludes with future scope in Sect. 7.

2 Literature review

ABC is recently introduced and most popular swarm intelligence algorithm that simulates the foraging behavior of honey bees. ABC is proposed by Karaboga in 2005 (2005). ABC has shown competitive results when applied to solve many real time applications. Initially ABC was introduced to handle unconstrained optimization problems (Karaboga and Basturk 2007b) which later modified to handle and solve constrained optimization problems (Karaboga and Basturk 2007a, b; Karaboga and Akay 2011). However like other swarm intelligence algorithms it also suffers with slow convergence rate (Mezura-Montes and Coello Coello 2011). Therefore several researchers have modified the basic ABC structure and presented the hybrid variants of ABC in order to improve the efficiency and efficacy of the ABC by providing a tradeoff between exploration and exploitation (Diwold et al. 2011; Bansal and Sharma 2012; Kumar et al. 2013; Jadon et al. 2015). This section carries the literature review of the hybrid variants of ABC for solving constrained optimization problems (COP).

In 2011, Brajevic et al. (2011) presented a SC-ABC (simple constrained ABC) where the best solutions are carried into new run as initial solutions. This makes each run dependent on the previous run. All the infeasible solutions found so far are replaced by randomly generated solutions in Scout bee phase. Mezura-Montes and Cetina-Domłnguez (2012), presented a modified variant M-ABC (modified ABC). They have done four modifications in the structure of ABC: embedded tournament selection in onlooker phase and smart flight operator in scout phase and handling boundary constraints. Also dynamic tolerance is used for equality constraints. Brajevic and Tuba (2013), introduced an ABC variant called upgraded ABC (UABC) for solving engineering design constrained problems. In UABC the parameter, modification rate (MR), is fine tuned as well as the scout bee number is dynamically adapted using a new control parameter, ISPP to enhance exploration. In 2014, Li and Yin (2014) presented a Self adaptive ABC (SACABC) algorithm that uses Deb's rule for generating feasible solution in the employed bee phase while onlooker bee phase utilizes the multi-objective optimization method. Tsai (2014) proposed a hybrid variant of ABC named as ABC-BA which integrates ABC and Bees Algorithm. In ABC-BA a constraint handling method similar to penalty method is introduced where fitness value of a feasible solution is treated as objective function value and fitness value of infeasible solution is considered as violation. Brajevic (2015) presented five modifications in ABC. It uses two modified search operators in the employed and onlooker phase and random search is replaced with crossover in scout phase of ABC to improve exploitation capability. Secondly equality constraints are handled using dynamic tolerance while boundary constraints are handled with some improved mechanism. Ayan et al. (2015) proposed chaotic ABC to solve optimal power flow problem. Pan (2016) proposed a co-evolutionary ABC (CCABC) to solve steelmaking continuous casting problem. In CCABC there are two sub swarms that address each sub problem. Bhambu et al. (2018) presented a MGABC (modified Gbest-guided ABC) that adjusts solutions step size in each iterations.

Sharma and Pant (2017) proposed a hybrid variant that integrates ABC and Shuffled Frog Leaping (SFL) algorithm and named it as Shuffled-ABC. Shuffled-ABC initiates with generation of initial population using randomly distribution and chaotic system. Then this population is divided in to two parts based on their fitness values. The first group contains the superior population and applies ABC to process where as second having inferior population applies SFLA to process. Akay and Karaboga (2017) presented nine variants of ABC with modifications in various phases of employed, onlooker and scout along with mutation, crossover operators of Differential Evaluation algorithm. Liang et al. (2017) introduced a modified variant of ABC named as I-ABC (improved artificial bee colony) to solve constrained optimization problems, in which Deb's constraint handling rules are used to handle the constraints. Liang suggested few modifications in the basic structure of ABC. At initial better feasible solutions are selected and then new selection strategy based on rank selection is introduced. Later search equation is modified by embedding the information of best solution found so far that adapts control parameters to justify exploration and exploitation. The algorithm was tested over a set of 14 benchmark constrained functions considered from the literature. Sundar et al. (2017) proposed a hybrid ABC algorithm to solve job scheduling problem where local search mechanism is used to improvise the quality of the solutions.

In 2018, Liu et al. (2018) proposed modified ABC that uses dynamic penalty method to handle constraints; Employed bee phase is modified using Lévy flight along with logistic map. Further onlooker bee phase uses best and its two neighboring solutions and boundary constraints are handled using best solutions. Wang and Jiao-Hong (2018) proposed a hybrid method that encompasses Krill Herd and

ABC for exchanging the information that results in balancing exploration and exploitation process. Wang et al. (2019) propose a method based on ANN and SFLA for selection of outsourcing service in cement equipment manufacturing company. A brief overview of ABC proposals for handling constraints are discussed in Table 1.

3 Brief introduction to artificial bee colony

Artificial Bee Colony simulates the foraging process of natural honey bees. The bee colony family in ABC consists of three members: employed, onlooker and scout bees. Scout bees' initiates searching of food sources randomly, once the potential food sources are identified by scout they become employed bees. Then food sources are exploited by employed bees that also shares the information about the quality and quantity of food sources to the onlooker (bees resting at hive and waiting for the information from employed bees). A specific "waggle dance" is performed to share food information. The ABC algorithm is presented below:

3.1 Initialization of random food sources

The random food sources (FS) are generated in the search space using following Eq. (1):

$$x_{ij} = max_j + rand(0,1) \times \left(max_j - min_j\right), \tag{1}$$

where i represents the FS and j denotes the jth dimension. max and min denote the upper and lower bounds.

3.2 Employed bee process

The search equation involved in this phase and also performs the global search by introducing new food sources $V_i = (v_{i1}, v_{i2}, \ldots v_{iD})$ corresponding to $X_i = (x_{i1}, x_{i2}, \ldots x_{iD})$ is discussed below:

$$v_{ij} = x_{ij} + rand(-1,\ 1) \times (x_{ij} - x_{kj}), \tag{2}$$

where k is selected randomly and distinct from i. Greedy selection mechanism is performed to select the population to store in a trail vector. In case v_{ij} fails corresponding to boundary constraints then they are handled using following Eq. (3):

$$v_{ij} = \begin{cases} max_j & if\ v_{ij} > max_j, \\ min_j & if\ v_{ij} < min_j. \end{cases} \tag{3}$$

From the Eq. (3), new solution may generated then there will be greedy selection in Eqs. (3) and (4)

$$x_i = \begin{cases} v_i & if\ fit(v_i) > fit\left(x_i\right), \\ x_i & otherwise. \end{cases} \tag{4}$$

where $fit()$ represents the fitness value which is defined in Eq. (5) (for minimization case):

$$fit(x_i) = \begin{cases} \frac{1}{(1+f(x_i))} & if\ f(x_i\ ) > 0 \\ 1 + abs\, f\left(x_i\right) & if\ f(x_i\ ) \le 0, \end{cases} \tag{5}$$

where $f()$ represents the objective function value.

Table 1 Brief overview of ABC & modified ABC proposals handling constrained optimization functions

Algorithm (Author(s), Year)	Constrained handling methodology
ABC (Bernardino et al. 2007)	Deb's Rule
SM-ABC (Erbatur et al. 2000)	Deb's Rule
Elitist-ABC (Eskandar et al. 2012)	Deb's Rule
Modified ABC (Bernardino et al. 2008)	Deb's Rule
SC-ABC (Brajevic 2015)	Deb's Rule
M-ABC (Brajevic and Tuba 2013)	Deb's Rule
U-ABC (Brajevic et al. 2011)	Deb's Rule
SACABC (Brajević and Ignjatović 2018)	Multiobjective and Deb's Rule
ABC-BA (Cagnina et al. 2008)	Penalty Method
CB-ABC (Chen and Chuang 2018)	Deb's Rule
CABC (Coelho 2010)	Penalty Method
CCABC (Coello Coello 2000)	Penalty Method
i-ABC (Gandomi et al. 2013)	Deb's Rule
MGABC (Coello Coello and Becerra 2004)	Deb's Rule
Shuffled-ABC (Coello Coello and Montes 2002)	Penalty Method
ABCV1-9 (Coello Coello and Landa-becerra 2003)	Deb's Rule
I-ABC (D'Apice et al. 2014)	Deb's Rule

3.3 Onlooker bee process

Onlooker bee carry out local search in the region of the food sources shared by employed bee. Equation (6) is used to choose the food source by Onlooker bee from a set of *FS* solutions. Probability P_i is used to choose the food source (solution).

$$P_i = fit(x_i) / \sum_{i=1}^{FS} fit(x_i). \tag{6}$$

Onlooker bee chooses the food source having better probability, then Eq. (2) is used to exploit the food source and new food source is generated. After this a greedy process is followed using Eq. (4).

3.4 Scout bee process

If the food source does not improve in the fix number of trials (*limit a* control parameter) then employed bees turns into scout bees and randomly forage for the new food sources.

Initially ABC was designed to handle unconstrained optimization problems where it has shown competitive edge over PSO, DE and GA (Karaboga and Akay 2009). Later ABC was modified to handle and solve COPs (Karaboga and Akay 2011) by adding one more parameter, modification rate (*MR*) in employed and onlooker phase, constraints were handled using Deb's rule (Deb 2000) and thirdly another control parameter named *SPP* is added along with *limit* in scout bee phase that controls the abandoned food source, if it exceeds limit. If it exceeds limit then a scout production process is carried out. SPP ensures that the new food source randomly generated by the scout bee replace the proposed food source. Deb's rule suggests that:

(a) Feasible solution is selected over infeasible solution.
(b) In case two feasible solutions are there then the solution having best objective function value would be considered.
(c) If both the solutions are infeasible then the one violating minimum number of constraints would be preferred.

Following Eq. (7) is used by the employed and onlooker bees to generate the new food source.

$$v_{ij} = \begin{cases} x_{ij} + \varphi_{ij}\left(x_{ki} - x_{ij}\right) & if\ R_j < MR \\ x_{ij} & Otherwise \end{cases}, \tag{7}$$

where φ_{ij} is a random number in the range [− 1,1] and *MR* controls the modification in x_{ij} *and* $R \in [0,1]$.

4 I-ABC *greedy*: proposed scheme

The following proposal has been proposed by Sharma and Pant (2013) for solving unconstrained optimization problems. This study is extended here to implement the proposed algorithms on COP's.

4.1 Motivation and concept involved in the proposed Algorithm: I-ABC *greedy*

In order to enhance the solution diversity as well as convergence rate the basic ABC is improved by embedding Opposition based learning (OBL) concept while initialing the initial solution and later the search equation (Eq. 2) is modified by inserting a greedy concept. The concept of OBL is discussed below:

4.1.1 Opposition based learning (OBL)

In normal process, the initial population is randomly generated and we initiate towards the solution with this random guess which may or may not be in the vicinity of the exact solution. Sometimes, considering the worst scenario, the solution may lie in opposite direction then the process of search may take comparatively more time. Also not having the prior information about the solution, it is not possible to make an initial best guess. Therefore, simultaneously searching process must be performed in all directions or in opposite direction. This also provides solutions diversity. This is why the concept of OBL, proposed by Rahnamayan et al. (2008) is embedded in the proposal.

Opposite solutions to initially random generated populations are generated using the concept of opposite numbers. Following Eq. (8) is used to generate opposite solutions:

Let $x \in$ in a certain defined interval $x \in [l_b,\ u_b]$, then opposite number $\bar{x}$ is calculated as:

$$\bar{x} = l_b - u_b - x \tag{8}$$

4.2 Steps involved in proposed scheme

4.2.1 Initialization process

The basic structure of ABC is modified while initializing initial set of solutions (food sources). Following steps are involved to generate the initial solutions (food sources):

- Firstly, solutions are generated using the uniformly distributed random numbers, say Pop_1, where Pop_1 represents $\{U_1, U_2,..., U_N\}$. Evaluate their fitness value.
- Secondly, apply the concept of OBL and generate the corresponding opposite solutions, say Pop_2, where Pop_2 represents $\{O_1, O_2,\ldots, O_N\}$. Evaluate their fitness value.

There will be 2N populations, one from randomly generated and other will be from opposite numbers.

- Then the mean of Pop_1 and Pop_2 named as *intermediate locations* (*IL*) (solutions) based on fitness values is computed Eq. (9):

$$IL = \{(U_1 + O_1)/2,\ (U_2 + O_2)/2, \ldots,\ (U_N + O_N)/2\}. \tag{9}$$

After this the elite *N*, based on fitness values are taken as initial solutions (X_{IL}) or food sources.

Sphere and Griekwank functions are used to demonstrate the generation of initial solution using the above discussed scheme in Fig. 1.

4.2.2 Modification in the search mechanism

Now in order to enhance the exploitation capability as well as the convergence rate, search equation (Eq. 2) is modified by bounding the search in the vicinity of the *best* solution i.e. $x_{best,j}$ is selected over x_{ij} *in* Eq. (2). This also helps in managing the balance between exploration and exploitation process. Hence Eqs. (2) and (7) are modified as:

$$v_{ij} = x_{best,j} + \varphi_{ij}\left(x_{best,j} - x_{kj}\right) \tag{10}$$

$$v_{ij} = \begin{cases} x_{best,j} + \varphi\left(x_{best,j} - x_{kj}\right) & if\ R_j < MR \\ x_{ij} & Otherwise \end{cases} \tag{11}$$

These two modifications in basic ABC defines I-ABC *greedy* algorithm.

4.3 Constraint handling

Constraints are handled using following Deb's rules (2000):

(a) Feasible solution is selected over infeasible solution.
(b) In case two feasible solutions are there then the solution having best objective function value would be considered.
(c) If both the solutions are infeasible then the one violating minimum number of constraints would be preferred.

The notations are same as discussed in Eq. (2). The flow graph of the proposals is demonstrated in Fig. 2.

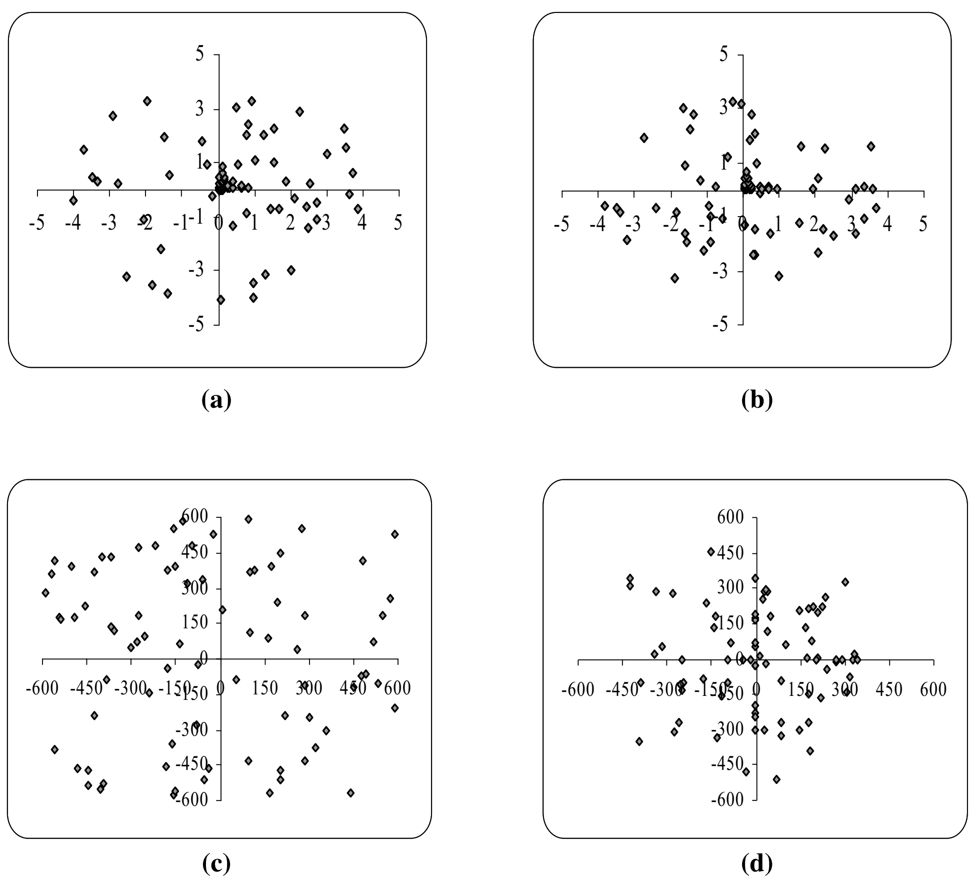


Fig. 1 Demonstration of initial population generation for Sphere in **a** uniformly random distribution **b** using the proposed concept of OBL and for Griekwank in **c** uniformly random distribution **d** using the proposed concept of OBL (Babaeizadeh and Ahmad 2016)

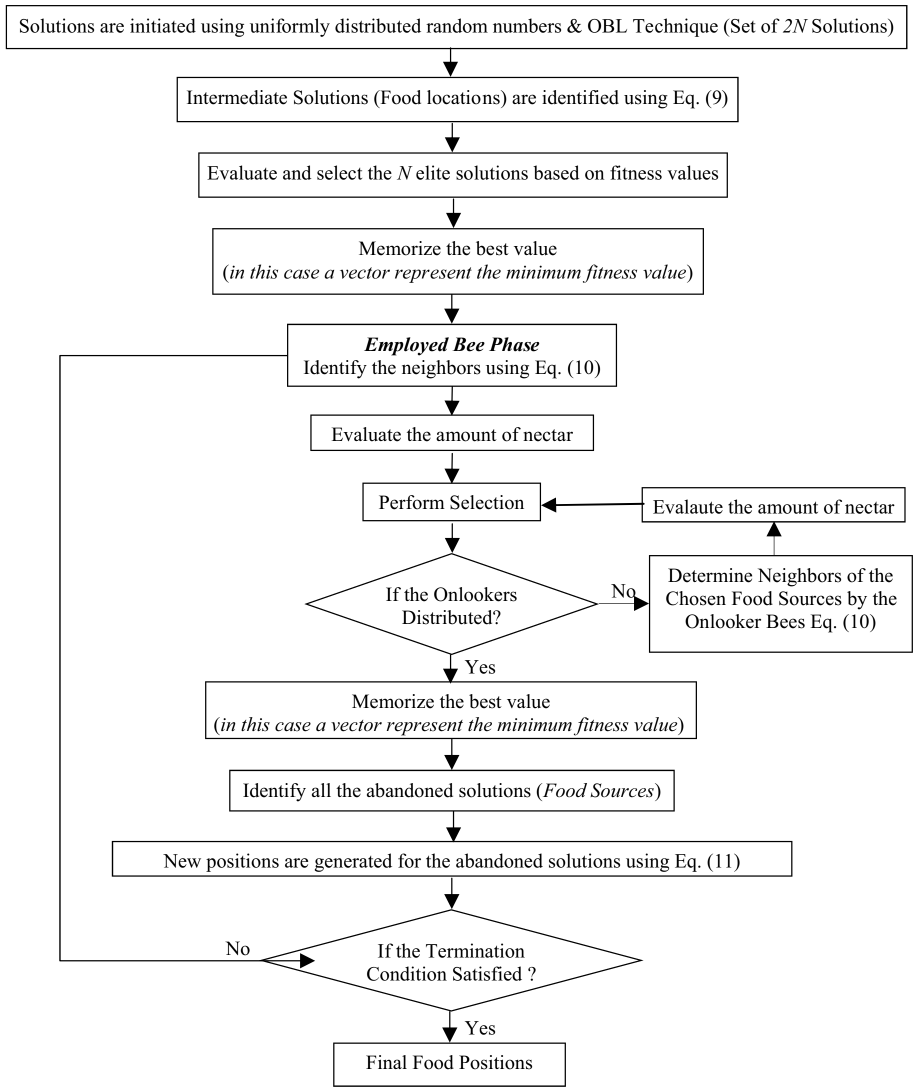


Fig. 2 Flow graph of the proposed I-ABC *greedy* algorithm

5 Parameter settings and evaluation criterion

- *Stopping criterion* 500,000 number of number of function evaluations (*NFE's*)
- *Runs*: Each algorithm performed 25 independent runs
- Statistical results in terms of Best, Mean, Median, Worst and Standard Deviation (SD) have been computed (Liang et al. 2006).
- Deb's rule are used to handle the constraints

5.1 Control parameters

Algorithms are highly sensitive and plays a significant role in search mechanism. Also for unbiased results comparison parameters are to be tuned accordingly. In order to make fair comparison the algorithms considered for simulated result comparison are adopted with their original parameter setting as mentioned in the original articles (Coello Coello 2000; Coello Coello and Montes 2002; Lampinen 2002; Ray and Liew 2003; Coello Coello and Becerra 2004; Krohling and Coelho 2006; He and Wang 2007a, b; Yuan and Qian 2010; Zahara and Kao 2009; Yang 2010; Kashan 2011; Eskandar et al. 2012; Akay and Karaboga 2012; Sadollah et al. 2013; Gong et al. 2014; Baykasolu and Ozsoydan 2015; Guedria 2016; Yi et al. 2016; Brajević and Ignjatović 2018). For ABC the parameters are considered from (Akay and Karaboga 2012) and for I-ABC *greedy* following is the adjustment of control parameters (Table 1):

All the algorithms are executed in Dev C++ with the following machine configuration:

Processor Intel(R) Core (TM) i3-5005U CPU @2.00 GHz having 4 GB RAM. An inbuilt *rand* () function in C++ is used to initialize the random numbers.

Table 2 Control parameters of I-ABC *greedy* algorithm

Colony size (CS)	100 (50 Onlooker & 50 employed)
Modification rate (MR)	0.8
Limit	$0.5 \times \mathrm{CS} \times D$
SPP	$0.5 \times \mathrm{CS} \times D$

The parameter settings for all the algorithms considered for statistical results comparisons are kept same as mentioned in the literature (see Table 2).

6 Mechanical engineering design optimization problems and result analysis

6.1 Welded beam design (WBD) problem (Rao 1996)

This problem is taken from that aims to optimize the cost of design of welded beam. The key objective of the problem is to optimize the cost of fabrication. The problem has four decision variables with seven restraints presenting shear stress (α), beams bending stress (β), deflection on the beam (ϑ) and the bar (Δ_c) having buckling load. The problem is detailed below

$$f(Z) = 1.10471 z_1 z_2^2 + 0.04811 z_3 z_4 (14 + z_2).$$

w.r.t. restraints

$$h_1(Z) = \alpha(Z) - \alpha_{max} \le 0$$

$$h_2(Z) = \beta(z) - \beta_{max} \le 0$$

$$h_3(Z) = z_1 - z_4 \le 0$$

$$h_4(Z) = 0.10471 z_1^2 + 0.04811 z_3 z_4 \left(14 + z_2\right) - 5 \le 0$$

$$h_5(Z) = 0.125 - z_1 \le 0$$

$$h_6(Z) = \vartheta(z) - \vartheta_{max} \le 0$$

$$h_7(Z) = \Delta - \Delta_c(z) \le 0$$

$$Z = (z_1,\ z_2,\ z_3,\ z_4)^T; 0.1 \le z_1; z_4 \le 2; 0.1 \le z_2; z_3 \le 10,$$

where

$$\alpha(Z) = \sqrt{\alpha'^2 + \frac{2\alpha'\alpha'' z_2}{2R} + \alpha''^2}$$

$$\alpha' = \frac{\Delta}{\sqrt{2} z_1 z_2}$$

$$\alpha'' = \frac{MR}{J}$$

$$M = \Delta(14 + \frac{z_2}{2})$$

$$R = \sqrt{\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2}$$

$$J = 2\left\{ \sqrt{2} z_1 z_2 \left[\frac{z_2^2}{12} + \left(\frac{z_1 + z_3}{2}\right)^2 \right] \right\}$$

$$\beta(Z) = \frac{504000}{z_4 z_3^2}$$

$$\vartheta(Z) = \frac{65856000}{30 \times 10^6 z_4 z_3^3}$$

$$\Delta_c = \frac{4.013 \times \left(30 \times 10^6\right)}{196} \sqrt{\frac{z_3^2 z_4^6}{36}} \times \left[1 - \left(\frac{z_3 \sqrt{\frac{30\times10^6}{4\times(12\times10^6)}}}{28} \right) \right]$$

$$\Delta = 6000 lb; L = 14 inch; E = 30 \times 10^6\ psi; G = 12 \times 10^6\ psi$$

$$\alpha_{\max} = 13600\ psi;\ \beta_{\max} = 30000\ psi;\ \vartheta_{\max} = 0.25\ inch.$$

The illustration of welded beam is given in Fig. 3.

The simulated statistical results of I-ABC *greedy* are compared with the state-of-art algorithms and are presented in Tables 3, 4 and 5. The considered algorithms for result comparison are variants of Genetic Algorithm (GA) i.e. GA1 (GA based Co-evolution Algorithm) (Coello Coello 2000); GA2 (GA using dominance based tournament) (Coello Coello and Montes 2002); DE (Lampinen 2002); SC (Ray

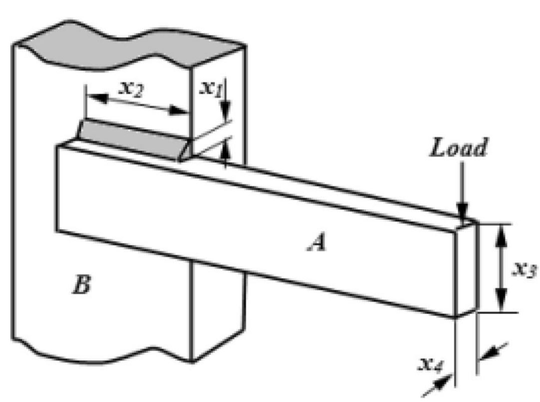


Fig. 3 Illustration of WBD problem

Table 3 The best simulated results for WBD problem attained by GA2, CPSO, CAEP, HGA, NM-PSO, WCA, MBA, APSO, ABC, IAPSO, UFA, MHS-PCLS and the proposal I-ABC *greedy*

D.V.	z_1	z_2	z_3	z_4	$h_1(Z)$	$h_2(Z)$	$h_3(Z)$	$h_4(Z)$	$h_5(Z)$	$h_6(Z)$	$h_7(Z)$	$f(Z)$
GA2	0.205986	3.471328	9.020224	0.20648	−0.103049	−0.231747	−5.00E04	−3.430044	−0.080986	−0.235514	−58.64689	1.728226
CPSO	0.202369	3.544214	9.04821	0.205723	−13.655547	−78.814077	−3.35E03	−3.424572	−0.077369	−0.235595	−4.472858	1.728024
CAEP	0.2057	3.4705	9.0366	0.2057	1.988676	4.481548	0	−3.433213	−0.0807	−0.235538	2.603347	1.724852
HGA	0.2057	3.4705	9.0366	0.2057	1.988676	4.481548	0	−3.433213	−0.0807	−0.235538	2.603347	1.724852
NM-PSO	0.20583	3.468338	9.036624	0.20573	−0.02525	−0.053122	0.0001	−3.433169	−0.08083	−0.23554	−0.031555	1.724717
WCA	0.205728	3.470522	9.03662	0.205729	−0.034128	−3.49E05	−1.19E06	−3.43298	−0.080728	−0.23554	−0.013503	1.724856
MBA	0.205729	3.470493	9.036626	0.205729	−0.001614	−0.016911	−2.40E07	−3.432982	−0.080729	−0.23554	−0.001464	1.724853
APSO	0.202701	3.574272	9.040209	0.2059215	−117.467062	−51.712981	−0.003221	−3.421741	−0.077701	−0.235571	−18.36701	1.736193
ABC	0.205,730	3.470,489	9.036624	0.205,730	0.000000	−0.000002	0.000000	−3.432984	−0.080730	−0.235540	0.000000	1.724,852
IAPSO	0.2057296	3.47048866	9.03662391	0.20572964	−1.05E10	−6.941E10	−7.66E15	−3.43298378	−0.080729639	−0.235540323	-5.80E10	1.7248523
UFA	0.20572964	3.4704886	9.03662391	0.20572963	−8.10E7	−8.10E7	−3.96E11	−3.48	−0.08	−0.24	−5.48E7	1.724852308
MHS-PCLS	0.205730	3.470489	9.036624	0.205730	1.724852	NA	NA	NA	NA	NA	NA	NA
I-ABC *greedy*	0.2057294	3.47048861	9.03662389	0.20572876	−3.12E09	−1.67E08	−6.98E14	−3.43298167	−0.08072915	−0.235539	−7.86E8	1.7248210

and Liew 2003); CAEP (Coello Coello and Becerra 2004); CPSO-GD (Krohling and Coelho 2006); HPSO (He and Wang 2007a, b); HGA (Hybrid GA) (Yuan and Qian 2010); NM-PSO (Zahara and Kao 2009); APSO (Yang 2010); LCA (Kashan 2011); WCA (Eskandar 2012); ABC (Akay and Karaboga 2012); MBA (Sadollah et al. 2013); rank-iMDDE (Gong et al. 2014); AFA (Baykasolu and Ozsoydan 2015); IAPSO (Guedria 2016); MHS-PCLS (Yi et al. 2016); and UFA (Brajević and Ignjatović 2018). In Table 3, best obtained results for all decision variables (DV), restraints and objective function value are presented. Tables 4 and 5, demonstrate the statistically simulated results in terms of worst, mean, best, SD and NFE's for I-ABC *greedy* and other optimizers considered for the comparisons. I-ABC *greedy* has shown the competitive results and able to solve this problem efficiently with lesser NFE's. The same is illustrated in Fig. 4.

6.2 Pressure vessel design (PVD) problem

In the series next problem is taken from (Sandgren 1990; Kannan and Kramer 1994; Rao 1996). The problem is mixed non linear problem. The problem is named as pressure Vessel Design (PVD). This optimization problem aims to obtain the value of materials, forming and welding costs in order to minimize the total manufacturing cost of a pressure vessel. The problem has four decision variables namely z_1, z_2, z_3 and z_4 along with four restraints. z_1 and z_2 are discrete where as z_3 and z_4 are continuous variables. z_1, z_2, z_3 and z_4 represent the shell thickness, head thickness, inner side radius and length of the cylinder respectively. The figure of PVD is illustrated in Fig. 5. To handle the problem of mixed non linear, the discrete variables (z_1 and z_2) are rounded off to their integer part by multiplying with 0.0625. The problem of PVD is formulated below:

$$\textit{Minimize}\ \ f(Z) = 0.6224 z_1 z_3 z_4 + 1.7781 z_2 z_3^2 + 3.1661 z_1^2 z_4 + 19.84 z_1^2 z_3$$

w.r.t. restraints

$$h_1(Z) = -z_1 + 0.0193 z_3 \le 0$$

$$h_2(Z) = -z_2 + 0.00954 z_3 \le 0$$

$$h_3(Z) = -\pi z_3^2 z_4^2 - \frac{4}{3}\pi z_3^3 + 1296000 \le 0$$

$$h_4(Z) = z_4 - 240 \le 0$$

where

$$1 \times 0.0625 \le z_1; z_2 \le 99 \times 0.0625; 10 \le z_3; z_4 \le 200.$$

To verify and compare the statistical results of I-ABC *greedy* algorithm, in this study GA1 (Coello Coello 2000);

Table 4 The statically simulated results of WBD problem attained (worst, mean, best, SD and NFE's) by GA1 &2, CAEP, CPSO, HPSO, PSO-DE, NM-PSO, SC, DE, WCA, LCA, MBA

Opti.	GA1	GA2	CAEP	CPSO	HPSO	PSO-DE	NM-PSO	SC	DE	WCA	LCA	MBA
Worst	1.785835	1.993408	3.179709	1.782143	1.814295	1.724852	1.733393	6.399678	1.824105	1.744697	1.7248523	1.724853
Mean	1.771973	1.792654	1.971809	1.748831	1.74904	1.724852	1.726373	3.002588	1.768158	1.726427	1.7248523	1.724853
Best	1.748309	1.728226	1.724852	1.728024	1.724852	1.724852	1.724717	2.385434	1.733461	1.724856	1.7248523	1.724853
S.D	1.12E02	7.47E02	4.43E01	1.29E02	4.01E02	6.70E16	3.50E03	9.60E01	2.21E02	4.29E03	7.11E15	6.94E19
NFEs	900,000	80,000	50,020	240,000	81,000	66,600	80,000	33,095	204,800	46,450	15,000	47,340

Table 5 The statically simulated results of WBD problem attained (worst, mean, best, SD and NFE's) by APSO, ABC, rank-iMDDE, UFA, IAPSO, UFA, MHS–PCLS and I-ABC *greedy*

Opti.	APSO	ABC	Rank-iMDDE	AFA	IAPSO	UFA	MHS-PCLS	I-ABC *greedy*
Worst	1.993999	NA	1.724852309	1.724852	1.7248624	1.7248523090	**1.724852**	1.724910
Mean	1.877851	1.741913	1.724852309	1.724852	1.7248528	1.7248523088	**1.724852**	1.724865
Best	1.736193	1.724852	1.724852309	1.724852	1.7248523	1.7248523087	1.724852	**1.724852**
S.D	0.076118	3.1E02	7.71E11	**0.000000**	2.02E06	7.96E 11	8.11e–10	1.92E-05
NFEs	50,000	30,000	15,000	50,000	12,500	**2000**	10,000	14,500

The best results are highlighted in bold

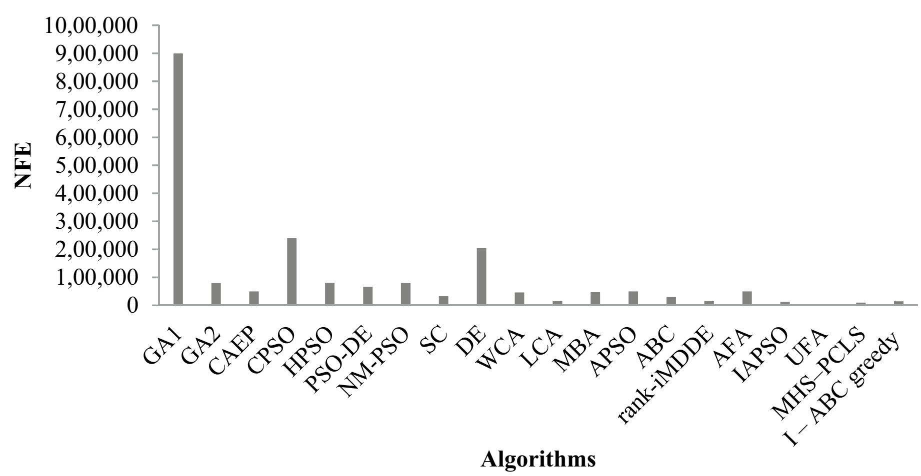


Fig. 4 NFE's used by the optimizers to obtain optimal value in WBD Problem

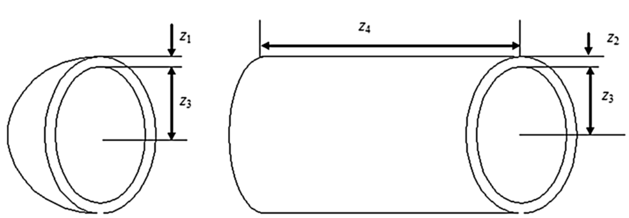


Fig. 5 Illustration of PVD problem

GA2 (Coello Coello and Montes 2002); CPSO (Krohling and Coelho 2006); HPSO (He and Wang 2007a, b); co-evolutionary differential evolution (CDE) (Huang et al. 2007); NM–PSO (Zahara and Kao 2009); G-QPSO (Coelho 2010); QPSO (Coelho 2010); LCA (Kashan 2011), WCA (Eskandar et al. 2012); PSO (Sadollah et al. 2013); MBA (Sadollah et al. 2013); rank-iMDDE (Gong et al. 2014); AFA (Baykasolu and Ozsoydan 2015); IAPSO (Guedria 2016); MHS-PCLS (Yi et al. 2016); and UFA (Brajević and Ignjatović 2018) are considered. The results are presented in Tables 6, 7 and 8. In Table 4 the best results achieved for the decision variables and restraints are given. Also in Tables 7 and 8 the simulated results in terms of best, worst, mean, SD and the NFE along with other optimizers are presented. It can be clearly noticed from the tables that I-ABC *greedy* performed at par and able to achieve the optimized results in lesser NFE as compared to others considered for comparison

Table 6 The best simulated results for PVD problem attained by CDE, GA1, CPSO, HPSO, NM-PSO, G-QPSO, WCA, MBA, APSO, ABC, IAPSO, UFA, MHS-PCLS and I-ABC *greedy*

D.V.*	z_1	z_2	z_3	z_4	$h_1(Z)$	$h_2(Z)$	$h_3(Z)$	$h_4(Z)$	$f(Z)$
CDE	0.8125	0.4375	42.0984	176.6376	− 6.67E–07	− 3.58E–02	− 3.705123	− 63.3623	6059.734
GA1	0.8125	0.4375	42.0974	176.654	− 2.01E–03	− 3.58E–02	− 24.7593	− 63.346	6059.9463
CPSO	0.8125	0.4375	42.0913	176.7465	− 1.37E–06	− 3.59E–04	− 118.7687	− 63.2535	6061.0777
HPSO	0.8125	0.4375	42.0984	176.6366	− 8.80E–07	− 3.58E–02	3.1226	− 63.3634	6059.7143
NM-PSO	0.8036	0.3972	41.6392	182.412	3.65E–05	3.79E–05	− 1.5914	− 57.5879	5930.3137
G-QPSO	0.8125	0.4375	42.0984	176.6372	− 8.79E–07	− 3.58E–02	− 0.2179	− 63.3628	6059.7208
WCA	0.7781	0.3846	40.3196	200	− 2.95E–11	− 7.15E–11	− 1.35E–06	− 40	5885.3327
MBA	0.7802	0.3856	40.4292	198.4964	0	0	− 86.3645	− 41.5035	5889.3216
APSO	0.8125	0.4375	42.0984	176.6374	− 9.54E–07	− 3.59E–02	− 63.3626	− 0.9111	6059.72418
ABC	0.8125	0.4375	42.098446	176.636596	0.000000	− 0.035881	− 0.000226	− 63.363404	6059.714339
IAPSO	0.8125	0.4375	42.0984	176.6366	− 4.09E–13	−3.58E–2	−1.39E–07	−63.363.34	6059.71433
UFA	0.8125	0.4375	42.098445	176.63659	−9.76E–13	−0.04	−3.60E–7	NA	6059.71433
MHS–PCLS	0.8125	0.4375	42.098446	176.636596	−1.6928E–11	−3.59E–02	−1.7664E–05	−63.3634	6059.71433
I-ABC *greedy*	0.8125	0.4375	42.0984	176.6369	− 1.27E–11	− 3.58E–02	− 1.26E–06	− 63.2963	6059.7124

DV decision variables

Table 7 The statically simulated results of PVD problem attained (worst, mean, best SD and NFE's) by I-ABC *greedy*, GA1, GA2, CPSO, HPSO, NM-PSO, G-QPSO, QPSO, PSO, CDE, WCA, LCA

Opti	GA1	GA2	CPSO	HPSO	NM-PSO	G-QPSO	QPSO	PSO	CDE	WCA	LCA
Worst	6308.497	6469.322	6363.8041	6288.677	5960.0557	7544.4925	8017.2816	14076.324	6371.0455	6590.2129	6090.6114
Mean	6293.8432	6177.2533	6147.1332	6099.9323	5946.7901	6440.3786	6440.3786	8756.6803	6085.2303	6198.6172	6070.5884
Best	6288.7445	6059.9463	6061.0777	6059.7143	5930.3137	6059.7208	6059.7209	6693.7212	6059.734	5885.3327	6059.8553
SD	7.4133	130.9297	86.45	86.2	9.161	448.4711	479.2671	1492.567	43.013	213.049	11.37534
NFEs	900,000	80,000	240,000	81,000	80,000	8000	8000	8000	204,800	27,500	24,000

Table 8 The statically simulated results of PVD problem attained (worst, mean, best SD and NFE's) by I-ABC *greedy*, MBA, APSO, ABC, rank-iMDDE, UFA, IAPSO, UFA, MHS–PCLS and I-ABC *greedy*

Opti	MBA	APSO	ABC	Rank-iMDDE	AFA	IAPSO	UFA	MHS-PCLS	I-ABC *greedy*
Worst	6392.5062	7544.49272	NA	6059.714335	6090.52614259	6068.78539	**6059.7143352069**	6059.71439	6086..982
Mean	6200.64765	6470.71568	6245.308144	6059.714335	6064.33605261	6068.7539	**6059.714335100**	6059.71434	6067.816
Best	5889.3216	6059.7242	6059.714736	6059.714335	6059.71427196	6059.7143	6059.7143350561	6059.71433	**6059.7142**
SD	160.34	326.9688	205.1332	7.57E–07	11.28785324	14.0057	**3.47E-08**	1.28120E–05	19.044
NFEs	70,650	200,000	30,000	15,000	50,000	**7,500**	**2000**	10, 000	8000

The best results are highlighted in bold

else IAPSO. The same is depicted in Fig. 6. This shows that I-ABC *greedy* converges faster than others in comparison. All the results for other optimizers are taken from their original research.

6.3 Tension/compression spring design (T/CSD) optimization problem (Ha and Gao 2017)

The problem focuses on optimization of the weight of the compression string. The problem consists of three decision variables namely z_1, z_2 and z_3 with four restraints defining deflection, frequency, shear stress and outside diameter limits. There are separate conditions for each variable. The decision variables z_1, z_2 and z_3 denotes diameter of wire, mean coil diameter and active coils respectively. The diagram of (T/CSD) is illustrated in Fig. 7. The mathematical formulation of the problem is stated below.

$$Minimize\, f(Z) = (z_3 + 2)z_2 z_1^2$$

w.r.t. restraints

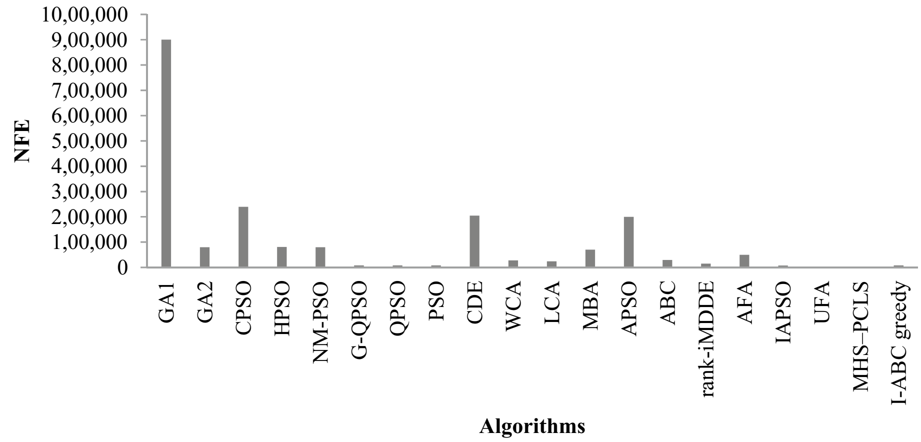


Fig. 6 NFE's used by the optimizers to obtain optimal value in PVD Problem

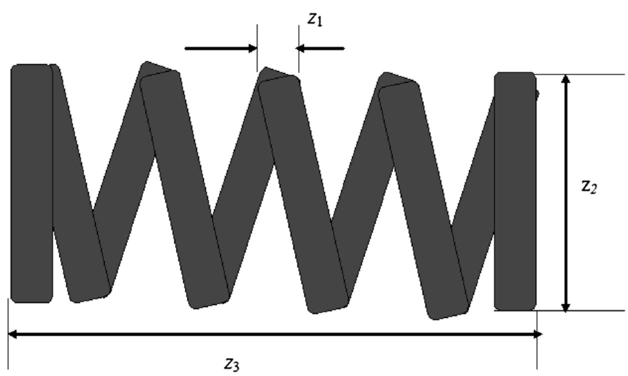


Fig. 7 Tension/compression spring design (T/CSD) optimization problem

$$h_1(Z) = 1 - \frac{z_2^3 z_3}{7.1785 z_1^4} \leq 0$$

$$h_2(Z) = \frac{4z_2^2 - z_1 z_2}{12.566\left(z_2 z_1^3\right) - z_1^4} + \frac{1}{5.108 z_1^2} - 1 \leq 0$$

$$h_3(Z) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \leq 0$$

$$h_4(Z) = \frac{z_2 + z_1}{1.5} - 1 \leq 0$$

where

$0.05 \leq z_1 \leq 2; .25 \leq z_2 \leq 1.3; 2 \leq z_3 \leq 15.$

For the analysis over results, the proposal is compared with other optimizers refereed from the literature. The optimizers are GA1(Coello Coello 2000); DE (Lampinen 2002); GA2 (Coello Coello and Montes 2002); SC (Ray and Liew 2003); CPSO (Coello Coello and Becerra 2004); (μ + λ)-ES (Mezura-Montes and Coello Coello 2005); HPSO (He and Wang 2007a, b); ABC (Karaboga and Basturk 2007a); DEDS (Zhang et al. 2008); NM–PSO (Zahara and Kao 2009); HEAA (Wang et al. 2009); G-QPSO (Coelho 2010); APSO (Yang 2010); QPSO (Coelho 2010); PSO–DE (Liu et al. 2010); DELC (Wang and Li 2010); PSO (Sadollah 2013); rank-iMDDE (Gong et al. 2014); AFA (Baykasolu and Ozsoydan 2015); IAPSO (Guedria 2016); MHS-PCLS (Yi et al. 2016); and UFA (Brajević and Ignjatović 2018). Table 9 presents the best simulated and comparative results of T/CSD problem attained by G-QPSO, DEDS, HEAA, NM–PSO, DELC, WCA, LCA, MBA, APSO, ABC, IAPSO, UFA, MHS–PCLS and I-ABC *greedy*. Also in Tables 10 and 11 (divided in two tables as there are several optimizers considered for results comparisons) the statically simulated results of T/CSD problem attained (worst, mean, best SD and NFE's) by DE, DELC, DEDS, HEAA, PSO–DE, SC, (μ + λ)-ES, ABC, LCA, WCA, MBA, APSO, GA1,GA2, CAEP, CPSO, HPSO, NM–PSO, G-QPSO, QPSO, PSO, ABC, IAPSO, UFA, MHS–PCLS and I-ABC *greedy* are presented. It can be analyzed that G-QPSO, QPSO, PSO, IAPSO and I-ABC *greedy* took 2000 NFEs only to attain the global optimal value. The same is depicted in Fig. 8.

6.4 Gear train design (GTD) optimization problem (Ha and Gao 2017)

The objective of the problem is to optimize the cost incurred in gear ratio in GTD. The design is presented in Fig. 9. The problem has only four integer decision variables namely z_1, z_2, z_3, and z_4 with only one limit condition. The variables correspond to the gear teeth numbers of A, B. D & F. The value of decision variables ranges from 12 to 60. The mathematical formulation of GTD is given below.

$$Minimize\, f(Z) = \left(\left(\frac{1}{6.931}\right) - \left(\frac{z_2 z_3}{z_1 z_4}\right)\right)^2,$$

Table 9 Best simulated and comparative results of T/CSD problem attained by I-ABC *greedy*, G-QPSO, DEDS, HEAA, NM-PSO, DELC, WCA, LCA, MB, APSO, ABC, IAPSO, UFA, MHS-PCLS and I-ABC *greedy*

D.V.	z_1	z_2	z_3	$h_1(Z)$	$h_2(Z)$	$h_3(Z)$	$h_4(Z)$	$f(Z)$
G-QPSO	0.051515	0.352529	11.538862	– 4.83E–05	– 3.57E–05	−4.0455	– 0.73064	0.012665
DEDS	0.051689	0.356717	11.288965	1.45E–09	– 1.19E–09	– 4.053785	– 0.727728	0.012665
HEAA	0.051689	0.356729	11.288293	3.96E–10	– 3.59E–10	– 4.053808	– 0.72772	0.012665
NM–PSO	0.05162	0.355498	11.333272	1.01E–03	9.94E–04	– 4.061859	– 0.728588	0.01263
DELC	0.051689	0.356717	11.288965	– 3.40E– 09	2.44E–09	– 4.053785	– 0.727728	0.012665
WCA	0.05168	0.356522	11.30041	– 1.65E–13	– 7.9E–14	– 4.053399	– 0.727864	0.012665
LCA	0.051689	0.356718	11.28896	NA	NA	NA	NA	0.01266523
MB	0.051656	0.35594	11.344665	0	0	– 4.052248	– 0.728268	0.012665
APSO	0.052588	0.378343	10.138862	– 1.55E–04	– 8.33E–04	– 4.089171	– 1.069069	0.0127
ABC	0.051749	0.358179	11.203763	0.000000	0.000000	– 4.056663	– 0.726713	0.012665
IAPSO	0.051685	0.356629	11.294175	– 1.97E–10	– 4.64E–10	– 4.053610	– 1.091686	0.01266523
UFA	0.05168967	0.3567324	11.2881015	−5.98E – 10	−1.63E–10	−4.054	−0.728	0.0126652328
MHS–PCLS	0.05168918	0.35672077	11.288788	−2.2128E–10	−4.5078E–11	−4.0538	−0.7277	0.0126652
I-ABC *greedy*	0.051686	0.356014	11.202765	– 6.98E–05	– 2.08E–05	– 4.053.610	– 1.091.686	0.012665

w.r.t. restraints

$12.0 \le z_i \le 60.0.$

The obtained simulated results are presented in Table. To test a13nd validate the efficiency the proposal is compared with results obtained by CSA (Cuckoo Search Algorithm) (Gandomi et al. 2013); UPSO (Unified PSO) (Wang et al. 2005); ABC (Karaboga and Basturk 2007a); APSO (Yang 2010); MBA (Sadollah et al. 2013); IAPSO (Guedria 2016); and UFA (Brajević and Ignjatović 2018). In Table 12, the value obtained for all the decision variables is given and compared with CSA, MBA, APSO, IAPSO and UFA. In Table 13 comparative statistical results obtained in terms of worst, mean, best, SD and NFSs are presented. It can be analyzed from the Tables that I-ABC *greedy* is competitive in solving the problem of GTD. I-ABC *greedy* converges fast and took only 795 NFEs to achieve the optimized results. Also the comparative analysis is depicted in Fig. 10.

6.5 Speed reducer design (SRD) problem

The problem of speed reducer is taken form Golinski (1973) and Rao (1996) with an objective of optimizing the weight of speed reducer with respect to the restraints and conditions. The problem has seven decision variables with eleven restraints and boundary conditions for seven variables. All the variables are continuous except z_3. There are two cases of the problem, with a difference of only one boundary condition of z_5. This problem is complex and mixed integer. The diagram of SED is shown in Fig. 11 and mathematical formulation is presented below:

Case 1

$$\begin{aligned} \textit{Minimize}\ f(Z) &= 0.7854 z_1 z_2^2 \left(3.3333 z_3^2 + 14.9334 z_3 - 43.0934\right) - 1.508 z_1 \left(z_6^2 + z_7^2\right) \\ &+ 7.4777\left(z_6^3 + z_7^3\right) + 0.7854(z_4 z_6^2 + z_5 z_7^2) \end{aligned}$$

$$h_1(Z) = \frac{27}{z_1 z_2^2 z_3} - 1 \le 0$$

$$h_2(Z) = \frac{397.5}{z_1 z_2^2 z_3^2} - 1 \le 0$$

$$h_3(Z) = \frac{1.93 z_4^3}{z_2 z_3 z_6^4} - 1 \le 0$$

$$h_4(Z) = \frac{1.93 z_5^3}{z_2 z_3 z_7^4} - 1 \le 0$$

Table 10 The statically simulated results of T/CSD problem attained (worst, mean, best SD and NFE's) by DE, DELC, DEDS, HEAA, PSO-DE, SC, (μ+λ)-ES, ABC, LCA, WCA, MBA, APSO, ABC, GA1

Opti.	DE	DELC	DEDS	HEAA	PSO-DE	SC	(μ+λ)-ES	LCA	WCA	MBA	APSO	ABC	GA1
Worst	0.01279	0.012665	0.012738	0.012665	0.012665	0.016717	NA	0.0126667	0.012952	0.0129	0.014937	NA	0.012822
Mean	0.012703	0.012665	0.012669	0.012665	0.012665	0.012922	0.013165	0.0126654	0.012746	0.012713	0.013297	0.012709	0.012769
Best	0.01267	0.012665	0.012665	0.012665	0.012665	0.012669	0.012689	0.0126652	0.012665	0.012665	0.0127	**0.012665**	0.012704
S.D	2.7E−05	1.3E−07	1.3E−05	1.4E−09	1.2E−08	5.9E−04	3.9E−04	3.88E−07	8.06E−05	6.30E−05	6.85E−04	0.012813	3.94E−05
NFEs	204,800	20,000	24,000	24,000	24,950	25,167	30,000	15,000	11,750	7650	120,000	30,000	900,000

Table 11 The statically simulated results of T/CSD problem attained (worst, mean, best SD and NFE's) by I-ABC *greedy*, GA2, CAEP, CPSO, HPSO, NM–PSO, G-QPSO, QPSO, PSO, rank-iMDDE, AFA, IAPSO, UFA and MHS–PCLS

Opti.	GA2	CAEP	CPSO	HPSO	NM-PSO	G-QPSO	QPSO	PSO	rank-iMDDE	AFA	IAPSO	UFA	MHS–PCLS	I-ABC *greedy*
Worst	0.012973	0.015116	0.012924	0.012719	0.012633	0.017759	0.018127	0.071802	0.0126674	0.0000128	0.0178286	0.01266523	**0.0126652**	0.01812453
Mean	0.012742	0.013568	0.01273	0.012707	0.012631	0.013524	0.013854	0.019555	0.01266529	0.012677	0.01367652	0.012665232	**0.0126652**	0.013731147
Best	0.012681	0.012721	0.012674	0.012665	0.01263	0.012665	0.012669	0.012857	**0.0126652**	0.0126653	**0.0126652**	**0.01266523**	**0.0126652**	**0.012665**
SD	5.90E−05	8.42E−04	5.20E−04	1.58E−05	8.47E−07	0.001268	0.001341	0.011662	8.48E−07	0.0127116	1.573E−3	1.15E−10	**3.6008e-11**	1.12E−06
NFEs	80,000	50,020	240,000	81,000	80,000	2000	2000	2000	10,000	50,000	2000	**2000**	10,000	**2000**

The best results are highlighted in bold

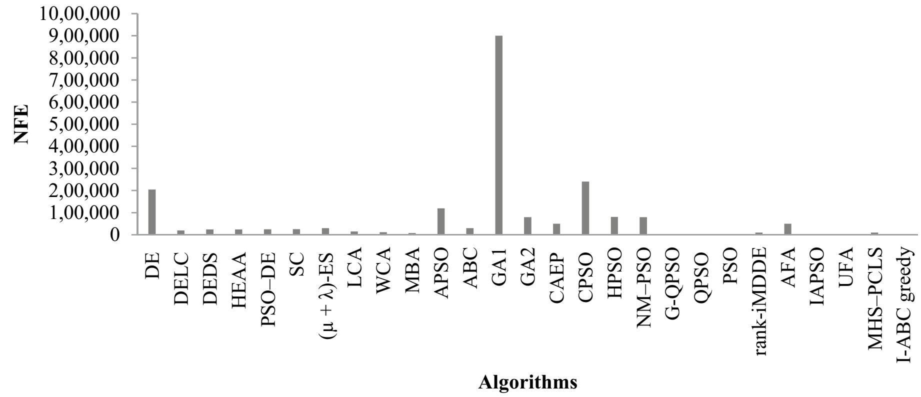


Fig. 8 NFE's used by the optimizers to obtain optimal value in T/CSD Optimization Problem

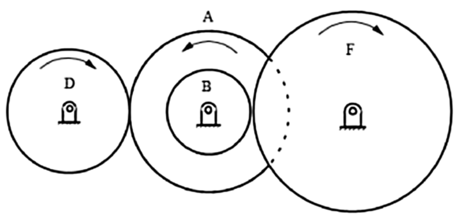


Fig. 9 Illustration of Gear Train Design (GTD) Optimization problem

$$h_5(Z) = \frac{1.0}{110z_6^3}\sqrt{\left(\frac{745.0z_4}{z_2z_3}\right)^2 + 16.9\times 10^6} - 1 \le 0$$

$$h_6(Z) = \frac{1.0}{85z_7^3}\sqrt{\left(\frac{745.0z_4}{z_2z_3}\right)^2 + 157.5\times 10^6} - 1 \le 0$$

$$h_7(Z) = \frac{z_2z_3}{40} - 1 \le 0$$

$$h_8(Z) = \frac{5z_2}{z_1} - 1 \le 0$$

$$h_9(Z) = \frac{z_1}{12z_2} - 1 \le 0$$

$$h_{10}(Z) = \frac{1.5z_6 + 1.9}{z_4} - 1 \le 0$$

$$h_{11}(Z) = \frac{1.5z_7 + 1.9}{z_5} - 1 \le 0.$$

Table 12 Best and comparative results of attained by GTD problem obtained by CS, MBA, APSO, ABC, IAPSO, UFA and I-ABC *greedy*

D.V.	z_1	z_2	z_3	z_4	$f(Z)$
CS	43	16	19	49	2.70E–12
MBA	43	16	19	49	2.70E–12
APSO	43	16	19	49	2.70E–12
ABC	49	16	19	43	0
IAPSO	43	16	19	49	2.700857E–12
UFA	49	16	19	43	2.700857E–12
I-ABC greedy	43	16	19	49	2.70E–12

Table 13 The statically simulated results of GTD problem attained (worst, mean, best SD and NFE's) by MBA, UPSO, CS, APSO, ABC, IAPSO, UFA and I-ABC *greedy*

Optimizer	MBA	UPSO	CS	APSO	ABC	IAPSO	UFA	I-ABC *greedy*
Worst	2.062904E–08	N.A	2.36E–09	7.07E–06	NA	1.827380E–08	**1.361649E-09**	1.68E–08
Mean	2.471635E–09	3.80562E–08	1.98E–09	4.78E–07	3.641339E–10	5.492477E–09	**2.953672E-10**	6.452E–09
Best	2.700857E–12	2.700857E–12	2.70E–12	2.70E–12	2.700857E–12	2.700857E–12	2.700857E–12	**2.702E–12**
SD	3.94E–09	1.09E–07	3.55E–09	1.44E–06	5.525811E–10	6.36E–09	**3.75E-10**	5.29E–10
NFSs	1120	100,000	5,000	8,000	60	800	450	**60**

The best results are highlighted in bold

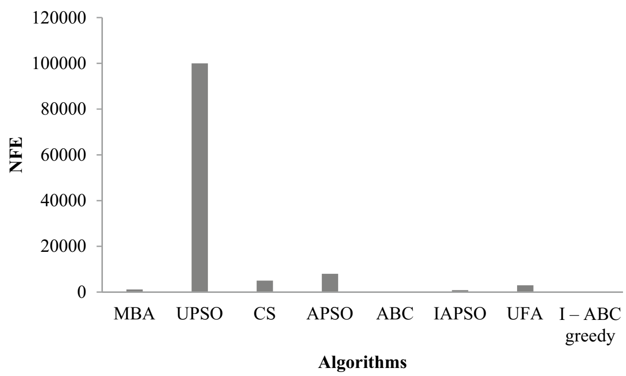


Fig. 10 NFE's used by the optimizers to obtain optimal value in GTD Optimization Problem

where

$$2.6 \le z_1 \le 3.6; 0.7 \le z_2 \le 0.8; 17 \le z_3 \le 28; 7.3 \le z_4 \le 8.3; 7.8 \le z_5 \le 8.3; 2.9 \le z_6 \le 3.9; 5.0 \le z_7 \le 5.5$$

***Case* 2** The problem differs in the bound of variable z_5 that changes to:

$$7.3 \le z_5 \le 8.3.$$

The simulated results of SRD problem are presented in Tables 14, 15, 16 and 17. The results of the proposal are also compared with SES (Coello Coello and Landa-becerra 2003); COPSO (Aguirre et al. 2007); DES (Kim et al. 2007); SiC-PSO (Cagnina et al. 2008); MBFOA (Mezura-Montes and Hernández-Ocaña 2009a; Mezura-Montes and Cetina-Domínguez 2009b); LCA (Kashan 2011); rank-iMDDE (Gong et al. 2014); AFA (Baykasolu and Ozsoydan 2015); IAPSO (Guedria 2016); MHS-PCLS (Yi et al. 2016); and UFA (Brajević and Ignjatović 2018). I-ABC *greedy* has shown the effective performance in optimizing the results of the problem. It can be observed form the results in Table 14 that I-ABC *greedy* took 6500 NFEs to obtain the optimal results where as UFA took only 3000 NFE's to achieve the optimal result. UFA converges faster in comparison to the proposed algorithm but the statistical results presents the efficacy of I-ABC *greedy* algorithm in comparison to other Metaheuristics considered. The same is depicted in Fig. 12.

6.6 Cantilever beam design

This problem is refereed from Erbatur et al. (2000). In this problem volume is to be minimized. There are ten decision variables ($b_1,\ldots,b_5$; $h_1,\ldots,h_5$) that correspond to height and breadth of each rectangular cross section. The problem is defined as:

$$Minimize\, f(Z) = 100 \sum_{k=1}^{5} h_i b_i.$$

w.r.t. restraints

$$h_k(Z) = \alpha_k \le 14000\ N/cm^2$$

$$h_{k+5}(Z) = \frac{h_k}{b_k} \le 20$$

$$h_{11}(Z) = \beta \le 2.7\ cm,$$

dip reflection in vertical direction.

Where b_1, h_1 are integer variables; b_2, b_3 have the discrete values and generally considered from 2.40, 2.60, 2.80 and 3.10; also h_2, h_3 have the discrete values and generally considered from 45.00, 50.00, 55.00 and 60.00. Variables b_4, b_5, h_4 and h_5 are continuous in nature. 200GPa is the Young's modulus of the material. The cantilever beam is illustrated in Fig. 13.

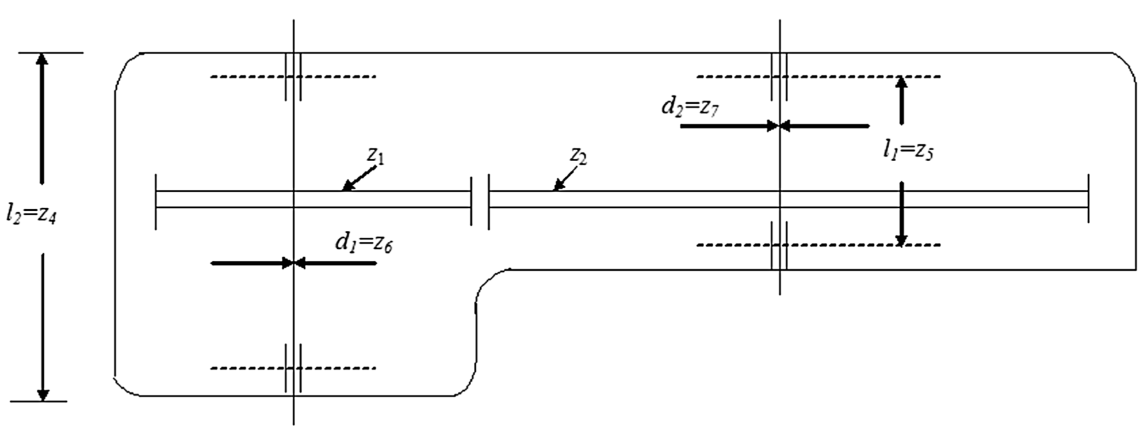


Fig. 11 Illustration of SRD problem

Table 14 The statically simulated results of SRD problem attained (worst, mean, best SD and NFE's) by SiC-PSO, COPSO, MBFOA, DES, SES, LCA, APSO, ABC, IAPSO, UFA and I-ABC *greedy* (*Case* – 1)

Method	SiC-PSO	COPSO	MBFOA	DES	SES	LCA	APSO	ABC	IAPSO	UFA	I-ABC *greedy*
Worst	NA	NA	NA	NA	3226.248291	2996.348165	4677.005187	NA	2996.34816497	2996.3481649986	**2996.348165**
Mean	2996.4085	2996.408525	3014.759	2996.348	3088.777816	2996.348165	3855.581557	2997.058412	2996.34816497	2996.3481649885	**2996.348165**
Best	2996.34816	2996.372448	2999.264	2996.348	3025.005127	2996.348165	3177.530771	2997.058412	2996.34816497	2996.3481649760	**2996.348165**
SD	0	2.867E–02	11	7.54E–06	47.36189	2.63E–12	473.767	0	**688E–13**	4.51E–09	6.25E–12
NFE's	24,000	30,000	30,000	36,000	36,000	24,000	30,000	30,000	6000	3000	6500

The best results are highlighted in bold

Table 15 The best simulated results for SRD problem attained by different Optimizers (*Case* – 2)

D.V.	z_1	z_2	z_3	z_4	z_5	z_6	z_7	$f(Z)$
DEDS	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
DELC	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
HEAA	3.500022	0.7	17.000012	7.300427	7.715377	3.35023	5.286663	2994.499107
MDE	3.50001	0.7	17	7.300156	7.800027	3.350221	5.286685	2996.356689
PSO-DE	3.5	0.7	17	7.3	7.8	3.350214	5.2866832	2996.348167
WCA	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
MBA	3.5	0.7	17	7.300033	7.715772	3.350218	5.286654	2994.482453
LCA	3.5	0.7	17	7.3	7.8	3.350214666	5.28668323	2994.471066
APSO	3.501313	0.7	18	8.127814	8.042121	3.352446	5.287076	3187.630486
ABC	3.499999	0.7	17	7.3	7.8	3.350215	5.287800	2997.058412
IAPSO	3.5	0.7	17	7.3	7.7153199	3.350214666096	5.286654464979	2994.4710661459
UFA	3.5	0.7	17	7.3	7.8	3.35021466610	5.2866832297	2996.34816
MHS–PCLS	3.5	0.7	17	7.3	7.7153199	3.3502146	5.2866545	2994.471068
I-ABC *greedy*	3.50021	0.7	17	7.30	7.71531189	3.350214689	5.2866554	2994.4710315

Table 16 The statically simulated results of SRD problem attained (worst, mean, best SD and NFE's) by SC, PSO-DE, DELC, DEDS, HEAA, MDE, (μ+λ)-ES, ABC, WCA, LCA (*Case* – 2)

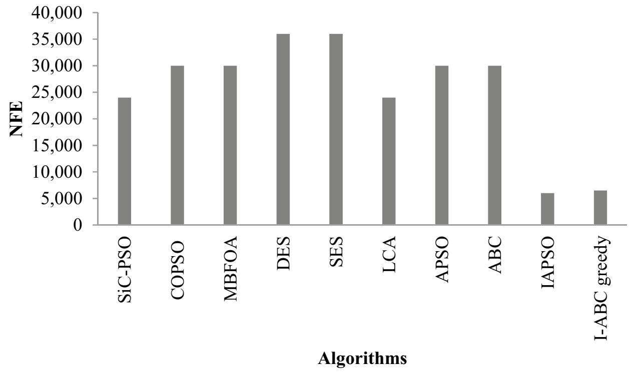
Opti	SC	PSO-DE	DELC	DEDS	HEAA	MDE	(μ+λ)-ES	ABC	WCA	LCA
Worst	3009.964736	2996.348204	2994.47107	2994.47107	2994.752311	NA	NA	NA	2994.505578	2994.471066
Mean	3001.758264	2996.348174	2994.47107	2994.47107	2994.613368	2996.36722	2996.348	2997.058	2994.474392	2994.471066
Best	2994.744241	2996.348167	2994.47107	2994.47107	2994.499107	2996.356689	2996.348	2997.058	2994.471066	2994.471066
S.D	4	6.4E -06	1.9E -12	3.6E–12	7.0E–02	8.2E -03	0	0	7.4E–03	2.66E–12
NFSs	54,456	54,350	30,000	30,000	40,000	24,000	30,000	30,000	**15,150**	24,000

The best result is highlighted in bold

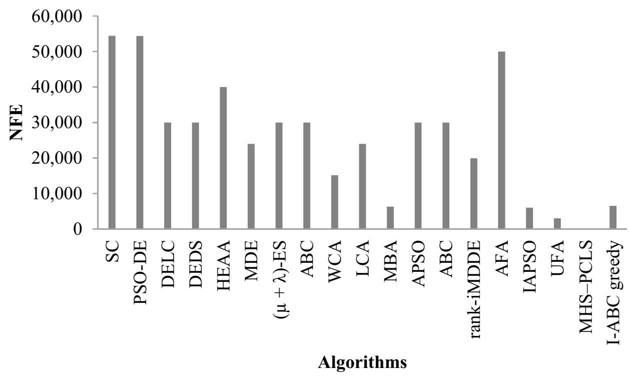
Table 17 The statically simulated results of SRD problem attained (worst, mean, best SD and NFE's) by MBA, APSO, ABC, rank-iMDDE, AFA, IAPSO, UFA, MHS-PCLS and I-ABC *greedy* (*Case* – 2)

Opti	MBA	APSO	ABC	rank-iMDDE	AFA	IAPSO	UFA	MHS-PCLS	I-ABC *greedy*
Worst	2999.6524	4443.01764	NA	2994.471066	2996.669016	2994.47106615489	2994.4710662200	**2994.471106**	2994.902
Mean	2996.769	3822.64062	2995.6531	2994.471066	2996.514874	2994.47106614777	2994.4710661872	**2994.471077**	2994.6631
Best	2994.4825	3187.63049	2994.47243	2994.471066	2996.372698	2994.47106614598	2994.4710661647	2994.471068	**2994.4710**
S.D	1.56	366.146	2.98E–12	**7.93E–13**	0.09	2.65E -09	1.53E–08	7.142949E–06	**1.87E–12**
NFSs	6300	30,000	30,000	19,920	50,000	6000	**3000**	10, 000	6500

The best results are highlighted in bold



(a) Case 1



(b) Case 2

Fig. 12 NFE's used by the optimizers to obtain optimal value in SRD Problem. **a** Case 1, **b** Case 2

6.6.1 Result discussion of Cantilever design problem

The statistical simulated results are presented in the Tables 18 and 19. The results of I-ABC *greedy* are compared with that of basic ABC, GAOS-Erbatur (Erbatur et al. 2000); AIS-GA & AIS-GA^{C} (Bernardino et al. 2007); (AIS-GA^{H}) & (APM^{bc}) (Bernardino et al. 2008) and SR (Runarsson and Yao 2000). The parametric setting for all the algorithms are same as mentioned in the research articles. In Table 18, results found in terms of Best, Median, Average (Avg.), Standard Deviation (SD), Worst and *NFE* are presented. I-ABC *greedy* has shown the efficiency in evaluating the best optimized value which is very close to the value achieved by SR. Also I-ABC *greedy* took minimum number of *NFE* in achieving the optimized volume. In other cases *NFE* is fixed. Figure 14 depicts the NFEs comparisons. Table 19 presented the value of the decision variables ($b_1,\ldots b_5$ & $h_1,\ldots,h_5$) and the estimated value of the volume achieved.

Table 18 Details of volume achieved for Cantilever beam design by I-ABC *greedy*, GAOS-Erbatur, AIS-GA, AIS-GA^{C}, AIS-GA^{H}, APM^{bc}, SR, APM^{rc}

Opti	GAOS-Erbatur	AIS-GA	AIS-GA^{C}	AIS-GA^{H}	APM^{bc}	SR	APM^{rc}	ABC	I-ABC *greedy*
Best	64,815	65559.6	66533.47	64834.7	66030.05	64599.65	64647.82	64599.67	**64599.65**
Median	-	-	-	74987.16	79466.1	70508.33	76721.19	66298.52	**66291.87**
Avg	-	70857.12	71821.69	76004.24	83524.21	71240.03	79804.77	67995.18	**68263.83**
S.D	-	-	-	6.93E+03	1.44E+04	3.90E+03	1.63E+04	2.09E+04	**5.97E+03**
Worst	-	77272.78	76852.86	102981.06	151458.17	83968.45	162089.24	69895.75	**69877.91**
NFE's	10,000	35,000	35,000	35,000	35,000	35,000	35,000	30,000	**9700**

The best results are highlighted in bold

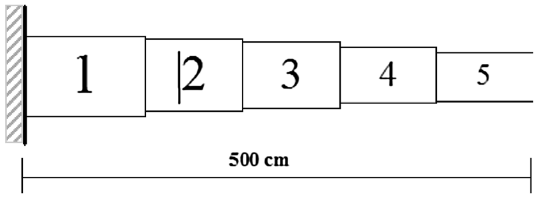


Fig. 13 Cantilever Beam design

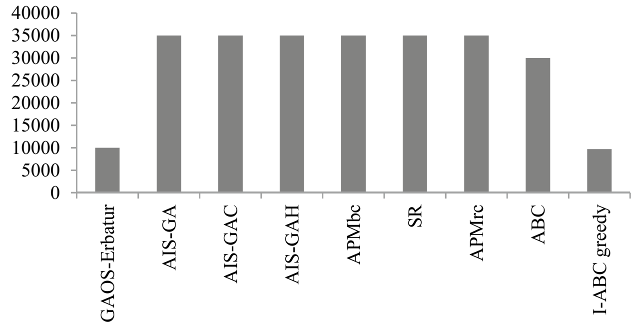


Fig. 14 NFE's used by the optimizers to obtain optimal value in Cantilever design Problem

6.7 Optimization problem of design of Multiple Disk Clutch Brake (MDCB)

This problem is taken from Osyczka (2002). The objective of the problem is to minimize the mass of MDCB. The problem has five decision discrete variables namely disc thickness, actuating force, inner & outer radius as well as friction surfaces that need to be computed while satisfying eight restraints. These variables are represented by γ_0, γ_1, γ_2, γ_3 and γ_4. These variables are conditioned to select the following values:

$$\gamma_0 = 60,\ 61,\ 62,\ 63\ldots,\ 80;\ \gamma_1 = 90,\ 91,\ 92,\ 93...,110;\ \gamma_2 = 1,\ 1.5,\ 2.0,\ 2.5\ldots,3;$$
$$\gamma_3 = 600,\ 610,\ 620,\ 630\ldots,1000 \text{ and } \gamma_4 = 2,\ 3,\ 4,\ 5\ldots,\ 9.$$

The problem is defined below and the illustration of MDCB is shown in Fig. 15.

$$Minimize\ f(Z) = \pi\left(\gamma_0^2 - \gamma_i^2\right) \times t \times (Q+1)\rho$$

w.r.t. restraints

$$h_1(Z) = \gamma_0 - \gamma_i - \Delta r \geq 0$$

$$h_2(Z) = l_{max} - (Q+1)(t+\delta) \geq 0$$

$$h_3(Z) = P_{max} - P_{rz} \geq 0$$

$$h_4(Z) = P_{max} \times \vartheta_{sr\ max} - P_{rz} \times \vartheta_{sr} \geq 0$$

$$h_5(Z) = \vartheta_{sr\ max} - \vartheta_{sr} \geq 0$$

$$h_6(Z) = T_{max} - T \geq 0$$

$$h_7(Z) = M_h - sM_s \geq 0$$

$$h_8(Z) = T \geq 0$$

where

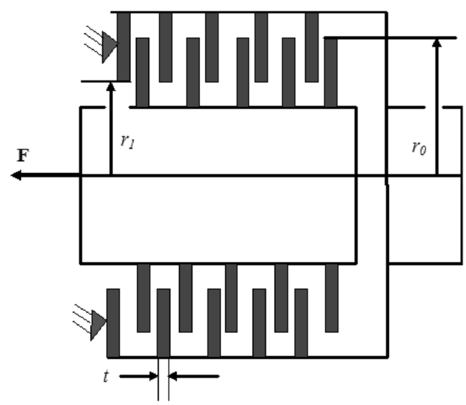


Fig. 15 Illustration of MDCB

Table 19 Simulated comparative best results for the decision variables in Cantilever design problem by by I-ABC *greedy*, GAOS-Erbatur, AIS-GA, AIS-GA^C, AIS-GA^H, APM^{bc}, SR, APM^{rc}

D.V.	b_1	b_2	b_3	b_4	b_5	h_1	h_2	h_3	h_4	h_5	V
GAOS-Erbatur	3	3.1	2.6	2.3	1.8	60	55	50	45.5	35	64,815
AIS-GA	3	3.1	2.8	2.2348	2.0038	60	55	50	44.3945	32.878708	65559.6
AIS-GA^C	3	3.1	2.6	2.3107	2.2254	60	60	50	43.1857	31.250282	66533.47
AIS-GA^H	3	3.1	2.6	2.2947	1.825	60	55	50	45.2153	35.1191	64834.7
APM^{bc}	3	3.1	2.6	2.2094	2.0944	60	60	50	44.0428	31.9867	66030.05
SR	3	3.1	2.6	2.2837	1.7532	60	55	50	45.5507	35.0631	64599.65
APM^{rc}	3	3.1	2.6	2.2978	1.7574	60	55	50	45.5037	34.9492	64647.82
ABC	3	3.1	2.6	2.2977	1.7575	60	55	50	45.5508	24.9786	64599.65
I-ABC *greedy*	3	3.1	2.6	2.291	2.0671	60	55	50	45.5487	35.0912	64599.64

Table 20 The statically simulated results of MDCBD problem attained (worst, mean, best SD and NFE's) by ABC, I-ABC *greedy*, TLBO, WCA, APSO, IAPSO

Optimizers	TLBO	WCA	APSO	ABC	IAPSO	I-ABC *greedy*
Worst	0.392071	0.313656	0.716313	0.313677	**0.313656**	**0.313656**
Mean	0.327166	0.313656	0.506829	0.313659	**0.313656**	**0.313656**
Best	0.313657	0.313656	0.337181	0.313657	**0.313656**	**0.313656**
S.D	NA	1.69E–16	0.09767	5.97E–15	**1.13E–16**	1.27E–16
NFEs	> 900	500	2000	1500	**400**	750

The best results are highlighted in bold

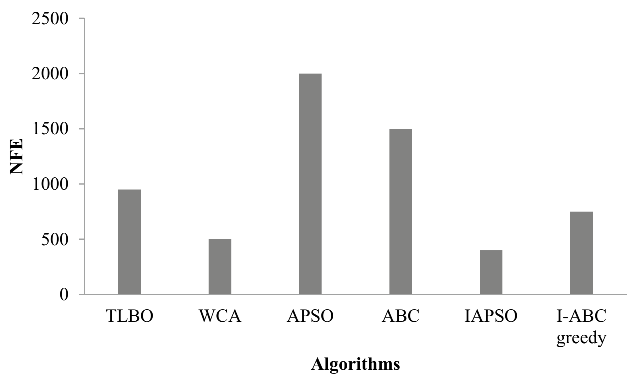


Fig. 16 NFE's used by the optimizers to obtain optimal value in MDCBD problem

$$M_h = \frac{2}{3}\mu FZ\frac{\gamma_0^3 - \gamma_i^3}{\gamma_0^2 - \gamma_i^2}$$

$$P_{rz} = \frac{F}{\pi(\gamma_0^2 - \gamma_i^2)}$$

$$v_{rz} = \frac{2\pi n\left(\gamma_0^3 - \gamma_i^3\right)}{90\left(\gamma_0^2 - \gamma_i^2\right)}$$

$$T = \frac{I_2 \pi n}{30(M_h - M_f)}$$

$$t_{min} = 1.5;\ t_{max} = 3;\ \Delta r = 20\ mm;$$
$$I_z = 55kgmm^2;\ F_{min} = 600;\ Z_{min} = 2;$$
$$Z_{max} = 9;\ P_{max} = 1Mpa;\ F_{max} = 1000N;\ T_{max} = 15s;\ \mu = 0.5;\ s = 1.5;$$
$$M_s = 40Nm;\ M_f = 3Nm;\ n = 250rpm;\ v_{sr\ max} = \frac{10m}{s};\ l_{max} = 30mm;\ \gamma_{0\ min} = 90;\ \gamma_{0\ max} = 110;$$
$$\gamma_{i\ min} = 60;\ \gamma_{i\ max} = 80;$$

The simulated results of MDCB are presented in the Tables 20 and 21. The results of I-ABC *greedy* are compared with the simulated results of TLBO, WCA, APSO, ABC and IAPSO in terms of Worst, Mean, Best, SD and NFEs. I-ABC greedy took 750 NFEs where as IAPSO achieved the optimal results within 400 NFEs. Further best and comparative results of attained by MDCBD problem obtained by ABC, I-ABC greedy, NSGA-II (Deb and Srinivasan 2006); TLBO (Rao et al. 2011); WCA (Eskandar et al. 2012); APSO (Yang 2010) and IAPSO (Guedria 2016) are presented in Table 21 (see Fig. 16).

6.8 Analyses

In Fig. 17 a radar presentation of the mean values obtained by all the seven considered problems by different optimizers are presented. Radar basically displays values to the centre point. It can be analyzed that I-ABC *greedy* is able to solve all the considered problems competitively with best convergence rate. In case of GTD problem the graph is plotted

Table 21 Best and comparative results of attained by MDCBD problem obtained by ABC, I-ABC *greedy*, NSGA-II, TLBO, WCA and APSO

D.V.	z_1	z_2	z_3	z_4	z_5	$h_1(Z)$	$h_2(Z)$	$h_3(Z)$	$h_4(Z)$	$h_5(Z)$	$h_6(Z)$	$h_7(Z)$	$h_8(Z)$	$f(Z)$
NSGA-II	70	90	1.5	1000	3	0	22	0.9005	9.7906	7.8947	3.3527	60.625	11.6473	0.4704
TLBO	70	90	1	810	3	0	24	0.919427	9830.371	7894.6965	0.702013	37706.25	14.297986	0.313656
WCA	70	90	1	910	3	0	24	0.90948	9.809429	7.894696	2.231421	49.768749	12.768578	0.313656
APSO	76	96	1	840	3	0	24	0.922273167	9.824211285	7.738378002	1.3966105	48.8483721	13.60338949	0.337181
ABC	70	90	1	900	3	0	24	0.9114743	9.8219772	7.8956441	1.3895732	48.678478	13.658382	0.317652
IAPSO	70	90	1	900	3	0	24	0.910475344	9.8115234375	7.89469659	1.359771388	48.5625	13.64022861	0.313656
I-ABC *greedy*	70	90	1	900	3	0	24	0.9108733	9.8207371	7.8946555	1.3619831	48.57545	13.641874	0.313766

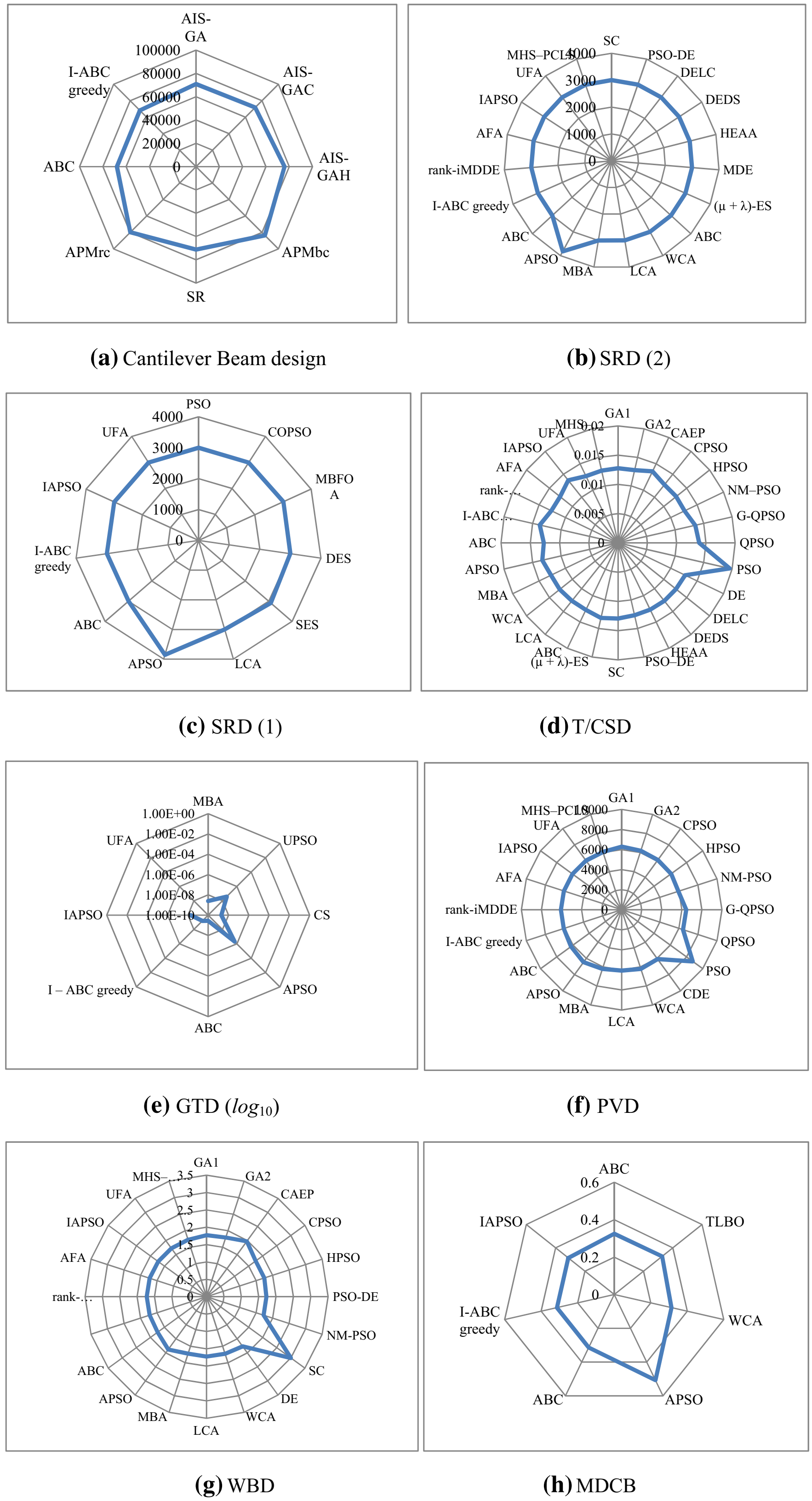


Fig. 17 Radar illustration of mean values of all the seven problems

Table 22 Statistical results based on performance of Optimizers for Seven mechanical engineering design problems

	Best	Mean	Worst	NFE	SD
Welded beam design (WBD) problem	LCA/ABC/ I-ABC *greedy*	MHS-PCLS	MHS–PCLS	UFA	AFA
Pressure Vessel Design (PVD) Problem	HPSO/ I-ABC *greedy*	UFA	UFA	UFA	UFA
Tension/Compression Spring Design (T/CSD) optimization problem	MHS–PCLS/UFA/I-ABC *greedy*	I-ABC *greedy*	NM-PSO	UFA	LCA
Gear Train Design (GTD) Optimization problem	Rank-iMDDE/ IAPSO/UFA/ MHS-PCLS/I-ABC greedy	MHS–PCLS	MHS–PCLS	ABC/ I-ABC *greedy*	MHS-PCLS
Speed Reducer Design (SRD) problem	LCA/ I-ABC *greedy*	UFA	UFA	ABC/I-ABC *greedy*	UFA
Cantilever Beam design	I-ABC *greedy*	I-ABC *greedy*	I-ABC *greedy*	I-ABC *greedy*	I-ABC *greedy*
Optimization problem of design of Multiple Disk Clutch Brake (MDCB)	I-ABC *greedy*	I-ABC *greedy*	I-ABC *greedy*	IAPSO	IAPSO

using logarithm of base 10 as the values are very close to zero. Also, in Table 22 the statistical results based on performance of optimizers for seven mechanical engineering design problems are discussed.

7 Conclusions

In this study a variant of ABC named I-ABC *greedy* is presented. I-ABC *greedy* incorporates OBL concept for maintaining the population diversity and the modified searching behavior of employed and onlooker bee to enhance exploitation. So, I-ABC *greedy* balances exploration and exploitation process and assist in accelerating convergence rate. I-ABC greedy uses Debs technique for handling the constraints. The proposal is validated on a set of seven mechanical engineering optimization problems is taken from the literature. The results of I-ABC *greedy* are compared with the results of the other optimizers refereed from the literature. The statically simulated results on all the problems justify the effectiveness of I-ABC *greedy* algorithm in comparison to other considered optimizers. Also I-ABC *greedy* demonstrates the efficient convergence rate in achieving optimal results.

In future the constrained handling techniques affect can be examined on I-ABC greedy as well as multi-objective optimization problems would be considered to evaluate the efficiency of I-ABC *greedy* algorithm.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Aguirre AH, Zavala AEM, Villa E, Hern A, Mu AE (2007) COPSO: constrained optimization via PSO algorithm. Comunicación Técnica No I-07-04/22-02-2007 (CC/CIMAT)

Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf 23:1001–1014

Akay B, Karaboga D (2017) Artificial bee colony algorithm variants on constrained optimization. Int J Optim Control Theories Appl 7(1):98–111

Ayan K, Kılıç U, Baraklı B (2015) Chaotic artificial bee colony algorithm based solution of security and transient stability constrained optimal power flow. Int J Electr Power Energy Syst 64:136–147

Babaeizadeh S, Ahmad R (2016) An improved artificial bee colony algorithm for constrained optimization. Res J Appl Sci 11(1):14–22

Bansal JC, Sharma H (2012) Cognitive learning in differential evolution and its application to model order reduction problem for single-input single-output systems. Memet Comput 4(3):209–229

Baykasolu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164

Bernardino H, Barbosa H, Lemonge A (2007) A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In Proceedings of the 2007 IEEE congress on evolutionary computation. IEEE Press, Singapore, pp 646–653

Bernardino H, Barbosa H, Lemonge A, Fonseca L (2008) A new hybrid AIS-GA for constrained optimization problems in mechanical engineering. In Proceedings of the 2008 ieee congresson evolutionary computation. IEEE Press, Hong Kong, pp 1455–1462

Bhambu P, Sharma S, Kumar S (2018) Modified Gbest Artificial Bee Colony algorithm. In: Pant M, Ray K, Sharma T, Rawat S, Bandyopadhyay A (eds) Soft computing: theories and applications. Advances in intelligent systems and computing, vol 583. Springer, Singapore

Brajevic I (2015) Crossover-based artificial bee colony algorithm for constrained optimization problems. Neural Comput Appl 26(7):1587–1601

Brajevic I, Tuba M (2013) An upgraded artificial bee colony (abc) algorithm for constrained optimization problems. J Intell Manuf 24(4):729–740

Brajevic I, Tuba M, Subotic M (2011) Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. Int J Math Comput Simul 5(2):135–143

Brajević I, Ignjatović J (2018) An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems. J Intell Manuf. https://doi.org/10.1007/s10845-018-1419-6, (2018)

Cagnina LC, Esquivel SC, Coello Coello CA (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer. Inform 32:319–326

Chen T, Chuang YH (2018) Fuzzy and nonlinear programming approach for optimizing the performance of ubiquitous hotel recommendation. J Ambient Intell Human Comput 9(2):275–284
Coelho LDS (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683
Coello Coello CA (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127
Coello Coello CA, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36:219–236
Coello Coello CA, Landa-becerra R (2003) Engineering optimization using a simple evolutionary algorithm, pp 149–156
Coello Coello CA, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 16:193–203
D'Apice C, Nicola CD, Manzo R, Moccia V (2014) Optimal scheduling for aircraft departures. J Ambient Intell Human Comput 5(6):799–807
Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Method Appl M 186(2):311–338
Deb K, Srinivasan A (2006) Innovization. In: Proc. 8th Annu. Conf. Genet. Evol. Comput.–GECCO'06, ACM Press, New York, New York, USA, pp 1629–1636
Diwold K, Aderhold A, Scheidler A, Middendorf M (2011) Performance evaluation of artificial bee colony optimization and new selection schemes. Memet Comput 3(3):149–162
Erbatur F, Hasançebi O, Tütüncü I, Kilç H (2000) Optimal design of planar and space structures with genetic algorithms. Comput Struct 75:209–224
Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166
Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
Golinski J (1973) An adaptive optimization system applied to machine synthesis. Mech Mach Theory 8:419–436
Gong W, Cai Z, Liang D (2014) Engineering optimization by means of an improved constrained differential evolution. Comput Methods Appl Mech Eng 268:884–904
Guedria NB (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. Appl Soft Comput 40:455–467
Ha M, Gao Z (2017) Optimization of water allocation decisions under uncertainty: the case of option contracts. J Ambient Intell Human Comput 8(5):809–818
He Q, Wang L (2007a) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422
He Q, Wang L (2007b) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
Holland CJ (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356
Jadon SS, Bansal JC, Tiwari R, Sharma H (2015) Accelerating artificial bee colony algorithm with adaptive local search. Memet Comp 7:215–230
Kannan BK, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411
Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department
Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
Karaboga D, Akay B (2011) A modified artificial bee colony (ABC) algorithm for constrained optimization problems. Appl Soft Comput 11(3):3021–3031
Karaboga D, Basturk B (2007a) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39(3):459–471
Karaboga D, Basturk B (2007b) Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. In: Melin P, Castillo O, Aguilar LT, Kacprzyk J, Pedrycz W (eds) Foundations of fuzzy logic and soft computing. IFSA 2007. Lecture notes in computer science, vol 4529. Springer, Berlin, Heidelberg
Kashan AH (2011) An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA). Comput Des 43:1769–1792
Kim H-KKH-K, Chong J-KCJ-K, Park K-YPK-Y, Lowther DA (2007) Differential evolution strategy for constrained global optimization and application to practical engineering problems. IEEE Trans Magn 43:1565–1568
Krohling RA, Coelho LDS (2006) Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern B Cybern 36:1407–1416
Kumar S, Kumar P, Sharma TK, Pant M (2013) Bi-level thresholding using PSO, artificial bee colony and MRLDE embedded with Otsu method. Memetic Comp 5(4):323–334
Lampinen J (2002) A constraint handling approach for the differential evolution algorithm. In: Proc. 2002 Congr. Evol. Comput. CEC'02 (Cat. No.02TH8600), IEEE, pp. 1468–1473
Li X, Yin M (2014) Self-adaptive constrained artificial bee colony for constrained numerical optimization. Neural Comput Appl 24(3–4):723–734
Liang J, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan P, Coello CC, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. J Appl Mech 41:1–8
Liang Y, Wan Z, Fang D (2017) An improved artificial bee colony algorithm for solving constrained optimization problems. Int J Mach Learn Cyber 8(3):739–754
Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput J 10:629–640
Liu F, Sun Y, Gai-ge W, Wu T (2018) An artificial bee colony algorithm based on dynamic penalty and Lévy flight for constrained optimization problems. Arab J Sci Eng. https://doi.org/10.1007/s13369-017-3049-2
Mezura-Montes E, Cetina-Domınguez O (2012) Empirical analysis of a modified artificial bee colony for constrained numerical optimization. Appl Math Comput 218(22):10943–10973
Mezura-Montes E, Coello Coello CA (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms (eds) MICAI 2005. Lect Notes Artif Int 3789:652–662
Mezura-Montes E, Coello Coello CA (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1(4):173–194
Mezura-Montes E, Hernández-Ocaña B (2009a) Modified bacterial foraging optimization for engineering design. In: Dagli CH, Bryden KM, Corns SM, Gen M, Tumer K, Süer G (eds) Intelligent engineering systems through artificial neural networks. ASME Press, New York. https://doi.org/10.1115/1.802953.paper45

Mezura-Montes E, Cetina-Domínguez O (2009b) Exploring promising regions of the search space with the Scout Bee in the Artificial Bee Colony for constrained optimization. In: Proceedings of the artificial neural networks in enginnering conference (ANNIE'2009). https://doi.org/10.1115/1.802953.paper32
Osyczka (2002) Evolutionary algorithms for single and multicriteria design optimization: studies in fuzzyness and soft computing. Physica-Verlag, Heidelberg, p 218
Pan Q-K (2016) An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. Eur J Oper Res 250(3):702–714
Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Opposition-based differential evolution. IEEE Trans Evol Comput 12(1):64–79
Rajpurohit J, Sharma TK, Abraham A, Vaishali (2017) Glossary of metaheuristic algorithms. Int J Comput Inf Syst Ind Manag Appl 9:181–205
Rao SS (1996) Engineering optimization. Wiley, New York
Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Comput Des 43:303–315
Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7:386–396
Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294
Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput J 13:2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
Safarzadeh S, Shadrokh S, Salehian A (2018) A heuristic scheduling method for the pipe-spool fabrication process. J Ambient Intell Human Comput 9(6):1901–1918
Salomon R (1998) Evolutionary algorithms and gradient search: similarities and differences. IEEE Trans Evol Comput 2(2):45–55
Sandgren (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223–229
Sharma TK, Pant M (2011) Enhancing the food locations in an artificial bee colony algorithm. In: IEEE swarm intelligence symposium (SIS), pp 119–123
Sharma TK, Pant M (2013) Enhancing the food locations in an artificial bee colony algorithm. Soft Comput 17(10):1939–1965
Sharma TK, Pant M (2017) Shuffled artificial bee colony algorithm. Soft Comput 21(20):6085–6610
Sundar S, Suganthan PN, Jin CT, Xiang CT, Soon CC (2017) A hybrid artificial bee colony algorithm for the job-shop scheduling problem with no-wait constraint. Soft Comput 21(5):1193–1202
Tang L, Zhao Y, Liu J (2014) An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-Continuous Casting Production. IEEE Trans Evol Comput 18(2):209–225
Tsai HC (2014) Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. Inf Sci 258:80–93
Wang H, Jiao-Hong Y (2018) An improved optimization method based on krill herd and artificial bee colony with information exchange. Memet Comp 10(2):177–198
Wang L, Li LP (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41:947–963
Wang L, Chen K, Ong YS (2005) Unified particle swarm optimization for solving constrained engineering optimization problems. In: Advances in natural computation, lecture notes in computer science volume, 3612, pp 582–591
Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37:395–413
Wang L, Guo C, Li Y, Du B, Guo S (2019) An outsourcing service selection method using ANN and SFLA algorithms for cement equipment manufacturing enterprises in cloud manufacturing. J Ambient Intell Human Comput 10(3):1065–1079
Yang XS (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley, Hoboken, 2010
Yi J, Li X, Chu CH, Gao L (2016) Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization. J Intell Manuf. https://doi.org/10.1007/s10845-016-1255-5
Yuan Q, Qian F (2010) A hybrid genetic algorithm for twice continuously differentiable NLP problems. Comput Chem Eng 34:36–41
Zahara E, Kao YT (2009) Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 36:3880–3886
Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci (NY) 178:3043–3074