Soft Computing (2021) 25:5455–5481
https://doi.org/10.1007/s00500-020-05545-9

METHODOLOGIES AND APPLICATION

Solving engineering optimization problems using an improved real-coded genetic algorithm (IRGA) with directional mutation and crossover

Amit Kumar Das[1] · Dilip Kumar Pratihar[1]

Published online: 9 January 2021

Abstract
Genetic algorithm (GA) is used to solve a variety of optimization problems. Mutation operator also is responsible in GA for maintaining a desired level of diversity in the population. Here, a directional mutation operator is proposed for real-coded genetic algorithm (RGA) along with a directional crossover (DX) operator to improve its performance. These evolutionary operators use directional information to guide the search process in the most promising area of the variable space. The performance of an RGA with the proposed mutation operator and directional crossover (DX) is tested on six benchmark optimization problems of different complexities, and the results are compared to that of the RGAs with five other mutation schemes. The proposed IRGA is found to outperform other RGAs in terms of accuracy in the solutions, convergence rate, and computational time, which is established firmly through statistical analysis. Furthermore, the performance of the proposed IRGA is compared to that of a few recently proposed optimization algorithms. The proposed IRGA is seen to yield the superior results compared to that of the said techniques. It is also applied to solve five constrained engineering optimization problems, where again, it has proved its supremacy. The proposed mutation scheme using directional information leads to an efficient search, and consequently, a superior performance is obtained.

Keywords Real-coded genetic algorithm · Directional mutation · Directional crossover · Computational time

List of symbols

N	Population size
p_c	Crossover probability
d	Dimensions of an optimization problem
α	Multiplying factor
fe	Number of function evaluation
t_{avg}	Average CPU time
Q_{min}	Minimum frequency
D	Mean diameter
τ	Shear stress
P_c	Buckling load on the beam
T_s	Shell thickness
r_i	Inner radius
b	Face width
z	Number of teeth on the pinion
d_1, d_2	Diameters of the first and second shafts, respectively
r_m	Mean radius of particles
MRR	Material removal rate
ρ_w	Density of work material
H_{dw}	Dynamic hardness of ductile work material
$(R_a)_{max}$	Permissible surface roughness
p_m	Mutation probability
p_{cv}	Variablewise crossover probability
p_{best}	Current best solution
p_{mean}	Average of two parents p_1 and p_2
max_gen	Maximum number of generations
fe_{avg}	Average number of function evaluations
d_w	Wire diameter
n	Number of active coils
θ	Bending stress
δ	End deflection of a bar
T_h	Thickness of head
l	Length of the cylindrical vessel
m	Module of teeth

✉ Dilip Kumar Pratihar
dkpra@mech.iitkgp.ac.in

Amit Kumar Das
amit.das@iitkgp.ac.in

[1] Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, India

L_1, L_2	Lengths of the first and second shafts between bearings, respectively
$\dot{M}_a$	Mass flow rate of abrasives
v_a	Velocity of abrasives
ρ_a	Density of abrasives
δ_{cw}	Critical plastic strain of ductile work material
ζ	Amount of plastically deformed indentation volume
α'	Significance level in statistical tests

1 Introduction

During the last few decades, various optimization techniques had been used to solve a variety of problems belonging to the domains of engineering, science, and others (Cao et al. 2018; Koh et al. 2006; Schutte et al. 2005; Sobieszczanski-Sobieski and Haftka 1997; Venter and Haftka 2009). In general, real-world optimization problems are found to be nonlinear in nature, and an efficient, robust, and reliable optimization technique is needed to tackle these types of problems. Traditional deterministic optimization tools are applicable to solve comparatively well-conditioned and smooth objective functions. In addition, these techniques do not use random numbers during their iterative search and the optimal solution depends on the chosen initial solution. On the other hand, the stochastic approaches use random numbers, due to which different optimum solutions may be obtained in varying runs with the same initial population. In addition, these techniques are basically population-based approaches and there is no need to determine the gradient of the objective functions. Therefore, these techniques may be implemented to solve all kinds of optimization problems and these are generally found to be robust. The major differences between these deterministic and stochastic methods are illustrated in Fig. 1.

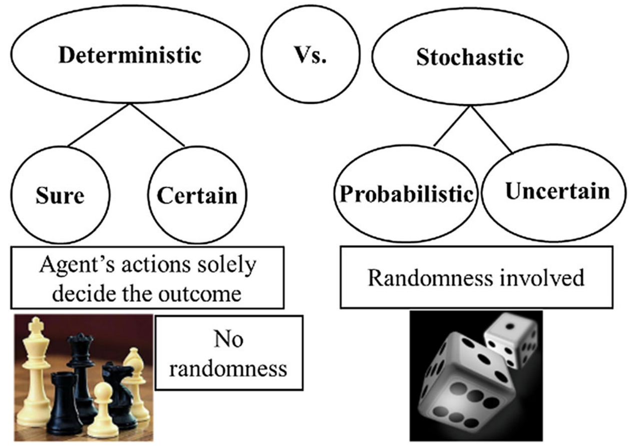


Fig. 1 Deterministic vs. stochastic methods

Due to several advantages of the stochastic methods compared to the deterministic ones, both the theoretical and application-related studies of these stochastic approaches have received more and more attention of the researchers. In the literature, the theoretical contributions related to these techniques have been reported in three major directions. In the first case, researchers had tried to further improve the performances of the existing algorithms by either modifying or proposing various types of stochastic operators for them (Kogiso et al. 1994a, b; Le Riche and Haftka 1995; Nagendra et al. 1996; Soremekun et al. 2001; Todoroki and Haftka 1998). On the other hand, in the second case, several optimization techniques had been fused together to obtain the merits of each of these algorithms and simultaneously overcome their inherent limitations (Barroso et al. 2017; Kazemzadeh Azad 2017; Raj and Setia 2017). However, a number of new optimization techniques, such as social spider optimization (SSO) (Cuevas and Cienfuegos 2014), search group algorithm (SGA) (Gonçalves et al. 2015), sine–cosine algorithm (SCA) (Mirjalili 2016), cricket algorithm (Canayaz and Karci 2016), water evaporation optimization (WEO) (Kaveh and Bakhshpoori 2016), and grasshopper optimization algorithm (GOA) (Saremi et al. 2017), had been proposed as the third research direction.

Genetic algorithm (GA) (Holland 1992) is found to be an efficient and robust meta-heuristic used for optimization. It mimics the basic theory of natural selection, i.e., survival of the fittest. Due to its several advantages, GA had been found to have a number of optimization-related applications in different spheres of engineering and science (Chen et al. 2017; Corbera Caraballo et al. 2017; Furuya and Haftka 1995; Jafari et al. 2018; Nagendra et al. 1993; Wang et al. 2018). Among various available coding schemes, such as binary-code (Goldberg 1989), grey code (Press et al. 2007), integer genes (MacKay and Mac Kay 2003), and others, the real-coded genetic algorithm (RGA) (Das and Pratihar 2017, 2018) is found to be more popular due to its inherent capability to be applied for the optimization problems involving continuous parameters. An RGA is basically a population-based optimization tool. This works in a cycle consisting of three main evolutionary operators, namely selection, crossover, and mutation. There had been a lot of efforts to propose new crossover operators, so that the performance of an RGA could be improved. However, the development of efficient mutation schemes for the RGA could not reach enough attention of the researchers. Both crossover and mutation operators have immense roles in the performance of a GA. Keeping this in view, an improved real-coded genetic algorithm (IRGA) has been proposed in this study to enhance the performance of a GA. This IRGA is equipped with a newly proposed directional mutation (DM) and a recently proposed directional crossover (DX) (Das and Pratihar 2019) operators.

2 Literature review

In an RGA, the mutation operator is used to bring sudden variations in the obtained solutions after the implementation of crossover operator. It incorporates a certain amount of diversity in the population, which helps to overcome the issue of local optima trapping of an RGA. Mutation operator primarily helps an RGA to increase its exploration or diversification power during an evolution. This capability of an RGA assists the search process to spread over widely in the design variables' space. However, a mutation-assisted search is found to be random in nature and the strength of mutation is dependent on the parameters used in the scheme. Several researchers had developed various mutation schemes, which operate based on different mechanisms. The ideas of uniform and non-uniform mutations were proposed by Michalewicz (1996). In the uniform mutation, a mutated solution is found to have a random number in the range of upper and lower boundary limits, whereas a step size (which reduces over the iterations) is used in case of a non-uniform mutation to control the diversification of the search process. Therefore, for the non-uniform mutation, a comparatively more diversified search is performed by an RGA at the initial stage of an evolution and it is seen to be reduced at the later stages. The boundary mutation (Michalewicz and Schoenauer 1996) is a special case of a uniform mutation, where a design variable is assigned a value equal to either the upper or lower boundary limit after the mutation. In Gaussian mutation (Schwefel 1987), two parameters such as mean and standard deviation of the distribution are used. Here, the mutation strength of the scheme depends on the standard deviation value. Later on, a self-adaptive Gaussian mutation (Hinterding 1995) was suggested, where the step size of the proposed scheme was found to be self-configuring. Apart from this, a modified GA (Hinterding et al. 1996) was developed, where the population size and the mutation strength were self-adjusted.

In case of a point directed (PoD) mutation (Berry and Vamplew 2004), a coevolving bunch of simple directions had been implemented for each chromosome. The concept of mutation-with-momentum for an RGA was introduced by Temby et al. (2005), and the performance of the same was compared to that of a Gaussian mutation on a set of six optimization functions. During the comparison, the said operator outperformed the other one. Moreover, a hybridization of these two mutation schemes was done and the performance of the hybrid operator was seen to be better compared to that of both the individual schemes. Motivated by the principal component analysis (PCA) strategy, a new mutation operator, namely PCA mutation (Munteanu and Lazarescu 1999), was introduced. A GA with the said operator was applied to solve a design optimization problem of IIR filters, and the obtained results were compared and found to be better compared to that of the other classical mutation schemes. The concept of a polynomial mutation was proposed by Deb and Goyal (1996). It had been used to solve a number of optimization problems successfully (Deb 2001). This is a very popular and commonly applied mutation scheme. Furthermore, a new mutation scheme was developed by Mäkinen et al. (1999). It was applied to solve a problem related to shape optimization. This operator was given a name as Makinen, Periaux, and Toivanen mutation (MPTM) (Deep and Thakur 2007). Inspired by the wavelet theory, another mutation operator, namely wavelet mutation, was developed by Ling and Leung (2007). This strategy used the Morlet wavelet, as its base. An RGA with the average bound crossover and wavelet mutation was implemented to solve a few classical benchmark functions, and the performance of this RGA was seen to be better compared to that of the other RGAs. The concept of power mutation (PM) was proposed by Deep and Thakur (2007), where the mutation strength was controlled by a parameter, namely index of mutation (p). The performance of the said operator was checked on a set of twenty classical optimization functions, and the results were compared to that of the other two mutation schemes, such as MPTM and non-uniform mutation.

In the process of optimization, search direction is an important factor. It is quite possible that an optimization method will able to reach globally optimum solution easily, if the search direction information is properly fed to the algorithm. Directional information increases the success rate of an optimization algorithm probabilistically. However, most of the stochastic operators do not use directional information in their search mechanisms. Therefore, in this study, an effort is made to improve the performance of an RGA with a newly proposed mutation operator, namely directional mutation (DM) and a recently proposed directional crossover (DX) (Das and Pratihar 2019). Directional information is collected in each generation, and then, it is fed to both DX and DM operators. The obtained directional information helps the operators to find the most promising search region and create new solutions in that area with the higher probability. The performance of the proposed IRGA has been tested on a set of six classical optimization problems of varying complexities, and the obtained results are compared to that yielded by RGAs with five other existing mutation schemes. Moreover, its performance has been compared to that of other six recently developed optimization algorithms. Finally, the proposed algorithm has been applied to solve five constrained engineering optimization problems, where the obtained best solutions are compared to that available in the literature.

The rest of the paper has been arranged as follows: Sect. 3 describes the principles of the proposed IRGA. Results and discussion are provided in Sect. 4, whereas Sect. 5 deals with the applications of the proposed IRGA for solving some constrained and real-world optimization problems. At the end, some concluding remarks are made in Sect. 6.

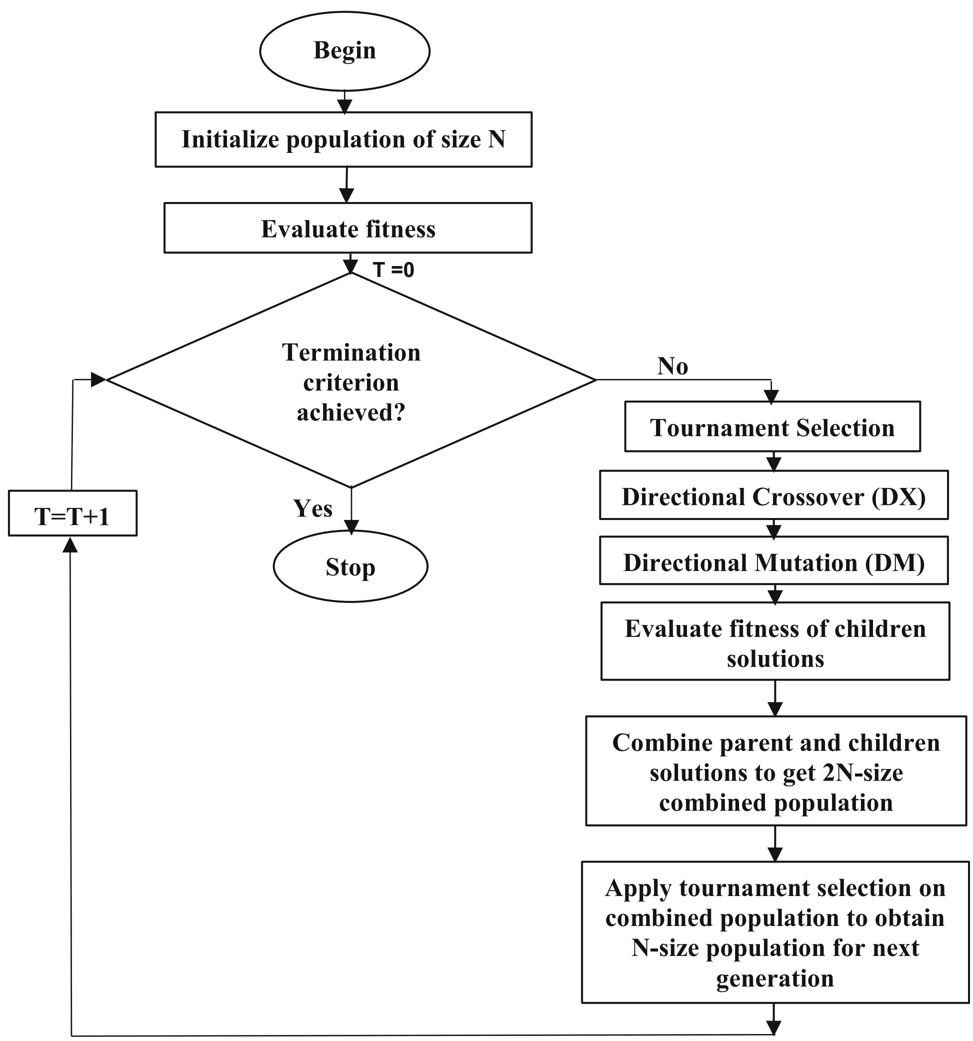
3 Proposed IRGA

The proposed IRGA consists of mainly three operators, such as tournament selection, directional crossover (DX), and directional mutation (DM) (refer to Fig. 2). Here, we discuss the operating mechanism of the proposed directional mutation (DM) and recently developed DX operator for a real-coded genetic algorithm (RGA). Both the operators are influenced by the directional information of the given optimization problem. The said information directs the search process to the most potential areas in the variable space, where the chance of getting the better solutions is high. The details of this proposed mutation scheme are explained in Sect. 3.1.

3.1 Detailed descriptions of DM

Let us assume that the population size and total number of variables of an optimization problem are found to be equal to N and d, respectively. The mutation operation is

Fig. 2 Flowchart of the proposed IRGA



performed variablewise. Let us consider a parent and its mutated solutions, which are denoted by y_i^j and y_m, respectively, where i and j vary from 1 to N and 1 to d, respectively. Now, the parent solution (y_i^j) is allowed to participate in the mutation operation, if a random number (say, r_1), created in the range of (0,1), is found to be either equal or less than the mutation probability (p_m). If the parent solution is permitted to participate in mutation, then the mutated solution is created under the guidance of the directional information. Otherwise, there will be no change in the value of the parent solution.

As the proposed DM operator uses the directional information, we need to collect the information before a mutated solution is created. This is done by comparing the parent solution to that of the variable value present in the current best solution obtained so far (say, p_{best}^j). When the value of p_{best}^j is found to be either greater or equal to the value of y_i^j, the mutated solution (y_m) is created using Eq. (3), as follows:

$$\beta_1 = e^{\left(2r - \frac{2}{r}\right)}, \tag{1}$$

$$\beta_2 = e^{\left(r - \frac{2}{r}\right)}, \tag{2}$$

$$y_m = \begin{cases} y_i^j + \beta_1 \times \left(y_u^j - y_i^j\right), & \text{if } r_2 \le p_d \\ y_i^j - \beta_2 \times \left(y_i^j - y_l^j\right), & \text{otherwise} \end{cases} \tag{3}$$

where β_1 and β_2 are the two intermediate parameters and they are used to determine the y_m. Moreover, r and r_2 are two different random numbers created in the range of (0,1) and $r \neq 0$. y_u^j and y_l^j are the upper and lower limits of the jth variable, respectively. p_d represents the directional probability, and in general, its value is set in between 0.5 and 1. In another case, where p_{best}^j is seen to be less than y_i^j, the mutated solution (y_m) is generated using Eq. (4), as follows:

$$y_m = \begin{cases} y_i^j - \beta_1 \times \left(y_i^j - y_l^j\right), & \text{if } r_2 \le p_d \\ y_i^j + \beta_2 \times \left(y_u^j - y_i^j\right), & \text{otherwise} \end{cases} \tag{4}$$

From Eqs. (3) and (4), it is easy to understand that the mutated solutions are going to be created always within the boundary limits of the variables, and therefore, there is no need to apply any boundary constraint handling technique, in this case. A pseudo-code of the proposed DM has been provided in Fig. 3.

Input: Parent solution (y_i^j), DM parameters (p_m and p_d)

```
if r1 ≤ pm
    β1 ← using Eq.(1)
    β2 ← using Eq.(2)
    if p_best^j ≥ y_i^j
        ym ← using Eq.(3)
    else
        ym ← using Eq.(4)
    endif
else
    ym = y_i^j
endif
```

Fig. 3 Pseudo-code of the proposed DM operator

3.2 Settings and influence of the parameters of DM

The proposed DM operator has two user-defined parameters, such as mutation probability (p_m) and directional probability (p_d). Theoretically, both the parameters can vary in the range of (0, 1). However, normally, the value of p_m is set to a low value (<0.1) to ensure an efficient search by the RGA. On the contrary, the parameter p_d is recommended to assign a value either greater than or equal to 0.5.

In the proposed mutation scheme, it is assumed that if the mutated solution is created in the direction of the current best solution with respect to the parent solution, a better solution can be found out with the higher probability (>0.5) and this probability is termed as the directional probability (p_d). Moreover, this parameter is found to have a high value (close to 1.0), whenever a unimodal and comparatively easy optimization problem, where relatively less diversity in the population is required, is solved. However, for the multimodal and complex problem, it is recommended to assign a low value (close to 0.5) for the same, so that more diversity can be incorporated in the population.

In Fig. 4, the positional distributions of the mutated solutions are shown, when (a) $y_{\text{best}}^j \ge y_i^j$, and (b) $y_{\text{best}}^j < y_i^j$, where y_{best}^j and y_i^j are the jth variables of the current best and ith solutions, respectively. In this example, the value of y_i^j is considered to be equal to 0.0, and the upper and lower boundary limits for the jth variable are taken to be equal to 10 and -10, respectively. From the figure, one important observation is to be noted here that the mutated solutions are seen to be very close to the parent solution, when the random number r varies approximately in the range of 0.0 to 0.4. However, for the higher value of r, the mutated solutions is found to be more distant from the parent. From this observation, it can be stated that nearly for the 40% of the total range of the random number r (i.e., approximately 0.0 to 0.4), the proposed DM is found to create new solutions near to the parents (i.e., less diversity is incorporated in the population). On the other hand, for the remaining part of the range of r (nearly 0.4 to 1.0), the

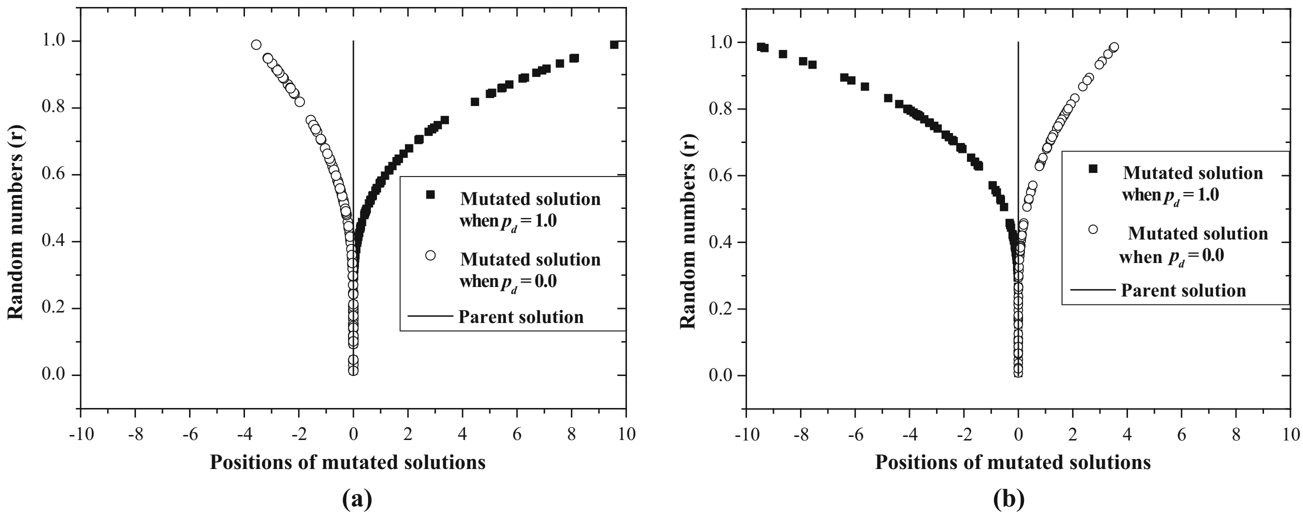


Fig. 4 Positional distributions of the mutated solutions with the random number (r), when: (**a**) $y^j_{\text{best}} \geq y^j_i$, and (**b**) $y^j_{\text{best}} < y^j_i$

search process becomes a global one, as children solutions are generated to be far from the parent positions.

Another finding to be stated here is that for a value of r equals to 1.0, the distance between the parent and mutated solutions are found to be more (when p_d is seen to be equal to 1.0) compared to that, when p_d is found to have a value equal to 0.0. This has been done purposefully in the proposed DM. The basic assumption of this proposed DM is that the obtained directional information is correct with a high probability value of p_d. When this information is assumed to be correct, i.e., a random number r_2 is found to be either less or equal to p_d, a mutated solution can be generated anywhere in between the parent solution and the variable boundary limit (either lower or upper depending upon the direction). However, if the directional information is considered to be incorrect, i.e., r_2 is seen to be greater than p_d, the maximum distance between the parent and mutated solution is restricted to a certain amount.

In summary, some important aspects of the proposed DM operator can be stated to show how it does theoretically support an RGA to search for the globally optimum solution as follows:

- The proposed DM operator is guided by the directional information of the problem, due to which the search process is directed towards the most promising areas in the variable space.
- The proposed mutation operator has been developed in such a way that it can either concentrate or diversify the search depending upon the value of the random number (r). This is an advantage of the proposed mutation scheme.
- The parameter, directional probability (p_d), has been used to keep a nice balance between the population diversity and selection pressure.
- The mutation strength of the proposed operator is found to be dependent on the random number (r).
- Finally, it can be stated that the proposed DM operator can be made responsible for both intensification (approximately, the probability is near to 0.4) and diversification of the searches (approximately, the probability is close to 0.6) depending upon the situations.

3.3 A brief description of DX

Similar to the proposed DM operator, the directional crossover (DX) (Das and Pratihar 2019) operator has been seen to have an influence of the directional information of an optimization problem. Moreover, the way to get this information is also quite similar to that of the proposed DM scheme. DX has mainly four parameters, namely crossover probability (p_c), variablewise crossover probability (p_{cv}), directional probability (p_d), and multiplying factor (α). This DX operator can be described briefly as follows:

Let us assume that two mating parents (say, p^j_1 and p^j_2, where j varies from 1 to d) are allowed to participate in crossover and they are not equal. Moreover, p^j_{mean} and p^j_{best} are the mean of the two parents and jth variable of the current best solution, respectively. Now, if p^j_{best} is seen to be either greater than or equal to p^j_{mean}, the two children solutions, represented as c_1 and c_2, respectively, are created as follows:

$$\mathrm{val} = 1 - (0.5)^{e^{\left[\frac{|p_1^j - p_2^j|}{(y_u^j - y_l^j)}\right]}}, \tag{5}$$

$$\beta = \frac{r_3}{\alpha^2}, \tag{6}$$

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_1^j + p_2^j\right) + \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 \le p_d \end{aligned} \tag{7}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_1^j + p_2^j\right) - \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 \le p_d \end{aligned} \tag{8}$$

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_1^j + p_2^j\right) - \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 > p_d \end{aligned} \tag{9}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_1^j + p_2^j\right) + \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 > p_d \end{aligned} \tag{10}$$

where r_3 and r_4 are the two different random numbers created in the range of (0,1). *val* and β are the two intermediate parameters. y_u^j and y_l^j are the upper and lower limits of the jth variable, respectively. α has been termed as a multiplying factor. In another situation, where p_{best}^j is observed to be less than p_{mean}^j, the children are yielded as follows:

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_1^j + p_2^j\right) - \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 \le p_d \end{aligned} \tag{11}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_1^j + p_2^j\right) + \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 \le p_d \end{aligned} \tag{12}$$

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_1^j + p_2^j\right) + \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 > p_d \end{aligned} \tag{13}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_1^j + p_2^j\right) - \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left|p_1^j - p_2^j\right| \\ & \text{if } r_4 > p_d \end{aligned} \tag{14}$$

Now, if the mating parents are found to have the equal values and $p_{\mathrm{best}}^j \ne p_{\mathrm{mean}}^j$, then the children solutions are produced as follows:

$$\mathrm{val} = 1 - (0.5)^{e^{\left[\frac{|p_{\mathrm{best}}^j - p_{\mathrm{mean}}^j|}{\left(y_u^j - y_l^j\right)}\right]}}, \tag{15}$$

$$\beta = \frac{r_3}{\alpha^2}, \tag{16}$$

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_{\mathrm{best}}^j + p_{\mathrm{mean}}^j\right) + \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left(p_{\mathrm{best}}^j - p_{\mathrm{mean}}^j\right) \\ & \text{if } r_4 \le p_d \end{aligned} \tag{17}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_{\mathrm{best}}^j + p_{\mathrm{mean}}^j\right) - \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left(p_{\mathrm{best}}^j - p_{\mathrm{mean}}^j\right) \\ & \text{if } r_4 \le p_d \end{aligned} \tag{18}$$

$$\begin{aligned} c_1 = {} & \mathrm{val} \times \left(p_{\mathrm{best}}^j + p_{\mathrm{mean}}^j\right) - \alpha^{r_3} \times e^{(1-\beta)} \times (1 - \mathrm{val}) \\ & \times \left(p_{\mathrm{best}}^j - p_{\mathrm{mean}}^j\right) \\ & \text{if } r_4 > p_d \end{aligned} \tag{19}$$

$$\begin{aligned} c_2 = {} & (1 - \mathrm{val}) \times \left(p_{\mathrm{best}}^j + p_{\mathrm{mean}}^j\right) + \alpha^{(1-r_3)} \times e^{(-\beta)} \times \mathrm{val} \\ & \times \left(p_{\mathrm{best}}^j - p_{\mathrm{mean}}^j\right) \\ & \text{if } r_4 > p_d \end{aligned} \tag{20}$$

where r_3 and r_4 are two separate random numbers generated in the range of (0,1). In other cases, the children solutions will be the same as the parents. The newly created children solutions will be constrained by the variable boundary limits. This is nothing but applying the variable boundary constraints.

Moreover, c_1 is recognized as either first child or second child with a probability value equal to 0.5. If c_1 is chosen as the first child, c_2 will be recognized as the second child and vice versa. This step is known as child recognition conditions. A pseudo-code of DX is given in Fig. 5. This is how, the DX operator works. For more details, interested readers may refer to Das and Pratihar (2019).

3.4 Settings and influence of parameters of IRGA

In the proposed IRGA, there are two common parameters, such as population size (N) and stopping criterion, and four algorithm-specific parameters, namely crossover probability (p_c), variablewise crossover probability (p_{cv}), multiplying factor (α), and mutation probability (p_m). Population size (N) is generally set equal to five to ten times of the number of variables (d). For a simpler objective function to solve, N may be taken as a small value. However, N is recommended to take a large value in order to solve a complex optimization problem. Either the maximum number of generations (max_gen) or maximum number of function evaluations can be selected as a stopping criterion. The stopping criterion may be set as follows: During the evolution, if the change in the best obtained fitness is found to be less than a very small value (say, 10^{-8}) for a certain

Fig. 5 Pseudo-code of the directional crossover (DX) of IRGA

```
Input: Two parents (p_1 and p_2) with d dimensions, p_c, p_cv, p_d, p_best, α
Output: offspring (Ch_1 and Ch_2)
----------------------------------------------------------------
if rand ≤ p_c
    for j = 1 to d
        if r_1 ≤ p_cv    (% r_1 is a random number created in the range (0, 1))
            if |p_1^j − p_2^j| > 0
                Determine p_mean^j
                Determine val and β ← Apply Eq.(5) and Eq.(6)
                    if (p_best^j ≥ p_mean^j)
                        if r_4 ≤ p_d
                        c_1 and c_2 ← Apply Eq.(7) and Eq.(8)
                        else
                        c_1 and c_2 ← Apply Eq.(9) and Eq.(10)
                    endif
                    else
                        if r_4 ≤ p_d
                        c_1 and c_2 ← Apply Eq.(11) and Eq.(12)
                        else
                        c_1 and c_2 ← Apply Eq.(13) and Eq.(14)
                        endif
                    endif
                Use boundary constraint
                Use child recognition conditions
            else
                p_mean^j = p_1^j
                if (p_best^j ≠ p_mean^j)
                    val and β ← Apply Eq.(15) and Eq.(16)
                    if r_4 ≤ p_d
                        c_1 and c_2 ← Apply Eq.(17) and Eq.(18)
                    else
                        c_1 and c_2 ← Apply Eq.(19) and Eq.(20)
                    endif
                    Use boundary constraint
                    Use child recognition conditions
                else
                        Ch_1^j = p_1^j
                        Ch_2^j = p_2^j
                endif
            endif
        else
            Ch_1^j = p_1^j
            Ch_2^j = p_2^j
        endif
    end of for loop
else
    Ch_1^j = p_1^j
    Ch_2^j = p_2^j
endif
```

number of generations, the algorithm may be stopped there and the number of generations at that condition may be set as max_gen. The other parameters like p_c and p_m can be set in a similar way, as in the case of a standard genetic algorithm. Mutation operator, as it increases diversity in the population, helps the algorithm to avoid premature convergence of the search process. p_{cv} may vary in the range of (0, 1). However, similar to p_c, it is generally set to a higher value (near to 1) for the better performance of the algorithm. For α, it is suggested to assign a value less than or equal to one, if a simple optimization problem is considered. In this case, the children solutions will be created near their parents. Therefore, the search process will focus more on the promising regions. However, for solving a complex optimization problem, α is generally set to a value greater than 1 to inject more diversity in the population (Das and Pratihar 2019). It is to be noted here that directional probability (p_d) is not a user-defined parameter of IRGA. It is set in the algorithm, as discussed in Sect. 4.1.1.

3.5 Limitations of IRGA

It is to be noted here that the proposed algorithm is a real-coded genetic algorithm. Therefore, it is unable to handle variables in binary form. Moreover, the present form of IRGA is suitable for solving optimization problem with continuous variables only. For discrete and integer variable programming, IRGA is not suitable. Nevertheless, the proposed IRGA is found to be not applicable to solving NP hard optimization problems and that involving uncertain parameters.

4 Results and discussion

To measure the performance of the proposed IRGA, several experiments have been carried out and the obtained results are compared to that of other efficient metaheuristics. These experiments are described below, in detail.

4.1 The first experiment

The RGA, where the proposed DM is used, has other evolutionary operators, like tournament selection with tournament size of two, a directional crossover (DX) (Das and Pratihar 2019), and a replacement operation suggested by (Deb 2000). This RGA is named as IRGA in this paper. Similarly, we have other five RGAs with the same evolutionary operators, except the mutation one. Five popular mutation schemes, such as polynomial mutation (PLM) (Deb and Goyal 1996), power mutation (PWM) (Deep and Thakur 2007), MTP mutation (Mäkinen et al. 1999), Gaussian mutation (GM) (Schwefel 1987), and uniform mutation (UM) (Michalewicz 1996), are used in these RGAs, and these are termed as RGA-PLM, RGA-PWM, RGA-MTP, RGA-GM, and RGA-UM, respectively.

In the first experiment, using the above-mentioned RGAs, a set of six benchmark optimization functions with different attributes has been solved and the results are compared. The mathematical formulations and the variable limits are given in Table 1, where d represents dimensionality of a problem. In this experiment, we have considered four different levels of dimensions (d), such as 30, 60, 90, and 120 for each test function. Moreover, each test function with a particular dimensionality level has been solved for 50 times and the average of the obtained 50 best finesses has been calculated. Among the test functions, the first three are of unimodal type, whereas the rests are multimodal in nature. In a unimodal problem, there is only one globally optimum point with no other locally optimum basin (refer to Fig. 6a). However, several locally optimum points along with that of the globally one can coexist in the objective function space of a multimodal problem (see Fig. 6b).

4.1.1 Parameters' settings

The set of parameters for the RGAs is selected after several trial experiments. It is to be noted that for a fair comparison, the common parameters for all the RGAs are assigned the same values and these are as follows: maximum number of generations, max_gen $= 500$, population size, $N = 5 \times d$, crossover probability, $p_c = 0.9$, mutation probability, $p_m = \frac{1}{d}$, variablewise crossover probability, $p_{cv} = 0.9$, and multiplying factor $\alpha = 0.95$. Moreover, the directional probability (p_d) used in the DX operator has been set as follows: Whenever an improvement is found in the current best fitness value compared to that of the previous generation, p_d is assigned a value equal to 0.75. Otherwise, it is seen to have a value equal to 0.5. Here, one thing is to be remembered that the values of the parameters, like N and p_m, are seen to be dependent on the dimensionality (d) of an optimization problem of interest. For example, if d is considered to be equal to 30, the values of N and p_m are seen to be equal to 150 and (1/30), respectively, whereas these are found to have values equal to 300 and (1/60), respectively, if d is set to be equal to 60.

Apart from these, the mutation scheme-specific parameters for the six RGAs used in the experiment are as follows:

- IRGA: Directional probability (p_d) has been assigned a value in the same way as in case of DX operator.
- RGA-PLM: Distribution index of the polynomial mutation is set to be equal to 20.
- RGA-PWM: Index of mutation, $P = 0.5$.
- RGA-MTP: Index of mutation, $b = 4$.
- RGA-GM: The values of both the parameters, namely scale and shrink, are set to be equal to 0.75.
- RGA-UM: There is no such parameter.

Table 1 Mathematical formulations and variable bound of the six test functions

Name of the function	Mathematical expression	Variable bounds
F01: Sphere	$f(x) = \sum_{i=1}^{d} x_i^2$	$[-100, 100]^d$
F02: Sum of different powers	$f(x) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	$[-100, 100]^d$
F03: Bent ciger	$f(x) = x_1^2 + 10^6 \sum_{i=2}^{d} x_i^2$	$[-10, 10]^d$
F04: Rastrigin	$f(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12, 5.12]^d$
F05: Alpine	$f(x) = \sum_{i=1}^{d} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$[-10, 10]^d$
F06: Schaffer F7	$f(x) = \frac{1}{d-1} \sum_{i=1}^{d-1} [(x_i^2 + x_{i-1}^2)^{0.25} + (x_i^2 + x_{i+1}^2)^{0.25} \sin^2\left(50\left(x_i^2 + x_{i+1}^2\right)^{0.1}\right)]$	$[-100, 100]^d$

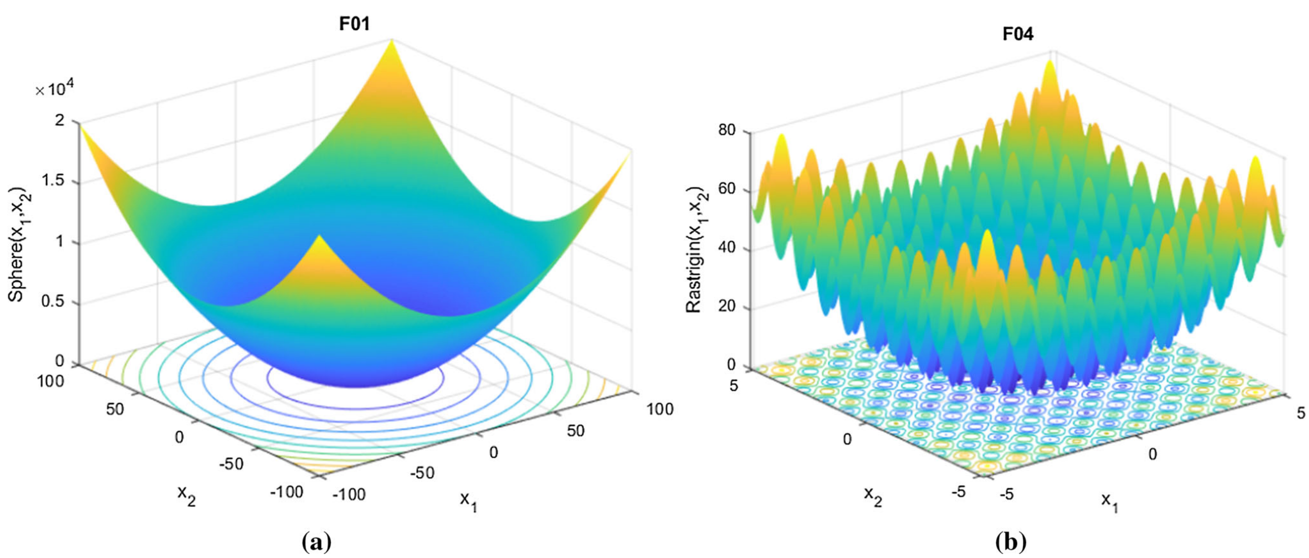


Fig. 6 Plots of the objective functions with two variables: **a** Sphere (F01), and **b** Rastrigin (F04)

The stopping criterion for the RGAs is set as the maximum number of generations to be run. The obtained results are given in Table 2, where the best results are marked in bold. From these results, it is clear that the RGA with the proposed DM operator (IRGA) has outperformed the other ones in most of the cases in terms of the accuracy in solution. Moreover, the evolutions for the average current best finesses of all the runs have been depicted in Fig. 7, for all the test functions. From the figure, it has been observed that IRGA is able to reach the better accuracy of the solution in less number of generations compared to the other RGAs. This fact ensures that the IRGA has the faster convergence rate compared to that of the other ones. Furthermore, the evolutions of the average of first variable's values in the population are shown in Fig. 8 for the IRGA. This is done to observe the exploitation and exploration behaviours of the algorithm during the evolutions.

4.1.2 Exploitation analysis

During the experiments, three unimodal functions, such as Sphere (F01), Sum of different powers (F02), and Bent ciger (F03), have been solved by the RGAs. It is known that these unimodal functions have only one globally optimum point with no other locally optimum basins. This basic characteristic of the objective function space of these

Table 2 Comparison of average results for the test functions (F01-F06) with 30, 60, 90, and 120 dimensions (d)

Function	IRGA	RGA-PLM	RGA-PWM	RGA-MTP	RGA-GM	RGA-UM
$d = 30$						
F01	**2.894E − 12**	6.495E − 10	8.240E − 09	6.131E − 09	7.859E − 09	4.681E − 09
F02	**8.517E − 19**	1.136E − 14	6.043E + 01	8.855E − 06	5.730E − 07	2.274E − 08
F03	**1.514E − 08**	2.711E − 06	5.163E − 05	3.403E − 05	3.628E − 05	2.085E − 05
F04	**6.403E − 02**	9.350E − 01	1.984E + 01	4.392E + 00	1.864E + 00	2.766E + 00
F05	**1.205E − 08**	2.531E − 07	1.138E − 06	8.833E − 07	9.135E − 07	7.321E − 07
F06	**3.118E − 03**	1.087E − 02	1.778E − 02	2.116E − 02	1.515E − 02	1.900E − 02
$d = 60$						
F01	**5.586E − 06**	1.850E − 04	2.364E − 03	1.897E − 03	1.831E − 03	1.231E − 03
F02	**2.725E + 01**	4.518E + 02	6.557E + 16	1.694E + 12	9.637E + 10	3.624E + 08
F03	**4.161E − 02**	1.297E + 00	1.608E + 01	1.313E + 01	1.117E + 01	9.559E + 00
F04	**1.879E + 01**	2.687E + 01	4.983E + 01	3.171E + 01	2.802E + 01	2.840E + 01
F05	**1.107E − 04**	8.801E − 04	4.077E − 03	3.460E − 03	3.082E − 03	2.865E − 03
F06	**3.543E − 02**	7.383E − 02	1.302E − 01	1.251E − 01	1.208E − 01	1.095E − 01
$d = 90$						
F01	**4.324E − 03**	4.715E − 02	5.631E − 01	4.367E − 01	4.196E − 01	3.243E − 01
F02	6.744E + 23	**1.964E + 23**	7.998E + 40	8.056E + 36	7.389E + 33	1.540E + 34
F03	**3.535E + 01**	3.801E + 02	4.379E + 03	3.394E + 03	3.313E + 03	2.330E + 03
F04	**4.560E + 01**	5.298E + 01	9.105E + 01	6.835E + 01	6.632E + 01	6.906E + 01
F05	**5.795E − 03**	2.490E − 02	6.700E − 02	6.097E − 02	5.827E − 02	5.591E − 02
F06	**1.440E − 01**	2.423E − 01	4.115E − 01	3.968E − 01	3.868E − 01	3.718E − 01
$d = 120$						
F01	**2.018E − 01**	9.269E − 01	1.057E + 01	9.027E + 00	7.684E + 00	6.187E + 00
F02	2.660E + 52	**2.093E + 50**	2.807E + 69	3.711E + 65	3.827E + 63	1.915E + 63
F03	**1.766E + 03**	8.990E + 03	9.181E + 04	7.808E + 04	7.166E + 04	5.651E + 04
F04	**8.019E + 01**	9.802E + 01	1.448E + 02	1.262E + 02	1.182E + 02	1.212E + 02
F05	**4.844E − 02**	9.270E − 02	1.740E − 01	1.655E − 01	1.655E − 01	1.528E − 01
F06	**3.238E − 01**	4.504E − 01	7.734E − 01	7.496E − 01	7.152E − 01	6.928E − 01

problems has made them suitable for checking the exploitation capability of an optimization algorithm (Mirjalili et al. 2014). The superior results yielded by using the IRGA (refer to Table 2) ensure that it has the better exploitation ability compared to the other RGAs have. The use of directional information helps the search process to concentrate on the most promising areas of the variable space, and consequently, the search process becomes more efficient. Moreover, as already discussed (refer to Sect. 3), the proposed DM operator has also a certain amount of contributions in promoting the local search by the IRGA, which, in other sense, increases the exploitation capability of the IRGA.

4.1.3 Exploration analysis

It is suggested that the multimodal problems are more suitable to measure the exploration capability of an optimization algorithm (Mirjalili et al. 2014). This type of problem is found to have several locally optimum basins in the objective function space, which makes it difficult for an optimization algorithm to find the globally optimal solution. In this experiment, three multimodal functions, like Rastrigin (F04), Alpine (F05), and Schaffer F7 (F06), have been considered. These functions are comparatively difficult to solve, and moreover, this difficulty level increases with the increase in number of variables of a problem. It is to be noted that in our experiment, four different dimensionality levels (such as$d = 30, 60, 90,$ and 120) have been considered for each of the optimization functions. Therefore, it can be assumed that the exploration power of an optimization algorithm can be well tested through this experiment. The superior results yielded by the IRGA (given in Table 2) prove that the proposed algorithm acquires the better exploration ability compared to that of the others. The uses of directional probability (p_d) and other mechanisms in the proposed DM operator help in promoting the exploration strength of the IRGA, and consequently, it has yielded the superior results compared to other RGAs.

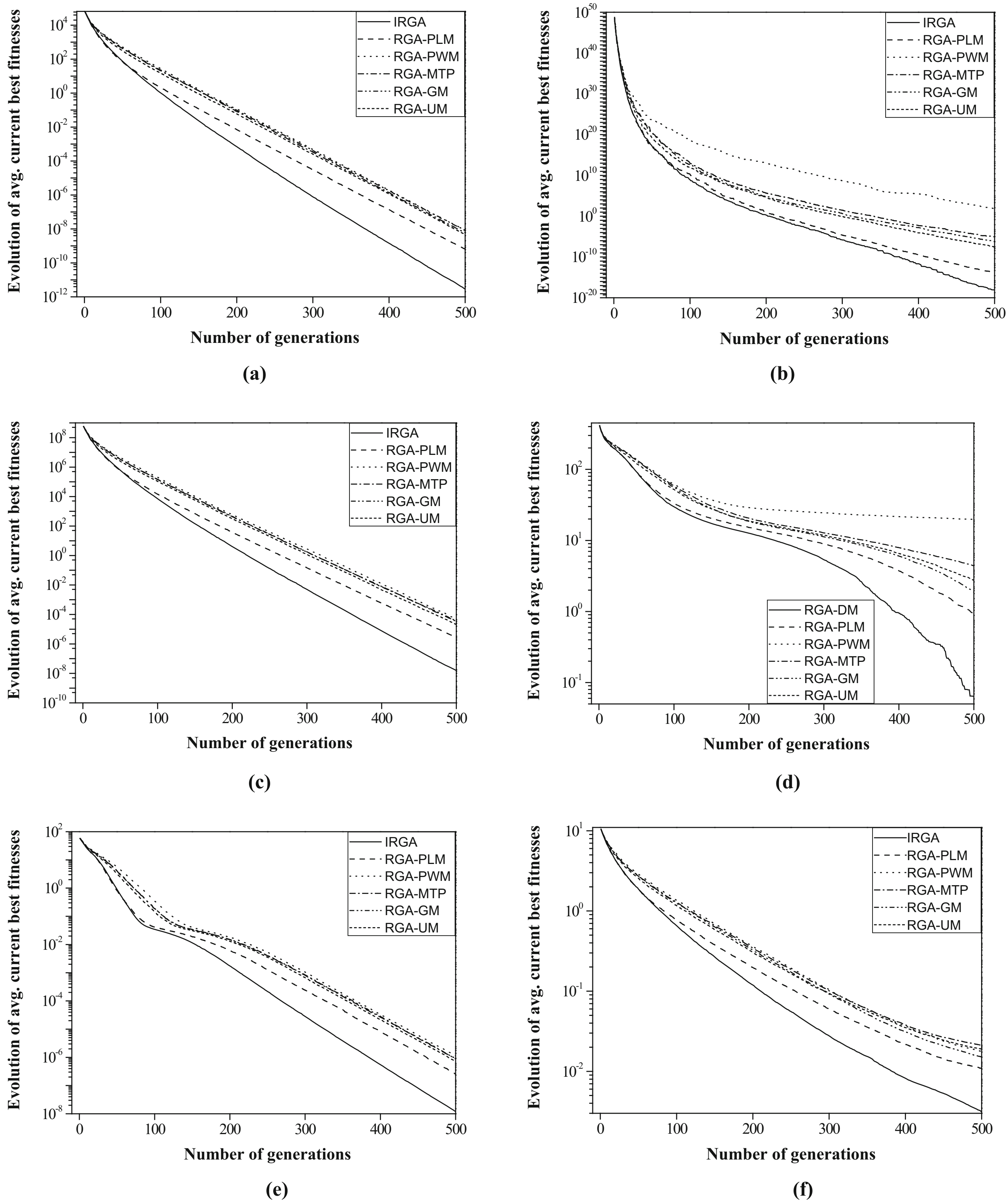


Fig. 7 Evolutions of average current best fitness values of all the runs using six RGAs for the functions: **a** F01, **b** F02, **c** F03, **d** F04, **e** F05, and **f** F06

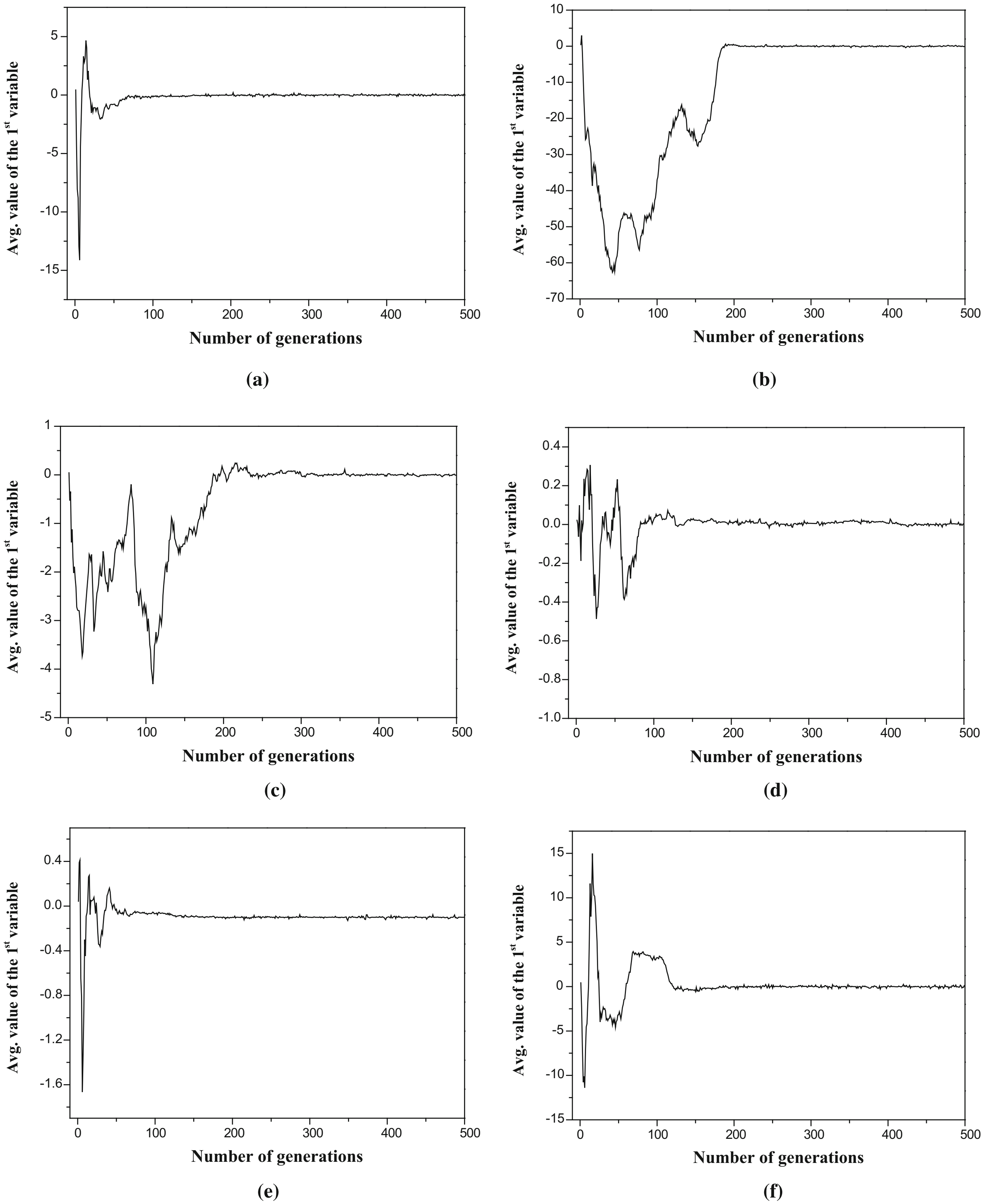


Fig. 8 Evolutions of average value of the first variable in the population for the functions: **a** F01, **b** F02, **c** F03, **d** F04, **e** F05, and **f** F06

4.1.4 Convergence analysis

It is debated (Van Den Bergh and Engelbrecht 2006) that abrupt and comparatively large changes in the values of the variables are found, during the initial phase of search, and it is seen to be gradually decreasing at the later stages. From this, it can be inferred that in general, the exploration capability of an optimization algorithm is observed to be dominating over the exploitation at the initial phase of an evolution and the converse is true during the later stages. Figure 8 presents the evolution of the average of first variable values in the population for all the test functions. It is clear from the figure that at the initial generations, average values of the first variable fluctuate abruptly, which denotes a phase, where the exploration factor of the IRGA is found to be dominating. However, these fluctuations are seen to be reduced over the generations and the algorithm is observed to converge to an optimal solution. This behaviour is in line with the convergence criterion of an optimization algorithm suggested by Van Den Bergh and Engelbrecht (2006). Therefore, it can be concluded that the IRGA is able to converge to an optimum solution after exploring the variable space.

4.1.5 The statistical analysis

It is sometimes becoming difficult to decipher the differences in performances among several algorithms by merely observing the obtained results, as given in Table 2. Moreover, a method of quantifying the performance differences of the algorithms is also to be developed by the researchers. Therefore, the obtained results of Table 2 have been analysed statistically to investigate the possibilities of finding the significant differences in performances among the RGAs. Mainly, two types of nonparametric tests, such as pairwise and multiple comparisons, have been carried out. Moreover, for pairwise comparisons, two popular tests, such as Sign and Wilcoxon's tests (Derrac et al. 2011), have been performed, whereas Friedman, aligned Friedman, and Quade tests along with four post hoc procedures (namely Holland, Rom, Finner, and Li) (Derrac et al. 2011; García et al. 2010) are considered in the multiple comparisons.

The results obtained applying pairwise comparisons between the IRGA and other RGAs are given in Table 3. The first row of the table represents the number of wins and losses of the IRGA compared to other RGAs. During the comparison between two algorithms, the one, which gives the better optimum result in terms of accuracy of the solution for a particular function, is called the winner. In the first experiment, there are a total of 24 instances. Therefore, to be an overall winner in a pairwise comparison, an algorithm has to win at least in 18 cases compared to the other ones with a significance level of $\alpha' = 0.05$ (Derrac et al. 2011). Considering the results provided in Table 2, the IRGA can be declared as the overall winner in all the pairwise comparisons. Moreover, the p-values yielded in the Sign test are seen to be less than 0.05. This signifies that there exists a considerable amount of differences in the performances between the IRGA and others. Using Wilcoxon's test also, similar observations have been obtained, which confirms supremacy in the performance of the proposed IRGA compared to other RGAs.

To make the multiple comparisons, a CONTROLTEST Java package (Derrac et al. 2011), obtained from the SCI2S website (http://sci2s.ugr.es/sicidm), has been used. According to the Friedman, aligned Friedman, and Quade tests, the relative ranks of the RGAs are provided in Table 4. A lower value of the relative rank signifies the better performance for an algorithm compared to others. It has been seen that the proposed IRGA occupies the lowest rank for all three tests, which establishes the supremacy of the IRGA compared to other RGAs. Moreover, the numerical values of the statistics and p-values of three tests are reported in the last two rows of Table 4. For the Friedman and aligned Friedman tests, the statistics are found to be distributed according to the Chi-squared distribution with 5 degrees of freedom, whereas that for the Quade test is seen to follow the F-distribution with 5 and 115 degrees of freedom. Nevertheless, the p-values obtained in the three statistical tests are found to be less than 0.05. These observations confirm that the performance of the proposed IRGA is significantly better than that of the other RGAs.

The results obtained from the contrast estimation are given in Table 5. This test has the capability to measure the difference in performance between two algorithms in

Table 3 Pairwise comparison results: IRGA vs. other RGAs

IRGA vs.	RGA-PLM	RGA-PWM	RGA-MTP	RGA-GM	RGA-UM
Wins/losses	22/2[a]	24/0[a]	24/0[a]	24/0[a]	24/0[a]
Sign test p-value	3.59E − 05	1.19E − 07	1.19E − 07	1.19E − 07	1.19E − 07
Wilcoxon p-value	3.25E − 03	1.82E − 05	1.82E − 05	1.82E − 05	1.82E − 05

[a]Level of significance = 0.05

Table 4 Friedman, aligned Friedman, and Quade ranks of the RGAs

Algorithm	Friedman	Aligned Friedman	Quade
IRGA	1.083	36.375	1.157
RGA-PLM	1.917	38.333	1.843
RGA-PWM	5.917	113.625	5.960
RGA-MTP	4.833	86.500	4.913
RGA-GM	3.917	81.333	3.730
RGA-UM	3.333	78.833	3.397
Statistic	110.857	18.887	81.805
p- value	9.67E − 11	2.02E − 03	2.22E − 16

numeric form. This method is fruitful in determining how much an optimization technique is performing better compared to the other (Derrac et al. 2011). In Table 5, each row represents the results of comparisons between the algorithm mentioned in the first column and the rest of the approaches. When the measured result is found to be positive, then it is assumed that the performance of the method present in the first column is comparatively better than that of the other, and in a similar way, the converse is also seen to be true, when the estimated result is found to be a negative one. Moreover, the higher the measured value, the higher will be the difference in performance between two optimization techniques. In Table 5, all the determined values corresponding to the proposed IRGA are observed to be positive. This proves supremacy in the performance of the IRGA compared to that of the others.

In addition, to see the difference in performance, a few parameters such as z-values, unadjusted p-values, and adjusted p-values using four post hoc procedures of the Friedman, aligned Friedman, and Quade tests are calculated, as given in Table 6. In this comparison, the proposed IRGA has been assigned as the control method. Using the respective z-values on a normal distribution N (0, 1) (Derrac et al. 2011), the unadjusted p-values are determined. From these p-values of the three tests, it can be claimed that the proposed IRGA has significantly better performance compared to that of the other RGAs, except the RGA-PLM. However, the unadjusted p-values are susceptible to be inaccurate, as some family error acquiring issues may occur. Therefore, some post hoc procedures, such as Holland, Rom, Finner, and Li, are recommended to obtain the adjusted p-values (Derrac et al. 2011) and the results obtained by these methods are reported in the said table. According to all three tests (Friedman, aligned Friedman, and Quade), the proposed IRGA is found to perform significantly better compared to the other RGAs, except the RGA-PLM.

To make a convergence performance analysis of the RGAs, a nonparametric statistical method, namely Page's trend test, has been utilized (Derrac et al. 2014). This test carries out the comparison of convergence performance pairwise considering the fact that an optimization technique having the better convergence rate should move faster towards the optimum solution compared to that with the slower convergence rate. This statistical approach is going to find the differences in the objective function values between two optimization algorithms at the various number of generations (cut points). In our experiment, a total of 10 cut points have been considered starting from the 50th to 500th generations with an increment of 50. Although there exist two versions of the test, such as basic and alternative, the latter one has been applied in our case, as it is found to be more efficient (Derrac et al. 2014). The obtained p-values through the convergence test are reported in Table 7 and the values, which are found to be less than 0.1, are marked in bold. Here, the null hypothesis is considered, as both the algorithms have the equal convergence rates. However, the alternative hypothesis says that the algorithm appears in the row is seen to possess the better convergence rate compared to that present in the column. The p-values for the IRGA are found to be less than 0.1. It means that the proposed IRGA has the best convergence rate among the RGAs with a significant level equal to 0.1. Moreover, it is also concluded that in terms of convergence rate, the IRGA is found to be the fastest followed by the algorithms, such as RGA-PLM, RGA-UM, RGA-MTP, and RGA-GM. In addition, the RGA-PWM is seen to have the worst convergence performance among the six RGAs.

From this statistical analysis, it is obvious that the proposed IRGA has performed the best among all the

Table 5 Contrast estimation results of the first experiment

	IRGA	RGA-PLM	RGA-PWM	RGA-MTP	RGA-GM	RGA-UM
IRGA	0	0.1209	3.649	1.115	0.9778	0.8225
RGA-PLM	− 0.1209	0	3.528	0.994	0.8569	0.7016
RGA-PWM	− 3.649	− 3.528	0	− 2.534	− 2.671	− 2.826
RGA-MTP	− 1.115	− 0.994	2.534	0	− 0.137	− 0.2924
RGA-GM	− 0.9778	− 0.8569	2.671	0.137	0	− 0.1553
RGA-UM	− 0.8225	− 0.7016	2.826	0.2924	0.1553	0

Table 6 Statistical results of post hoc procedures over all RGAs with IRGA as control method at a significance level equal to 0.05

Procedure	i	Algorithm	z-value	Unadjusted p-value	Adjusted p-value			
					p_{Holl}	p_{Rom}	p_{Finn}	p_{Li}
Friedman	1	RGA-PWM	8.9496	3.57E − 19	0.00E + 00	1.70E − 18	0.00E + 00	4.07E − 19
	2	RGA-MTP	6.9437	3.82E − 12	1.53E − 11	1.46E − 11	9.55E − 12	4.36E − 12
	3	RGA-GM	5.2463	1.55E − 07	4.66E − 07	4.66E − 07	2.59E − 07	1.77E − 07
	4	RGA-UM	4.1662	3.10E − 05	6.19E − 05	6.19E − 05	3.87E − 05	3.53E − 05
	5	RGA-PLM	1.5430	0.1228	0.1228	0.1228	0.1228	0.1228
Aligned Friedman	1	RGA-PWM	6.4153	1.41E − 10	7.03E − 10	6.68E − 10	7.03E − 10	1.09E − 09
	2	RGA-MTP	4.1627	3.15E − 05	1.26E − 04	1.20E − 04	7.86E − 05	2.43E − 04
	3	RGA-GM	3.7336	1.89E − 04	5.66E − 04	5.66E − 04	3.15E − 04	1.46E − 03
	4	RGA-UM	3.5260	4.22E − 04	8.44E − 04	8.44E − 04	5.27E − 04	3.26E − 03
	5	RGA-PLM	0.1626	0.8708	0.8708	0.8708	0.8708	0.8708
Quade	1	RGA-PWM	4.9209	8.61E − 07	4.31E − 06	4.10E − 06	4.31E − 06	1.66E − 06
	2	RGA-MTP	3.8486	1.19E − 04	4.75E − 04	4.53E − 04	2.97E − 04	2.29E − 04
	3	RGA-GM	2.6363	0.0084	0.0249	0.0251	0.0139	0.0159
	4	RGA-UM	2.2948	0.0217	0.0430	0.0435	0.0271	0.0403
	5	RGA-PLM	0.7035	0.4818	0.4818	0.4818	0.4818	0.4818

Table 7 Convergence results (p-values) of the first experiment

	IRGA	RGA-PLM	RGA-PWM	RGA-MTP	RGA-GM	RGA-UM
IRGA	–	**0.000311**	**0**	**0**	**0**	**0**
RGA-PLM	0.999706	–	**0**	**0**	**0**	**0**
RGA-PWM	1	1	–	1	1	1
RGA-MTP	1	1	**0**	–	**0.05362**	1
RGA-GM	1	1	**0.000001**	0.947979	–	1
RGA-UM	1	1	**0**	**0**	**0**	–

optimization algorithms used in this study to search for the globally optimum solutions with the fastest convergence rate. Moreover, it is statistically proved that these differences in the performances are significant.

4.2 The second experiment

It is debated that when the ranges of variables, within which the initial population is generated, do not cover the globally optimal solution, it becomes more difficult to solve the optimization problem, especially the multimodal ones (Deb and Beyer 2001). Considering this concept, the second experiment has been designed in such a way that the globally optimum point should not be bracketed by the initial population. Therefore, the first experiment has been repeated with only one dimensionality level ($d = 30$) and the initial population for each of the problems has been created in the variable range of (− 5, − 10). Moreover, the variable boundary handling technique has not been used in this experiment. The obtained average best finesses of 50 runs yielded by using the RGAs are provided in Table 8, where the best results are written in bold. From the results, it is observed to comprehend that the proposed IRGA has shown a superior performance compared to that of other RGAs.

4.3 Performance comparisons of IRGA with other recently proposed optimization algorithms

To get an idea of the state-of-the-art algorithms, the performance of the proposed IRGA has also been compared to that of a few recently proposed optimization techniques, such as social spider optimization (SSO) (Cuevas and Cienfuegos 2014), search group algorithm (SGA) (Gonçalves et al. 2015), sine–cosine algorithm (SCA) (Mirjalili 2016), cricket algorithm (Canayaz and Karci 2016), grasshopper optimization algorithm (GOA) (Saremi et al. 2017), and hybrid firefly and particle swarm optimization (HFPSO) (Aydilek 2018). The six test functions (refer to Table 1) with 30 variables ($d = 30$) are solved for 50 times

Table 8 Comparison of average results of 50 runs for the test functions (F01–F06) with different variable boundaries

Function	IRGA	RGA-PLM	RGA-PWM	RGA-MTP	RGA-GM	RGA-UM
F01	1.006E−13	2.145E−11	1.521E−10	2.565E−10	7.010E−10	1.636E−10
F02	2.688E−26	3.842E−22	1.559E−13	7.145E−13	7.820E−07	6.857E−13
F03	6.715E−08	1.421E−05	2.439E−04	3.300E−04	5.106E−03	1.810E−04
F04	2.499E+00	7.042E+00	1.372E+01	1.175E+01	1.087E+01	9.208E+00
F05	1.951E−06	7.175E−05	7.888E−06	4.086E−05	7.761E−05	6.483E−05
F06	1.643E−03	5.865E−02	1.555E−02	1.155E−02	2.564E−02	9.968E−03

each by using these recently developed algorithms. The common parameters, such as the maximum number of generations (max_gen $= 500$) and population size ($N = 5 \times d = 150$), are kept the same, as mentioned in Sect. 4.1.1. Moreover, for these methods, the algorithm-specific parameters are selected through some trial experiments and these are as follows:

- SSO — Lower and upper female per cent factors are considered to be equal to 0.65 and 0.9, respectively
- SGA — Initial value for perturbation factor = 2, minimum value of the perturbation factor = 0.01, search group ratio = 0.1, global iteration ratio = 0.3, and number of mutated individuals of the search group = 5
- SCA — The value of constant to obtain the range of sine and cosine has been taken to be equal to 2
- CA — minimum frequency ($Q_{\min} = 0$), $\beta_{\min}$ = 0.2
- GOA — Maximum and minimum values for the coefficient to shrink the comfort zone, repulsion zone, and attraction zones are 1 and 0.00004, respectively
- HFPSO — Acceleration coefficient ($c_1, c_2 = 1.49445$); maximum velocity (V_{max}) = 0.1 × Search Range; minimum velocity (V_{min}) = -V_{max}; inertia weight (w), w_i = 0.9, w_f = 0.5; for firefly algorithm (FA), = 0.2; B_0 = 2; = 1

The obtained average best finesses of the 50 runs are reported in Table 9. From the results, it is clear that the proposed IRGA has outperformed the other six recently developed optimization approaches for all the test functions considered in this experiment.

4.4 Scale-up study

It is a common experience that with the increase in number of variables, the optimization problems, especially multi-modal functions, become more and more complex and difficult to solve (Deb and Beyer 2001). Therefore, it is recommended to perform a scalability study to see whether the proposed algorithm is able to find the globally optimum solution in a large variable space. Moreover, another purpose of this study is to get an approximate idea of how the required number of function evaluations to reach a particular objective function value vary with the number of variables of the given problem. To serve these two purposes, we have carried out an experiment on the set of six test functions mentioned in Table 1. These functions have been solved using the proposed IRGA, and the number of variables for each of the problems has been varied from 20 to 200 with an increment of 20. The parameters' setting for the IRGA has been taken the same as that mentioned in Sect. 4.1.1, except the stopping criterion. Here, the proposed algorithm stops, when the current best fitness value is found to be either less than or equal to 0.01. Each of the problems has been solved for ten times, and after each run, the required number of function evaluations (fe) has been noted. Furthermore, a logarithmic plot has been drawn for each of the test functions (refer to Fig. 9), where the X and Y axes represent the values of $\ln(d)$ and $\ln(\text{fe})$, respectively. In addition, the slope of each curve is mentioned in the figure for all the test functions. It is to be noted from this study that even for the large variable space, the proposed IRGA is capable of hitting the globally optimum point. Moreover, the measured parameter fe varies following an approximate polynomial distribution with d in

Table 9 Comparisons of average results of 50 runs for the test functions (F01-F06) obtained by the proposed IRGA and other recent optimization algorithms

Function	IRGA	SSO	SGA	SCA	CA	GOA	HFPSO
F01	2.894E−12	8.006E−02	1.976E−03	4.457E−01	1.824E+02	6.555E−02	1.685E−10
F02	8.517E−19	4.890E+08	1.833E+03	1.695E+06	7.957E+41	2.021E+36	6.427E−13
F03	1.514E−08	7.357E+04	2.265E+01	1.738E+03	1.857E+03	8.080E+00	4.000E+00
F04	6.403E−02	5.041E+01	2.692E+01	2.153E+01	1.236E+02	1.097E+02	4.055E+01
F05	1.205E−08	2.378E−01	7.106E−02	4.629E−01	9.962E+00	4.010E+00	5.350E−04
F06	3.118E−03	1.251E+00	2.551E+00	7.733E−02	6.674E+00	2.770E+00	6.012E−01

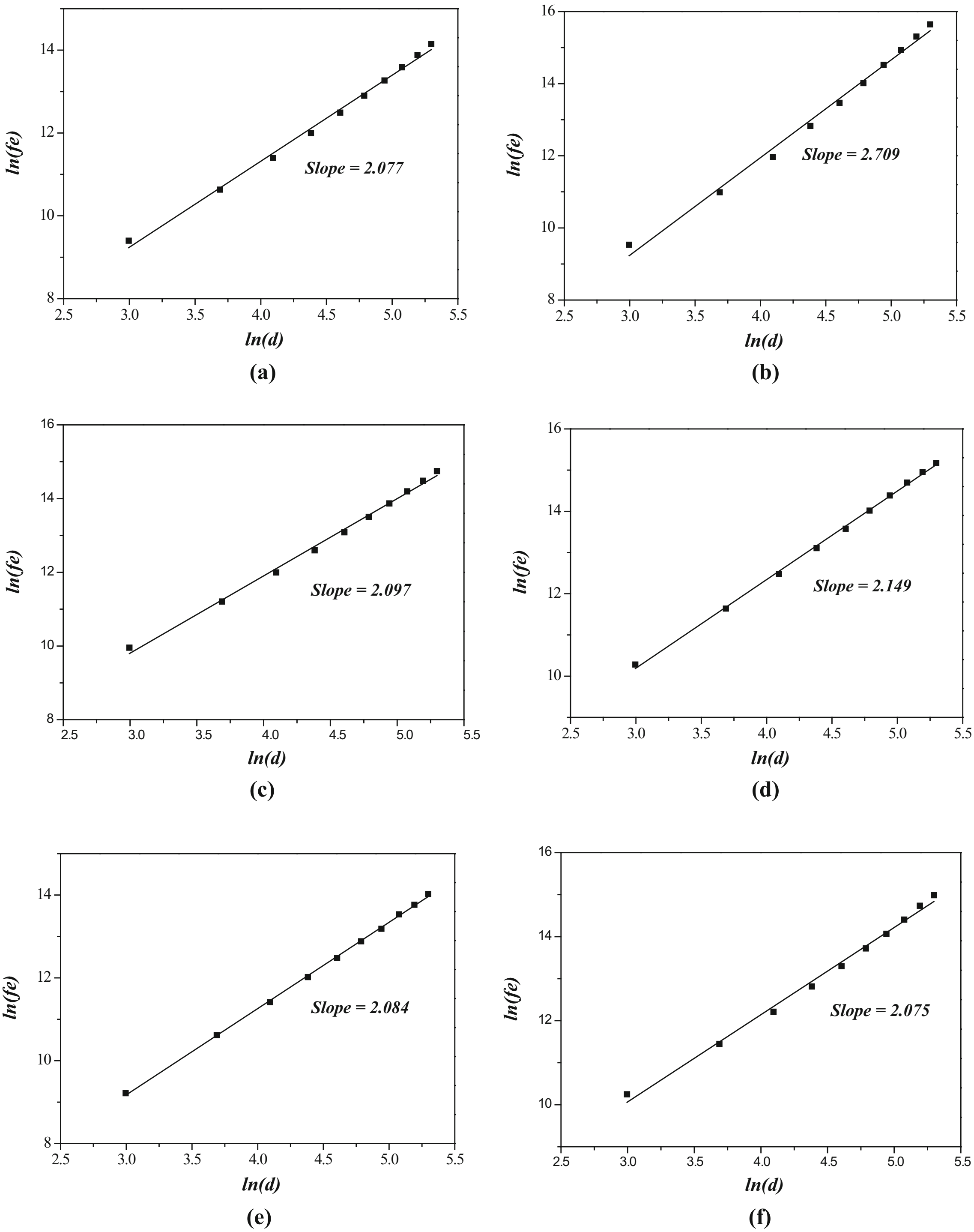


Fig. 9 Scale-up study for the functions: **a** F01, **b** F02, **c** F03, **d** F04, **e** F05, and **f** F06

the total range of the number of variables considered in this experiment. From these values of the slopes, it has been seen that the required number of function evaluations for the F02 is more compared to that of the other functions. This may be due to a reason that this function may become more complex with the higher number of variables compared to the other ones. However, the slopes for the other functions are found to be close to 2. This is a great advantage of the proposed IRGA that it can handle complex optimization problem in an efficient manner and takes the smaller number of function evaluations to reach the globally optimum solution.

Table 10 Comparisons of the results of the time study

Function	IRGA		RGA-PLM		RGA-PWM		RGA-MTP		RGA-GM		RGA-UM	
	t_{avg} (in Sec.)	fe_{avg}	t_{avg} (in Sec.)	fe_{avg}	t_{avg} (in Sec.)	fe_{avg}	t_{avg} (in Sec.)	fe_{avg}	t_{avg} (in Sec.)	fe_{avg}	t_{avg} (in Sec.)	fe_{avg}
F01	0.4624	19,497	0.5445	22,839	0.7035	30,093	0.6968	29,481	0.6975	29,688	0.6609	28,392
F02	0.8310	29,310	0.9090	32,049	1.9426	68,613	1.3974	49,497	1.3173	46,845	1.2195	43,722
F03	0.8854	38,031	1.0906	45,945	1.2515	53,841	1.2279	52,506	1.2241	52,863	1.1862	51,348
F04	1.4707	61,830	1.7907	74,643	NA	NA	2.6866	112,482	2.2213	93,699	2.0520	86,760
F05	0.2656	10,887	0.2820	11,580	0.4430	18,414	0.4039	16,728	0.4067	16,890	0.3863	15,879
F06	1.2740	31,455	1.5780	38,454	1.8441	45,789	1.7898	44,406	1.7486	44,196	1.7489	43,761

4.5 Time study

A time study has also been carried out to analyse both computation and convergence performance of the proposed IRGA and to make a comparison with the other five RGAs. In this experiment, each of the test functions (refer to Table 1) with 30 dimensions has been solved for 50 times and the parameters' settings for the RGAs are considered to be the same, as mentioned in Sect. 4.1.1, except the stopping criterion. An algorithm is allowed to stop, only when it reaches an objective function value equal to 0.1.

After the completion of each run, the required CPU time (in seconds) and the number of function evaluations are captured. Moreover, the average values of the CPU times (t_{avg} in seconds) and number of function evaluations (fe_{avg}) of the 50 runs are calculated for each of the optimization functions and these are reported in Table 10. It is to be noted that all the experiments have been performed on an Intel Core i5 Processor with 3.20 GHz speed and 16 GB RAM under Windows 10 platform using MATLAB 2017b programming language. For F04, the RGA-PWM has not been seen to reach the desired solution accuracy up to 10^5 number of generations, and therefore, the data for the RGA-PWM related to this function have been depicted as not available (NA) in the table. From the results provided in Table 10, it has been found that the proposed IRGA has taken the minimum CPU time and less number of function evaluations to reach the desired accuracy of the solution for all the test functions compared to that of other RGAs. This finding also proves that the proposed IRGA has the fastest convergence rate of all the algorithms used in this study.

5 Real-world applications of the proposed IRGA

Finally, the performance of the proposed IRGA has been examined on some real-world problems. We have selected four engineering constrained optimization problems related to tension/compression spring, welded beam, pressure vessel, and speed reducer gearing system. These applications had been already solved using other several optimization methods, and therefore, a comparison of the obtained results has been done. Moreover, to solve these constrained optimization problems, a simple scalar penalty function approach (Kuo and Lin 2013) has been adopted for the IRGA.

5.1 Tension/compression spring design problem

This design problem is related to minimization of the weight of a tension/compression spring and this is subjected to the constraints on various attributes, like minimum deflection of the spring, surge frequency, shear stress, and the ranges of design variables (refer to Fig. 10). Here, the wire diameter (d_w), mean coil diameter (D), and the number of active coils (n) are the three design variables of the problem.

This optimization problem is mathematically expressed as follows:

$$\text{Minimize } f(X) = (n+2)Dd_w^2 \tag{21}$$

subject to

$$g_1(X) = 1 - \frac{D^3 n}{71785 d_w^4} \le 0$$

$$g_2(X) = \frac{4D^2 - d_w D}{12566\left(Dd_w^3 - d_w^4\right)} + \frac{1}{5108 d_w^2} - 1 \le 0$$

$$g_3(X) = 1 - \frac{140.45 d_w}{D^2 n} \le 0$$

$$g_4(X) = \frac{(D + d_w)}{1.5} - 1 \le 0$$

and $0.05 \le d_w \le 2.00$, $0.25 \le D \le 1.30$, $2.0 \le n \le 15.0$.

The said problem had been solved by both traditional and non-traditional optimization tools and these are as follows: constraint correction at constant cost (Arora 2004), an approach based on mathematical optimization

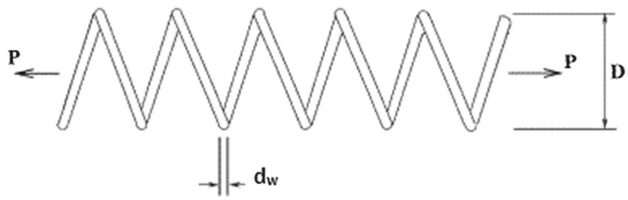


Fig. 10 Schematic view of tension/compression spring design (Coello 2000)

(Belegundu and Arora 1985), genetic algorithm (GA) (Coello 2000), particle swarm optmization (PSO) (He and Wang 2007), differential evolution (DE) (Huang et al. 2007), evolution strategy (ES) (Mezura-Montes and Coello 2008), harmony search (HS) (Mahdavi et al. 2007), grey wolf optimizer, and gravitational search algorithm (GSA) (Mirjalili et al. 2014). The results obtained by these methods and that yielded utilizing the proposed IRGA (with a precision up to six digits after the decimal point) have been reported in Table 11 (best solution has been marked in bold). From this table, it is seen that the IRGA has given the best optimum and feasible result (minimum weight value of 0.012665 with no constraint violation) for the problem compared to that of the other methods.

5.2 Design optimization of a welded beam

The aim of this optimization problem is to minimize the cost of fabrication for the welded beam (refer to Fig. 11) and there are seven functional constraints related to the aspects, namely shear stress (τ), bending stress (θ), buckling load on the beam (P_c), and end deflection of the bar (δ). The four design parameters: h, l, t, and b, are represented by x_1, x_2, x_3 and x_4, respectively.

The mathematical formulation of the optimization problem is given as follows:

$$\text{Minimize } f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{22}$$

subject to

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

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

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

$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5.0 \le 0$

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

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

$g_7(X) = P - P_c(X) \le 0$

and

$0.1 \le x_1, x_4 \le 2.0,\ 0.1 \le x_2, x_3 \le 10.0$

where

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

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

$\sigma(X) = \frac{6PL}{x_4x_3^2}$, $\delta(X) = \frac{4PL^3}{Ex_3^3x_4}$, $P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$, $P = 6000$ lb, $L = 14$ in., $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13,600$ psi, $\sigma_{\max} = 30,000$ psi, $\delta_{\max} = 0.25$ in.

To solve this problem, several traditional optimization methods, such as Richardson's random method, Davidon–Fletcher–Powell, simplex method, Griffith and Stewart's successive linear approximation method (Ragsdell and Phillips 1976), and various meta-heuristic techniques, like GA1 (Coello Coello 2000), GA2 (Deb 1991), GA3 (Deb 2000), HS (Lee and Geem 2005), GSA, and GWO

Table 11 Comparison of results for the design problem of tension/compression spring

Algorithm	Best obtained solution			Optimum weight
	d_w	D	n	
IRGA	0.051,686	0.356644	11.293294	0.012665
GWO	0.05169	0.356737	11.28885	0.012666
GSA	0.050276	0.32368	13.52541	0.0127022
PSO	0.051728	0.357644	11.244543	0.0126747
ES	0.051989	0.363965	10.890522	0.012681
GA	0.05148	0.351661	11.632201	0.0127048
HS	0.051154	0.349871	12.076432	0.0126706
DE	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization	0.053396	0.39918	9.1854	0.0127303
Constraint correction	0.05	0.3159	14.25	0.0128334

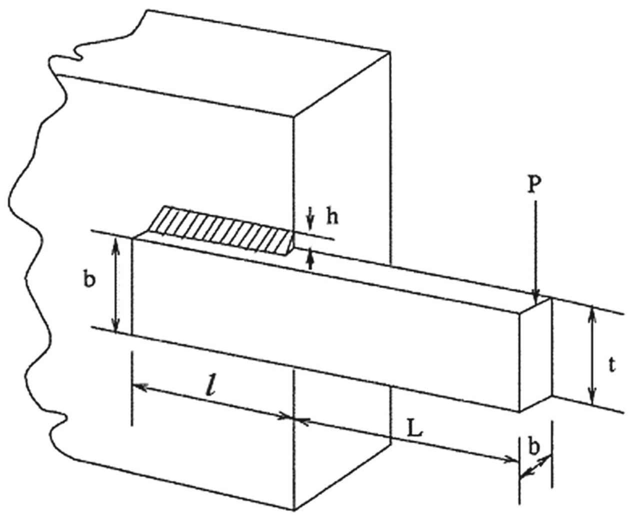


Fig. 11 Schematic view of a welded beam (Deb 1991)

(Mirjalili et al. 2014), had been applied. These reported results along with that yielded by the proposed IRGA (with a precision up to the six digits after the decimal point) are provided in Table 12, where IRGA has shown its superior performance compared to the other methods. (Best solution is shown in bold.)

5.3 Design optimization of a pressure vessel

This problem deals with the optimal design of a cylindrical pressure vessel capped with the head of hemispherical shape (see Fig. 12). Here, the total fabrication cost, including the cost of the material, welding, and forming, is to be minimized after satisfying four functional constraints of the problem. There are four design variables, such as the thickness of the shell (T_s), thickness of the head (T_h), inner radius (r_i), and length of the cylindrical vessel excluding the head portion (l). It is to be noted that the design parameters, such as T_s and T_h, should have the values that are integer multiples of 0.0625 inch (as these thickness values are available for the rolled steel plate) (Kannan and Kramer 1994). However, the other two variables, such as r_i and l, are found to be continuous in nature. Mathematically, this optimization problem can be expressed as follows:

$$\text{Take}, X = [x_1 x_2 x_3 x_4] = [T_s T_h r_i l],$$

$$\text{Minimize}\, f(X) = 0.6224 x_1 x_2 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \quad (23)$$

subject to

$$g_1(X) = -x_1 + 0.0193 x_3 \leq 0$$

$$g_2(X) = -x_2 + 0.00954 x_3 \leq 0$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$

$$g_4(X) = -x_4 - 240 \leq 0$$

and $0.0 \leq x_1, x_2 \leq 99.0$, $10.0 \leq x_3, x_4 \leq 200.0$.

This problem had been tried to solve using several optimization techniques, like GA1 (Coello 2000), GA2 (Coello and Montes 2002), GA3 (Deb 1997), PSO (He and Wang 2007), DE (Huang et al. 2007), ES (Mezura-Montes and Coello 2008), ant colony optimization (ACO) (Kaveh and Talatahari 2010), GSA, GWO (Mirjalili et al. 2014), augmented Lagrangian multiplier (Kannan and Kramer 1994), and branch-and-bound (Sandgren 1990). Their results along with that obtained using the proposed IRGA (with a precision level of six digits after the decimal) are given in Table 13. From this table, it has been observed

Table 12 Comparison of results for the design problem of welded beam

Algorithm	Best obtained solution				Best cost
	H	*l*	*t*	*b*	
IRGA	0.20573	3.470484	9.036616	0.20573	1.724854
GWO	0.205676	3.478377	9.03681	0.205778	1.72624
GSA	0.182129	3.856979	10	0.202376	1.879952
GA1	N/A	N/A	N/A	N/A	1.8245
GA2	N/A	N/A	N/A	N/A	2.38
GA3	0.2489	6.173	8.1789	0.2533	2.4331
HS	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.66	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815

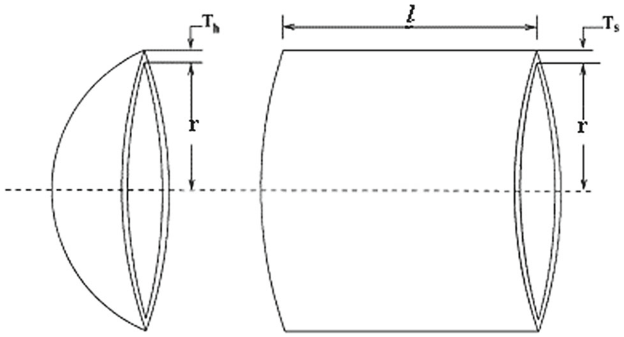


Fig. 12 Schematic view of pressure vessel (Kaveh and Talatahari 2010)

that GWO has yielded the minimum cost for the problem. However, this solution is found to be infeasible, as the obtained T_h value is not the integer multiple of 0.0625 inches. Thus, the solution obtained by the proposed IRGA is seen to be the best feasible one out of all, as shown in bold, in Table 13.

5.4 Design of a speed reducer gearing system

The goal of this optimization problem is to minimize the weight of a speed reducer (refer to Fig. 13) and it is subjected to eleven functional constraints on several attributes, such as bending stress of the gear teeth, transverse deflection of the shafts, surface stress, and the stresses created in the shafts. There are seven design variables in this optimization problem, such as face width (b), module of teeth (m), number of teeth on the pinion (z), lengths of the first and second shafts between bearings (L_1 and L_2, respectively), and diameters of the first and second shafts (d_1 and d_2, respectively). These parameters are represented using the notations x_1 through x_7, respectively. Moreover, all these design variables are found to be continuous, except for the third variable, which takes only the integer value. Hence, it is a mixed-integer optimization problem with eleven functional and seven side constraints. This makes the problem very difficult to solve (Ku et al. 1998).

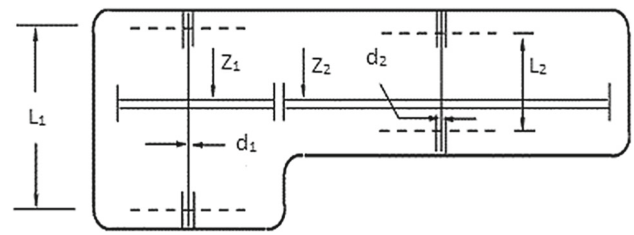


Fig. 13 Schematic view of the speed reducer (Sadollah et al. 2013)

Mathematically, this optimization problem can be written as follows:

$$\begin{aligned}\text{Minimize } f(X) = {} & 0.7845x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ & - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\ & + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)\end{aligned} \tag{24}$$

subject to

$$g_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \le 0$$

$$g_2(X) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0$$

$$g_3(X) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0$$

$$g_4(X) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0$$

Table 13 Comparison of results for the design problem of pressure vessel

Algorithm	Best obtained solution				Best cost
	T_s	T_h	r_i	l	
IRGA	0.8125	0.4375	42.098445	176.636604	6059.7144
GWO	0.8125	0.4345 (infeasible sol.)	42.089181	176.758731	6051.5639
GSA	1.125	0.625	55.9886598	84.4542025	8538.8359
PSO	0.8125	0.4375	42.091266	176.7465	6061.0777
GA1	0.8125	0.4345	40.3239	200	6288.7445
GA2	0.8125	0.4375	42.097398	176.65405	6059.9463
GA3	0.9375	0.5	48.329	112.679	6410.3811
ES	0.8125	0.4375	42.098087	176.640518	6059.7456
DE	0.8125	0.4375	42.098411	176.63769	6059.734
ACO	0.8125	0.4375	42.098353	176.637751	6059.7258
Lagrangian multiplier	1.125	0.625	58.291	43.69	7198.0428
Branch-and-bound	1.125	0.625	47.7	117.701	8129.1036

$$g_5(X) = \frac{[(745(x_4/x_2x_3))^2 + 16.9 \times 10^6]^{\frac{1}{2}}}{110x_6^3} - 1 \le 0$$

$$g_6(X) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.5 \times 10^6\right]^{\frac{1}{2}}}{85x_7^3} - 1 \le 0$$

$$g_7(X) = \frac{x_2x_3}{40} - 1 \le 0$$

$$g_8(X) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(X) = \frac{x_1}{12x_2} - 1 \le 0$$

$$g_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

and $2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4, x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5.0 \le x_7 \le 5.5$.

Through a thorough literature review, it is found that the problem had been solved using several optimization approaches, such as differential evolution with level comparison (DELC) (Wang and Li 2010), differential evolution with dynamic stochastic selection (DEDS) (Zhang et al. 2008), hybrid evolutionary algorithm with adaptive constraint handling (HEAA) (Wang et al. 2009), social behaviour-inspired optimization (SBO) technique (Ray and Liew 2003), modified differential evolution (MDE) (Mezura-Montes et al. 2006), and mine blast algorithm (Sadollah et al. 2013). Table 14 contains the optimized solutions obtained by these techniques and the proposed IRGA. It is to be noted that the result yielded by the IRGA has the precision level of ten digits after the decimal point (best results is reported in bold). After analysing the results, it is easy to say that the proposed IRGA has able to find the best feasible solution for the problem of interest compared to that of the other techniques.

5.5 Optimization of Abrasive Jet machining (AJM) process parameters

In AJM process, a jet of abrasive particles along with a carrier gas with high velocity comes out from a nozzle and impinges on the surface material to erode it. Here, a constrained optimization model of AJM process for ductile material is presented (Jain et al. 2007). There are three design variables in the optimization problem and these are as follows: mass flow rate of abrasives ($\dot{M}_a$ in kg/s), mean radius of particles (r_m in mm), and velocity of abrasives (v_a in mm/s). These variables are expressed as: $X = [x_1, x_2, x_3] = [\dot{M}_a, r_m, v_a]$. The goal is to maximize material removal rate (MRR in mm^3/s) ($f(X)$), while satisfying surface roughness constraint ($g(X)$). The optimization problem is formulated mathematically as follows:

$$\text{Maximize}\, f(X) = 1.0436 \times 10^{-6} \zeta \frac{\rho_w}{\delta_{cw}^2 H_{dw}^{1.5} \rho_a^{0.5}} x_1 x_3^3 \quad (25)$$

subject to

$$g(X) = 1.0 - \frac{25.82}{(R_a)_{max}} [\frac{\rho_a}{H_{dw}}]^{0.5} x_2 x_3 \ge 0.0$$

$0.0000167 \le x_1 \le 0.0005$, $0.005 \le x_2 \le 0.075$, and$150000 \le x_3 \le 400000$, where density of abrasives ($\rho_a = 2.48E - 6$ kg/mm^3); density of work material ($\rho_w = 2.7E - 6$ kg/mm^3); critical plastic strain of ductile work material ($\delta_{\mathrm{cw}} = 1.5$); dynamic hardness of ductile work material ($H_{\mathrm{dw}} = 1150$ MPa); amount of plastically deformed indentation volume ($\zeta = 1.6$); and permissible surface roughness ($(R_a)_{\max} = 2.0\ \mu$m).

This optimization problem has been solved using a genetic algorithm (GA1) with simulated binary crossover (SBX) and polynomial mutation operators, and proposed IRGA. Both the GAs are run for 50 times each. The common parameters for both the algorithms have been set as follows: maximum number generations (max_gen = 50), population size ($N = 50$), crossover

Table 14 Comparison of results for the speed reducer problem

D.V.	IRGA	DELC	DEDC	HEAA	SBO	MDE	MBA
x_1	3.5000000000	3.5000000000	3.5000000000	3.5000228993	3.5000681000	3.5000100000	3.500000
x_2	0.7000000000	0.7000000000	0.7000000000	0.7000003924	0.7000000100	0.7000000000	0.700000
x_3	17.0000000000	17.0000000000	17.0000000000	17.0000128592	17.0000000000	17.0000000000	17.000000
x_4	7.3000000000	7.3000000000	7.3000000000	7.3004277414	7.3276020500	7.3001560000	7.300033
x_5	7.7151697140	7.7153199115	7.7153199115	7.7153774494	7.7153217500	7.8000270000	7.715772
x_6	3.3502146661	3.3502146661	3.3502146610	3.3502309666	3.3502670200	3.3502210000	3.350218
x_7	5.2865179218	5.2866544650	5.2866544650	5.2866636970	5.2866545000	5.2866850000	5.286654
$f(X)$	2994.381034	2994.471066	2994.471066	2994.499107	2994.744241	2996.356689	2994.482453

probability ($p_c = 1.0$), and mutation probability ($p_m = 0.03$). For GA1, the indices for SBX and polynomial mutation are selected as 10 and 10, respectively, similar to that, as in (Jain et al. 2007). For IRGA, the other parameters have been set as mentioned in Sect. 4.1.1. This constrained optimization problem is solved using simple penalty function approach with a penalty factor kept equal to 10^{10}. The best, mean, worst, standard deviation (SD), and CPU time of the obtained results are shown in Table 15. (The best results are marked in bold.) From the results, it is clear that the proposed IRGA is able to yield the best optimal solution for the AJM problem, after satisfying the surface roughness constraint. Moreover, the proposed IRGA is found to be computationally faster than GA1 in terms of CPU time.

The superior performance of the proposed IRGA, in solving both the classical benchmark functions and constrained engineering problems, proves that the algorithm can maintain a good balance between the population diversity and selection pressure. The designed mechanism of the proposed mutation operator, which promotes both the global and local searches, is found to be effective in solving the unimodal and multimodal optimization problems. Moreover, the used directional information of the problem helps the proposed IRGA to converge to an optimal solution at the faster rate.

6 Conclusions

In this study, an effort has been made to improve the capability and performance of an RGA with a proposed directional mutation operator (DM) along with the directional crossover. The proposed mutation scheme utilizes the directional information of the search process, so that it can focus on the most potential areas of the variable space in order to find the better solutions in next iteration. In addition, a directional probability term (p_d) has been introduced in the proposed operator to keep a proper balance between the exploration and exploitation capability of the IRGA. For both the unimodal and multimodal optimization problems, the proposed IRGA is seen to outperform the other five RGAs in terms of accuracy in the solution. It happens due to the fact that the proposed algorithm is able to maintain a superior balance between the diversity and selection pressure compared to the others. In the statistical analysis of the obtained results, the proposed IRGA is also proved to be the best performer with the fastest convergence rate among all the algorithms used in this study. Scale-up study reveals that the proposed IRGA can also handle complex optimization problems in an efficient manner and it can reach the globally optimum solutions with the smaller number of function evaluations. Moreover, the excellent performance of the proposed IRGA to solve an optimization problem even with a large variable space confirms that it is equally efficient to handle the problems with the large dimensions. The outcomes of the time study show that the proposed IRGA has the better convergence performance and it is also found to be computationally less expensive compared to other RGAs. Furthermore, the comparisons of the results of the proposed IRGA with that of a few other recently developed optimization algorithms strengthen the foundation of the new proposal in the present state of the art of the optimization algorithms. In addition, the successful implementations of the proposed IRGA in the realm of real-world optimization problems ensure its supremacy in solving the constrained optimization problems with varying complexities.

The use of directional information ensures the search method of the proposed IRGA to be efficient. Moreover, the internal mechanism of the proposed mutation operator (DM) has been designed in such a way that it can perform both the global as well as local search. Overall, the proposed IRGA is able to maintain a good balance between the diversification and intensification phenomena, which consequently enables it to perform in a better way compared to others. In future, the proposed IRGA will be applied to solve even more complex real-world optimization problems and the performance of IRGA will be compared to that of some efficient hybrid approaches, like firefly and

Table 15 Comparison of results for the AJM problem

		IRGA	GA1
$f(X)$	Best	**0.6056**	0.6019
	Mean	**0.6056**	0.5091
	Worst	**0.6056**	0.1596
	S.D.	8.351E−06	9.836E−02
Decision variables (D.V.) for the best $f(X)$	x_1	0.0005	0.0005
	x_2	0.0050	0.0050
	x_3	333600.6077	332916.0141
Constraint violation (C.V.)		0.00	0.00
Total CPU time for 50 runs (in seconds)		**5.9146**	5.9413

cuckoo (Elkhechafi et al. 2018); hybrid firefly with chaotic simulated annealing (Tayal and Singh 2017, 2018), and others.

Compliance with ethical standards

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

References

Arora J (2004) Introduction to optimum design. Academic Press, San Diego. https://doi.org/10.1016/B978-012064155-0/50001-X

Barroso ES, Parente E, Cartaxo de Melo AM (2017) A hybrid PSO-GA algorithm for optimization of laminated composites. Struct Multidiscip Optimiz 55:2111–2130. https://doi.org/10.1007/s00158-016-1631-y

Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part I: theory. Int J Numer Meth Eng 21:1583–1599. https://doi.org/10.1002/nme.1620210904

Berry A, Vamplew P (2004) PoD can mutate: a simple dynamic directed mutation approach for genetic algorithms. In: The 2nd international conference on artificial intelligence in science and technology (AISAT 2004). pp 200–205

Canayaz M, Karci A (2016) Cricket behaviour-based evolutionary computation technique in solving engineering optimization problems. Appl Intell 44:362–376. https://doi.org/10.1007/s10489-015-0706-6

Cao H, Qian X, Zhou Y (2018) Large-scale structural optimization using metaheuristic algorithms with elitism and a filter strategy. Struct Multidiscip Optim 57:799–814. https://doi.org/10.1007/s00158-017-1784-3

Chen S-y, Shui X-f, Huang H (2017) Improved genetic algorithm with two-level approximation using shape sensitivities for truss layout optimization. Struct Multidiscip Optim 55:1365–1382. https://doi.org/10.1007/s00158-016-1583-2

Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127. https://doi.org/10.1016/S0166-3615(99)00046-9

Coello Coello CA (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civil Eng Syst 17:319–346. https://doi.org/10.1080/02630250008970288

Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–200. https://doi.org/10.1016/S1474-0346(02)00011-3

Corbera Caraballo S, Olazagoitia Rodríguez JL, Lozano Ruiz JA, Álvarez Fernández R (2017) Optimization of a butterfly valve disc using 3D topology and genetic algorithms. Struct Multidiscip Optim 56:941–957. https://doi.org/10.1007/s00158-017-1694-4

Cuevas E, Cienfuegos M (2014) A new algorithm inspired in the behavior of the social-spider for constrained optimization. Expert Syst Appl 41:412–425. https://doi.org/10.1016/j.eswa.2013.07.067

Das AK, Pratihar DK (2017) A direction-based exponential crossover operator for real-coded genetic algorithm. Paper presented at the seventh international conference on theoretical, applied, computational and experimental mechanics (ICTACEM 2017), IIT Kharagpur, India

Das AK, Pratihar DK (2018) Performance improvement of a genetic algorithm using a novel restart strategy with elitism principle. Int J Hybrid Intell Syst (pre-press): 1–15. https://doi.org/10.3233/his-180257

Das AK, Pratihar DK (2019) A directional crossover (DX) operator for real parameter optmization using genetic algorithm. Appl Intell 49:1841. https://doi.org/10.1007/s10489-018-1364-2

Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015

Deb K (1997) GeneAS: A robust optimal design technique for mechanical component design. In: Dasgupta D, Michalewicz Z (eds) Evolutionary algorithms in engineering applications. Springer, Berlin, pp 497–514. https://doi.org/10.1007/978-3-662-03423-1_27

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338. https://doi.org/10.1016/S0045-7825(99)00389-8

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester

Deb K, Beyer H-g (2001) Self-adaptive genetic algorithms with simulated binary crossover. Evol Comput 9:197–221. https://doi.org/10.1162/106365601750190406

Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inf 26:30–45

Deep K, Thakur M (2007) A new mutation operator for real coded genetic algorithms. Appl Math Comput 193:211–230. https://doi.org/10.1016/j.amc.2007.03.046

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Derrac J, García S, Hui S, Suganthan PN, Herrera F (2014) Analyzing convergence performance of evolutionary algorithms: a statistical approach. Inf Sci 289:41–58. https://doi.org/10.1016/j.ins.2014.06.009

Elkhechafi M, Hachimi H, Elkettani Y (2018) A new hybrid cuckoo search and firefly optimization. Monte Carlo Methods Appl 24(1):71–77

Furuya H, Haftka RT (1995) Placing actuators on space structures by genetic algorithms and effectiveness indices. Struct Optim 9:69–75. https://doi.org/10.1007/bf01758822

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180:2044–2064. https://doi.org/10.1016/j.ins.2009.12.010

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc, Boston

Gonçalves MS, Lopez RH, Miguel LFF (2015) Search group algorithm: a new metaheuristic method for the optimization of truss structures. Comput Struct 153:165–184. https://doi.org/10.1016/j.compstruc.2015.03.003

He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99. https://doi.org/10.1016/j.engappai.2006.03.003

Hinterding R (1995) Gaussian mutation and self-adaption for numeric genetic algorithms. In: Proceedings of international conference on evolutionary computation (ICEC 1995). pp 384–389. https://doi.org/10.1109/icec.1995.489178

Hinterding R, Michalewicz Z, Peachey TC (1996) Self-adaptive genetic algorithm for numeric functions. Parallel problem solving from nature—PPSN IV. Springer, Berlin, pp 420–429

Holland JH (1992) Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence, 2nd edn. MIT Press, Cambridge
Huang F-z, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356. https://doi.org/10.1016/j.amc.2006.07.105
Jafari M, Moussavian H, Chaleshtari MHB (2018) Optimum design of perforated orthotropic and laminated composite plates under in-plane loading by genetic algorithm. Struct Multidiscip Optim 57:341–357. https://doi.org/10.1007/s00158-017-1758-5
Jain N, Jain VK, Deb K (2007) Optimization of process parameters of mechanical type advanced machining processes using genetic algorithms. Int J Mach Tools Manuf 47:900–919. https://doi.org/10.1016/j.ijmachtools.2006.08.001
Kannan B, 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. https://doi.org/10.1115/1.2919393
Kaveh A, Bakhshpoori T (2016) A new metaheuristic for continuous structural optimization: water evaporation optimization. Struct Multidiscip Optim 54:23–43. https://doi.org/10.1007/s00158-015-1396-8
Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182. https://doi.org/10.1108/02644401011008577
Kazemzadeh Azad S (2017) Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables. Struct Multidiscip Optim 55:2159–2180. https://doi.org/10.1007/s00158-016-1634-8
Kogiso N, Watson LT, Gürdal Z, Haftka RT (1994a) Genetic algorithms with local improvement for composite laminate design. Struct Optim 7:207–218. https://doi.org/10.1007/bf01743714
Kogiso N, Watson LT, GÜRdal Z, Haftka RT, Nagendra S (1994b) Design of composite laminates by a genetic algorithm with memory. Mech Compos Mater Struct 1:95–117. https://doi.org/10.1080/10759419408945823
Koh B-I, George AD, Haftka RT, Fregly BJ (2006) Parallel asynchronous particle swarm optimization. Int J Numer Meth Eng 67:578–595. https://doi.org/10.1002/nme.1646
Ku K, Rao S, Chen L (1998) Taguchi-aided search method for design optimization of engineering systems. Eng Optim 30:1–23. https://doi.org/10.1080/03052159808941235
Kuo H-C, Lin C-H (2013) A directed genetic algorithm for global optimization. Appl Math Comput 219:7348–7364. https://doi.org/10.1016/j.amc.2012.12.046
Le Riche R, Haftka RT (1995) Improved genetic algorithm for minimum thickness composite laminate design. Compos Eng 5:143–161. https://doi.org/10.1016/0961-9526(95)90710-S
Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933. https://doi.org/10.1016/j.cma.2004.09.007
Ling SH, Leung FHF (2007) An improved genetic algorithm with average-bound crossover and wavelet mutation operations. Soft Comput 11:7–31. https://doi.org/10.1007/s00500-006-0049-7
MacKay DJ, Mac Kay DJ (2003) Information theory, inference and learning algorithms. Cambridge University Press, Cambridge
Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579. https://doi.org/10.1016/j.amc.2006.11.033
Mäkinen RA, Périaux J, Toivanen J (1999) Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms. Int J Numer Meth Fluids 30:149–159. https://doi.org/10.1002/(SICI)1097-0363(19990530)30:2%3c149:AID-FLD829%3e3.0.CO;2-B
Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37:443–473. https://doi.org/10.1080/03081070701303470
Mezura-Montes E, Coello CC, Velázquez-Reyes J (2006) Increasing successful offspring and diversity in differential evolution for engineering design. In: Proceedings of the seventh international conference on adaptive computing in design and manufacture (ACDM 2006), pp 131–139
Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, New York
Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evol Comput 4:1–32. https://doi.org/10.1162/evco.1996.4.1.1
Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
Munteanu C, Lazarescu V (1999) Improving mutation capabilities in a real-coded genetic algorithm. In: Evoworkshops: evolutionary image analysis, signal processing and telecommunications. Berlin Heidelberg, pp 138–149
Nagendra S, Haftka R, Gurdal Z (1993) Design of a blade stiffened composite panel by genetic algorithm. In: 34th structures, structural dynamics and materials conference. La Jolla, CA, pp 2418–2436. https://doi.org/10.2514/6.1993-1584
Nagendra S, Jestin D, Gürdal Z, Haftka RT, Watson LT (1996) Improved genetic algorithm for the design of stiffened composite panels. Comput Struct 58:543–555. https://doi.org/10.1016/0045-7949(95)00160-I
Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) Section 22.3, gray codes, numerical recipes: the art of scientific computing, 3rd edn. Cambridge University Press, New York
Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Eng Industry 98:1021–1025. https://doi.org/10.1115/1.3438995
Raj KH, Setia R (2017) Quantum seeded evolutionary computational technique for constrained optimization in engineering design and manufacturing. Struct Multidiscip Optim 55:751–766. https://doi.org/10.1007/s00158-016-1529-8
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. https://doi.org/10.1109/TEVC.2003.814902
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 13:2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223–229. https://doi.org/10.1115/1.2912596
Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
Schutte JF, Koh B, Reinbolt JA, Haftka RT, George AD, Fregly BJ (2005) Evaluation of a particle swarm algorithm for biomechanical optimization. J Biomech Eng 127:465–474
Schwefel HP (1987) Collective phenomena in evolutionary systems. Universität Dortmund, Abteilung Informatik

Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: survey of recent developments. Struct Optim 14:1–23. https://doi.org/10.1007/bf01197554

Soremekun G, Gürdal Z, Haftka RT, Watson LT (2001) Composite laminate design optimization by genetic algorithm with generalized elitist selection. Comput Struct 79:131–143. https://doi.org/10.1016/S0045-7949(00)00125-5

Tayal A, Singh SP (2017) Designing flexible stochastic dynamic layout: an integrated firefly and chaotic simulated annealing-based approach. Glob J Flex Syst Manag 18:89–98. https://doi.org/10.1007/s40171-016-0140-6

Tayal A, Singh SP (2018) Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem. Ann Oper Res 270(1–2):489–514

Temby L, Vamplew P, Berry A (2005) Accelerating real-valued genetic algorithms using mutation-with-momentum. In: Advances in artificial intelligence (AI 2005). Springer, Berlin, pp 1108–1111

Todoroki A, Haftka RT (1998) Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy. Compos B Eng 29:277–285. https://doi.org/10.1016/S1359-8368(97)00030-9

Van Den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176:937–971. https://doi.org/10.1016/j.ins.2005.02.003

Venter G, Haftka RT (2009) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 40:65. https://doi.org/10.1007/s00158-009-0380-6

Wang L, Li L-p (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41:947–963. https://doi.org/10.1007/s00158-009-0454-5

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. https://doi.org/10.1007/s00158-008-0238-3

Wang Y, Zhao W, Zhou G, Gao Q, Wang C (2018) Optimization of an auxetic jounce bumper based on Gaussian process metamodel and series hybrid GA-SQP algorithm. Struct Multidiscip Optim 57:2515–2525. https://doi.org/10.1007/s00158-017-1869-z

Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178:3043–3074. https://doi.org/10.1016/j.ins.2008.02.014