Scientia Iranica E (2021) 28(3), 1750–1764

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

An improved red deer algorithm for addressing a direct current brushless motor design problem

A.M. Fathollahi-Fard[a,*], **M. Niaz Azari**[b], **and M. Hajiaghaei-Keshteli**[c]

a. *Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran.*
b. *Department of Electrical Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran.*
c. *Department of Industrial Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran.*

Received 27 September 2018; received in revised form 6 June 2019; accepted 10 August 2019

KEYWORDS
Red Deer Algorithm (RDA);
Metaheuristic;
Global optimization;
DC brushless motor;
Benchmarks.

Abstract. Red Deer Algorithm (RDA) is one of the recent metaheuristic algorithms inspired by the behavior of red deer in the breeding season. RDA has exhibited viable performance in dealing with a variety of combinatorial optimization problems in different real-world applications. In this paper, the parameters and operators of RDA were modified using some adaptive strategies to improve its performance. To prove the efficiency of the new Improved RDA (IRDA), a number of benchmarked functions were utilized and, also, a Direct Current (DC) brushless motor design was considered, which is one of the real-world engineering design issues. The results of the developed IRDA are compared with other algorithms existing the literature. This comparative study confirms that the proposed IRDA outperforms other algorithms and provides very competitive results.

1. Introduction

Metaheuristic algorithms represent a type of stochastic optimization in nature that have become increasingly applicable to many engineering design issues [1]. A number of recently introduced optimizers exist for solving complex and non-linear problems. For example, the following are some of the successful algorithms in recent years: Social Engineering Optimizer (SEO) inspired by the rules of social engineering [2]; Moth-Flame Optimizer (MFO) inspired by the swarm behavior of flying moth in the night with respect to the moon [3]; Sine Cousin Algorithm (SCA) inspired by the logic of sine and cousin functions [4]; Exchange Market Algorithm (EMA) inspired by different techniques of marketing in a bazar [5]; and Red Deer Algorithm (RDA) inspired by the evolutionary strategies of red deer in the breeding season [6]. Generally, high efficiency of many metaheuristics in solving challenging engineering design issues can be verified with a large number of local solutions [2]. In the case of real-world optimization problems, upon increasing the size of the designed model, it is difficult to achieve a global solution using the current metaheuristics [7]. Therefore, such algorithms may need some improvements to solve these NP-*hard* problems better [3].

One of interesting real-world optimization engineering problems is designing a Direct Current (DC) brushless motor. The advantages of DC motor are becoming more common among both academics and industrial practitioners due to the highlighted characteristics, especially its fast control designs and high efficiency [8]. Accordingly, the present generation of brushes and commentator may be considered as a set of major demerits of these motors due to rapid

*. *Corresponding author.*
E-mail address:
amirmohammad.fathollahifard.1@ens.etsmtl.ca (A.M. Fathollahi-Fard)

doi: 10.24200/sci.2019.51909.2419

erosion of their components [9]. Nowadays, growth in safety hazards and increase in the maintenance costs of such motors can be envisaged over a last few decades [8]. This problem has been formulated as a Brushless DC (BLDC) motor. The difficulty of solving this problem serves as a motivation for several studies to apply numerous metaheuristics [9]. Based on the No Free Lunch theorem [10], it is always possible for a new optimizer to outperform the existing algorithms. From another point of view, modifying the solution algorithms may be useful in finding the global optimum instead of local solutions [11]. This reason has motivated us to employ RDA as one of the recently developed bio-optimizer for the first time in the area of BLDC. The main contribution of our work is to propose an improved version of RDA to solve the BLDC design problem better.

According to the scientific studies in the area of BLDC, this literature is generally very rich in employing recent and well-known metaheuristics. According to a recent study by Duan and Gan [8], a BLDC was applied by developing an Orthogonal Multi-Objective Chemical Reaction Optimizer (OMOCRO). The aim of this work was to maximize the efficiency of BLDC model by minimizing the material cost. They compared OMOCRO with the state-of-the-art optimizers including Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective of Particle Swarm Optimization (MOPSO). Another novel optimizer was proposed by Lee et al. [9]. They proposed the Search Region Management (RSM) algorithm to obtain a global solution instead of a set of local solutions. Their optimal design of BLDC was tested using Fuzzy Evolutionary Algorithm (FEA) to optimize the torque ripple. In 2016, a novel Multi-Objective Krill Herd (MOKH) was proposed by Ayala et al. [12]. They also applied beta distribution using the inertia weight collaboration. They showed that their proposed algorithm could outperform Electromagnetic-like Algorithm (EMA). Ishikawa et al. discussed the viability of optimization tools for solving the BLDC [13]. They considered a new Genetic Algorithm (GA) based on the topology optimization to solve the stator teeth in a BLDC motor to minimize torque ripple. In 2016, another BLDC motor was optimized by a recent population algorithm called Interstellar Search Method (ISM) considering mesh adaptive direct search in the work of Son et al. [14]. Their goal was to minimize the torque ripple as the objective function. In another research, Yoon and Kwon [15] considered the optimal design of BLDC with the cost-efficiency of ferrite magnets to optimize the flux density and the torque ripple, simultaneously. Similarly, Kim et al. [16] optimized the anisotropic ferrite magnet shape and magnetization direction of an interior permanent magnet BLDC motor. They employed a simulation-based optimization approach to solve this complicated optimization model. The 2D analytical optimization case of BLSC motor was suggested by Liu et al. [17]. Their aim was to predict the magnetic field distribution and compare the results with the best existing outputs from the literature. A comprehensive discussion on the outer rotor type motor design using a blower system of BLDC vehicle was done by Lee et al. [18]. In 2017, Azari et al. [19] applied Cuckoo Optimization Algorithm (COA), GA, and Particle Swarm Optimization (PSO) to achieve an optimal design for BLDC motor. Their aim was to find the best parameters of motor geometric functions. In 2018, Xu and Deng [20] recently proposed a Pigeon-Inspired Optimizer (PIO) to solve a set of power components for BLDC design motor by merging adjacent-disturbances and integrated dispatching.

Generally, it can be observed that recently developed optimizers exhibit better efficiency than the well-known and state-of-the-art algorithms. The aforementioned papers in the literature can be divided into three categories. A number of studies have added different new factors to formulate a BLDC model. Since this problem is NP-*hard*, a number of researchers have mainly considered applying recent metaheuristics to reach an optimal design of BLDC motor. Another category considered both modeling and solution approaches. This study proposes an Improved Red Deer Algorithm (IRDA) for the first time in this area to better solve a BLDC design motor adopted from [19]. An extensive comparative study is performed on not only the BLDC design motor but also some standard benchmarked functions.

The rest of this paper can be summarized as follows: Section 2 overviews RDA with its successful literature as well as the steps and details. Section 3 proposes the developed IRDA with its formulations and explanations. Section 4 makes a comprehensive comparison between the proposed algorithm and other applied algorithms to identify the pros and cons of the proposed methodology. In fact, the conclusion and future recommendations are suggested in Section 5.

2. Red Deer Algorithm (RDA)

RDA, first introduced by Fathollahi Fard and Hajiaghaei-Keshteli [6], is one of recent evolutionary optimizers inspired by the competition of red deer in the breeding season. This algorithm enjoys high efficiency in tackling several engineering design issues known as NP-*hard*. For example, Golmohamadi et al. [21] addressed a fuzzy fixed-charge transportation problem considering batch transferring by RDA and a hybrid version employing Simulated Annealing (SA). They compared the efficiency of the proposed RDA with those of Variable Neighborhood Search (VNS) and Imperialist Competitive Algorithm (ICA). In another work, in the area of supply chain network design, Samadi et al. [22] proposed a sustainable supply chain network design problem. They developed three heuristics as the initial solutions of their applied metaheuristics including RDA and GA. Their results ensured the performance of heuristic-based RDA in conclusion. A similar work reported by Hajiaghaei-Keshteli and Fathollahi-Fard [23] developed a sustainable closed-loop supply chain considering discount supposition. They addressed different solution approaches based on recent metaheuristics including Keshtel Algorithm (KA), ICA, and RDA. In another comparative study, Sahebjamnia et al. [24] illustrated a sustainable tire closed-loop supply chain network design problem for the first time. They solved their model using hybrid metaheuristics based on recent algorithms, i.e., Water Wave Optimization (WWO) and RDA. Similarly, considering uncertainty in this area, Fathollahi-Fard et al. [25] considered a multi-objective stochastic model to formulate a closed-loop supply chain network design considering social impacts. They solved this NP-*hard* problem by three hybrid optimizers based on KA and RDA. Concerning truck scheduling problems, Mohammadzadeh et al. [26] applied WWO, Virus Colony Search (VCS), and RDA to address a truck scheduling problem in a cross-docking system. They revealed that RDA outperformed other metaheuristics existing in the literature. Overall, the aforementioned papers [21–26] are only some examples among different other studies employing RDA and its different variants to show that RDA is one of the more successful recent optimizers than other recent and well-known ones to address different engineering design issues and NP-*hard* problems.

Similar to evolutionary optimizers, the RDA starts with a population of random solutions called Red Deers (RDs). These solutions are divided into two types: "male RDs" as the best solutions and "hinds" as the rest of solutions. Generally, roaring, fighting, and mating operations constitute the three main behaviors of RDs in the breeding season. First of all, male RDs roar strongly to show their power against other males to attract hinds. After that, some successful males are selected as the commanders based on their proven superiority (fitness of solutions). Another main action of this algorithm is the fighting process between the commanders and the rest of males called stags. During this competition, better solutions as the winners are chosen again as the commanders to form harems composed of a group of hinds. Based on the power of each commander, some hinds are selected to be in the harem. The greater the power of commander, the larger the number of hinds in the harem. After generating harems, an amazing mating behavior occurs. In this regard, first, the commanders should mate with a number of hinds in the harem and a few others in another harem to extent this territory. Then, the stags can mate with the nearest hind regardless of the restriction of harems. In terms of the evolutionary concepts in the RDA, some better solutions will be chosen as the next generation of this algorithm through roulette wheel selection or tournament selection mechanisms. At the end, the stop condition of this algorithm based on the maximum number of iterations should be satisfied.

Like other metaheuristics, a balance between exploitation and exploration phases is very important. RDA enjoys exploitation properties through the roaring and fighting of males as well as mating of stags with the nearest hinds. As such, the main exploration phase involves generating some harems for each commander, the mating operator with a harem, and a randomly selected one. To summarize the results of these procedures, Figure 1 reveals the flowchart of this metaheuristic. Accordingly, the blue boxes specify the intensification phase, whereas the red boxes maintain the diversification characteristics of this algorithm. For the green box, to break away from the local optimum solutions, the next generation

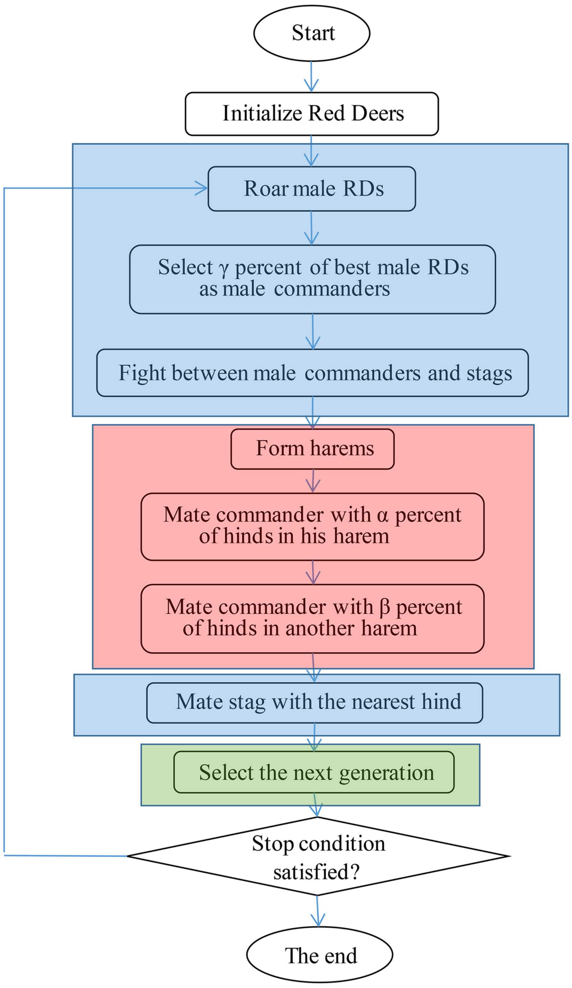

Start
Initialize Red Deers
Roar male RDs
Select γ percent of best male RDs as male commanders
Fight between male commanders and stags
Form harems
Mate commander with α percent of hinds in his harem
Mate commander with β percent of hinds in another harem
Mate stag with the nearest hind
Select the next generation
Stop condition satisfied?
The end

Figure 1. Flowchart of Red Deer Algorithm (RDA).

will be selected based on the evolutionary concept. To understand the RDA coding better, its pseudo-code is provided in Appendix A. For the simplicity of RDA and related formulas, readers are referred to the main source [6] and other related papers in this area [21–26].

By and large, RDA is characterized by more pros and cons than other recent metaheuristic algorithms. This algorithm is very good at balancing the exploration and exploitation phases [6]. The majority of case studies can obtain a global optimum rather than a local one [6,21–26]. The convergence rate of this metaheuristic is generally better than several well-known and recent optimizers such as ICA, GA, PSO, WWO, VCS, etc. based on recently published papers [21–26]. However, the main disadvantage of this algorithm is its difficulty to be tuned due to some controlling parameters. This study aims to find a way to reduce the main parameters of this algorithm using some adaptive strategies. To solve this drawback, the parameters and operators of this algorithm will be updated per iteration to search the feasible space better and improve its capability to find global solutions.

3. Improved RDA (IRDA)

Here, a novel metaheuristic based on the RDA as an improved version is proposed. In this regard, an adaptive version of RDA is developed to ease the calibration of parameters and improve its capability to search the feasible area better. Since the RDA has 6 input parameters to tune ($Maxit$ = the maximum number of iterations; $nPop$ = the number of population; $nMale$ = the number of males; $Hind = nPop - nMale$; $alpha$ = the percentage of mating in a harem; $beta$ = the percentage of mating of a commander with another in the harem; $gamma$ = the percentage of commandeers), this reason makes the tuning of RDA very difficult. The proposed IRDA not only uses an adaptive strategy to ease the tuning but also reveals a better performance than the original version.

In the proposed IRDA, there is a set of rules to formulate and value the number of parameters including alpha, beta, and gamma. First of all, the number of commanders should be planned. Each commander generates a harem. From another point of view, the number of commanders is the number of best solutions that promotes the intensification phase. However, the number of harems is directly affected by their diversification properties. Therefore, the number of commanders is very important in manipulating both of the search phases. Overall, the following formula has been proposed to update the percentage of commanders among all males:

$$gamma = \left(0.1 + 0.9 \times \frac{it}{Maxit}\right), \tag{1}$$

where it is the current iteration of running the algorithm. By considering the above formula, this number increased during the iterations. The minimum possible value of this parameter is 0.1. This rate fluctuates from the mentioned value to 1 as the highest value of this parameter. In our experiments, the user enjoys a chance to manipulate both exploitation and exploration properties. To calculate alpha and beta, an elitist strategy has been applied. The average of fitness for the hinds in this harem and that in a randomly selected harem have been computed. In case of minimization, if the average of fitness in the harem belonging to the considered commander is lower than another, it appears that the hinds of this harem are better than those of the other harem. Thus, the percentage of mating for the commander is computed as follows:

$$alpha = \left(0.5 + 0.5 \times \frac{it}{Maxit}\right). \tag{2}$$

Conversely, the beta is considered as follows:

$$beta = 1 - alpha. \tag{3}$$

Notably, if the average fitness of this harem is higher than that of another harem, the values of these two parameters should be exchanged. It means that the rate of beta should be calculated through Eq. (2). Similarly, Eq. (3) should be considered for calculating the rate of alpha. The main reason behind this strategy is to promote the intensification properties of the proposed methodology due to an increase in the number of iterations. To facilitate a better understanding of the proposed IRDA, the pseudo-code of the algorithm is reported in Appendix B. Notably, its flowchart is exactly similar to RDA in general, as seen in Figure 1.

Taken together, the developed IRDA has only three simple parameters: $Maxit$, $nPop$, and $nMale$. To update the other main parameters of RDA, an adaptive technique is taken into consideration to manage the search mechanism of this algorithm more efficiently. This improved metaheuristic enjoys higher efficiency than the original RDA based on the results provided in the next section.

4. Experimental results

In this section, the experimental analyses of the proposed IRDA for some standard benchmarked functions and an interesting engineering design issue called BLDC are provided. First of all, the datasets required to generate the benchmarks are reported. Consequently, the calibration of the parameters for algorithms is taken into consideration to enhance the performance of the metaheuristics to make a fair comparison. Afterwards, two comparative studies have evaluated the proposed IRDA with its original idea and the best existing results from the literature. Notably, all treatments have been run on the same computer with the same processors as those of previous studies. These computations have been run on a laptop with a Core 2 Duo-2.26 GHz processor.

4.1. Data sets

Since two different problems have been employed in this study, the definitions and settings of data are clearly specialized. Herein, the following sub-sections have been provided to address the employed benchmarked data comprehensively.

4.1.1. Standard benchmarked functions

Normally, to evaluate a novel optimizer, some standard benchmarked functions should be utilized. The literature reports that there are more than 60 different benchmarked functions [2–6]. In this regard, 12 standard benchmarked functions taken from the Congress on Evolutionary Computation (CEC) and adopted from recent studies have been chosen [2],[5,6]. They are numbered as P1 to P12. All of them belong to an optimized case of minimization. The global optimum for all treatments is zero. Their formulas and distributions are reported in Appendix C (as can be seen in Table C1). Furthermore, to analyze the proposed algorithms in different cases, both low- and high-dimensional situations have been utilized with 30 and 100 decision variables, respectively. Based on their characteristics, they are divided into unimodal, multimodal, and separable and non-separable classifications. Since these test problems have been addressed in most of previous works, more details of the used benchmarked functions can be found in [2–6].

4.1.2. BLDC motor structure

The proposed BLDC design motor in this study was benchmarked in [19,27]. Accordingly, the definitions and related formulations of the considered BLDC motor as well as the related design factors can be referred to [19,27]. From an overall view, Figure 2 shows the studied BLDC motor in this study. Generally, there are 10 important decision variables for the considered BLDC motor. The notations, definitions, and feasible ranges of these variables are reported in Appendix D (see Table D1).

4.2. Tuning of optimizers

Given that both RDA and IRDA have some controlling parameters, they need to be tuned satisfactorily. If a metaheuristic is not calibrated very well, its behavior will not be efficient [28–30]. In this subsection, these optimizers are tuned for each problem, comprehensively [31–34]. Accordingly, each parameter associated with the algorithms should be considered as a factor accompanied by its corresponding level (suggested values). Regarding the RDA, there are 6 factors and for each of them, 3 levels are considered. Based on the full-factorial calibration, there are $3^6 = 729$ treatments for each run. According to the IRDA, there are only 3 factors and for each of them, 4 levels are suggested. From each run, there are $4^3 = 64$ in total. In addition, due to the stochastic nature of metaheuristics and the reliability of results, for each treatment, it is required to run the experiment for at least 30 times. Therefore, there are many run times for each metaheuristic in each problem. To handle this dilemma, Taguchi experimental design method [28] offers a set of orthogonal arrays to shorten the run times and tune the algorithms in an efficient way. Since the Taguchi experimental design method for calibration is a well-known methodology and it has been employed in many papers, more explanations about this method have not been provided for the sake of brevity; thus, for more information, one should refer to [1,21,25,32–34].

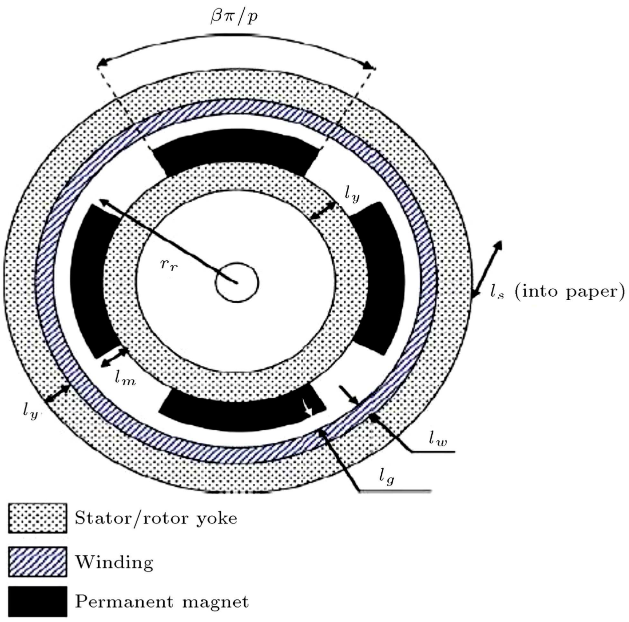


Figure 2. Brushless DC (BLDC) motor structure considered by this study [27].

This methodology uses a transformative regulation to measure the variation of algorithms among the best existing cases to select a robust parameter design of the optimizers [35,36]. Overall, there are two well-known transformation measurements in this method: Signal-to-Noise (S/N) and Relative Percentage Deviation (RPD). S/N denotes the variation value of the response variables of Taguchi. In this regard, the higher value of S/N is more preferable. In case of minimization, the following formula can be considered:

$$S/N = -10\log(Z^2), \tag{4}$$

where Z is the response variable adopted from the objective function of the problem. Similarly, RPD is considered to specify the efficiency of algorithms. The lower value of RPD is more preferable. In case of minimization, it can be formulated as follows:

$$RPD = \frac{\mathrm{Alg}_{sol} - Best_{sol}}{Best_{sol}}, \tag{5}$$

where $Best_{sol}$ is the minimum solution ever found during 30 run times and Alg_{sol} is the solution for each run. Notably, the average of RPD for 30 runs will be used for the final evaluation of collaborations between the algorithms.

Overall, the parameters associated with the algorithms and their candidate levels are given in Table 1. Finally, to perform the treatments, the orthogonal arrays should be considered in decreasing the total number of experiments. Regarding RDA, L_{27} is selected. Accordingly, the number of experiments is reduced from 729 to 27. As such, Taguchi proposes L_{16} for the IRDA. It appears that the total number of treatments will decrease from 64 to 16. Overall, it can be observed that the Taguchi method helps users save time to do the tuning.

Since there are two different optimization problems, both optimizers should be tuned for each problem, separately. For the case of BLDC motor deign problem, the results of mean RPD and S/N ratio for both of the proposed algorithms are provided in Figures 3–6. In conclusion, the calibrated parameters for both optimizers are given in Table 2.

4.3. Evaluation with standard benchmarked functions

Here, a comparative study based on standard benchmarks has been applied. As mentioned before, 12 standard benchmarks have been adopted from the literature. This comparison has been inspired by the work of Fathollahi-Fard et al. [2]. Accordingly, there are a number of well-known (i.e., GA, SA, and PSO), state-of-the-art (i.e., L-SHADE), and recent (i.e., Artificial Bee Colony (ABC), ICA, Firefly Algorithm (FA), and RDA) optimizers to make this comparison. Due to 30 run times, the best, worst, average, and the

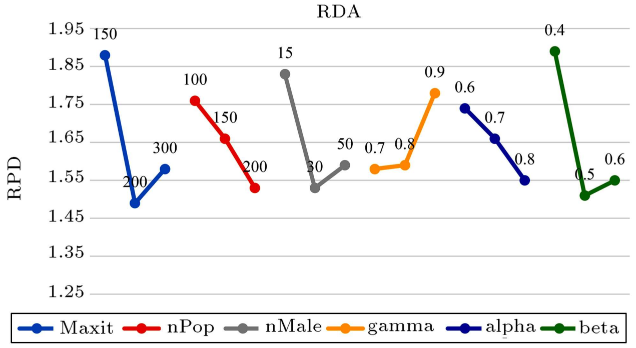


Figure 3. Mean Relative Percentage Deviation (RPD) for Red Deer Algorithm (RDA).

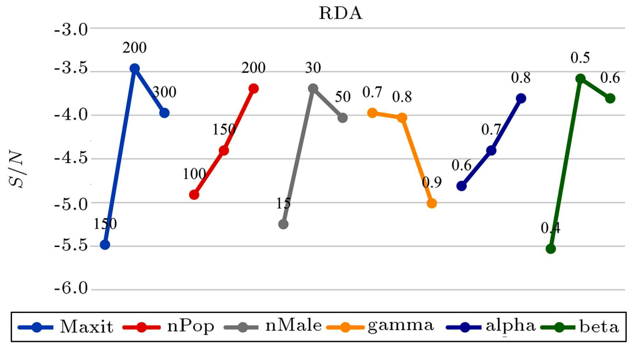


Figure 4. S/N ratio for Red Deer Algorithm (RDA).

Table 1. The factors associated with optimizers and their levels.

Optimizer	Factor	Levels			
		1	2	3	4
RDA	A: Maximum iteration ($Maxit$)	150	200	300	–
	B: Number of population ($nPop$)	100	150	200	–
	C: Number of males ($nMale$)	15	30	50	–
	D: Percentage of commander ($gamma$)	0.7	0.8	0.9	–
	E: Percentage of mating in each harem ($alpha$)	0.6	0.7	0.8	–
	F: Percentage of mating of commander in another harem ($beta$)	0.4	0.5	0.6	
IRDA	A: Maximum iteration ($Maxit$)	100	200	300	500
	B: Number of population ($nPop$)	50	100	150	200
	C: Number of males ($nMale$)	20	30	50	75

Table 2. Tuned values of optimizers.

Optimizer	Calibration for standard functions	Calibration for BLDC motor
RDA	$Maxit = 300; nPop = 150; nMale = 30;$ $gamma = 0.7; alpha = 0.7; beta = 0.5$	$Maxit = 200; nPop = 200; nMale = 30;$ $gamma = 0.7; alpha = 0.8; beta = 0.5$
IRDA	$Maxit = 300; nPop = 150; nMale = 50$	$Maxit = 500; nPop = 200; nMale = 50$

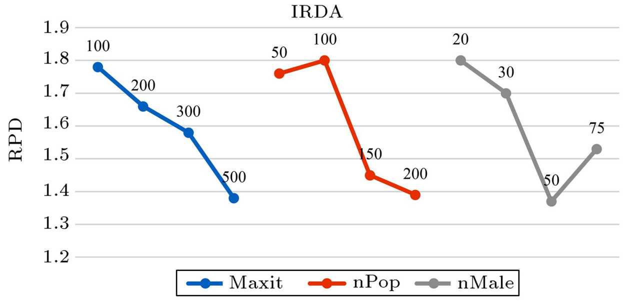


Figure 5. Mean Relative Percentage Deviation (RPD) for Improved RDA (IRDA).

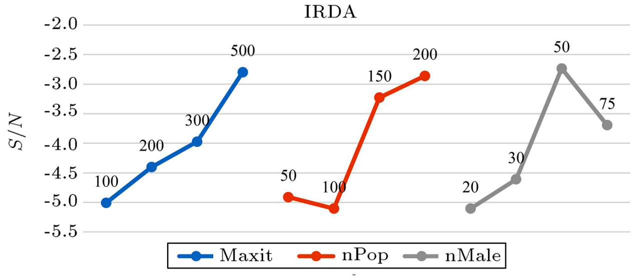


Figure 6. S/N ratio for Improved RDA (IRDA).

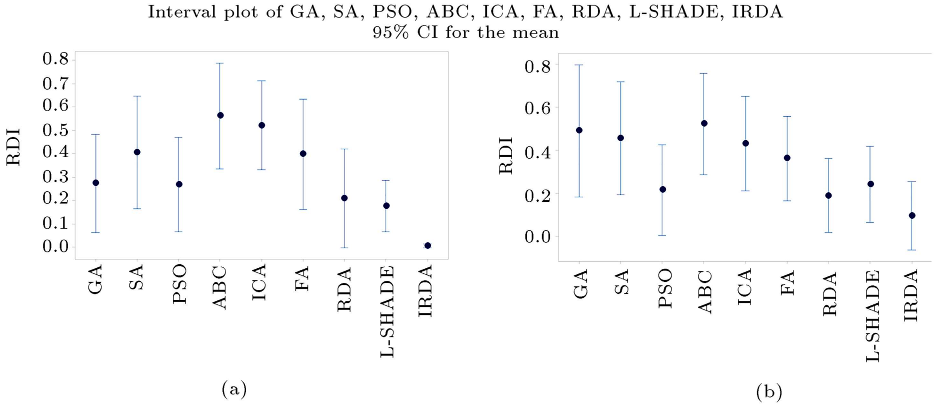
standard deviation employed to evaluate the robustness of algorithms are computed. Based on these results, the algorithms are ranked. All reports are given in Table 3. For the final evaluation, a well-known transformation metric is utilized to perform a set of statistical analyses. Based on the standard deviation, the results of algorithms are computed by the Relative Deviation Index (RDI) utilized according to the following equation:

$$RDI = \frac{|\text{Alg}_{sol} - Best_{sol}|}{Max_{sol} - Min_{sol}}, \tag{6}$$

where Max_{sol} and Min_{sol} are the maximum and minimum values of all algorithms' solutions, respectively. In case of minimization, $Best_{solis}$ the minimum value among all outputs. As such, Alg_{sol} is the solution of the applied algorithms. Notably, the lower value of RDI ensures better performance, ranging between zero and one. Overall, the interval plots based on the Least Significant Differences (LSD) for the RDI of optimizers in both low (i.e., 30 variables) and high (i.e., 100 variables) dimensions are given in Figure 7.

According to the results provided by Table 3, it can be observed that the IRDA will obtain the global solution in five benchmarks. Based on the final rank of optimizers, IRDA shows 1.69 as the best rank. Most of RDA outputs are shown in bold. It can be implied that this algorithm reveals a set of robust solutions in a majority of benchmarks. From the low-dimensional perspective, IRDA achieved the best results; similarly, from the high-dimensional perspective, it provided very competitive solutions. Of note, a general RDA exhibits the best behavior.

According to Figure 7, the statistical analyses have been performed using RDI metric. At first, in low dimensions (Figure 7(a)), there is a clear difference between the performance of the proposed algorithm



Individual standard deviations are used to calculate the intervals

Figure 7. Interval plot and Least Significant Differences (LSD) chart based on the Relative Deviation Index (RDI) for low (a) and high (b) dimensional evaluation.

Table 3. Comparison of optimizers based on the benchmark functions during 30 run times (B = best, W = worst, M = average, SD = standard deviation, D = dimension, R = rank).

Function	D		[2]								This study
			GA	**SA**	**PSO**	**ABC**	**ICA**	**FA**	**RDA**	**L-SHADE**	**IRDA**
P1	30	W	9.58E-02	2.78E-01	6.37E-03	8.37E-01	1.37E-02	4.93E-0 4	3.52E-04	1.92E-02	**1.85E-05**
		M	8.16E-07	2.15E-02	5.78E-05	2.51E-02	5.36E-03	6.91E-05	1.73E-06	1.17E-07	**4.27E-08**
		B	2.19E-08	4.28E-03	1.89E-06	3.16E-04	2.51E-04	3.78E-08	2.86E-11	5.48E-09	**3.81E-11**
		SD	0.004783	0.050924	0.000972	0.46176	0.393635	0.000534	0.000136	1.20E-03	**4.62E-04**
		R	5	9	6	7	8	4	2	3	1
	100	W	1.22	4.78	3.65E-02	2.42	5.84	2.76E-01	**4.28E-03**	2.39E+00	3.82E-02
		M	4.67E-03	0.8514	8.35E-01	0.7534	4.32	5.47E-04	5.95E-04	1.22E-01	**5.27E-04**
		B	6.18E-04	0.0251	4.28E-03	0.4627	2.17	2.81E-05	1.53E-05	8.37E-03	**5.37E-07**
		SD	**0.01854**	1.0894	0.06854	1.0325	3.2817	0.05894	0.07634	2.18E-01	6.27E-01
		R	4	7	5	8	9	3	2	6	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	6.45E-04	5.81E-04
		M	6.82E-07	5.32E-05	5.73E-09	2.68E-05	2.17E-06	3.18E-07	3.85E-08	8.87E-06	**6.28E-09**
		B	2.18E-09	3.11E-06	4.35E-10	3.19E-06	3.12E-07	2.57E-08	1.25E-11	4.44E-07	**5.12E-13**
		SD	3.17E-02	4.28E-02	3.47E-04	2.64E-03	8.53E-03	4.81E-03	5.73E-04	1.43E-02	**3.82E-05**
		R	4	8	3	9	6	5	2	7	1
	100	W	2.81E-01	5.18E-02	**5.38E-04**	3.18E-02	5.15E-02	1.75E-02	3.27E-01	1.73E-02	6.72E-01
		M	2.57E-05	7.48E-03	**8.13E-07**	7.29E-03	3.59E-04	3.21E-04	4.38E-06	2.49E-03	5.73E-06
		B	6.36E-07	1.94E-04	5.42E-08	5.42E-05	2.88E-05	2.72E-06	5.74E-08	2.77E-05	**6.11E-10**
		SD	5.28E-02	6.53E-01	1.82E-03	4.18E-01	3.15E-02	3.22E-03	3.16E-03	1.31E-01	**4.81E-04**
		R	4	9	2	8	7	5	3	6	1
P3	30	W	2.16E-02	4.17E-04	**2.54E-06**	3.17E-04	2.16E-03	3.15E-02	5.93E-03	7.93E-05	6.72E-06
		M	3.63E-09	2.64E-10	3.72E-07	2.62E-05	3.71E-06	4.71E-06	8.15E-07	4.37E-06	**6.84E-13**
		B	1.22E-11	3.14E-12	5.01E-10	4.83E-08	4.85E-08	1.85E-09	3.15E-12	9.66E-09	**0**
		SD	7.52E-03	8.53E-03	2.17E-02	5.77E-03	3.15E-03	2.71E-02	8.52E-03	1.44E-03	**4.37E-06**
		R	4	2	5	8	9	6	3	7	1
	100	W	2.45E-01	2.51E-03	**7.18E-05**	6.43E-02	4.36E-02	2.58E-01	4.26E-02	9.19E-03	3.85E-03
		M	3.81E-08	**2.78E-09**	3.18E-06	3.17E-04	1.27E-04	2.83E-05	4.27E-06	1.06E-04	2.86E-08
		B	2.51E-09	3.22E-10	5.17E-09	1.68E-06	3.19E-06	7.12E-07	1.29E-09	2.40E-07	**1.33E-10**
		SD	4.87E-01	2.56E-02	**2.15E-04**	1.25E-01	1.78E-01	2.16E-01	2.81E-02	1.56E-02	5.16E-03
		R	5	2	4	8	9	6	3	7	1
P4	30	W	78.93	8.18	22.91	9.54	6.19	7.52	5.01	**1.19**	2.86
		M	10.53	1.2794	5.73	0.8659	0.8134	0.5894	**4.82E-04**	1.08E-01	2.79E-03
		B	3.8217	2.81E-02	3.91E-02	1.29E-02	5.28E-03	2.71E-05	**2.77E-06**	3.23E-03	2.85E-05
		SD	9.4521	3.62	8.61	4.28	3.994	2.71E-00	3.82E-01	1.43E+00	**3.44E-02**
		R	9	7	8	6	5	2	1	4	3
	100	W	95.62	10.64	34.17	15.85	12.95	11.37	**7.83**	3.19E+01	8.943
		M	11.84	4.26	7.95	3.1854	5.8728	0.8623	3.61E-02	5.92	**2.81E-02**
		B	6.842	3.81E-01	2.18	1.6809	0.7854	3.72E-03	**5.87E-05**	1.14	5.81E-03
		SD	24.81	3.72	2.0823	1.7905	2.6847	3.78	1.9923	8.27	**3.85E-01**
		R	9	4	8	6	5	2	1	7	3
P5	30	W	98.47	95.27	99.15	114.46	119.76	102.54	87.54	**39.9**	48.6
		M	28.65	**25.94**	44.75	98.72	91.45	82.18	29.58	35.7	38.76
		B	19.75	18.75	32.15	30.99	31.58	30.85	24.61	**15.8**	25.67
		SD	44.87	**30.98**	486.74	1845.62	6254.91	1392.57	196.74	2.08E+03	89.71
		R	3	2	9	6	8	7	4	1	5
	100	W	123.25	105.89	114.95	156.75	142.68	116.75	95.47	**57.5**	59.86
		M	32.67	29.81	52.68	115.63	99.74	86.53	31.47	**37.6**	44.52
		B	20.64	19.41	36.27	32.16	34.82	33.81	26.91	**15.1**	38.93
		SD	**56.14**	32.8162	549.62	1974.25	754.38	1473.89	215.48	215.3	162.37
		R	3	2	8	5	7	6	4	1	9
P6	30	W	3	3	1	3	7	0	0	0	0
		M	0.0008	1.89E-03	8.24E-18	2.35E-09	2.57E-01	0	0	0	0
		B	0	0	0	0	0	0	0	0	0
		SD	0	6.045955	7.348469	62.7283	18.54724	5.95219	1.06066	0	0
		R	3	6	7	9	8	5	4	1	1
	100	W	24	21	57	87	23	13	21	1.90E+01	7
		M	2.65E-02	1.75E-01	2.64E-04	2.19E-05	2.16E-00	1.75E-15	**3.72E-10**	8.80E-05	7.83E-08
		B	1	1	1	1	3	**0**	1	**0**	**0**
		SD	1.65E-01	2.85E-02	1.87E-01	3.28E-02	1.57E-01	3.82E-02	3.16E-02	6.23E-02	**5.71E-06**
		R	8	6	9	4	7	2	5	3	1

The best values are shown in bold.

Table 3. Comparison of optimizers based on the benchmark functions during 30 run times (B = best, W = worst, M = average, SD = standard deviation, D = dimension, R = rank) (continued).

Function	**D**		[2]								**This study**
			GA	**SA**	**PSO**	**ABC**	**ICA**	**FA**	**RDA**	**L-SHADE**	**IRDA**
P7	30	W	13.87	14.56	13.87	13.69	12.54	9.76	9.81	11.75	**4.78**
		M	7.89	10.25	10.57	10.86	10.25	8.46	8.26	9.86	**1.75**
		B	6.54	7.81	7.89	6.47	8.92	7.15	6.38	7.65	**2.75E-01**
		SD	1.584	2.71	1.47	1.5843	1.3672	1.91	2.685	1.36	**0.89**
		R	4	7	8	3	9	5	2	6	1
	100	W	18.95	17.24	15.49	15.86	14.74	12.85	**10.86**	12.85	10.93
		M	8.918	11.1288	11.92	11.63	10.54	9.86	9.24	11.76	**6.82**
		B	8.25	8.25	9.63	8.25	8.18	8.82	8.10	10.22	**1.85**
		SD	1.67	2.86	1.75	1.79	1.64	2.56	2.89	3.11	**1.52**
		R	4	6	8	5	3	7	2	9	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	4.90E-03	**5.82E-05**
		M	3.72E-05	1.46E-04	5.37E-06	1.48E-04	5.78E-04	6.24E-07	5.87E-05	7.30E-05	**3.81E-08**
		B	2.76E-07	5.81E-06	2.14E-08	3.27E-09	3.11E-08	6.82E-11	5.46E-09	2.91E-06	**0**
		SD	2.81E-03	8.26E-02	3.81E-02	2.64E-02	4.18E-02	1.82E-03	7.53E-04	4.13E-02	**3.17E-05**
		R	7	9	5	3	6	2	4	8	1
	100	W	1.53E-02	8.53E-01	2.84E-02	4.17E-01	5.83E-01	8.42E-03	5.17E-02	2.92E-01	**3.82E-03**
		M	1.23E-04	3.32E-06	6.54E-06	1.54E-02	4.83E-02	1.28E-04	**4.37E-07**	2.42E-02	1.58E-05
		B	5.92E-04	1.48E-08	4.79E-07	5.38E-05	5.92E-04	6.19E-05	8.29E-08	2.96E-04	**0.84E-12**
		SD	0.000564	1.03E-05	0.000109	0.284115	0.088928	0.036815	0.000181	2.96E-02	**5.82E-06**
		R	7	2	4	5	8	9	3	6	1
P9	30	W	43.82	42.81	39.81	38.29	43.81	35.92	32.18	**19.1**	20.85
		M	18.36	30.92	22.59	28.11	17.85	29.32	16.931	14.1	**12.64**
		B	17.4938	17.4938	12.4821	22.5918	16.3917	16.3917	8.5474	7.51	**5.87**
		SD	9.461557	4.940765	12.68426	10.67775	18.2485	25.73	20.563	**3.56**	4.56
		R	8	7	4	8	5	6	3	2	1
	100	W	52.86	59.15	48.72	51.23	52.19	43.64	42.58	**29.6**	32.57
		M	28.71	34.82	27.91	30.82	24.71	32.81	19.76	**17.4**	19.84
		B	16.83	20.73	18.91	18.16	12.76	14.83	11.56	10.4	**10.27**
		SD	8.745	9.15	7.84	6.32	8.182	7.93	6.15	**3.05**	8.39
		R	6	9	8	7	4	5	3	2	1
P10	30	W	3.82E-02	0.8126	6.81E-01	2.71	1.28	3.88E-03	8.53E-04	9.03E-01	**5.82E-06**
		M	6.32E-04	1.86E-02	4.82E-02	7.81E-02	4.82E-02	5.82E-05	6.26E-07	3.91E-02	**6.82E-10**
		B	1.54E-06	3.66E-04	5.77E-04	2.97E-02	8.31E-02	5.88E-08	1.43E-09	9.90E-03	**0**
		SD	2.78E-02	1.66E-01	3.22E-02	4.37E-01	2.36E-01	6.82E-02	3.82E-03	1.46E-01	**4.82E-04**
		R	4	5	6	8	9	3	2	7	1
	100	W	2.867	7.84	2.78	6.34	3.91	0.684	0.523	2.61E+00	**1.86E-03**
		M	2.91E-02	3.82E-01	3.82E-02	4.71E-02	5.81E-02	1.57E-04	**5.81E-06**	1.91E-01	4.86E-05
		B	2.84E-04	1.94E-02	8.24E-03	3.66E-04	2.55E-05	9.51E-06	8.94E-07	9.70E-03	**3.86E-07**
		SD	3.71E-01	4.14E-01	5.11E-02	3.81E-01	2.48E-02	1.753-03	2.51E-02	2.07E-01	**4.61E-03**
		R	5	9	7	6	4	3	2	8	1
P11	30	W	0.85	0.85	0.81	0.43	0.875	0.36	0.89	4.05E-01	**5.92E-03**
		M	5.92E-03	2.81E-03	5.72E-03	6.81E-03	1.92E-04	6.42E-04	8.92E-03	2.86E-03	**6.81E-09**
		B	7.83E-05	6.83E-05	5.82E-04	7.22E-05	7.61E-06	4.91E-06	7.61E-06	1.94E-04	**0**
		SD	0.0438	0.025072	0.001411	0.0619	0.08662	0.071474	0.027377	7.06E-04	**5.16E-05**
		R	7	5	9	6	3	2	4	8	1
	100	W	4.81	2.75	7.84	4.92	5.37	2.81	3.97	2.69	**5.28E-01**
		M	2.67E-01	8.43E-01	8.53E-01	5.76E-01	6.89E-01	4.76E-02	1.85E-02	3.45E-01	**3.82E-06**
		B	4.82E-04	4.82E-03	4.11E-03	5.92E-03	3.88E-04	5.82E-04	6.11E-04	1.29E-04	**2.66E-10**
		SD	0.83	0.57	0.0411	0.2119	0.366	0.274	0.377	1.83E-01	**4.82E-02**
		R	4	8	7	9	3	5	6	2	1
P12	30	W	2.51	1.45	8.29E-01	1.25	1.45	1.25	**5.82E-02**	7.25E-01	5.83E-02
		M	5.62E-07	3.78E-02	7.88E-05	3.82E-03	7.82E-05	5.83E-02	6.84E-09	3.91E-05	**6.18E-09**
		B	8.99E-09	5.86E-05	7.81E-06	5.18E-04	4.29E-04	6.23E-04	2.84E-11	2.15E-04	**0**
		SD	**0.0048**	12.647	1.575753	52.6121	21.88676	26.9302	0.899861	7.30E+00	4.81E-03
		R	3	5	4	8	7	9	2	6	1
	100	W	18.74	8.95	4.73	9.12	10.96	2.85	**0.9372**	2.98E+00	1.86
		M	1.62E-06	0.3447	2.38E-04	1.54E-02	5.66E-01	9.79E-01	**1.26E-07**	1.72E-01	5.27E-05
		B	6.39E-08	0.0083	7.38E-05	6.38E-04	4.29E-04	6.19E-03	6.39E-08	2.77E-03	**2.44E-09**
		SD	0.138	18.447	11.5753	59.21	45.86	34.02	0.9861	6.15E+00	0.3982
		R	2	9	4	6	5	8	3	7	1
Average of Rank			5.130435	6	6.086957	6.73913	6.304348	4.869565	2.956522	5.130435	**1.695652**

The best values are shown in bold.

and those of other optimizers. The other algorithms reveal a set of similarities between their performances. Among them, L-SHADE functions generally better than others except IRDA. In this case, ABC is the worst optimizer. According to high dimensions (Figure 7(b)), there are some similarities concerning the efficiency of optimizers. Although the proposed IRDA outperforms the others, each of the other algorithms achieved a comparable result. Followed by IRDA, PSO achieved the best result. GA exhibited the weakest behavior, too. Taken together, the proposed IRDA is completely better than its original idea in both of the low- and high-dimensional assessments.

In conclusion, the proposed IRDA not only is of the best rank but also finds the global optimum in most of the cases. The statistical analyses confirm that the developed methodology is generally better than all algorithms from the literature.

4.4. Comparison of applied optimizers in the BLDC motor design problem

To assess the proposed IRDA in a real-world engineering design problem, a comparative study based on the BLDC motor was employed. In this subsection, based on GA and PSO applied by Rahideh et al. [27] and the COA employed by Azari et al. [19], both of the proposed RDA and IRDA were considered in making this comparison. Hence, the design factors associated with the BLDC motor were benchmarked to evaluate the best optimal solution found by RDA and IRDA in this study. A feasible range of designs is given in Appendix D. Similar to previous treatments, each optimizer was run 30 times to ensure the reliability of algorithm results. The final results of design factors are given in Table 4. In all factors except F1, F2, and F9, the lower value ensures a better capability for the algorithm. Otherwise, regarding these three factors, the higher value is more preferable. From the best optimal results provided, a statistical analysis based on RDI metric similar to the last subsection was conducted, too. Hence, the behaviors of all optimizers, i.e., GA, PSO, COA, RDA, and IRDA, are given in Figure 8.

Table 4 shows that the proposed optimizers clearly outperform the previous ones existing in the literature. Except GA in two factors, RDA and IRDA achieved the best values among other algorithms.

Figure 8 indicates a set of clear differences between the performance of two considered optimizers and other algorithms. The behaviors of GA, PSO, and COA share a set of similarities among them. However, the PSO is slightly better than GA and COA.

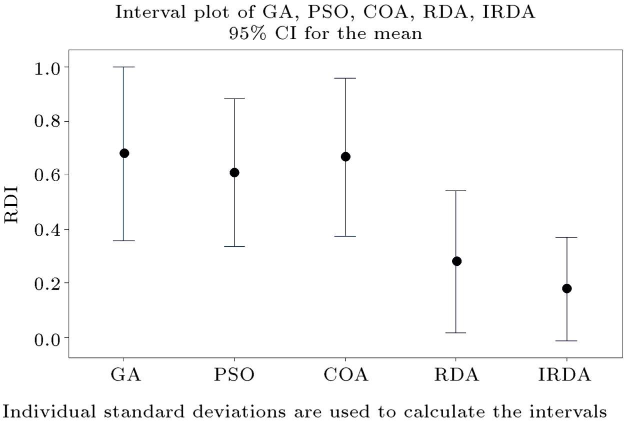


Figure 8. Interval plot for comparing the algorithms for Brushless DC (BLDC) motor based on Relative Deviation Index (RDI).

Table 4. Comparison of optimal values of Brushless DC (BLDC) design factors.

No. of factors	[27]		[19]	This study	
	GA	PSO	COA	RDA	IRDA
F1	5	5	5	**5.255**	5.1566
F2	2	1.9982	1.9551	1.9982	**2**
F3	0.7	0.6950	0.6912	0.6944	**0.6855**
F4	0.0130	0.0124	0.0120	0.0116	**0.0110**
F5	0.0060	0.0058	0.0081	**0.002**	0.0054
F6	0.0035	0.0034	0.0033	0.002	**0.0018**
F7	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
F8	0.0595	0.059	0.0592	0.049	**0.021**
F9	**5800000**	5784573	5819800	5673183	**5800000**
F10	0.0756	0.0732	0.0730	**0.05**	0.068

The best values are shown in bold.

As such, the behaviors of RDA and IRDA are the same. However, it can be observed that the developed IRDA is clearly better than its general idea in this comparison, too.

5. Conclusion and future remarks

The present paper proposed an enhanced version of the recently developed Red Deer Algorithm (RDA). The main contribution here was to introduce a set of adaptive strategies for the main parameters of RDA to facilitate the calibration of parameters and ensure a better interaction between the search phases, i.e., exploration and exploitation, in metaheuristics. To compare the efficiency of the proposed optimizers, not only 12 standard benchmarked functions were utilized but also a real engineering design issue concerning Brushless DC (BLDC) motor was applied. After introducing the general idea of RDA and its literature, the proposed improvement based on adaptive strategies was detailed. To ensure a fair comparison, the algorithms' calibration was addressed by Taguchi method. A comparative study of the best existing results from the literature and a comprehensive statistical analysis were carried out to highlight the efficiency of the proposed Improved RDA (IRDA). The developed methodology achieved global solutions in the majority of previous studies. It can reduce the time consumption and provide better efficiency than other algorithms. According to the results, discussion, and the main findings of this paper, it can be concluded that the proposed IRDA significantly outperforms its original idea and other existing algorithms in the literature.

This study can open several new directions for future works. It is worth employing the proposed IRDA in different other standard functions and additional real-world engineering design problems. The proposed IRDA may be hybridized with other traditional and recent optimizers to improve its efficiency. It is required to test the calibration of parameters for other engineering applications. Considering the effect of tuning of IRDA in other cases would be interesting as a continuation of this study.

References

1. Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., and Tavakkoli-Moghaddam, R. "A bi-objective green home health care routing problem", *Journal of Cleaner Production*, **200**, pp. 423–443 (2018).

2. Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., and Tavakkoli-Moghaddam, R. "The Social Engineering Optimizer (SEO)", *Engineering Applications of Artificial Intelligence*, **72**, pp. 267–293 (2018).

3. Mirjalili, S. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm", *Knowledge-Based Systems*, **89**, pp. 228–249 (2015).

4. Mirjalili, S. "SCA: a sine cosine algorithm for solving optimization problems", *Knowledge-Based Systems*, **96**, pp. 120–133 (2016).

5. Ghorbani, N. and Babaei, E. "Exchange market algorithm", *Applied Soft Computing*, **19**, pp. 177–187 (2014).

6. Fathollahi Fard, A.M. and Hajighaei-Keshteli, M. "Red deer algorithm (RDA); a new optimization algorithm inspired by red deer's mating", *International Conference on Industrial Engineering, IEEE*, **12**, pp. 331–342 (2016).

7. Faris, H., Mafarja, M.M., and Heidari, A.A., et al. "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems", *Knowledge-Based Systems*, **154**, pp. 43–67 (2018).

8. Duan, H. and Gan, L. "Orthogonal multiobjective chemical reaction optimization approach for the brushless DC motor design", *IEEE Transactions on Magnetics*, **51**(1), pp. 1–7 (2015).

9. Lee, T.-Y., Trung, P.X., Kim, J.-W., Kim, Y.-J., and Jung, S.-Y. "Search region management method for local search algorithm employing design optimization of brushless dc motor", *IEEE Transactions on Magnetics*, **52**(3), pp. 1–6 (2016).

10. Wolpert, D.H. and Macready, W.G. "No free lunch theorems for optimization", *IEEE Transaction on Evolutionary Computation*, **1**(1), pp. 67–82 (1997).

11. Fathollahi Fard, A.M. and Hajiaghaei-Keshteli, M. "A bi-objective partial interdiction problem considering different defensive systems with capacity expansion of facilities under imminent attacks", *Applied Soft Computing*, **68**, pp. 343–359 (2018).

12. Ayala, H.V., Segundo, E.H., Mariani, V.C., and Coelho, L.D.S. "Multiobjective krill herd algorithm for electromagnetic optimization", *IEEE Transactions on Magnetics*, **52**(3), pp. 31–48 (2016).

13. Ishikawa, T., Yonetake, K., and Kurita, N. "An optimal material distribution design of brushless DC motor by genetic algorithm considering a cluster of material", *IEEE Transactions on Magnetics*, **47**(5), pp. 1310–1313 (2011).

14. Son, B., Park, G.-J., Kim, J.-W., Kim, Y.-J., and Jung, S.-Y. "Interstellar search method with mesh adaptive direct search for optimal design of brushless DC motor", *IEEE Transactions on Magnetics*, **52**(3), pp. 1–4 (2011).

15. Yoon, K.-Y. and Kwon, B.-I. "Optimal design of a new interior permanent magnet motor using a flared-shape arrangement of ferrite magnets", *IEEE Transactions on Magnetics*, **52**(7), pp. 11–24 (2016).

16. Kim, H.-S., You, Y.-M., and Kwon, B.-I. "Rotor shape optimization of interior permanent magnet BLDC motor according to magnetization direction", *IEEE Transactions on Magnetics*, **49**(5), pp. 2193–2196 (2013).

17. Liu, X., Hu, H., Zhao, J., Belahcen, A., Tang, L., and Yang, L. "Analytical solution of the magnetic field and emf calculation in ironless BLDC motor", *IEEE Transactions on Magnetics*, **52**(2), pp. 1–10 (2016).

18. Lee, T.-Y., Seo, M.-K., Kim, Y.-J., and Jung, S.-Y. "Motor design and characteristics comparison of outer-rotor-type BLDC motor and blac motor based on numerical analysis", IEEE Transactions on Applied Superconductivity, **26**(4), pp. 1–6 (2016).

19. Azari, M.N., Samani, M., and Pahnekolaie, S.M.A. "Optimal design of a Brushless DC motor, by using the Cuckoo Optimization", *International Journal of Engineering, Transaction B: Applications*, **30**(5), pp. 668–677 (2017).

20. Xu, X. and Deng, Y. "UAV power component-DC brushless motor design with merging adjacent-disturbances and integrated-dispatching pigeon-inspired optimization", *IEEE Transactions on Magnetics*, **56**(5), pp. 1–7 (2018).

21. Golmohamadi, S., Tavakkoli-Moghaddam, R., and Hajiaghaei-Keshteli, M. "Solving a fuzzy fixed charge solid transportation problem using batch transferring by new approaches in meta-heuristic", *Electronic Notes in Discrete Mathematics*, **58**, pp. 143–150 (2017).

22. Samadi, A., Mehranfar, N., Fathollahi Fard, A.M., and Hajiaghaei-Keshteli, M. "Heuristic-based metaheuristic to address a sustainable supply chain network design problem", *Journal of Industrial and Production Engineering*, **35**(2), pp. 102–117 (2018).

23. Hajiaghaei-Keshteli, M. and Fathollahi Fard, A.M. "Sustainable closed-loop supply chain network design with discount supposition", *Neural Computing and Applications*, **31**(9), pp. 5543–5537 (2019).

24. Sahebjamnia, N., Fathollahi-Fard, A.M., and Hajiaghaei-Keshteli, M. "Sustainable tire closed-loop supply chain network design: Hybrid metaheuristic algorithms for large-scale networks", *Journal of Cleaner Production*, **196**, pp. 273–296 (2018).

25. Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., and Mirjalili S. "Multi-objective stochastic closed-loop supply chain network design with social considerations", *Applied Soft Computing*, **71**, pp. 505–525 (2018).

26. Mohammadzadeh, H., Sahebjamnia, N., Fathollahi-Fard, A.M., and Hajiaghaei-Keshteli, M. "New approaches in Metaheuristics to solve the truck scheduling problem in a cross docking center", *International Journal of Engineering, Transaction B: Applications*, **31**(8), pp. 1258–1266 (2018).

27. Rahideh, A., Korakianitis, T., Ruiz, P., Keeble, T., and Rothman, M. "Optimal brushless DC motor design using genetic algorithms", *Journal of Magnetism and Magnetic Materials*, **322**(22), pp. 3680–3687 (2010).

28. Taguchi, G., *Introduction to Quality Engineering: Designing Quality Into Products and Processes*, Springer, No. 658.562 T3 (1986).

29. Tian, G., Ren, Y., Feng, Y., Zhou, M., Zhang, H., and Tan, J. "Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony", *IEEE Transactions on Industrial Informatics*, **15**, pp. 2456–2468 (2019).

30. Esfandiyari, Z., Bashiri, M., and Tavakkoli-Moghaddam, R. "Resilient network design in a location-allocation problem with multi-level facility hardening", *Scientia Iranica*, **26**(2), pp. 996–1008 (2019).

31. Tian, G., Zhou, M., and Li, P. "Disassembly sequence planning considering fuzzy component quality and varying operational cost", *IEEE Transactions on Automation Science and Engineering*, **15**, pp. 748–760 (2018).

32. Safaeian, M., Fathollahi-Fard, A.M., Tian, G., et al. "A multi-objective supplier selection and order allocation through incremental discount in a fuzzy environment", *Journal of Intelligent & Fuzzy Systems*, **37**(1), pp. 1435–1455 (2019).

33. Fu, Y., Tian, G., Fathollahi-Fard, A.M., et al. "Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint", *Journal of Cleaner Production*, **226**, pp. 515–525 (2019).

34. Abdi, A., Abdi, A., Fathollahi-Fard, A.M., et al. "A set of calibrated metaheuristics to address a closed-loop supply chain network design problem under uncertainty", *International Journal of Systems Science: Operations & Logistics*, **8**(1), pp. 23–40 (2021).

35. Bahadori-Chinibelagh, S., Fathollahi-Fard, A.M., and Hajiaghaei-Keshteli, M. "Two constructive algorithms to address a multi-depot home healthcare routing problem", *IETE Journal of Research*, pp. 1–7 (2019). DOI: https://doi.org/10.1080/03772063.2019.1642802

36. Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M., and Mirjalili, S. "A set of efficient heuristics for a home healthcare problem", *Neural Computing and Applications*, **32**(10), pp. 6185–6205 (2020).

Appendix A

The Pseudo-code of RDA is shown in Figure A.1.

Appendix B

The Pseudo-code of IRDA is shown in Figure B.1.

Appendix C

The standard benchmarked functions as shown in Table C.1.

Appendix D

The Variables of BLDC motor as shown in Table D.1.

```
Initialize the Red Deers population.
Set the parameters (Maxit; nPop; nMale; alpha; beta; gamma)
Calculate the fitness and sort them and form the hinds (Hind) and male RDs (nMale).
X*=the best solution.
it=1;
while (it< Maxit)
    for each male RD
        A local search to update the position if better than the prior ones.
    end for
    Sort the males and also form the stags and the commanders as a gamma percentage of all males.
    for each male commander
        Fight between male commander and stag.
        Update the position of male commander and stag.
    end for
    Form harems.
    for each male commander
        Mate male commander with the selected hinds (alpha percentage) of his harem randomly.
        Select a harem randomly and name it k.
        Mate male commander with some of the selected hinds (beta percentage) of the harem randomly.
    end for
    for each stag
        Calculate the distance between the stag and all hinds and select the nearest hind.
        Mate stag with the selected hind.
    end for
    Select the next generation with roulette wheel selection.
    Update the X* if there is better solution.
    it=it+1;
end while
return X*
```

Figure A.1. Pseudo-code of RDA.

```
Initialize the Red Deers population.
Set the parameters (Maxit; nPop; nMale)
Calculate the fitness and sort them and form the hinds (Hind) and male RDs (nMale).
X*=the best solution.
it=1;
while (it< Maxit)
    for each male RD
        A local search to update the position if better than the prior ones.
    end for
    Sort the males and also form the stags and the commanders as a gamma percentage of all males.
    The gamma has been updated by Eq. (1).
    for each male commander
        Fight between male commander and stag.
        Update the position of male commander and stag.
    end for
    Form harems.
    for each male commander
        Select a harem randomly and name it k.
        Calculate the average of fitness for this harem and harem k.
        if the fitness average of this harem is lower than harem k
            Mate male commander with the selected hinds (alpha percentage as updated by Eq. (2)) of his harem randomly.
            Mate male commander with some of the selected hinds (beta percentage as updated by Eq. (3)) of the harem randomly.
```

Figure B.1. Pseudo-code of IRDA.

```
            else
                Mate male commander with the selected hinds (alpha percentage as updated by Eq. (3)) of his harem randomly.
                Mate male commander with some of the selected hinds (beta percentage as updated by Eq. (2)) of the harem randomly.
            endif
        end for
        for each stag
            Calculate the distance between the stag and all hinds and select the nearest hind.
            Mate stag with the selected hind.
        end for
        Select the next generation with roulette wheel selection.
        Update the X* if there is better solution.
        it=it+1;
end while
return X*
```

Figure B.1. Pseudo-code of IRDA (continued).

Table C.1. Standard benchmarked problems [5].

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	$f(x) = \max\{\lvert x_i \rvert, i = 1, ..., n\}$	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	Penalised function 1	$f(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2$ $\times[1 + 10\sin^2(\pi y_{i+1})]\} + \sum_{i=1}^{n} u(x_i, 10, 1000, 4)$ s.t. $y_i = 1 + (x_i + 1)/4$ $u(x_i, a, k, m) = \left\{\begin{array}{ll} k(x_i - a)^m & i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < a \end{array}\right\}$	M,N	$[-50, 50]$	$n = 30, 100$
P12	Penalised function 2	$f(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})]$ $+(x_n - 1)^2 \times [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	M,N	$[-50, 50]$	$n = 30, 100$

Note: C = Characteristic, U = Unimodal, M = Multimodal, S = Separable, N = Non-Separable, D = Dimension.

Table D.1. Design factors of BLDC motor [19].

No. of factors	Notation	Definitions	Range
F1	P	Number of pole pairs	[1, 6]
F2	A_c (mm^2)	Cross partitioning zone of winding	[0.1, 2]
F3	β	Pole-arc per pole-pitch ratio	[0.5, 1]
F4	l_m (m)	Magnet thickness	[0.001, 0.015]
F5	l_y (m)	Stator main thickness	[0.002, 0.01]
F6	l_w (m)	Winding thickness	[0.001, 0.0055]
F7	l_g (m)	Mechanical air gap	[0.001, 0.004]
F8	r_r (m)	Rotor radius	[0.005, 0.1]
F9	J_{cu} (Am^{-2})	Current density	$[3 \times 10^6, 6 \times 10^6]$
F10	l_s (m)	Wire gauge and stator axial length	[0.3, 2]

Biographies

Amir Mohammad Fathollahi-Fard was born and raised in Sari, Iran. He received his BSc degree (2016) and MSc degree (2018) in Industrial Engineering from University of Science & Technology of Mazandaran, Behshahr, Iran. He is currently a PhD candidate in Industrial Engineering in Amirkabir University of Technology, Tehran, Iran. He developed two novel metaheuristics, namely, Red Deer Algorithm (RDA) and Social Engineering Optimizer (SEO) in the early days of his bachelor and master degree. In addition to metaheuristics and optimization algorithms, the main focus of his research is in the area of healthcare management, supply chain management, green and sustainable logistics. He has published more than 50 scientific papers in high-ranked journals such as ASOC, JCLP, ANOR, EAAI, CAIE, INS and NCAA etc.

Milad Niaz Azari was born in Babol, Iran, in 1984. He received his BS degree in Electrical Engineering from Noshirvani University of Technology, Babol, Iran in 2007. He graduated with MS degree in the department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran in 2009. Also, he graduated with PhD degree of Electrical Engineering at Amirkabir University of Technology, Tehran, Iran in 2013. Since 2015, he has been at University of Science and Technology of Mazandaran, Behshahr, Iran, as an Assistant Professor in the department of Electrical Engineering. His areas of interest are electrical machines design and power electronics as well as metaheuristic algorithms.

Mostafa Hajiaghaei-Keshteli was born and raised in Babol, Iran. He earned his BSc degree from Iran University of Science & Technology, Tehran, Iran (2004), MSc degree from University of Science & Culture, Tehran, Iran (2006), and PhD degree from Amirkabir University of Technology, Tehran, Iran (2012), all in Industrial Engineering. He is currently an Associate Professor in Industrial Engineering at University of Science & Technology of Mazandaran, Behshar, Iran. He has over 10 years of experience in Business Development, System Analysis, Inventory and Project Management. He also has worked for many corporations in Iran and has held the positions of consulter, planning and project manager and VP. The main focus of his research is in the area of inventory control, supply chain network, transportation and meta-heuristics. He has published more than 100 scientific papers in high-ranked journals such as ESWA, CAIE, KNOSYS, JCLP, INS, NCAA, IEEE and ASOC etc.