Red deer algorithm (RDA): a new nature-inspired meta-heuristic

Article *in* Soft Computing · October 2020

DOI: 10.1007/s00500-020-04812-z

3 authors:


Amir M. Fathollahi-Fard
École de Technologie Supérieure


Mostafa Hajiaghaei-Keshteli
Tecnológico de Monterrey


Reza Tavakkoli-Moghaddam
University of Tehran

Soft Computing
https://doi.org/10.1007/s00500-020-04812-z

METHODOLOGIES AND APPLICATION

Red deer algorithm (RDA): a new nature-inspired meta-heuristic

Amir Mohammad Fathollahi-Fard[1] · Mostafa Hajiaghaei-Keshteli[1] · Reza Tavakkoli-Moghaddam[2,3]

Abstract

Nature has been considered as an inspiration of several recent meta-heuristic algorithms. This paper firstly studies and mimics the behavior of Scottish red deer in order to develop a new nature-inspired algorithm. The main inspiration of this meta-heuristic algorithm is to originate from an unusual mating behavior of Scottish red deer in a breading season. Similar to other population-based meta-heuristics, the red deer algorithm (RDA) starts with an initial population called red deers (RDs). They are divided into two types: hinds and male RDs. Besides, a harem is a group of female RDs. The general steps of this evolutionary algorithm are considered by the competition of male RDs to get the harem with more hinds via roaring and fighting behaviors. By solving 12 benchmark functions and important engineering as well as multi-objective optimization problems, the superiority of the proposed RDA shows in comparison with other well-known and recent meta-heuristics.

Communicated by V. Loia.

✉ Mostafa Hajiaghaei-Keshteli
mostafahaji@mazust.ac.ir

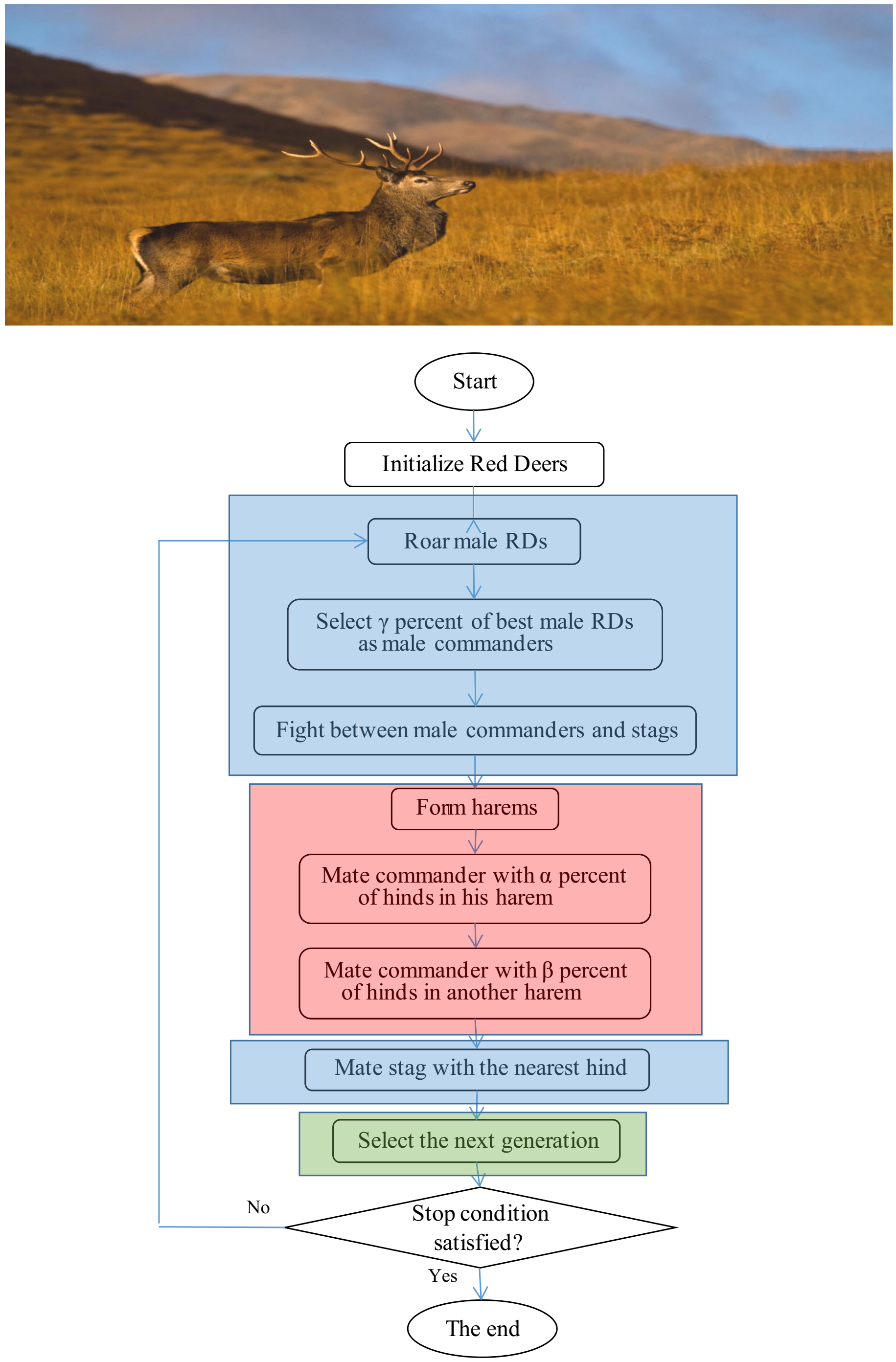
[1] Department of Industrial Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran

[2] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

[3] Arts et Métiers Paris Tech, LCFC, Metz, France

Published online: 10 March 2020

Graphic abstract



Keywords Red deer algorithm · Meta-heuristics · Real-world applications · Multi-objective optimization

1 Introduction

Over the last three decades, an enormous growth in the size and complexity of industrial organization occurred and consequently, the size and importance of the optimization problems increased (Cheraghalipour et al. 2018). Accordingly, managers want to find better ways or solutions, which help them to handle the whole organization. Meta-heuristic algorithms gave easy, cost-efficient and important tools to both researchers and managers to solve this dilemma (Mirjalili et al. 2014). In 1975, John Holland has developed the genetic algorithm (GA) to solve huge and complex problems, for the first time (Holland 1975). Consequently, a numerous meta-heuristics were developed which mostly have been inspired by nature or artificial processes, such as simulated annealing (SA) based on annealing process of metals (Kirkpatrick et al. 1983), ant colony optimization (ACO) inspired by the pheromone trail laying behavior of real ant colonies (Dorigo et al. 1996), particle swarm optimization (PSO) inspired by social behavior of bird flocking or fish schooling (Kennedy and Eberhart 1995), harmony search (HS) inspired by the improvising process of composing a piece of music (Geem et al. 2001), glow-worm swarm optimization (GSO) based on the flashing behavior of glow-worms (Krishnanand and Ghose 2005), imperialist competitive algorithm (ICA) based on imperialistic competition (Atashpaz-Gargari and Lucas 2007) and Keshtel algorithm (KA) based on Keshtels' feeding behavior (Hajiaghaei-Keshteli and Aminnayeri 2013). In 2016, whale optimization algorithm (WOA) is inspired by bubble-net hunting strategy of the social behavior of humpback whales (Mirjalili and Lewis 2016). Recently, Mirjalili et al. (2017) proposed the salp swarm algorithm (SSA) as an inspiration of the swarm behavior of slap' chain. Fathollahi-Fard et al. (2018) proposed a new meta-heuristic algorithm, namely social engineering optimizer (SEO) as an intelligent and single-solution algorithm. It starts with two initial solutions divided into an attacker and a defender and has four main steps and three simple parameters to be tuned.

In summary, each meta-heuristic algorithm has a set of distinct characteristics and advantages in comparison with each other as follows:

- The design of meta-heuristics is often based on some new mathematical theories inspired by natural laws or phenomena (Hussain et al. 2018).
- Meta-heuristics require only fewer mathematical conditions (Mirjalili 2015).
- The computational complexity of recent meta-heuristics is reduced effectively, an approximate and useful solution can be found in a reasonable time (Dong and Zhou 2017).

Regarding the applications of recent meta-heuristics along with their characteristics of search phases (i.e., intensification and diversification), recent years have seen a number of review papers aiming the recent trends of meta-heuristics (Beheshti and Shamsuddin 2013; Hosseini and Al Khaled 2014; Kar 2016; Kaveh 2016; Hussain et al. 2018). Regarding the intensification and diversification phases, the main feature of meta-heuristic techniques is that it does not ensure to find global optima, but a good enough solution instead. The intensification or exploitation phase only focuses on possible good areas in solution space to search carefully around them (Han et al. 2016). On the other hand, diversification or exploration phase wants to explore new areas and let the algorithm search in virgin areas. It also helps the algorithm to escape from local optima. Until now, most of the developed meta-heuristics focus on making a trade-off between intensification and diversification phases.

For example, crossover and mutation operators in GAs, work for diversification and intensification phases, respectively. The trade-off between them is carried by tuning their percentages (Wang et al. 2017). But, the dilemma in GAs is to operate on solutions without any logical plan. The operators just operate by chance and arbitrarily. In tuning step, the type of operators and its percentage are tuned. In SA, the intensification phase is done by neighboring, whereas the diversification phase is carried by changing the temperature and accepting rule (Fathollahi-Fard et al. 2019). Also, the SA gives almost guarantee to find the optimal solution if the cooling process is slow enough. Overall, the fine adjustment in parameters does alter the convergence rate of the optimization process, accordingly (Fathollahi-Fard et al. 2020).

In recently developed algorithms, the authors find the trade-off between two phases and focused on intelligent search. For instance, the ICA carries the diversification by forming different empires and does the exploitation of each empire. Similarly, in the KA, Keshtels search the potential places carefully by an intelligent way and do the exploration phase by moving and landing in the lake. The steps in the KA are developed in a way that the user can employ just one or two operators among the three ones (Hajiaghaei-Keshteli and Fathollahi Fard 2018). Additionally, in the WOA, a bubble-net hunting strategy of whales probes neighborhoods the best solutions and allows other search agents to explore the potential good areas. Like the WOA, the SSA considers the leader of slap' chain as the current best solution and moves the memberships of chain regarding the position of leader. Generally speaking, the theory of recent meta-heuristics is often based on the advantages of each other, which make them similar. Accordingly, the main idea of a new meta-heuristic should consider these advantages of the recent ones in a distinct way.

The main reason behind developing a new meta-heuristic refers to a no free lunch (NFL) theory. Regarding the NFL, there is no meta-heuristic optimization algorithm to solve all optimization problems (Wolpert and Macready 1997). This means that there is always the possibility that a new meta-heuristic algorithm shows a better performance in comparison with the current ones (Fathollahi-Fard and Hajiaghaei-Keshteli 2018). This fact motivates our attempt to contribute a new meta-heuristic algorithm, so-called red deer algorithm (RDA), which shows successful results to solve a set of standard benchmark functions, different engineering and multi-objective optimization problems [e.g., single-machine problem (SMP), traveling salesman problem (TSP), fixed-charge transportation problem (FCTP), vehicle routing problem (VRP) and nurse rostering problem (NRP)] in this study.

In this paper, a new optimization algorithm inspired by red deer mating is developed. The Scottish red deer (Cervus Elaphus Scoticus) is a sub-species of red deer, which lives in the British Isles (Clutton-Brock et al. 1979). In a breeding season, male red deer (RD) roars loudly and repeatedly. Roaring male RD causes to attract hinds. Females prefer a high to a low roaring rate (McComb 1991). Red deer mating patterns usually involve a dozen or more mating attempts before the first successful one. There may be several mating before the stag (mature male) will seek out another mate in his harem. A harem is a group of females, which mate with the head of the harem (male commander). The commander occupies the territory and protects the other hinds in his harem. The more the hinds in the harem, the more the power of the male commander. Like other meta-heuristics, the algorithm starts with an initial population, called RD. This population is divided them into two groups, namely male RD and hinds. After the roaring males, a number of the best ones will be selected as male commanders. The commander mates with a high percentage of females and makes a new generation. This procedure accompanies with a set of mating and fighting behaviors, which make our evolutionary algorithm.

The rest of this study is organized as follows. Section 2 investigates the inspiration of the algorithm originated from RD along with their wonderful mating behavior. Section 3 presents the RDA along with its steps. Section 4 not only tests the proposed algorithm with 12 standard benchmark functions, but also the performance of the proposed algorithm is validated by a set of real-world engineering applications along with a case of a multi-objective optimization problem. Finally, Sect. 5 summarizes the conclusions and future directions of this work.

2 Inspiration

The Scottish red deer (Cervus Elaphus Scoticus) is a sub-species of red deer which lives in the British Isles. Since the end of Pleistocene, red deer populations have existed in Britain especially Scotland. In most cases, they are found on the top of mountain forests, woodland and moorland (Clutton-Brock et al. 1979). The population of deer is divided into two types: males named stags and females called hinds.

One of the main amazing behaviors of this animal happens in a breeding season. Accordingly, males roar loudly and frequently. Hinds prefer a high to a low roaring rate, but not low-pitched roars. A roaring rate is positively associated with both reproductive success and fighting ability. Female choosing the males with high roaring rates may either reflect a selection pressure on females to mate with males that are successful in contests with other males or show that females mate with males that are easy to locate. The competition between the males to access and to mate can result in the same patterns of associative mating as the female choice. Regarding this fact, a number of males can be more successful than others. After absorbing a group of hinds by males, a harem will be formed. The strongest male will be the commander of the harem. Commanders protect the hinds in their harem and their territory. They do roaring so only during the autumn breeding season when they gather and defend harems (McComb 1991).

The competition of males to be a commander is considered by a predictable fight. Regarding the stags and commanders, one male approaches another and stands in a visible position, roaring toward him. The approacher responds and tows usually exchange roars for several minutes. If the contest enters a second stage, the approacher advances and moves to meet him. In the majority of cases, they walk side by side, usually at right angles to the direction from which the approacher has come. During the walk, either stag may invite contact by turning to face its opponent and lowering its antlers (Thouless and Guinness 1986). Accordingly, each animal attempts to turn his opponent so that the latter is below him on the slope and fighting stags will circle rapidly. Finally, the winner will be the commander of a harem and gets their hinds. This fact leads the rutting activities, which motivated several studies in the three last decades (e.g., Clutton-Brock et al. 1979; Thouless and Guinness 1986; McComb 1991; Krause et al. 2015; Blank et al. 2017)).

Another main characteristic of the red deer is their rutting behavior. The course of the rut is highly predictable from year to year. In mid-September, mature stags (over 5 years old), which have spent the previous

10 months in bachelor groups, become intolerant of each other and individuals move to traditional rutting areas where they collect and defend the groups of hinds (harems). Rutting activities of stag's peak during the same period and decline after October 20 (Clutton-Brock et al. 1979). One of the main scholars who studied Scottish red deer is Clutton-Brock. He eventually comes to the conclusion that the commander of the harem can be a maximum of 24 born father, for only a short time to be at the peak of their ability to stay (Thouless and Guinness 1986). Generally speaking, this study considers the roaring, fighting and mating behaviors of this animal. The inspiration for the above-mentioned activities motivates our attempt to consider these inspirations to propose a new optimizer.

3 Red deer algorithm (RDA)

Similar to other meta-heuristics, the RDA starts with an initial random population that is the counterpart to RDs. A number of the best RDs among the population is selected and named the "male RD" and the rest of them are called "hinds." First of all, the male RD should roar. Based on the power of a roaring phase, they are divided into two groups (i.e., commanders and stags). After that commanders and stags of each harem fight together in order to own their harem. Besides, harems are formed by commanders. The number of hinds in harems is directly related to the commanders' abilities in roaring and fighting process. Consequently, commanders mate with a number of hinds in harems. Note that the other males (i.e., stags) mate with the nearest hind without considering the limitation of the harem.

Generally, the mentioned steps of the RDA are designed in a way to consider the exploitation and exploration phases satisfactorily. The user can tune the phases regarding the used parameters and mathematical formulation. Accordingly, the roaring of male RD is the counterpart of local search in solution space to improve the exploitation properties. Similarly, the fighting between commanders and stags is also considered as local search; however, in this process, we only accept the better-observed solutions. This step mainly considered the exploitation characteristics as well. After that harems are formed and allocated to the commanders according to their power. This step helps the algorithm to do the exploration phase. Accordingly, the commander of a harem mates with a percentage of hinds in his harem and also with a percentage of hinds in another harem. These stages have also been improved the exploration properties. Note that regarding the breading season, all stags should mate with the nearest hind, that is, a stag mate with the hind with the minimum distance without considering of the limitation of the harem. This step also focuses on both exploration and exploitation phases, simultaneously. Another main phase of the RDA is the mating process, which leads to generating offspring of RDs. This phase is the counterpart of making new solutions in solution space. Finally, the next generation of the algorithm is done by giving a chance for weak solutions regarding the classification of the algorithm as an evolutionary one.

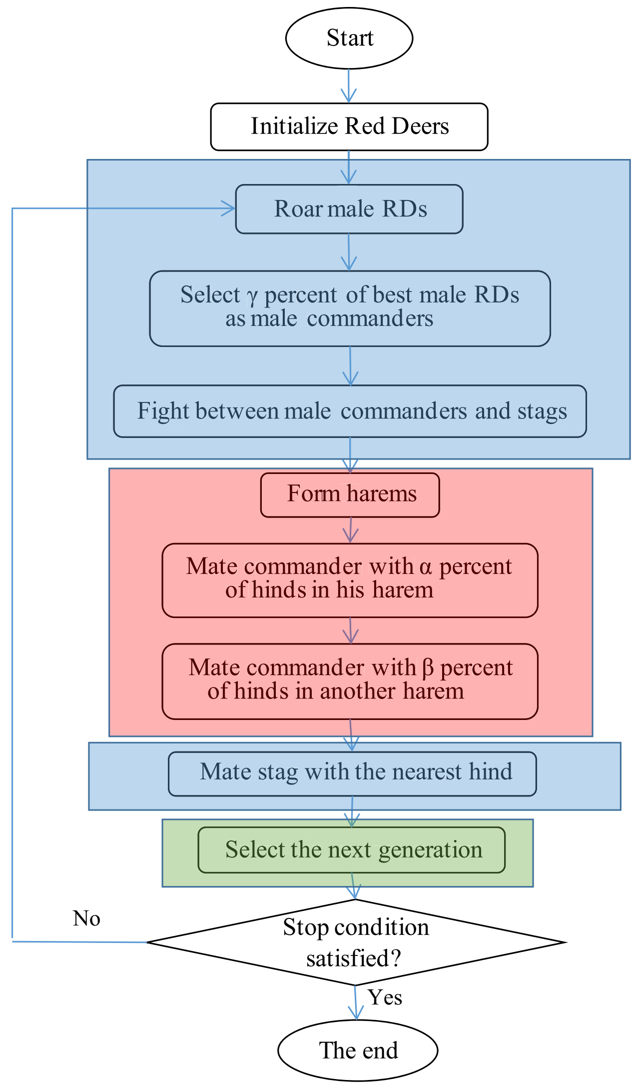


Fig. 1 Flowchart of the proposed RDA

Figure 1 shows the flowchart of the RDA in order to achieve more details about its structure. While the blue color boxes show the intensification phase, the red color boxes show the diversification phase. Due to the evolutionary concept of the algorithm, the green color box shows the part of the algorithm to escape from the local optimum. In conclusion, it seems that the tools of tuning the exploration and exploitation phases are available for a user to manipulate these steps regarding the characteristics of the considered problem. Most notably, the considered problem, for the general explanation of the proposed algorithm in this section, is an optimization model with continuous variables and no constraint. For example, the reader can refer to the standard benchmarks in Sect. 4.1.

Note that due to the novelty of the RDA, no existing study has treated a similar algorithm inspired by RD's mating. That is why there is no any mathematical formulation that is available for this research. In the following section, a mathematical theory regarding the steps of the RDA is developed. This theory is considered to solve a minimization problem. Regarding this theory, there exist three important parameters to control the exploration and exploitation phases. Notably, α and β control the diversification phase, while γ control the intensification phase. All of these parameters are between zero and one.

3.1 Generate an initial red deer

The goal of optimization is to find a near-optimal or global solution in terms of the variables of the problem. Here, we shape an array of variable values to be optimized. For example in GA, the terminology of this array is called "chromosome," but here in the RDA, this term, "Red Deer" is used for this array. Note that a "Red Deer" corresponds to a feasible solution X inside the solution space. Therefore, red deer is a counterpart of a solution. The dimensionality of this solution X is Nvar. Accordingly, in an Nvar $-$ dimensional optimization problem, a red deer is a 1 $\times$ Nvar array. This array is defined by:

$$\text{Red Deer} = [X1,\, X2,\, X3, \ldots, X_{N_{\text{var}}}]. \tag{1}$$

Notably, Eq. (1) shows the components or each dimension of X. Also, the function value can be evaluated for each RD as follows:

$$\text{Value} = f(\text{Red Deer}) = f(X1,\, X2,\, X3,\, \ldots, X_{N_{var}}). \tag{2}$$

To start the algorithm, we generate the initial population of size Npop. We select a set of the best RD to Nmale and the rest of them to Nhind (Nhind = Npop- Nmale). Note that the number of Nmale shows the elitist criterion of the algorithm. By another point of view, the number of Nmale maintains the intensification properties, while Nhind considers the diversification phase of the algorithm.

3.2 Roar male RDs

In this step, male RD is trying to increase their grace by roaring. Therefore, as happened in nature, the roaring process may be successful or faces with failure. Notably, male RDs are the good solutions in this algorithm. Regarding the solution space, we find the neighbors of the male RD, and if the objective functions of the neighbors are better than the male RD, we replace them with the prior ones. In fact, we permit every male RD to change their position. To update the position of males, the following equation is proposed:

$$\text{male}_{\text{new}} = \begin{cases} \text{male}_{\text{old}} + a_1 \times ((UB - LB) * a_2) + LB), & \text{if} a_3 \geq 0.5 \\ \text{male}_{\text{old}} - a_1 \times ((UB - LB) * a_2) + LB), & \text{if} a_3 < 0.5 \end{cases} \tag{3}$$

With regard to generate a feasible neighborhood solution of male, UB and LB limit the search space. They are the upper and lower bounds of search space, respectively. Note that male_{old} is the current position of male RD, and male_{new} is its updated position. Regarding the randomization of the roaring process in nature, a_1, a_2 and a_3 are generated randomly by a uniform distribution between zero and one. To create a link between this equation and the roaring process in nature, note that when a male RD roars, it tries to expand its territory. Therefore, it moves randomly. To illustrate a simple example of the roaring process, Fig. 2 shows two cases based on the solution space to show the effects of the roaring separately. Note that A and B happen generally in the roaring phase. In case A, the new position is accepted because the objective fitness (OF) of a solution is better than the prior one. However, in case B, we cannot accept the new solution. It should be noted that y-axis represents the OF and x-axis specifies the position of males in this example.

3.3 Select γ percent of the best male RD as male commanders

In nature, there is so much difference between male RDs. Some of them are more powerful, more attractive, or more successful in territory expanding than the others.

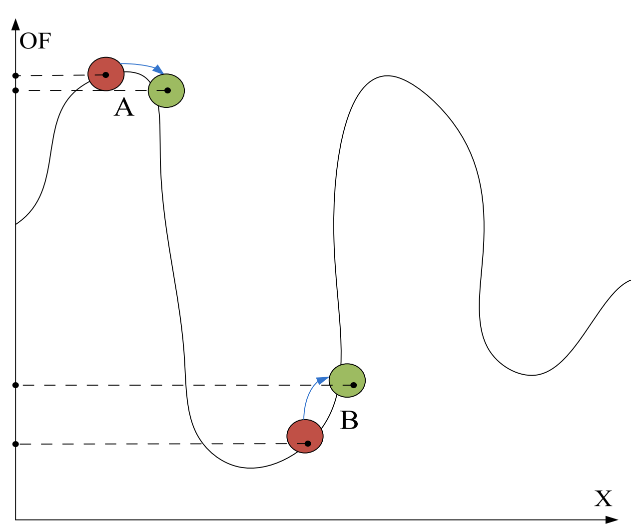


Fig. 2 Roaring process

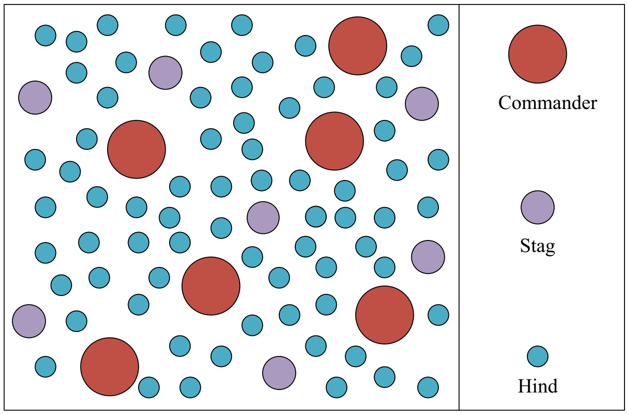


Fig. 3 Population of RD

Accordingly, RD is divided into two types, namely commanders and stags. The number of commander males is computed as follows:

$$N_{\mathrm{Com}} = \mathrm{round}\{\gamma . N_{\mathrm{male}}\} \quad (4)$$

where N_{Com} is the number of males, who are the commanders of harems. Note that γ is the initial value of the algorithm model. Its range of values is between zero and one. Finally, the number of stags is calculated by:

$$N_{\mathrm{stag}} = N_{\mathrm{male}} - N_{\mathrm{Com}} \quad (5)$$

where N_{stag} is the number of stags in regard to the population of males. Here, Fig. 3 shows the population of RD by considering all commanders, stags and hinds.

3.4 Fight between male commanders and stags

We let each commander fight with stags randomly. Regarding the solution space, we let a commander and a stag approach to each other. So, we obtain two new solutions and replace the commander by the better solution (i.e., having a better OF among four solutions, namely commander, stag and two new solutions obtained after approaching). Regarding the fighting process, two mathematical formulas are provided by:

$$\mathrm{New\,1} = \frac{(\mathrm{Com} + \mathrm{Stag})}{2} + b_1 \times ((UB - LB) * b_2) + LB) \quad (6)$$

$$\mathrm{New2} = \frac{(\mathrm{Com} + \mathrm{Stag})}{2} - b_1 \times ((UB - LB) * b_2) + LB) \quad (7)$$

where New1 and New2 are the two new solutions generated by the fighting process. Com and Stag represent the symbol of commanders and stags, respectively. Regarding the feasibility of new solutions, UB and LB limit the upper bound and lower bound of the search space. Due to the randomization of the fighting process, b_1 and b_2 are generated by a uniform distribution function between zero and one. Considering four solutions including Com, Stag, New1 and New2, only the best one regarding the OF will be selected. To show the connection of supposed formulas and the nature of male RDs, note that in the fighting, a commander and a stag are approached to each other. One of them will win this match and another will lose. Therefore, two new solutions are generated. One selects the winner and another is looser. Figure 4 depicts this procedure in a simple example. As shown in this figure, we have two new solutions. The OF of solution New1 is better than all solutions. The new commander is New1 accordingly.

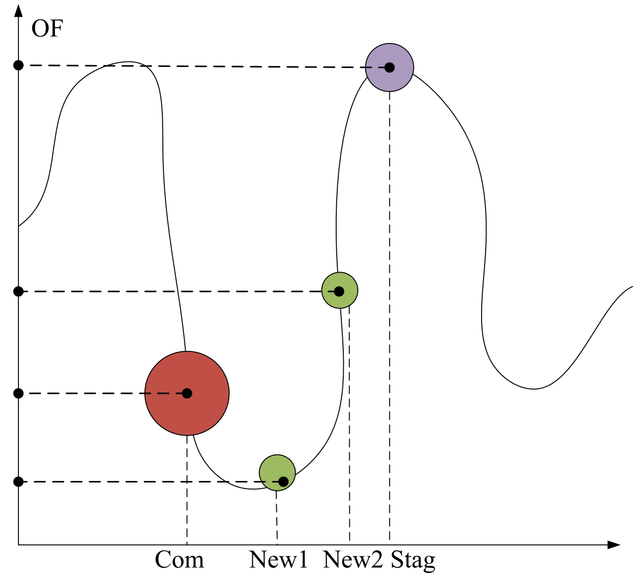


Fig. 4 Fighting process

3.5 Form harems

Here, we form the harems. A harem is a group of hinds in which a male commander seized them. The number of hinds in harems depends on the power of male commanders. We define the power of the commander as its OF. To form the harems, we divide hinds among commanders proportionally by:

$$V_n = v_n - \max_i\{v_i\} \quad (8)$$

where v_n refers to the power of the nth commander (i.e., its OF) and V_n is its normalized value. To calculate the normalized power of commanders, the following equation is provided to achieve this goal.

$$P_n = \left| \frac{V_n}{\sum_{i=1}^{N_{\mathrm{Com}}} V_i} \right| \quad (9)$$

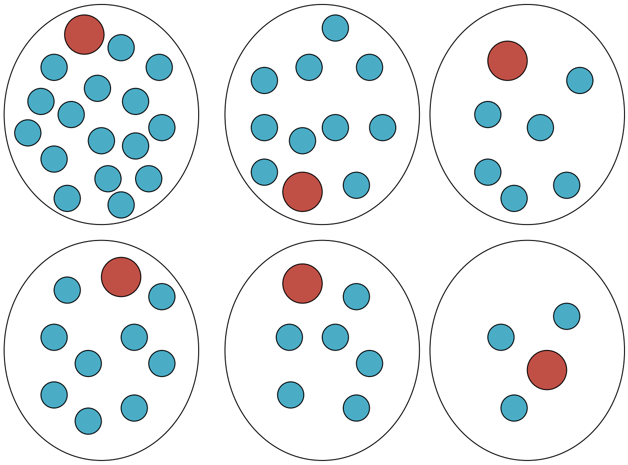


Fig. 5 Harems

From another point of view, the normalized power of a male commander is the portion of hinds that should be possessed by that male. Then, the number of hinds of a harem will be as follows:

$$N.\mathrm{harem}_n = \mathrm{round}\{P_n.N_{\mathrm{hind}}\} \tag{10}$$

where $N.\mathrm{harem}_n$ is the number of hinds in nth harem and Nhind is the number of all hinds. To divide the hinds to each male commander, we randomly choose $N.\mathrm{harem}_n$ of the hinds. These hinds along with the male will form the nth harem. In overall, the commander, which has a better fitness value, will get more hinds. Figure 5 shows the harems in this generation, in which the radius of each sphere in proportion refers to its fitness value.

3.6 Mate commander of a harem with α percent of hinds in his harem

Like other species in nature, deer mates with each other. This action is done by a commander, and α percent of hinds in his harem is the parents.

$$N.\mathrm{harem}_n^{\mathrm{mate}} = \mathrm{round}\{\alpha.N.\mathrm{harem}_n\} \tag{11}$$

where $N.\mathrm{harem}_n^{\mathrm{mate}}$ is the number of hinds of the nth harem that mate with their commander. In regard to the solution space, we randomly choose $N.\mathrm{harem}_n^{mate}$ of the $N.\mathrm{harem}_n$. Note that α is an initial parameter value for the model of the RDA. Its range value is between zero and one. Generally, the mating process is formulated by:

$$\mathrm{offs} = \frac{(\mathrm{Com} + \mathrm{Hind})}{2} + (UB - LB) \times c \tag{12}$$

where Com and Hind are the symbols of commanders and hinds, respectively. Offs is a new solution. Note that UB and LB are the upper and lower bounds, respectively. Also, c is generated randomly by a uniform distribution function between zero and one.

3.7 Mate commander of a harem with β percent of hinds in another harem

We select a harem randomly (name it k) and let male commander mate with β percent of hinds in this harem. In fact, the commander attack to another harem to expand his territory. Notably, β is an initial parameter value for the algorithm model. The range value of this parameter is between zero and one. The number of hinds in the harem, which mate with the commander, is computed by:

$$N.\mathrm{harem}_k^{\mathrm{mate}} = \mathrm{round}\{\beta.N.\mathrm{harem}_k\} \tag{13}$$

where $N.\mathrm{harem}_k^{\mathrm{mate}}$ is the number of hinds in the k-th harem, which mate with the commander. Note that the procedure of mating is done by considering Eq. (12).

3.8 Mate stag with the nearest hind

In this step, each stag mate with its closest hind. In breeding season, the male RD prefers to follow the handy hind. This hind may be his favorite hind among all hinds without consideration of harem territories. We let each stag mate with the nearest hind. To find the nearest hind, the distance between a stag and all hinds in J-dimension space should be calculated by:

$$d_i = \left(\sum_{j \in J}\left(\mathrm{stag}_j - \mathrm{hind}_j^i\right)^2\right)^{1/2} \tag{14}$$

where d_i is the distance between the i-th hind and a stag. Accordingly, the minimum value in this matrix represents the hind selected. The action after selecting a hind is the mating process. Note that this procedure will be tackled by the mathematical formula presented in Eq. (12). Similarly, instead of a commander, a stag should be considered in this formula.

3.9 Select the next generation

To select the next generation, two different strategies have been followed. In the first one, we keep all the male RD, all commander and stags (i.e., a percent of the best solutions out of all solutions). The second strategy refers to the remainder of the population in the next generation. We choose hinds out of all hinds and offspring generated by mating process regarding the fitness value by using the fitness tournament or roulette wheel mechanism. Since these mechanisms are well-known, the related mathematical formulation has not been provided. Accordingly, the

Fig. 6 Pseudo-code of the algorithm

```
Initialize the Red Deer population.
Calculate the fitness and sort them and form the hinds (N_hind) and male RDs (N_male).
X*=the best solution.
T1=clock;
While (t< maximum time of simulation)
   for each male RD
    Roar the male (Eq. 3).
    Update the position if better than the prior ones.
   end for
   Sort the males and also form the stags and the commanders (Eqs. 4 and 5).
   for each male commander
   Fight between male commanders and stags (Eqs. 6 and 7).
   Update the position of male commanders and stags
  end for
  Form harems (Eqs. 8, 9 and 10).
   for each male commander
   (Eq. 11)
   Mate a male commander with the selected hinds of his harem randomly (Eq. 12).
   Select a harem randomly and name it k.
   (Eq. 13)
   Mate a male commander with some of the selected hinds of the harem (Eq. 12).
   end for
   for each stag
  Calculate the distance between the stag and all hinds and select the nearest hind (Eq. 14).
  Mate stag with the selected hind (Eq. 12).
  end for
  Select the next generation with the roulette wheel selection.
   Update X* if there is a better solution.
   T2=clock;
   t=T2-T1;
end while
Return X*
```

interested readers can refer to Miller and Goldberg (1995) and Lipowski and Lipowska (2012);

3.10 Stopping condition

The stopping condition may be the number of iteration, the quality of the best solution ever found or a time interval. In conclusion, the main steps of the algorithm are summarized as shown in Fig. 6.

4 Experimental results

To prove the theoretical claimed pointed out in Sect. 3, a set of experiments is conducted. Firstly, 12 standard mathematical functions benchmarked from the literature (Ghorbani and Babaei 2014; Fathollahi-Fard et al. 2018) are used. Secondly, the engineering applications of the RDA are proven in a set of the computational engineering design issues related to scheduling, transportation and routing problems. The recent trends in meta-heuristics are stated that a number of new optimizers (e.g., ICA, KA, GWO and WOA, SEO) show a robust performance for the engineering design problems (Atashpaz-Gargari and Lucas 2007; Hajiaghaei-Keshteli and Aminnayeri 2013; Mirjalili et al. 2014; Mirjalili and Lewis 2016, Fathollahi-Fard et al. 2018). Generally speaking, it is formal to study the performance of a novel meta-heuristic algorithm with a number of benchmark functions. Contrary to the previous studies (Karaboga and Akay 2009; Ghorbani and Babaei 2014; Mirjalili 2015; Cheraghalipour et al. 2018), this study considers different practical engineering problems and a multi-objective optimization case study to reveal the efficiency and effectiveness of this new optimization algorithm.

4.1 Standard functions

Similar to other meta-heuristic papers, this paper utilizes a set of standard functions to evaluate the proposed RDA and compares with other popular and recent meta-heuristics. The literature reports that there are more than 50 standard functions (Karaboga and Akay 2009; He et al. 2009; Rao and Patel 2013; Mirjalili et al. 2014, 2017). Here, a set of standard functions adopted from CEC2005 (Liang and Suganthan 2005) and CEC2010 (Zhao et al. 2010) has been benchmarked from Ghorbani and Babaei (2014) and Fathollahi-Fard et al. (2018). These problems are numbered as P1 to P12. All of them are minimization problems. The optimum value for all of them is also equal to zero. The main reason behind selecting these functions is that each of them has a set of special characteristics to ease the evaluation of meta-heuristics. Accordingly, it is tried to assess different cases of the standard functions to explore the capabilities of the RDA. It is evident that P2, P3, P4 and P5 are unimodal and non-separable, in which P8, P10, P11 and P12 functions have multimodal and non-separable characteristics. As such, P5, P1 and P7 are unimodal and separable. Also, P9 is multimodal and separable. In overall, the details about these functions are given in Table 1.

The proposed RDA not only compares with a number of well-known meta-heuristics, but also the number of states of art optimizers is considered in this regard. The traditional meta-heuristics (i.e., GA and SA, particle swarm optimization (PSO) as one of the best techniques in swarm-based algorithms, artificial bee colony algorithm (ABC) (Karaboga and Akay 2009) and also a number of recent successful meta-heuristics (e.g., imperialist competitive algorithm (ICA) and firefly algorithm (FA) (Yang 2010)) are employed to solve the standard problems in order to compare with the RDA. Since these algorithms employed

Table 1 Benchmark functions (Ghorbani and Babaei 2014)

No. of problems	Function	Formulation	C	Range	D
P1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	U,S	$[-100, 100]$	n = 30, 100
P2	Schewfel 2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	U,N	$[-10, 10]$	n = 30, 100
P3	Schwefel 1.2	$f(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	U,N	$[-100, 100]$	n = 30, 100
P4	Schwefel 2.21	$f(x) = \max\{\lvert x_i \rvert, i = 1, \ldots, n\}$	U,N	$[-100, 100]$	n = 30, 100
P5	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	U,N	$[-30, 30]$	n = 30, 100
P6	Step	$f(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	U,S	$[-100, 100]$	n = 30, 100
P7	Quratic	$f(x) = \sum_{i=1}^{n} i \times x_i^4 + random[0, 1)$	U,S	$[-1.28, 1.28]$	n = 30, 100
P8	Greiwank	$f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	M,N	$[-600, 600]$	n = 30, 100
P9	Rastrigin	$f(x) = x_i^2 - 10\cos(2\pi x_i) + 10$	M,S	$[-5.12, 5.12]$	n = 30, 100
P10	Ackley	$f(x) = -20\exp(-0.2\sqrt{\sum_{i=1}^{n} x_i^2 \frac{1}{n}}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) +20 + e$	M,N	$[-32, 32]$	n = 30, 100
P11	Penalized function1	$f(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \times [1 + 10\sin^2(\pi y_{i+1})]\} + \sum_{i=1}^{n} u(x_i, 10, 1000, 4)$ s.t. $y_i = 1 + (x_i + 1)/4$, $u(x_i, a, k, m) = \left\{ \begin{array}{ll} k(x_i - a)^m & i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < a \end{array} \right\}$	M,N	$[-50, 50]$	n = 30, 100
P12	Penalized function2	$f(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] +(x_n - 1)^2 \times [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	M,N	$[-50, 50]$	n = 30, 100

C Characteristic, *U* Unimodal, *M* Multimodal, *S* Separable, *N* Non-Separable, *D* Dimension

Table 2 Tuned parameters of the algorithms

Algorithms	Parameters
GA	Initial population = 100, Maximum time of simulation = 10 (s), Mutation rate = 0.2, Mutation percentage = 0.05, Crossover rate = 0.5, Crossover percentage = 0.75
SA	Maximum time of simulation = 10 (s), Number of Sub-iteration = 50, Initial Temperature = 10, Rate of reduction = 0.99, Rate of creating a neighbor = 0.05
PSO	Initial population = 100, Maximum time of simulation = 10 (s), Personal learning coefficient = 2, Global learning coefficient = 2, Inertia rate damping ratio = 0.99, Initial weight = 1
ABC	Initial population = 100, Maximum time of simulation = 10 (s), Abandonment limit parameter: L = round(0.5*nVar*nPop), Acceleration coefficient upper bound = 0.3
ICA	Initial population = 100, Maximum time of simulation = 10 (s), Number of empires = 8, Number of colonies = Initial population-Number of empires, Assimilation coefficient = 0.2, Revolution probability = 0.1, Rate of revolution = 0.05, Colonies mean cost coefficient = 0.1
FA	Initial population = 100, Maximum time of simulation = 10 (s), Attraction coefficient base of value = 2, Light absorption coefficient = 1, Mutation coefficient = 0.2, Mutation Coefficient Damping Ratio = 0.99
RDA	Initial population = 100, Maximum time of simulation = 10 (s), Number of males = 15, Number of hinds = Initial population-Number of males, α = 0.9, β = 0.4, γ = 0.7

and proposed have several controlling parameters, the best value should be chosen to provide an unbiased comparison. Accordingly, the algorithms' parameters are tuned and the best values are given in Table 2. Due to the page limitation, the procedures of tuning are not presented.

To have a fair comparison between the algorithms, the comparison should be based on an equal number of fitness evaluations or equal process time (Kar 2016; Kaveh 2016; Hussain et al. 2018). Accordingly, for all methods in this study, the stopping condition is time interval equaled to 10 s regarding each standard problem. The main reason behind of this criterion refers to Mernik et al. (2015) and Draa (2015).

Also, the algorithms are run for 30 times. Accordingly, the best, worst, mean values and standard deviation of algorithms during 30 run times are given in Table 3. For both low (D = 30)- and high (D = 100)-dimensional models, the illustrated algorithms are compared to each other. From this table, the rank of algorithms in each problem is given. The methodology to rank the algorithms is based on the weights for the criteria. In each criterion, the rank of the algorithm is considered. For the best solution, the weight is 0.5 and the standard deviation is 0.3. For both best and worst criteria, the weight is 0.1. As obviously indicated, the summation of the weights equals to 1. In overall, the RDA reaches the best rank in most of the problems by the average of 1.33.

To analyze the behavior of algorithms in each function, Fig. 7 aims to compare the convergence rates of meta-heuristics with each other. As can be seen, x-axis shows the time interval based on seconds and y-axis shows the best fitness evaluations noted as the best cost found by algorithms. Note that the best run of algorithms has been utilized accordingly. Also, the low dimensional (D = 30) has been considered to achieve this goal. From this figure, it is evident that except P1, P5 and P8, the proposed RDA shows the best convergence behavior in this regard.

To conclude, a set of statistical analyses has been conducted to assess the performance of meta-heuristics. Here, means plot and least significant differences (LSD) intervals at 95% confidence level regarding both low- and high-dimensional functions are shown in Fig. 8. Generally, all statistical outputs state that RDA shows a robust behavior in most of the test problems.

More computationally, some sensitivity analyses are performed to evaluate the efficiency of the algorithms in different cases. Regarding the theoretical claimed in Sect. 3, the proposed RDA has three main parameters (i.e., α, β and γ). The ranges of these parameters are between zero and one. Here, it is aimed to consider a number of suitable values for each parameter to ease the tuning process of the RDA for the user. Nine random experiments are designed to evaluate the fitness evaluations of the RDA as given in Table 4. It should be noted that P1 is selected to these analyses in a low-dimensional model (D = 30). Also, to have a fair analysis, the number of initial population is equal to 100 and number of males is equal to 15. The stopping criterion is the maximum number of iterations equaled to 200 as well. The behavior of the RDA in all cases is shown in Fig. 9. It can be concluded that the effect of each parameter is significant. By increasing the amount of α and β, the efficiency of the RDA is increased as well.

The last sensitivity analysis is related to the run time of the algorithm. As mentioned earlier, the optimization algorithms have two phases: exploration and exploitation. The exploration phase is work to explore feasible areas in

Table 3 Final outputs of algorithms in benchmark functions during 30 run times (*B* best, *W* worst, *M* median, *SD* = standard deviation, *D* dimension, *R* rank)

Function		D	GA	SA	PSO	ABC	ICA	FA	RDA
P1	30	W	9.58E−02	2.78E−01	6.37E−03	8.37E−01	1.37E−02	4.93E−04	3.52E−04
		M	8.16E−07	2.15E−02	5.78E−05	2.51E−02	5.36E−03	6.91E−05	1.73E−06
		B	2.19E−08	4.28E−03	1.89E−06	3.16E−04	2.51E−04	3.78E−08	2.86E−11
		SD	0.004783	0.050924	0.000972	0.46176	0.393635	0.000534	0.000136
		R	2	7	5	6	4	3	1
	100	W	1.22	4.78	3.65E−02	2.42	5.84	2.76E−01	4.28E−03
		M	4.67E−03	0.8514	8.35E−01	0.7534	4.32	5.47E−04	5.95E−04
		B	6.18E−04	0.0251	4.28E−03	0.4627	2.17	2.81E−05	1.53E−05
		SD	0.01854	1.0894	0.06854	1.0325	3.2817	0.05894	0.07634
		R	3	5	4	6	7	2	1
P2	30	W	2.57E−03	5.16E−03	5.84E−06	1.25E−04	1.87E−04	2.35E−04	2.64E−04
		M	6.82E−07	5.32E−05	5.73E−09	2.68E−05	2.17E−06	3.18E−07	3.85E−08
		B	2.18E−09	3.11E−06	4.35E−10	3.19E−06	3.12E−07	2.57E−08	1.25E−11
		SD	3.17E−02	4.28E−02	3.47E−04	2.64E−03	8.53E−03	4.81E−03	5.73E−04
		R	3	6	2	7	5	4	1
	100	W	2.81E−01	5.18E−02	5.38E−04	3.18E−02	5.15E−02	1.75E−02	3.27E−01
		M	2.57E−05	7.48E−03	8.13E−07	7.29E−03	3.59E−04	3.21E−04	4.38E−06
		B	6.36E−07	1.94E−04	5.42E−08	5.42E−05	2.88E−05	2.72E−06	5.74E−08
		SD	5.28E−02	6.53E−01	1.82E−03	4.18E−01	3.15E−02	3.22E−03	3.16E−03
		R	3	7	1	6	5	4	2
P3	30	W	2.16E−02	4.17E−04	2.54E−06	3.17E−04	2.16E−03	3.15E−02	5.93E−03
		M	3.63E−09	2.64E−10	3.72E−07	2.62E−05	3.71E−06	4.71E−06	8.15E−07
		B	1.22E−11	3.14E−12	5.01E−10	4.83E−08	4.85E−08	1.85E−09	3.15E−12
		SD	7.52E−03	8.53E−03	2.17E−02	5.77E−03	3.15E−03	2.71E−02	8.52E−03
		R	3	1	4	7	6	5	2
	100	W	2.45E−01	2.51E−03	7.18E−05	6.43E−02	4.36E−02	2.58E−01	4.26E−02
		M	3.81E−08	2.78E−09	3.18E−06	3.17E−04	1.27E−04	2.83E−05	4.27E−06
		B	2.51E−09	3.22E−10	5.17E−09	1.68E−06	3.19E−06	7.12E−07	1.29E−09
		SD	4.87E−01	2.56E−02	2.15E−04	1.25E−01	1.78E−01	2.16E−01	2.81E−02
		R	3	2	4	6	7	5	1
P4	30	W	78.93	8.18	22.91	9.54	6.19	7.52	5.01
		M	10.53	1.2794	5.73	0.8659	0.8134	0.5894	4.82E−04
		B	3.8217	2.81E−02	3.91E−02	1.29E−02	5.28E−03	2.71E−05	2.77E−06
		SD	9.4521	3.62	8.61	4.28	3.994	2.71E−00	3.82E−01
		R	7	6	5	4	3	2	1
	100	W	95.62	10.64	34.17	15.85	12.95	11.37	7.83
		M	11.84	4.26	7.95	3.1854	5.8728	0.8623	3.61E−02
		B	6.842	3.81E−01	2.18	1.6809	0.7854	3.72E−03	5.87E−05
		SD	24.81	3.72	2.0823	1.7905	2.6847	3.78	1.9923
		R	7	3	6	5	4	2	1
P5	30	W	98.47	95.27	99.15	114.46	119.76	102.54	87.54
		M	28.65	25.94	44.75	98.72	91.45	82.18	29.58
		B	19.75	18.75	32.15	30.99	31.58	30.85	24.61
		SD	44.87	30.98	486.74	1845.62	6254.91	1392.57	196.74
		R	6	1	5	4	7	3	2
	100	W	123.25	105.89	114.95	156.75	142.68	116.75	95.47
		M	32.67	29.81	52.68	115.63	99.74	86.53	31.47
		B	20.64	19.41	36.27	32.16	34.82	33.81	26.91

Table 3 continued

Function		D	GA	SA	PSO	ABC	ICA	FA	RDA
		SD	56.14	32.8162	549.62	1974.25	754.38	1473.89	215.48
		R	2	1	7	4	6	5	3
P6	30	W	3	3	1	3	7	0	0
		M	0.0008	1.89E−03	8.24E−18	2.35E−09	2.57E−01	0	0
		B	0	0	0	0	0	0	0
		SD	0	6.045955	7.348469	62.7283	18.54724	5.95219	1.06066
		R	2	4	5	7	6	3	1
	100	W	24	21	57	87	23	13	21
		M	2.65E−02	1.75E−01	2.64E−04	2.19E−05	2.16E−00	1.75E−15	3.72E−10
		B	1	1	1	1	3	0	1
		SD	1.65E−01	2.85E−02	1.87E−01	3.28E−02	1.57E−01	3.82E−02	3.16E−02
		R	6	5	7	3	4	1	2
P7	30	W	13.87	14.56	13.87	13.69	12.54	9.76	9.81
		M	7.89	10.25	10.57	10.86	10.25	8.46	8.26
		B	6.54	7.81	7.89	6.47	8.92	7.15	6.38
		SD	1.584	2.71	1.47	1.5843	1.3672	1.91	2.685
		R	3	5	6	2	7	4	1
	100	W	18.95	17.24	15.49	15.86	14.74	12.85	10.86
		M	8.918	11.1288	11.92	11.63	10.54	9.86	9.24
		B	8.25	8.25	9.63	8.25	8.18	8.82	8.10
		SD	1.67	2.86	1.75	1.79	1.64	2.56	2.89
		R	3	4	7	5	2	6	1
P8	30	W	1.86E−04	1.47E−02	1.68E−03	5.42E−02	1.67E−02	2.72E−04	3.81E−03
		M	3.72E−05	1.46E−04	5.37E−06	1.48E−04	5.78E−04	6.24E−07	5.87E−05
		B	2.76E−07	5.81E−06	2.14E−08	3.27E−09	3.11E−08	6.82E−11	5.46E−09
		SD	2.81E−03	8.26E−02	3.81E−02	2.64E−02	4.18E−02	1.82E−03	7.53E−04
		R	6	7	4	3	5	1	2
	100	W	1.53E−02	8.53E−01	2.84E−02	4.17E−01	5.83E−01	8.42E−03	5.17E−02
		M	1.23E−04	3.32E−06	6.54E−06	1.54E−02	4.83E−02	1.28E−04	4.37E−07
		B	5.92E−04	1.48e-08	4.79E−07	5.38E−05	5.92E−04	6.19E−05	8.29E−08
		SD	0.000564	1.03E−05	0.000109	0.284115	0.088928	0.036815	0.000181
		R	6	1	3	4	7	5	2
P9	30	W	43.82	42.81	39.81	38.29	43.81	35.92	32.18
		M	18.36	30.92	22.59	28.11	17.85	29.32	16.931
		B	17.4938	17.4938	12.4821	22.5918	16.3917	16.3917	8.5474
		SD	9.461557	4.940765	12.68426	10.67775	0.182485	25.73	20.563
		R	6	5	2	7	4	3	1
	100	W	52.86	59.15	48.72	51.23	52.19	43.64	42.58
		M	28.71	34.82	27.91	30.82	24.71	32.81	19.76
		B	16.83	20.73	18.91	18.16	12.76	14.83	11.56
		SD	8.745	9.15	7.84	6.32	8.182	7.93	6.15
		R	4	6	7	5	2	3	1
P10	30	W	3.82E−02	0.8126	6.81E−01	2.71	1.28	3.88E−03	8.53E−04
		M	6.32E−04	1.86E−02	4.82E−02	7.81E−02	4.82E−02	5.82E−05	6.26E−07
		B	1.54E−06	3.66E−04	5.77E−04	2.97E−02	8.31E−02	5.88E−08	1.43E−09
		SD	2.78E−02	1.66E−01	3.22E−02	4.37E−01	2.36E−01	6.82E−02	3.82E−03
		R	2	4	5	7	6	3	1
	100	W	2.867	7.84	2.78	6.34	3.91	0.684	0.523

Table 3 continued

Function		D	GA	SA	PSO	ABC	ICA	FA	RDA
		M	2.91E−02	3.82E−01	3.82E−02	4.71E−02	5.81E−02	1.57E−04	5.81E−06
		B	2.84E−04	1.94E−02	8.24E−03	3.66E−04	2.55E−05	9.51E−06	8.94E−07
		SD	3.71E−01	4.14E−01	5.11E−02	3.81E−01	2.48E−02	1.753-03	2.51E−04
		R	5	7	6	4	3	2	1
P11	30	W	0.85	0.85	0.81	0.43	0.875	0.36	0.89
		M	5.92E−03	2.81E−03	5.72E−03	6.81E−03	1.92E−04	6.42E−04	8.92E−03
		B	7.83E−05	6.83E−05	5.82E−04	7.22E−05	7.61E−06	4.91E−06	7.61E−06
		SD	0.0438	0.025072	0.001411	0.0619	0.08662	0.071474	0.027377
		R	4	6	7	5	3	2	1
	100	W	4.81	2.75	7.84	4.92	5.37	2.81	3.97
		M	2.67E−01	8.43E−01	8.53E−01	5.76E−01	6.89E−01	4.76E−02	1.85E−02
		B	4.82E−04	4.82E−03	4.11E−03	5.92E−03	3.88E−04	5.82E−04	6.11E−04
		SD	0.83	0.57	0.0411	0.2119	0.366	0.274	0.377
		R	3	6	5	4	7	2	1
P12	30	W	2.51	1.45	8.29E−01	1.25	1.45	1.25	5.82E−02
		M	5.62E−07	3.78E−02	7.88E−05	3.82E−03	7.82E−05	5.83E−02	6.84E−09
		B	8.99E−09	5.86E−05	7.81E−06	5.18E−04	4.29E−04	6.23E−04	2.84E−11
		SD	0.0048	12.647	1.575753	52.6121	21.88676	26.9302	0.899861
		R	2	4	3	5	7	6	1
	100	W	18.74	8.95	4.73	9.12	10.96	2.85	0.9372
		M	1.62E−06	0.3447	2.38E−04	1.54E−02	5.66E−01	9.79E−01	1.26E−07
		B	6.39E−08	0.0083	7.38E−05	6.38E−04	4.29E−04	6.19E−03	6.39E−08
		SD	0.138	18.447	11.5753	59.21	45.86	34.02	0.9861
		R	2	7	4	5	6	3	1
Average of Rank			3.875	4.583	4.75	5.041	5.125	3.2916	1.3333

initial iterations and exploitation is work to find a good solution in last iterations. To activate both phases in the proposed RDA, we run the test problem P1 for different run times. In each analysis, the best, the worst and the average as well as the standard deviation during the run times are reported. As given in Table 5, the stopping condition is based on the 10 to 1000 s (s). The criteria to analyze the outputs are the same as Table 3. As indicated in the results, the RDA can reach the global solution if it has more time to search. After 400 s, the ideal solution has been identified. This analysis shows that both phases have been activated in the proposed algorithm if it has more time to search the solution space.

4.2 Real-world applications

This section applies the proposed RDA to several real-world problems as the engineering applications of this algorithm. In this regard, single-machine scheduling, traveling salesman, fixed-charge transportation and vehicle routing problems are used. To assess the efficiency and effectiveness of the RDA, two well-known and successful algorithms (i.e., GA and RDA) for these real-world applications are considered. Note that the comparison of meta-heuristics is also based on the time interval according to the complexity of problems. To achieve a reliable result, the algorithms are run for 30 times as well. All the algorithms are tuned for each real-world problem separately by the Taguchi experimental design method (Taguchi 1986).

4.2.1 Single-machine scheduling problem

A single-machine scheduling problem is one of the important and preliminary issues in the today area of scheduling. It tackles a single-machine or single-resource scheduling is to assign a number of tasks to a single machine or resource. These tasks are arranged due to their performance, which should be optimized. Regarding this fact, in production systems, there are shipping dates for orders, inventory carrying costs, which are incurred if the jobs (tasks) are finished earlier than the shipping dates (i.e., earliness). Accordingly, the shortage costs are incurred if

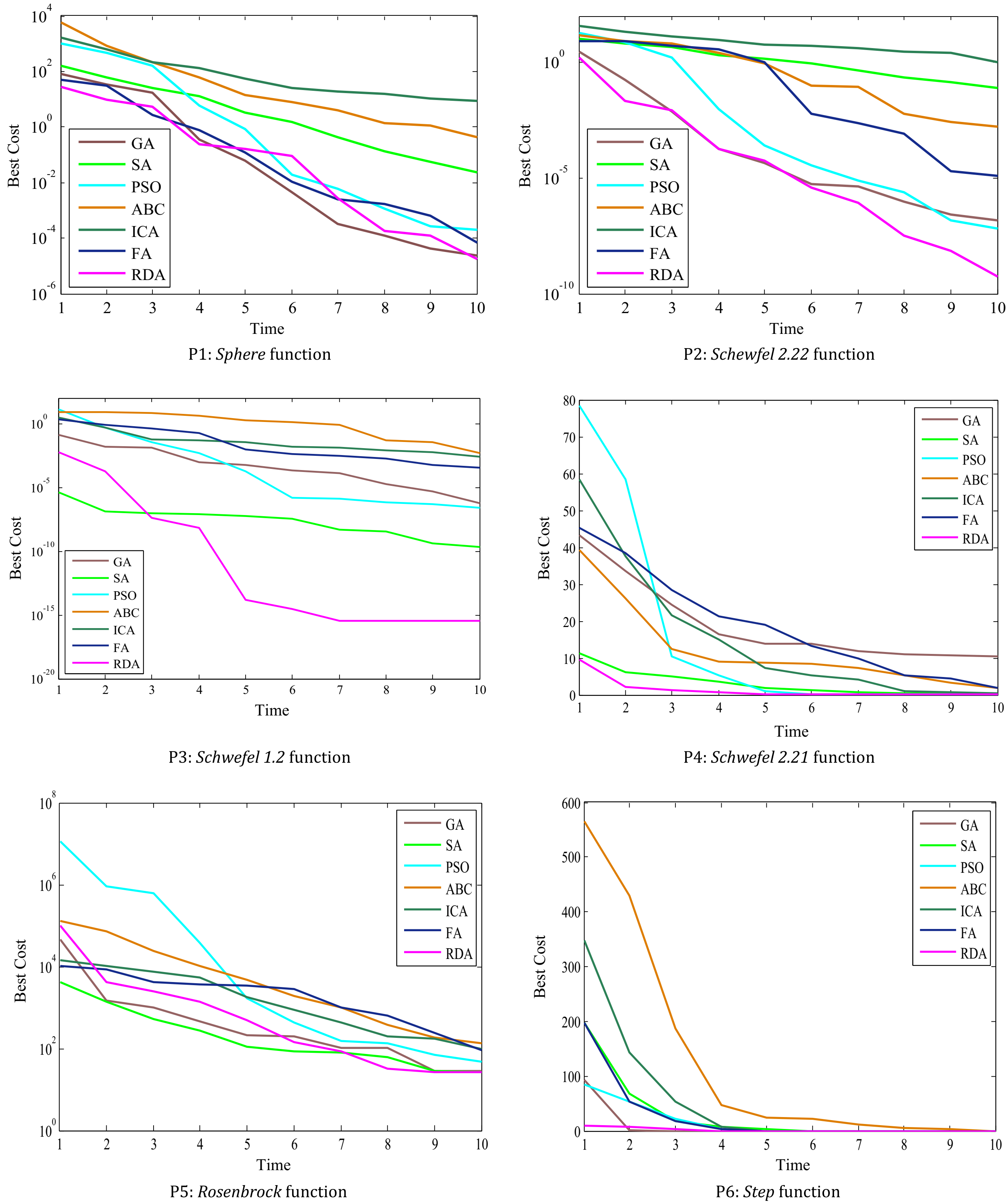
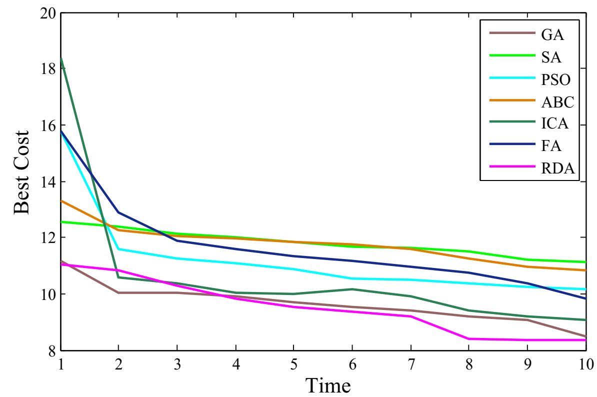
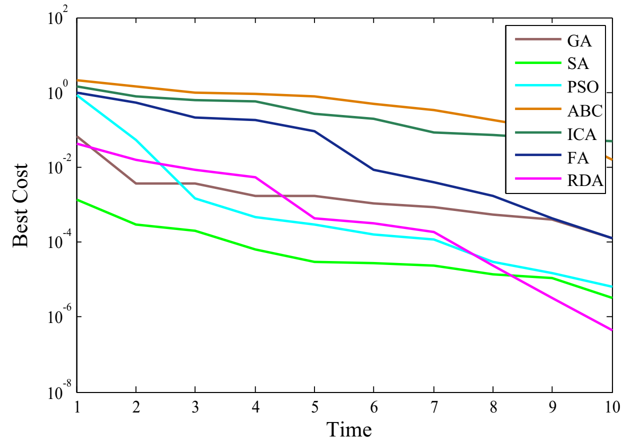


Fig. 7 Behavior of algorithms in each benchmark function

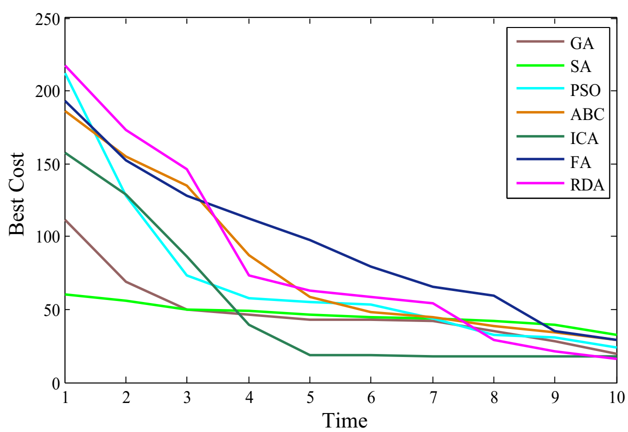
jobs are finished later than the shipping dates (i.e., tardiness). To minimize the total costs, both tardiness and earliness jobs should be minimized. To define this problem mathematically, the earliness and tardiness are considered as $\max_i(d_i - C_i, 0)$ and $\max_i(C_i - d_i)$, respectively, where d_i the due date and C_i is completion time of job i.



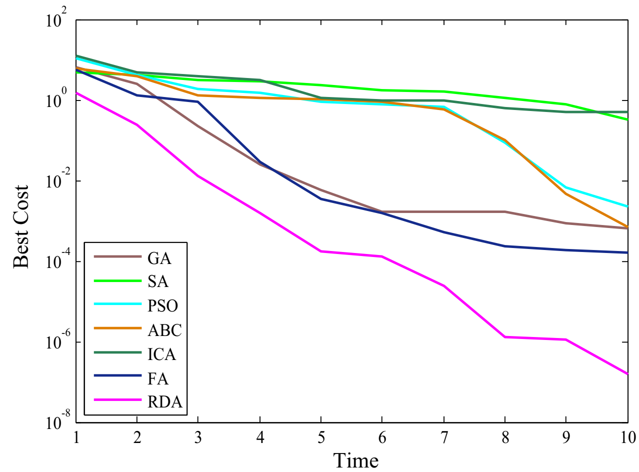
P7: *Quratic* function



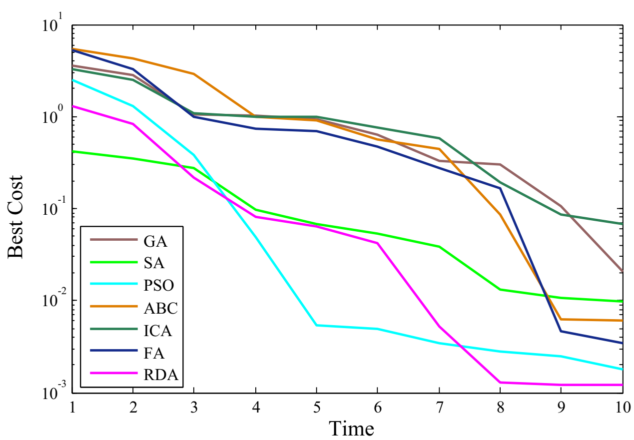
P8: *Greiwank* function



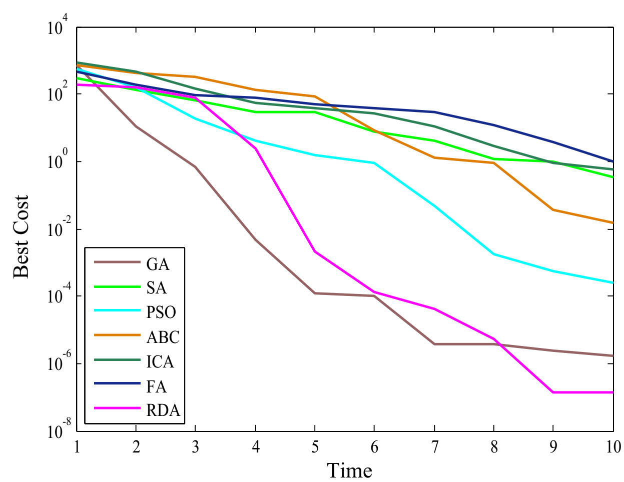
P9: *Rastrigin* function



P10: *Ackley* function



P11: *Penalised* function1



P12: *Penalised* function2

Fig. 7 continued

This problem is an NP-hard one (Hall et al. 1991; Hoogeveen and Van de Velde 1991). Accordingly, meta-heuristics are an efficient way to solve this problem and to achieve the global solution in a reasonable time. There are several meta-heuristics, which show their performance to tackle this problem satisfactorily [e.g., Tabu search (TS)

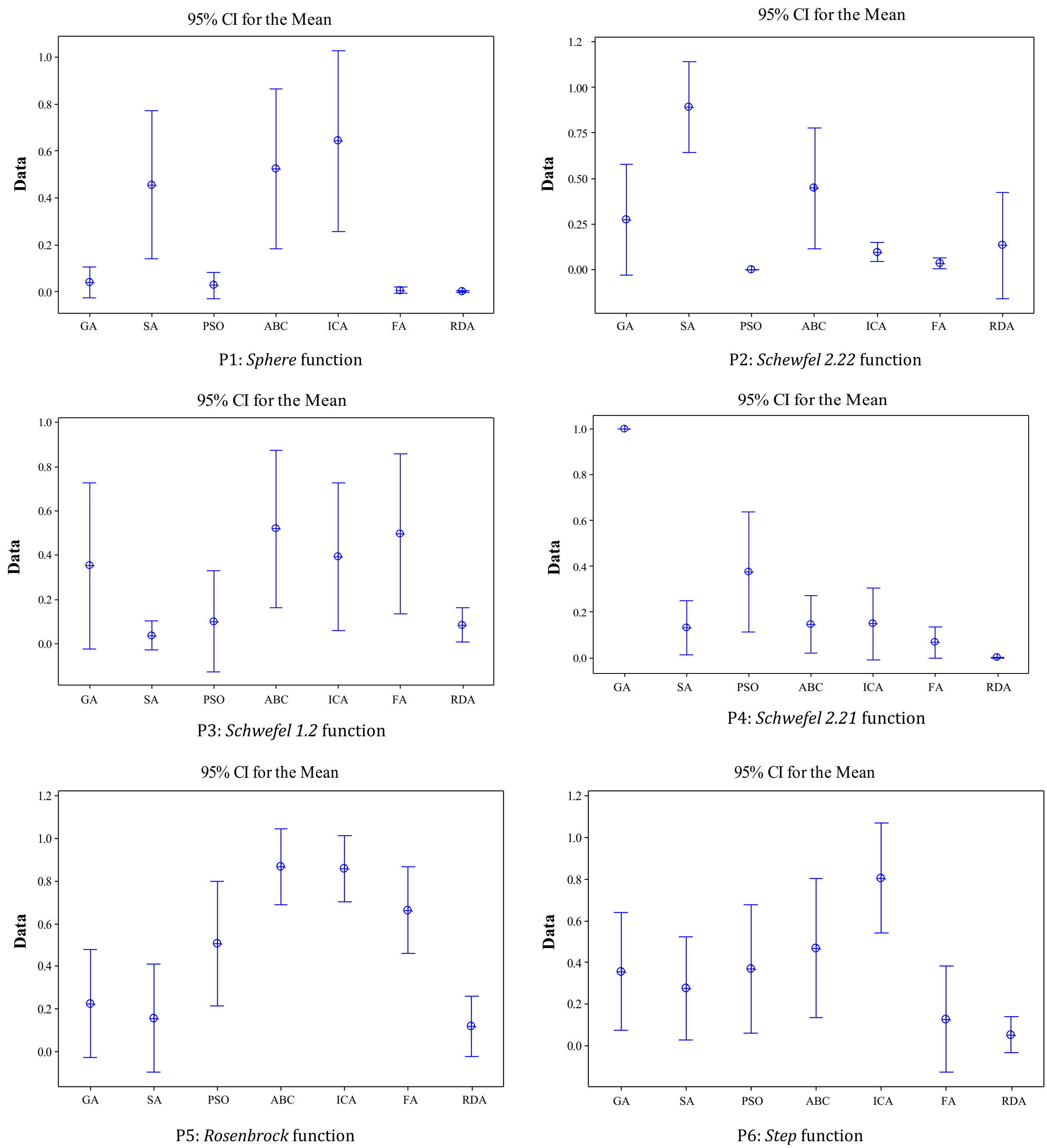


Fig. 8 Means plot and LSD intervals of algorithms in each benchmark function

(James 1997), parallel genetic algorithm (Lee and Kim 1995) and evolutionary search (ES), simulated annealing (SA) and threshold accepting (TA) (Feldmann and Biskup 2003) and steady-state genetic algorithm (Gonçalves et al. 2016) and the ICA (Yousefi and Yusuff 2013)].

Regarding the encoding plan for this problem, a two-stage methodology, called random key (RK) (Snyder and Daskin 2006), is utilized. In this method, first of all, a solution made by random numbers is generated by the algorithm's operators. In the next step, this solution is converted to a feasible one regarding the number of tasks, which should be arranged. Figure 10 shows a simple example accordingly.

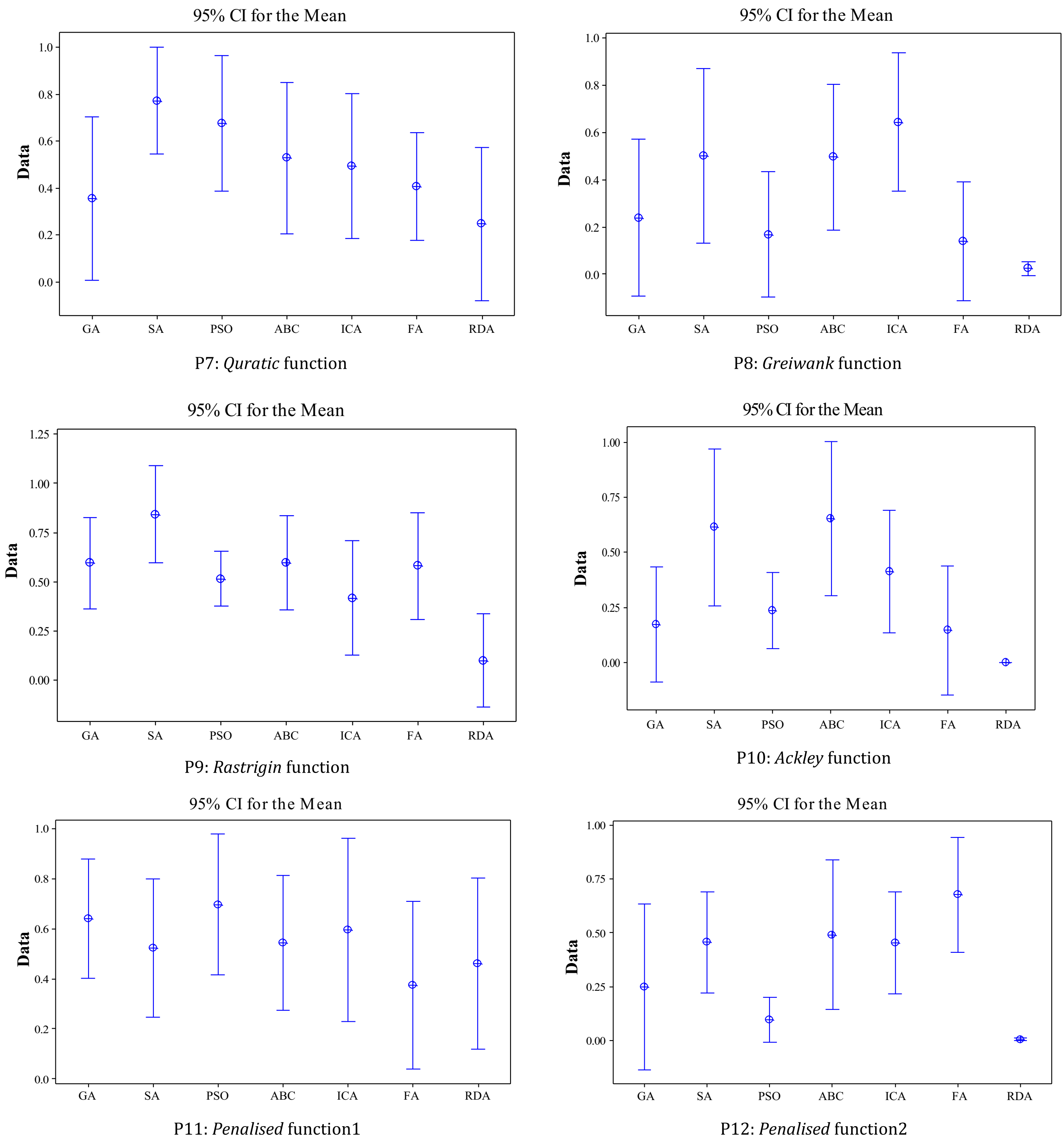


Fig. 8 continued

Six different sizes are applied to compare the results of three meta-heuristics (i.e., GA, ICA and RDA). The results are given in Table 6. Regarding the complexity of each test problem, the time criterion of algorithms comparison is updated and changed by increasing the model complexity. To analyze the performance of the algorithms, due to 30 run times, the relative percentage deviation (RPD) metric is used (Hajiaghaei-Keshteli and Fathollahi Fard 2018). The results of the RPD are shown in Fig. 11. From the table and figure, it is evident that the RDA has a better performance for this real-world application.

4.2.2 Traveling salesman problem

The traveling salesman problem (TSP) is one of the important, preliminary and strategic issues in the area of

Table 4 Result of effect of parameters to convergence (*FRFV* Final Reached Function Value)

Cases	α	β	γ	FRFV
C1	1	0	0.9	0.0011769
C2	0	1	0.9	0.067325
C3	0.8	0.4	0.7	0.038748
C4	0.9	0.3	0.8	0.05274
C5	1	0	0.6	0.029682
C6	0.8	0.3	0.7	0.03053
C7	0	0.4	0.6	20.7596
C8	0.9	1	0.8	0.0018387
C9	1	1	0.9	8.2148e-05

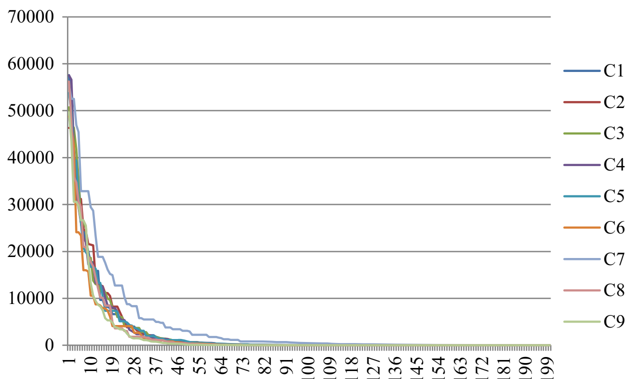


Fig. 9 Comparison of parameters analyses in different cases (x-axis is the number of iterations and y-axis is the fitness evaluations)

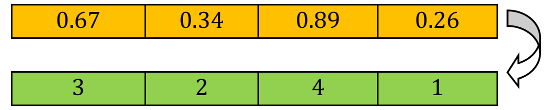


Fig. 10 Encoding scheme of meta-heuristics

Table 6 Results of a single-machine scheduling problem

Number of jobs	Time (s)	GA	ICA	RDA
50	5	8019	8912	8358
100	10	111940	118,537	114,255
150	15	256,639	254,402	254,517
200	20	962,543	995,432	998,197
250	25	1,636,638	1,622,396	1,609,515
300	30	2,294,702	2,327,774	2,308,051

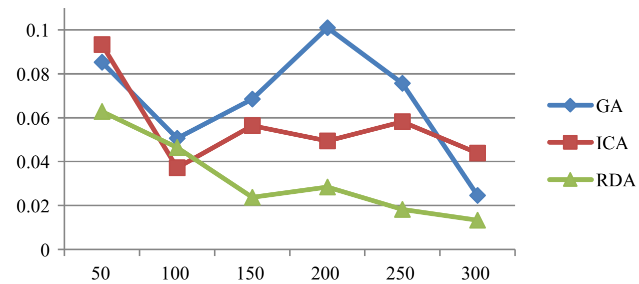


Fig. 11 Interaction between the problem size of a single-machine scheduling problem and the behavior of algorithms regarding the mean RPD during 30 time runs (x-axis show the problem size and y-axis specifies the RPD value)

transportation. Similar to other transportation problems, the goal is to minimize the distance traveled. This problem has a set of real-world applications for the postal routing (Laporte et al. 1996), computer file sequencing (Henrylab 1969), order picking in warehouses (Noon and Bean 1993), process planning for rotational parts (Ben-Arieh et al. 2003) and routing of clients through welfare agencies (Saskena 1970).

One of the main earliest works can be referred to Srivastava et al. (1969) and Henrylab (1969). They showed that this problem is difficult to solve when facing the large-scale tests. Accordingly, they proposed dynamic programming to tackle this problem. Developing a general mathematical formulation [i.e., mixed-integer linear programming (MILP)] for this problem was considered by

Table 5 Sensitivity analyses on the run times for RDA from 10 to 1000 s

Function	D		10 s	50 s	100 s	200 s	400 s	600 s	1000 s
P1	30	W	3.52E−04	6.93E−09	4.37E−16	2.77E−22	5.83E−44	0	0
		M	1.73E−06	3.14E−15	2.91E−23	6.81E−26	5.61E−47	0	0
		B	2.86E−11	3.24E−25	5.37E−38	0	0	0	0
		SD	0.000136	6.17E−07	3.12E−09	2.81E−12	3.37E−24	0	0
	100	W	4.28E−03	5.91E−05	8.15E−14	5.31E−26	5.71E−26	0	0
		M	5.95E−04	3.82E−08	3.19E−18	5.71E−28	4.81E−24	0	0
		B	1.53E−05	5.18E−19	2.81E−27	1.64E−42	0	0	0
		SD	0.07634	5.66E−04	5.14E−05	7.21E−09	5.81E−16	0	0

Table 7 Results of a TSP

Number of problems	Cities	Time (s)	GA	ICA	RDA
P1	25	10	969.1683	1176.8284	951.4579
P2	50	10	1078.5318	1286.3995	887.2729
P3	100	15	3507.1933	3085.9697	3032.4637
P4	150	20	5675.2574	5050.3815	4665.3509
P5	200	20	8379.4009	7275.0972	7192.8643
P6	300	30	11,850.0551	12,538.9371	10,994.7077

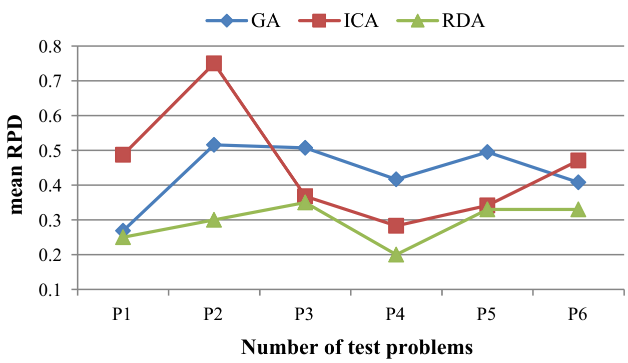


Fig. 12 Interaction between the test problem of the TSP and the performance of meta-heuristics regarding the mean RPD in 30 run times

Laporte and Nobert (1983) and Laporte et al. (1984). Since the problem is an NP-hard one, several meta-heuristics have been utilized in the literature (Fischetti et al. (1995, 1997); Noon and Bean 1993; Laporte et al. (1984); Ben-Arieh et al. (2003)). For instance, the GA, differential evolution (DE) and ICA have revealed a robust performance to solve this problem accurately. As such, a number of recent optimizers [e.g., worm optimization (WO)] is employed to address used this well-known problem (Arnaout 2016).

To implement a continuous search space of meta-heuristics (e.g., the RDA for a discrete optimization problem (i.e., TSP), a two-phase RK method is used. Similar to the solution representation of a single-machine scheduling problem, an encoding scheme is used (See Fig. 10). The second phase considers a sequence of cities for the salesman to travel between them.

To have a fair comparison, not only six different sizes used, but also the stopping condition of meta-heuristics is based on equal time. As such, the RPD is used to consider the reliability of meta-heuristics (i.e., GA, ICA and RDA). The average of results among thirty run times is given in Table 7. The results of the RPD for the algorithms are shown in Fig. 12. The results demonstrate that the RDA is better than the GA and ICA significantly in this real-world problem.

4.2.3 Fixed-charge transportation problem

The fixed-charge transportation problem (FCTP) includes a fixed cost as a setup cost along with a set of related variables to allocate a distribution center to a customer. This problem has been motivated by both business and industrial practitioners. Each supply chain system includes a number of FCTPs regarding each level. The FCTP is a special case of the fixed cost linear programming problem proposed by Farley and Richardson (1984). It has been shown that the FCTP is an NP-hard problem (Farley and Richardson 1984). Accordingly, Klose (2008) showed that the simplest case of the FCTP as a single-source supplier or distributer is NP hardness. These drawbacks motivate several researchers to contribute a number of meta-heuristics in the last decade. For instance, El-Sherbiny and Alhamali (2013) solved the FCTP by a hybrid PSO with artificial immune learning. Similarly, Molla-Alizadeh-Zavardehi et al. (2011) used the artificial immune system (AIS) and GA to solve this problem. In another similar study, Molla-Alizadeh-Zavardehi et al. (2013) solved a fuzzy fixed charge with a number of well-known meta-heuristics including SA and variable neighborhood search (VNS) as well as a hybrid method adopted from both algorithms. In addition, Xie and Jia (2012) formulated the FCTP using a mixed-integer programming model. Based on a steady-state GA, they proposed a hybrid GA, named NFCTP-HGA as the solution method of the model. Also, Lotfi and Tavakkoli-Moghaddam (2013) used a priority-based encoding for solving the FCTP by the pb-GA for both linear and nonlinear types of the problem. In a recent study, Sadeghi-Moghaddam et al. (2017) proposed an application of a recent nature-inspired algorithm, namely whale optimization algorithm (WOA) to solve the FCTP in a fuzzy environment. Note that there are a number of recent review papers to achieve the literature gaps in this area (e.g., Calvete et al. (2016) and Chen et al. (2017)).

Notably, we use the mathematical model of the FCTP considered by Sadeghi-Moghaddam et al. (2017). Also, the encoding scheme of presented meta-heuristics (i.e., GA, ICA and RDA) is also adopted from their study by using the RK method. Furthermore, Table 8 shows the data

Table 8 Data generation of the considered FCTP

Number of problems	Problem size	Time (s)	Total supply	Problem type	Range of variable costs		Range of fixed costs	
					Lower limit	Upper limit	Lower limit	Upper limit
P1	10 × 10	20	10,000	A	3	8	50	200
P2	10 × 20	20	15,000	B	3	8	100	400
P3	15 × 15	30	15,000	C	3	8	200	800
P4	10 × 30	35	15,000	D	3	8	400	1600
P5	50 × 50	70	50,000					
P6	30 × 100	90	30,000					
P7	50 × 200	120	50,000					

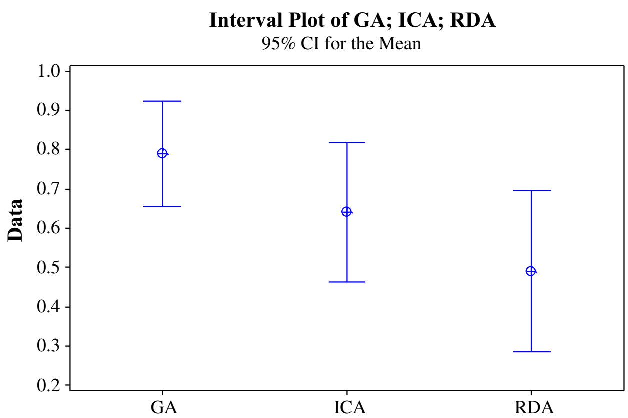


Fig. 13 Means plot and LSD intervals of the algorithms in the FCTP

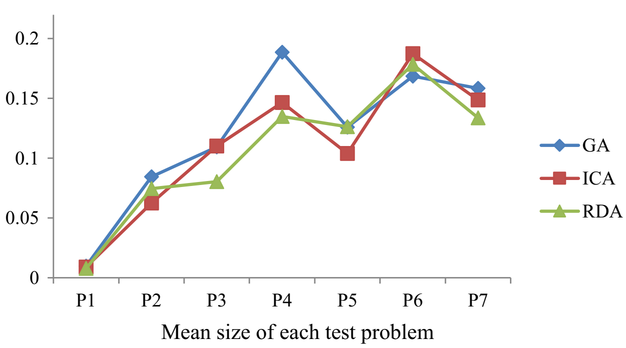


Fig. 14 Interaction between the problem size and the mean RPD of algorithms for the FCTP (x-axis shows the size of model and y-axis reveals the mean RPD)

generation of the FCTP. Regarding the test problems, the stopping criterion of meta-heuristics is considered as the same time interval.

To analyze the algorithms, a set of statistical analyses regarding 30 run times of algorithms is performed. The means plot and LSD intervals of algorithms (at the 95% confidence level) are shown in Fig. 13. From this figure, it is evident that the RDA is truly better than the GA and ICA in FCTP. Additionally, the RPD metric is utilized to assess the efficiency and effectiveness of used meta-heuristics. The results are shown in Fig. 14. According to this figure, the RDA reaches a set of reliable results which are better than other algorithms, except P2, P5 and P6.

4.2.4 Vehicle routing problem

Another basic problem in supply chain systems is the vehicle routing problem (VRP). Regarding this issue, a company aims to supply a number of customers by using the same vehicles. Similar to the TSP, the objective is to minimize the total travelled distance. The application of this real-world problem involves of several domains, such as the supply chain network design (Fathollahi-Fard and Hajiaghaei-Keshteli 2018), home health care (Fikar and Hirsch 2017) and emergency logistics and transportation (Rodríguez-Espíndola et al. 2018). Here, a basic model of the VRP presented by Laporte (1992) is utilized.

Academically, it has been proven that the VRP is known as an NP-hard problem. Accordingly, a number of heuristics and meta-heuristics have been proposed in the literature (Pasha et al. 2016). The most well-known and successful meta-heuristics in the literature are TS, SA, GA, ant colony optimization (ACO), VNS and ICA [Gendreau et al. (1994); Rego and Roucairol (1996); Barbarosoglu and Ozgur (1999); Hiquebran et al. (1993); Bullnheimer et al. (1999); Baker and Ayechew (2003); Crainic et al. (2015); Sarasola et al. (2016); Zhou et al. (2016); Jiang et al. (2017)].

The encoding scheme of the VRP for the employed meta-heuristics (i.e., GA, ICA and RDA) is simple and similar to the TSP illustrated in Sect. 4.2.2. Accordingly, the RK method is used. The problem is solved by different complexity levels of problem sizes. The results are given in Table 9. To analyze the reliability of results, the RPD metric is considered for the meta-heuristics. These effects are shown in Fig. 15. Generally, as can be seen, while the RDA shows the best performance, the ICA has the weakest performance accordingly.

Table 9 Results of the VRP

P_i	Cities	Vehicles	Time (s)	GA	ICA	RDA
P1	8	3	15	354.4722	354.4722	354.4722
P2	10	3	15	468.4243	506.4388	461.9244
P3	14	4	20	708.2543	534.0541	522.785
P4	20	4	20	667.3689	651.4673	594.0197
P5	25	5	40	1187.3328	815.8704	794.7334
P6	30	5	40	1026.0682	998.7305	975.1149
P7	40	6	50	1403.9785	1642.2008	1367.1802
P8	50	7	80	1608.58	1861.3722	1584.5904
P9	60	7	80	1859.7725	2319.0633	1748.1033
P10	70	8	90	2832.9527	3058.5962	2820.6076

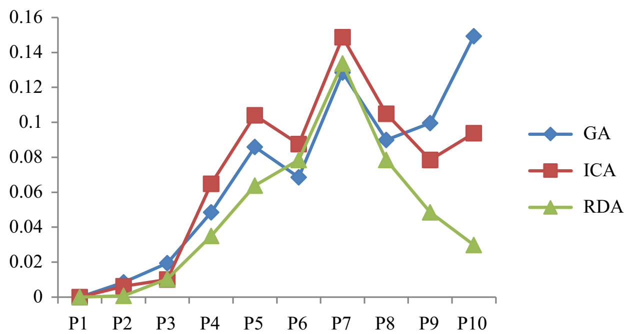


Fig. 15 Interaction between each algorithm and the problem size in the VRP during 30 time runs (x-axis shows the number of problems and y-axis is the mean RPD)

4.3 Multi-objective optimization

As its name says, multi-objective optimization deals with more than one objective. There are several real problems that aim to optimize more than one objective simultaneously. In this regard, all objectives should be optimized to address a multi-objective optimization problem (Mirjalili and Lewis 2015). A simple example of multi-objective optimization model can be formulated by (without the loss of generality):

$$\begin{aligned} &\min f_1 \\ &\max f_2 \\ &\text{s.t.} \\ &g_{(x)} \geq 0 \end{aligned} \tag{15}$$

where f_1 and f_2 are the first and second objective functions that aim to minimize the total costs and maximize the profits of a system, respectively. The nature of such problems is truly different for comparison of solutions regarding both single- and multi-objective optimization problems. Accordingly, the main procedure to compare two solutions considering a multi-objective model is named Pareto optimal dominance. To illustrate this fact, consider two solutions, a and b. Solution a is said to dominate another solution b when all the objectives of a are not worse than that of b, and there exists at least one objective of a that is better than b. Overall, the Pareto optimal set is the answer of this interaction between the objective functions. The interested readers to study more about the definitions of Pareto domination, and a Pareto optimal set can refer to Mirjalili et al. (2017).

Here, we firstly illustrate the multi-objective version of the proposed RDA. Secondly, four well-known assessment metrics are introduced to evaluate the Pareto optimal sets. After that, a real-world problem is considered to do a set of analyses by using four assessment metrics in the term of multi-objective optimization problems.

4.3.1 Multi-objective version of the RDA

Here, a multi-objective version of the RDA is considered. As mentioned earlier, the solution for a multi-objective problem is a set of solutions, called a Pareto optimal set. The RDA is able to drive the Pareto optimal solutions. Similar to other multi-objective evolutionary algorithms [e.g., NSGA-II (Deb et al. 2002)], the main differences between a single-objective form of the algorithm and a multi-objective one refer to select a number of solutions. Accordingly, after roaring and fighting of males, it is needed to define the Pareto optimal set for males. Additionally, to update the next generation, the strategy of selecting the non-dominated solutions should be considered. In overall, Fig. 16 shows the pseudo-code of the RDA in the term of multi-objective optimization problems.

4.3.2 Evaluation metrics for a Pareto optimal set

Note that the comparison of meta-algorithms in the term of multi-objective optimization problems is truly different from a single-objective form. Accordingly, a number of assessment metrics have recently been developed and

Fig. 16 Pseudo-code of the multi-objective RDA

```
Initialize the Red Deer population.
Calculate the fitness and sort them and form the hinds (N_hind) and male RD (N_male).
Form the Pareto-optimal sets.
T1=clock;
While (t< maximum time of simulation)
  for each male RD
   Roar the male (Eq. 3.3).
   Update the position if better than the prior ones.
  end for
  Update non-dominated male RD.
  Sort the males and also form the stags and the commanders (Eqs. 3.4 and 3.5).
  for each male commander
  Fight between male commanders and stags (Eqs. 3.6 and 3.6).
  Update the position of male commanders and stags
 end for
  Update the non-dominated commanders.
  Form harems (Eqs. 3.8, 3.9 and 3.10).
  for each male commander
   (Eq. 3.11)
   Mate a male commander with the selected hinds of his harem randomly (Eq. 3.12).
   Select a harem randomly and name it k.
   (Eq. 3.13)
   Mate male commander with some of the selected hinds of the harem (Eq. 3.12).
  end for
  for each stag
 Calculate the distance between the stag and all hinds and select the nearest hind (Eq. 3.14).
 Mate stag with the selected hind (Eq. 3.12).
 end for
 Select the next generation with the roulette wheel selection.
  Update the Pareto-optimal sets.
  T2=clock;
  t=T2-T1;
end while
Return the best Pareto-optimal set.
```

employed to compare the two sets of a Pareto front (Hajiaghaei-Keshteli and Fathollahi Fard 2018). Here, four popular assessment metrics are used. These metrics have been considered in several recent studies (e.g., Govindan et al. (2015) and Samadi et al. (2018)).

Number of Pareto Solution (NPS): This metric is important and reveals the properties of diversification phase in meta-heuristics. In this metric, the number of Pareto front solutions is computed for each algorithm (Govindan et al. 2015). It is evident that a higher value of this metric is more favorable.

Mean Ideal Distance (MID): This metric aims to consider the distance between the Pareto optimal sets. The formulation of this metric in a recent study (Samadi et al. 2018) is considered by:

$$MID = \frac{\sum_{i=1}^{NPS}\left(\sqrt{\sum_{j=1}^{n_{\text{obj}}}\left(\frac{f_i^j - f_{\text{best}}^j}{f_{\max}^j - f_{\min}^j}\right)^2}\right)}{NPS} \tag{16}$$

where NPS is the number of Pareto solutions for the algorithm. f_i^j is the i-th solution and j-th objective function. Also, f_{best}^j is the ideal point for the j-th objective function. $f_{\max}^j$ and $f_{\min}^j$ are the maximum and the minimum values among all Pareto solutions of every objective function, respectively. The lower value of the MID brings a better quality and performance (Govindan et al. 2015).

Spread of non-dominance solution (SNS): Regarding the MID, the SNS evaluates the diversity of Pareto sets. The higher value of this metrics brings a better performance of meta-heuristics. From a recent study, this metric is formulated by:

$$SNS = \sqrt{\frac{\sum_{i=1}^{NPS}\left(\text{MID} - \sum_{j=1}^{n_{\text{obj}}} f_i^j\right)^2}{NPS - 1}} \tag{17}$$

Maximum spread (MS): This metric aims to consider the extension of Pareto optimal sets (Samadi et al. 2018). The MS can be formulated by:

$$MS = \sqrt{\sum_{j=1}^{n_{\text{obj}}} (f_{\max}^{j} - f_{\min}^{j})^2} \quad (18)$$

Similar to other metrics, a higher value of MS brings a better capability of meta-heuristics (Hajiaghaei-Keshteli and Fathollahi Fard 2018).

4.3.3 Multi-objective nurse rostering problem

The nurse rostering problem (NRP) is one of the time-tabling problems. In recent years, the NRP has been widely investigated by the researchers in both terms of academics and healthcare practitioners. Generally, an NRP is described as an assignment of a set of qualified nurses to a different set of shifts over a predetermined scheduling period, subject to a set of hard and soft constraints. Similar to other scheduling problems, the NRP is an NP-hard problem. The literature of NRPs is rich in using of meta-heuristics to tackle this NP-hard problem, such as TS (Dowsland 1998), SA (Bailey et al. 1997), VNS (Burke et al. 2008), ACO (Gutjahr and Rauner 2007), GA (Moz and Pato 2007), PSO (Wu et al. 2015), HS algorithm (Awadallah et al. 2013; Nie et al. 2016) and ABC (Asaju et al. 2015). To study more about the literature review of NRPs, a number of the recent papers have been studied (e.g., Awadallah et al. (2017) and Nie et al. (2016)).

Here, a multi-objective NRP is considered, which has two conflict goals (i.e., minimizing the total cost of scheduling and maximizing the profits of nurses' effectiveness). The data of the developed model are taken from a case study in the Sari city located in the north of Iran.[1] The sets, parameters, decision variables and the proposed formulation are as follows.

Sets:

I Set of nurses $(i = 1, 2, \ldots, I).L$ Set of shifting time $(l_1 = \text{morning}, l_2 = \text{evening}, l_3 = \text{night})$.

D Set of days $d = 1, 2, \ldots, 28$.

Parameters:

$C_{i\ell}$ Wage of nurse i in shift $\ell R_{d\ell}$ All needed nurses in shift ℓ in day dP_i Priority level of nurse i that depends on his/her resume

Decision variables:

$X_{idl} : \{1 \text{ or } 0\}$ 1 if nurse i is allocated to shift l; 0, otherwise

Proposed model:

$$\min Z_1 = \sum_i \sum_d \sum_{L=1}^{2} C_{i\ell}.X_{id\ell} + 1/5 \sum_i \sum_d C_{i3}.X_{id3} \quad (19)$$

$$\max Z_2 = \sum_i \sum_d \sum_\ell p_i.X_{id\ell} \quad (20)$$

s.t.

$$\sum_\ell X_{id\ell} \le 1; \forall i, d \quad (21)$$

$$\sum_{i=1} X_{id2} = Rd_2; \forall d = 1, \ldots D \quad (22)$$

$$\sum_{i=1} X_{id3} = Rd_3; \forall d = 1, \ldots D \quad (23)$$

$$\sum_d X_{id\ell} \ge 1;\ \forall i, \ell \quad (24)$$

$$X_{i,d,3} + X_{i,d+2,3} + X_{i,d+3,3} \le 1;\ \forall i, d \quad (25)$$

$$X_{id\ell} = \{0, 1\} \quad (26)$$

The first and second objective functions of the model proposed are given in Eqs. (19) and (21), respectively. The first and second objectives aim to minimize the total cost of allocation of nurses and maximize the profits of the nurses' effectiveness, respectively. As such, Eq. (21) states that each nurse should be allocated to only one shift in each day. Equations (22)–(24) ensure that in all shift and days, the needed nurses should be allocated as well. Equation (25) states that if a nurse has worked in night shift, he/she is off for the next day and also should not be allocated for night shift in the next couple of days. Finally, Eq. (26) guarantees the binary variables decisions.

To illustrate problem better, an example of the model with 15 nurses is given in Table 10. Similar to other real-world cases, the multi-objective version of the RDA is employed and compared with the NSGA-II and multi-objective ICA. Similar to a single-machine scheduling problem, the solution representation is a sequence of nurses. Accordingly, the RK method is used to consider a continuous search space for the employed meta-heuristics.

Note that the parameters of meta-heuristics are tuned by the Taguchi experimental design method, like the pervious sections. The comparison of meta-heuristics is also based on equal computational time. First of all, the optimal values found by meta-heuristics are given in Table 11. Note that in this table, regarding the MID metric, the solution that has the lowest ideal distance among the Pareto optimal set is reported. Moreover, the results of meta-heuristics regarding four assessment metrics are given in Table 12. According to the results, the multi-objective RDA not only reaches optimal values in most of the test problems but also appears a robust behavior regarding the four assessment metrics in most of cases. Finally, an example of non-

[1] https://infosalamat.com/hospitals/hekmat-e-sari.

Table 10 An example of problem with 15 nurses

Nurses	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P_i	0.7	0.7	0.8	0.8	0.9	0.9	0.6	0.6	0.6	0.7	0.7	0.7	0.8	0.7	0.6
Shifting time	Waging cost c_{il}														
l_1	100	100	110	115	120	120	65	65	65	70	70	70	75	80	100
l_2	100	100	110	115	120	120	65	65	65	70	70	70	75	80	100
l_3	130	130	143	150	156	156	84	84	84	90	90	90	80	85	110

Days	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Shifting time	Number of nurses for each shift and day ($R_{(d,l)}$)																											
l_1	5	5	5	5	5	5	4	5	5	5	5	5	5	4	5	5	5	5	5	5	4	5	5	5	5	5	5	4
l_2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
l_3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 11 Results of multi-objective NRP

No. of problems	No. of nurses	Time (s)	NSGA-II		Multi-objective ICA		Multi-objective RDA	
			Z1	Z2	Z1	Z2	Z1	Z2
P1	15	20	2125	4.5	1848.5	3.9	2125	4.5
P2	30	30	8183	17.6	10,030	21.3	10,030	21.3
P3	50	40	13,983	45.9	13,983	45.9	2118.5	21.4
P4	70	50	5688	34.7	3868.5	33.4	21,011	71.4
P5	85	60	20,124	66.7	22,227	71.2	23,717.5	77
P6	100	70	8200.5	41.7	27,328	90.1	4663.5	42.8

Table 12 Results of assessment metrics for Pareto optimal sets

No. of problem	NSGA-II				Multi-objective ICA				Multi-objective RDA			
	NPS	MID	SNS	MS	NPS	MID	SNS	MS	NPS	MID	SNS	MS
P1	4	2.7834	363,533.66	199,573.8	5	2.3656	353,683.66	322,971.24	5	1.4909	242,655.17	360,138.18
P2	5	2.9856	700,352.33	371,253.15	6	2.1409	699,981.33	583,346.08	7	1.7844	745,847.24	629,121.45
P3	7	3.5849	1,091,855.54	536,673.68	8	3.0635	1,089,612.5	674,618.16	9	2.9208	948,759.11	725,389.07
P4	10	4.2781	1,702,788.11	611,408.12	10	4.6701	1,700,420.1	756,024.795	12	2.3252	1,318,978.17	609,693.58
P5	12	3.5903	2,360,089.14	859,254.95	11	2.9635	2,355,835.1	894,850.313	13	3.6888	2,265,444.76	619,643.54
P6	15	2.5784	2,707,964.92	788,505.79	14	5.7248	2,701,689.9	1,261,434.82	16	3.3281	2,578,609.05	1,743,193.31

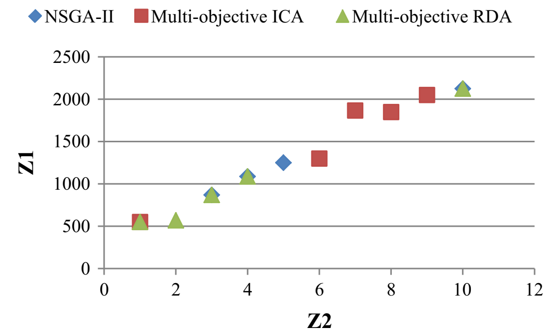


Fig. 17 Pareto optimal set of each meta-heuristic algorithm in P1 with 15 nurses

dominated solutions of employed meta-heuristics is shown in Fig. 17. As a result of this figure, the multi-objective RDA overcomes the other algorithms and shows the best performance accordingly.

5 Conclusion and future studies

This paper firstly studied and mimicked an unusual behavior of Scottish red deer (RD) to develop a new nature-inspired meta-heuristic algorithm, called red deer algorithm (RDA). The main inspiration of this proposed algorithm was originated from the mating behavior of RDs in a breading season. The algorithm started with an initial population of RD. This population was divided into two types: male RD and hinds. The better solutions were considered as the male RD. The main steps of this algorithm were focused on the competition of male RD to get the harem with more hinds by roaring and fighting behaviors.

To analyze the effectiveness and efficiency of the RDA, several analyses were conducted. Firstly, 12 standard functions were considered. Secondly, a number of real-world engineering problems including SMSP, TSP, FCTP and VRP were used. Furthermore, the multi-objective version of the RDA was considered and evaluated by a multi-objective NRP via four assessment metrics to analyze Pareto optimal sets. Regarding the results of experiments, it was observed that the proposed RDA could explore the promising search regions and find the global solutions in most of the cases. Generally, the RDA was simple to tune and could be ordered for different types of real-world problems. By changing or tuning the parameters of RDA, i.e., Nmale, Nhind, α, β and γ, an interaction between the intensification and diversification phases can be employed, easily, regarding the type and dimension of the problem.

This study opens several new directions for future studies. Firstly, more comprehensive analyses for both single- and multi-objective versions of the RDA should be performed. Secondly, the operators of roaring, fighting and mating can be enhanced by a set of local search strategies. Accordingly, the steps of the proposed RDA may need a set of modifications and hybridizations with other optimizers to enhance the properties of the proposed algorithm in the exploration and exploitation phases. Most notably, this study uses a continuous version of RDA for continuous variables. In this regard, an effort to develop a discrete version is highly suggested for future studies. More broadly, the application of the proposed RDA for other real-world engineering problems should be examined as well.

Acknowledgements The current paper is an extended version of our conference paper (Fathollahi-Fard and Hajiaghaei-Keshteli 2016) presented at the International Conference on Industrial Engineering (ICIE 2016) supported by the IEEE and held in Tehran on 15–16 January 2016.

Compliance with ethical standards

Conflict of interest The authors of this research certify that they have NO affiliation with or involvement in any organization or entity with financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

Arnaout JP (2016) Worm optimization for the traveling salesman problem. In: Heuristics, metaheuristics and approximate methods in planning and scheduling. Springer, Cham, pp 209–224

Asaju LAB, Awadallah MA, Al-Betar MA, Khader AT (2015) Solving nurse rostering problem using artificial bee colony algorithm. In: ICIT 2015 the 7th international conference on information technology. pp 32–38

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Proceedings of the IEEE congress on evolutionary computation, Singapore, 25–28 September 2007. pp 4661–4667

Awadallah MA, Khader AT, Al-Betar MA, Bolaji ALA (2013) Global best harmony search with a new pitch adjustment designed for nurse rostering. J King Saud Univ Comput Inf Sci 25(2):145–162

Awadallah MA, Al-Betar MA, Khader AT, Bolaji ALA, Alkoffash M (2017) Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem. Neural Comput Appl 28(3):463–482

Bailey RN, Garner KM, Hobbs MF (1997) Using simulated annealing and genetic algorithms to solve staff-scheduling problems. Asia Pac J Oper Res 14(2):27

Baker BM, Ayechew MA (2003) A genetic algorithm for the vehicle routing problem. Comput Oper Res 30(5):787–800

Barbarosoglu G, Ozgur D (1999) A tabu search algorithm for the vehicle routing problem. Comput Oper Res 26(3):255–270

Beheshti Z, Shamsuddin SMH (2013) A review of population-based meta-heuristic algorithms. Int J Adv Soft Comput Appl 5(1):1–35

Ben-Arieh D, Gutin G, Penn M, Yeo A, Zverovitch A (2003) Process planning for rotational parts using the generalized travelling salesman problem. Int J Prod Res 41(11):2581–2596

Blank DA, Ruckstuhl K, Yang W (2017) Development of juvenile goitered gazelle social behavior during the hiding period. Behav Proc 144:82–88

Bullnheimer B, Hartl RF, Strauss C (1999) An improved ant system algorithm for thevehicle routing problem. Ann Oper Res 89:319–328

Burke EK, Curtois T, Post G, Qu R, Veltman B (2008) A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. Eur J Oper Res 188(2):330–341

Calvete HI, Galé C, Iranzo JA (2016) An improved evolutionary algorithm for the two-stage transportation problem with fixed charge at depots. OR Spectr 38(1):189–206

Chen L, Peng J, Zhang B (2017) Uncertain goal programming models for bicriteria solid transportation problem. Appl Soft Comput 51:49–59

Cheraghalipour A, Hajiaghaei-Keshteli M, Paydar MM (2018) Tree growth algorithm (TGA): a novel approach for solving optimization problems. Eng Appl Artif Intell 72:393–414

Clutton-Brock TH, Albon SD, Gibson RM, Guinness FE (1979) The logical stag: adaptive aspects of fighting in red deer (*Cervus Elaphus* L.). Anim Behav 27:211–225

Crainic TG, Gajpal Y, Gendreau M (2015) Multi-zone multi-trip vehicle routing problem with time windows. INFOR Inf Syst Oper Res 53(2):49–67

Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dong W, Zhou M (2017) A supervised learning and control method to improve particle swarm optimization algorithms. IEEE Trans Syst Man Cybern Syst 47(7):1135–1148

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cyber Part B (Cybernetics) 26(1):29–41

Dowsland KA (1998) Nurse scheduling with tabu search and strategic oscillation. Eur J Oper Res 106(2–3):393–407

Draa A (2015) On the performances of the flower pollination algorithm–qualitative and quantitative analyses. Appl Soft Comput 34:349–371

El-Sherbiny MM, Alhamali RM (2013) A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem. Comput Ind Eng 64(2):610–620

Farley AA, Richardson KV (1984) Fixed charge problems with identical fixed charges. Eur J Oper Res 18(2):245–249

Fathollahi-Fard AM, Hajiaghaei-Keshteli M (2016) Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating. In: IEEE International Conference on Industrial Engineering, pp 33–34

Fathollahi-Fard AM, Hajiaghaei-Keshteli M (2018) A stochastic multi-objective model for a closed-loop supply chain with environmental considerations. Appl Soft Comput 69:232–249

Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R (2018) The social engineering optimizer (SEO). Eng Appl Artif Intell 72:267–293

Fathollahi-Fard AM, Ranjbar-Bourani M, Cheikhrouhou N, Hajiaghaei-Keshteli M (2019) Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system. Comput Ind Eng 137:106103

Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tian G, Li Z (2020) An adaptive Lagrangian relaxation-based algorithm for a

coordinated water supply and wastewater collection network design problem. Inf Sci 512:1335–1359
Feldmann M, Biskup D (2003) Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches. Comput Ind Eng 44(2):307–323
Fikar C, Hirsch P (2017) Home health care routing and scheduling: A review. Comput Oper Res 77:86–95
Fischetti M, González JJS, Toth P (1995) The symmetric generalized traveling salesman polytope. Networks 26(2):113–123
Fischetti M, Salazar González JJ, Toth P (1997) A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper Res 45(3):378–394
Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
Gendreau M, Hertz A, Laporte G (1994) A tabu search heuristic for the vehicle routing problem. Manage Sci 40(10):1276–1290
Ghorbani N, Babaei E (2014) Exchange market algorithm. Appl Soft Comput 19:177–187
Gonçalves TC, Valente JM, Schaller JE (2016) Metaheuristics for the single machine weighted quadratic tardiness scheduling problem. Comput Oper Res 70:115–126
Govindan K, Jafarian A, Nourbakhsh V (2015) Bi-objective integrating sustainable order allocation and sustainable supply chain network strategic design with stochastic demand using a novel robust hybrid multi-objective metaheuristic. Comput Oper Res 62:112–130
Gutjahr WJ, Rauner MS (2007) An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. Comput Oper Res 34(3):642–666
Hajiaghaei-Keshteli M, Aminnayeri M (2013) Keshtel Algorithm (KA); a new optimization algorithm inspired by Keshtels' feeding. In: Proceeding in IEEE conference on industrial engineering and management systems. pp 2249–2253
Hajiaghaei-Keshteli M, Fathollahi Fard AM, (2018) Sustainable closed-loop supply chain network design with discount supposition. Neural computing and applications (in press)
Hall NG, Kubiak W, Sethi SP (1991) Earliness–tardiness scheduling problems, II: deviation of completion times about a restrictive common due date. Oper Res 39(5):847–856
Han W, Xu J, Zhou M, Tian G, Wang P, Shen X, Hou E (2016) Cuckoo search and particle filter-based inversing approach to estimating defects via magnetic flux leakage signals. IEEE Trans Magn 52(4):1–11
He S, Wu QH, Saunders JR (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans Evol Comput 13(5):973–990
Henrylab Al (1969) Record balancing problem-a dynamic programming solution of a generalized traveling salesman problem. Revue Francaise D Informatique De Recherche Operationnelle 3(NB 2):43
Hiquebran DT, Alfa AS, Shapiro JA, Gittoes DH (1993) A revised simulated annealing and cluster-first route-second algorithm applied to the vehicle routing problem. Eng Optim 22(2):77–107
Holland J (1975) Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control and artificial intelligence
Hoogeveen JA, Van de Velde SL (1991) Scheduling around a small common due date. Eur J Oper Res 55(2):237–242
Hosseini S, Al Khaled A (2014) A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research. Appl Soft Comput 24:1078–1094
Hussain K, Salleh MNM, Cheng S, Shi Y (2018) Metaheuristic research: a comprehensive survey. Artificial intelligence review (in press)
James RJW (1997) Using tabu search to solve the common due date early/tardy machine scheduling problem. Comput Oper Res 24(3):199–208
Jiang X, Bai R, Atkin J, Kendall G (2017) A scheme for determining vehicle routes based on Arc-based service network design. INFOR Inf Syst Oper Res 55(1):16–37
Kar AK (2016) Bio inspired computing–A review of algorithms and scope of applications. Expert Syst Appl 59:20–32
Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
Kaveh A (2016) Advances in metaheuristic algorithms for optimal design of structures. Springer, Berlin
Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceeding of the IEEE international conference on neural networks, Perth, Australia. IEEE Service Center, Piscataway
Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
Klose A (2008) Algorithms for solving the single-sink fixed-charge transportation problem. Comput Oper Res 35(6):2079–2092
Krause J, James R, Franks DW, Croft DP (eds) (2015) Animal social networks. Oxford University Press, Oxford
Krishnanand KN, Ghose D (2005) Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In: Proceedings of the IEEE on swarm intelligence symposium, California, 8–10 June 2005. pp 84–91
Laporte G (1992) The vehicle routing problem: an overview of exact and approximate algorithms. Eur J Oper Res 59(3):345–358
Laporte G, Nobert Y (1983) Generalized travelling salesman problem through n sets of nodes: an integer programming approach. INFOR Inf Syst Oper Res 21(1):61–75
Laporte G, Desrochers M, Nobert Y (1984) Two exact algorithms for the distance-constrained vehicle routing problem. Networks 14(1):161–172
Laporte G, Asef-Vaziri A, Sriskandarajah C (1996) Some applications of the generalized travelling salesman problem. J Oper Res Soc 47(12):1461–1467
Lee CY, Kim SJ (1995) Parallel genetic algorithms for the earliness-tardiness job scheduling problem with general penalty weights. Comput Ind Eng 28(2):231–243
Liang J J, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer with local search. In: Proceedings of the IEEE congress on evolutionary computation, Edinburgh, UK, 2–4 September 2005. pp 522–528
Lipowski A, Lipowska D (2012) Roulette-wheel selection via stochastic acceptance. Phys A 391(6):2193–2196
Lotfi MM, Tavakkoli-Moghaddam R (2013) A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. Appl Soft Comput 13(5):2711–2726
McComb KE (1991) Female choice for high roaring rates in red deer, Cervus elaphus. Anim Behav 41(1):79–88
Mernik M, Liu SH, Karaboga D, Črepinšek M (2015) On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. Inf Sci 291:115–127
Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. Complex Syst 9(3):193–212
Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
Mirjalili S, Lewis A (2015) Novel performance metrics for robust multi-objective optimization algorithms. Swarm Evol Comput 21:1–23
Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191

Molla-Alizadeh-Zavardehi S, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R (2011) Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation. Expert Syst Appl 38(8):10462–10474

Molla-Alizadeh-Zavardehi S, Nezhad SS, Tavakkoli-Moghaddam R, Yazdani M (2013) Solving a fuzzy fixed charge solid transportation problem by metaheuristics. Math Comput Model 57(5):1543–1558

Moz M, Pato MV (2007) A genetic algorithm approach to a nurse rerostering problem. Comput Oper Res 34(3):667–691

Nie Y, Wang B, Zhang X (2016) Hybrid harmony search algorithm for nurse rostering problem. In: Harmony search algorithm. Springer, Berlin, Heidelberg, pp 109–120

Noon CE, Bean JC (1993) An efficient transformation of the generalized traveling salesman problem. INFOR Inf Syst Oper Res 31(1):39–44

Pasha U, Hoff A, Hvattum LM (2016) Simple heuristics for the multi-period fleet size and mix vehicle routing problem. INFOR Inf Syst Oper Res 54(2):97–120

Rao R, Patel V (2013) Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems. Int J Ind Eng Comput 4(1):29–50

Rego C, Roucairol C (1996) A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In: Meta-heuristics. Springer, Boston, MA, pp 661–675

Rodríguez-Espíndola O, Albores P, Brewster C (2018) Disaster preparedness in humanitarian logistics: A collaborative approach for resource management in floods. Eur J Oper Res 264(3):978–993

Sadeghi-Moghaddam S, Hajiaghaei-Keshteli M, Mahmoodjanloo M (2017) New approaches in metaheuristics to solve the fixed charge transportation problem in a fuzzy environment. Neural computing and applications (in press)

Samadi A, Mehranfar N, Fathollahi Fard AM, Hajiaghaei-Keshteli M (2018) Heuristic-based metaheuristic to address a sustainable supply chain network design problem. J Ind Prod Eng 35(2):102–117

Sarasola B, Doerner KF, Schmid V, Alba E (2016) Variable neighborhood search for the stochastic and dynamic vehicle routing problem. Ann Oper Res 236(2):425–461

Saskena JP (1970) Mathematical model of scheduling clients through welfare agencies. J Can Oper Res Soc 8:185–200

Snyder LV, Daskin MS (2006) A random-key genetic algorithm for the generalized traveling salesman problem. Eur J Oper Res 174(1):38–53

Srivastava SS, Kumar S, Garg RC, Sen P (1969) Generalized traveling salesman problem through n sets of nodes. CORS J 7(2):97

Taguchi G (1986) Introduction to quality engineering: designing quality into products and processes (No. 658.562 T3)

Thouless CR, Guinness FE (1986) Conflict between red deer hinds: the winner always wins. Anim Behav 34(4):1166–1171

Wang B, Xia X, Meng H, Li T (2017) Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems. IEEE/CAA J Autom Sin 4(1):143–153

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Wu TH, Yeh JY, Lee YM (2015) A particle swarm optimization approach with refinement procedure for nurse rostering problem. Comput Oper Res 54:52–63

Xie F, Jia R (2012) Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm. Comput Ind Eng 63(4):763–778

Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2(2):78–84

Yousefi M, Yusuff RM (2013) Minimising earliness and tardiness penalties in single machine scheduling against common due date using imperialist competitive algorithm. Int J Prod Res 51(16):4797–4804

Zhao SZ, Suganthan PN, Das S (2010) Self-adaptive differential evolution with modified multi-trajectory search for CEC'2010 large scale optimization. In: Proceedings of the international conference on swarm, evolutionary, and memetic computing, Chennai, India, 16–18 December 2010

Zhou Y, Luo Q, Xie J, Zheng H (2016) A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem. In: Metaheuristics and optimization in civil engineering. Springer, Cham, pp 255–276