Soft Computing
https://doi.org/10.1007/s00500-019-04410-8

METHODOLOGIES AND APPLICATION

Electron radar search algorithm: a novel developed meta-heuristic algorithm

Sajjad Rahmanzadeh[1] · Mir Saman Pishvaee[1]

Abstract
This paper introduces a new optimization algorithm called electron radar search algorithm (ERSA) inspired by the electron discharge mechanism. It is based on the natural phenomenon of electric flow as the form of electron discharge through a gas, liquid, or solid environment. When the voltage between separated electrodes (anode and cathode) increases, electrons tendency to emission from a low potential state to the higher potential condition is grown up. However, electrons are trying to find the best path with the least resistance in the medium. At each point, electrons evaluate the surrounding environment with a radar mechanism and least resistance path is selected for the next move. Hence, in this paper, a novel developed meta-heuristic algorithm based on the electrons' search approach is presented and the algorithm is benchmarked on 20 mathematical functions with four well-known methods for validation and verification tests. Moreover, the algorithm is implemented in two engineering design problems (tension/expression spring and welded beam design optimization) and the results demonstrate that the ERSA performs more efficiently for solving unknown search spaces and the algorithm found best solution in approximately 95% of the reviewed benchmark functions.

Keywords Electron radar search algorithm · Meta-heuristics · Electron discharge · Optimization

1 Introduction

In the area of mathematics and computer science, optimization problems are applied to find the best solution in the feasible region. Most classical methods (e.g., direct methods and gradient-based methods) are developed to find the optimum solutions in a finite or countably infinite set of potential solutions considering first- and second-order optimality conditions (Floudas 2000; Horst and Tuy 1996). However, problem-solving process will be more difficult when the inherent complexity of the problem increases (Daneshmand et al. 2015). Over the last several decades, many heuristics and meta-heuristic optimization techniques have been presented and successfully applied to different domains including computer science, mathematics, and engineering (Cacchiani and D'Ambrosio 2017; Goldenberg 2017; Ebrahimi et al. 2017; Christensen and Bastien 2015; Vidal et al. 2015). Generally, heuristic and meta-heuristic techniques are designed to solve a problem with an approximate solution and in many cases more quickly than the classical analytical methods. Heuristics are often specific and problem dependent, while meta-heuristic algorithms present a high-level problem-independent framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. Mirjalili et al. (2014) have presented four main reasons including (1) simplicity, (2) flexibility, (3) derivation-free mechanism, and (4) local optima avoidance, reinforcing the expansion and popularity of meta-heuristic algorithms in the last two decades. Efficient meta-heuristic algorithms use stochastic mechanisms to avoid trapping in the local optimum situation and to find an acceptable approximation for the global optimum solution (Gutjahr 2010). In the studied literature, several meta-heuristic algorithm classifications have been presented based on the type of search strategy, source of inspiration, particle population, search experience, and type of objective function. Additionally, Kaveh and Dadras (2017) classified three different types of meta-heuristic algorithms based on the source of inspiration, namely

Communicated by V. Loia.

✉ Mir Saman Pishvaee
pishvaee@iust.ac.ir

[1] School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Published online: 10 October 2019

evolutionary algorithms (EA) (Fonseca and Fleming 1995; Bäck and Schwefel 1995), Swarm algorithms (Deepa 2016; Shi and Eberhart 1999), and physical algorithms. Evolutionary algorithms like genetic algorithm (GA) (Holland 1992, 1975) are usually inspired by natural evolution such as reproduction, mutation, recombination, and selection. Some of the EAs are evolution programming (EP) (Yao and Liu 1996), evolution strategy (ES) (Pinebrook 1987), gene expression programming (GEP) (Byrne et al. 1994), genetic programming (GP) (Banzhaf et al. 1997), and differential evolution (DE) (Das and Suganthan 2010). Swarm algorithms oriented from the study of computational systems inspired by the collective intelligence emerge from the cooperation of homogeneous agents in the environment (Abraham et al. 2008). Ant colony optimization (ACO) (Maniezzo and Carbonaro 2002), grey wolf optimizer (GWO) (Mirjalili et al. 2014), firefly algorithm (FFA) (Yang 2009), bat search algorithm (BA or BSA) (Yang 2010), and particle swarm optimization (PSO) (Eberhart and Kennedy 1995) are popular swarm intelligence-based algorithms. Moreover, physical algorithms utilize physical laws in the optimization process. Simulated annealing (SA) proposed by Kirkpatrick et al. (1983), the Big Bang–Big Crunch algorithm (BB–BC) proposed by Erol and Eksin (2006), water wave optimization (WWO) proposed by Zheng (2015), and the gravitational search algorithm (GSA) presented by Rashedi et al. (2009) were introduced using physical phenomena. Additionally, meta-heuristic algorithms are divided into local search and global search categories considering the search strategy. Local search algorithms move from the candidate solution to a better neighbor solution in an iterative procedure until satisfying the terminating condition. Local search algorithms including Tabu search (TS) (Glover 1989; Du and Pardalos 1999) and SA are widely applied to numerous hard computational problems in computer science, bioinformatics, mathematics, and engineering. On the other hand, global search algorithms developed based on the population behaviors and explored new regions in the search space. PSO, ACO, and GA are the well-known and most popular global search algorithms. Meta-heuristic algorithms have been applied in a wide range of applications in structures and infrastructure issues. In this area, scheduling and planning problems (Xhafa and Abraham 2008; Pishvaee et al. 2010; Griffis et al. 2012), data mining and machine learning issues (Gandhi et al. 2012; Chen and Tsao 2009; Qureshi et al. 2017), and engineering design problems (Cantarella et al. 2015; Ranaboldo et al. 2015) are popular domains use meta-heuristic algorithms to find approximate solutions.

On the other hand, Sorensen (2015) challenges the novelty of some new meta-heuristic frameworks based on metaphors. Moreover, the author states that in a majority of cases, the metaphors meta-heuristics are not only unnecessary but also harmful to the scientific quality and the external appearance of the research field. However, the paper investigates that high-quality research in meta-heuristic algorithms including considerable differences with the classical ones will be appreciated in the literature. This paper proposes a new swarm intelligent algorithm called electron radar search algorithm (ERSA) which mimics electron discharge phenomena. Briefly, the ERSA can be characterized in (1) improvement and mutation (forking concept) of particles in the local search, (2) more emphasis on the better solutions during the iterations, and (3) an initiated approach that avoids being trapped in local minima. These characteristics are the outstanding points of the ERSA that make the algorithm different from the other common meta-heuristics including GA, PSO, SA, and ACO. Naturally, electrons search lower potential environment and move to the new position by the action of the electric forces. Electrons travel to the lower position with an indirect path method and multi-step mechanism. In each step, electrons scan the electric potential of the surrounding and lower potential point is selected for the electron movement. Therefore, electrons apply a multi-step mechanism to find the optimum solution with the least electric potential in an electron discharge phenomena. The ERSA is a population-based algorithm and avoids being trapped in local minima and able to explore globally for the better solution. Moreover, the algorithm benefits from the forking theory that increases exploitation within the neighborhood of previously visited points. Additionally, the algorithm was run for different benchmark functions and the results showed the ERSA presents the better solutions in approximately 95% of the functions.

The rest of the paper is organized as follows: Sect. 2 describes the electron discharge mechanism. Section 3 outlines the proposed ERSA algorithm. The results and discussion of benchmark functions to compare ERSA with some other popular optimization methods are presented in Sect. 4. Finally, conclusions are derived in Sect. 5.

2 Electric discharge mechanism

2.1 General description of electric discharge mechanism

Electric discharge is a complicated process which creates a conducting path between two points of different electric potential in the medium and occurs when the electric field exceeds some critical value (Fridman et al. 2005). Nowadays, many industries use electric discharge mechanism in various applications including ignition sources (Zaepffel et al. 2007), electrical discharge machining (EDM) (Ho and

Newman 2003; Abbas et al. 2007), chemical analysis (Walters 1969), and radio communications (Beauchamp 2001). Generally, in the electrical breakdown, when the potential voltage between electrodes increased and reached to a critical voltage, electrons are emitted from the cathode and transferred to the anode after ionizing the medium. Moreover, usually, each primary electron that drifts to the anode collides with the medium molecules and creates positive and negative ions during ionization and attachment process. Positive and negative ions move to the cathode and anode, respectively, and the potential voltage between the electrodes increases continuously (Keidar and Beilis 2013). Figure 1 presents a schematic view of the electron emission and medium breakdown process.

Generally, electron discharge occurs between two electrodes when the electric field strength exceeds sufficiently a certain threshold value. This creates a narrow ionized and conductive channel with very high electron current (up to 10^5 A) growing fast between cathode and anode (Fridman et al. 2005). To investigate electrical discharge process in more detail, it is essential to discuss some theoretical concepts and expressions defined in the theory of breakdown phenomena. These theoretical issues are mentioned in the following sections.

2.2 Electron avalanche mechanism

When a high voltage is applied between the electrodes, the primary electrons are emitted in the direction from cathode to the anode. The electrons collide with medium molecules, free additional electrons and consequently create an avalanche (Fridman et al. 2005; Keidar and Beilis 2013; Xiao 2016). Therefore, electron avalanche is defined as the multiplication of some primary electrons in cascade ionization process where ionization probability is higher than attachment probability. In more detail, suppose n_0 is the number of primary electrons emitted from the cathode and then for a given distance x, number of electrons added to the avalanche (n) will be equal to:

$$n = n_0 \times \exp(\alpha.x) \tag{1}$$

where α is the average number of electrons created per cm of the path called first Townsend coefficient. Therefore, the number of electrons in an avalanche is proportional to the number of primary electrons and the distance from the cathode. When an electron collides with the medium molecules, a positive ion is created if the electron has acquired enough energy before the collision. Furthermore, the released positive ions return to the cathode and free new electrons to join the discharge process. Therefore, (n-n_0) returned positive ions liberate $\gamma \times (n\text{-}n_0)$ additional electrons which γ represents the number of secondary electrons emitted by one positive ion. This means that the number of electrons participating in the avalanche process in distance x can be calculated as follows.

$$n = [n_0 + \gamma(n - n_0)] \times \exp(\alpha.x) \tag{2}$$

Figure 2 presents a schematic view of the electron avalanche process in the electric discharge phenomena.

As shown in Fig. 2, the primary electrons are emitted from the cathode and accelerated in the electric field. Then,

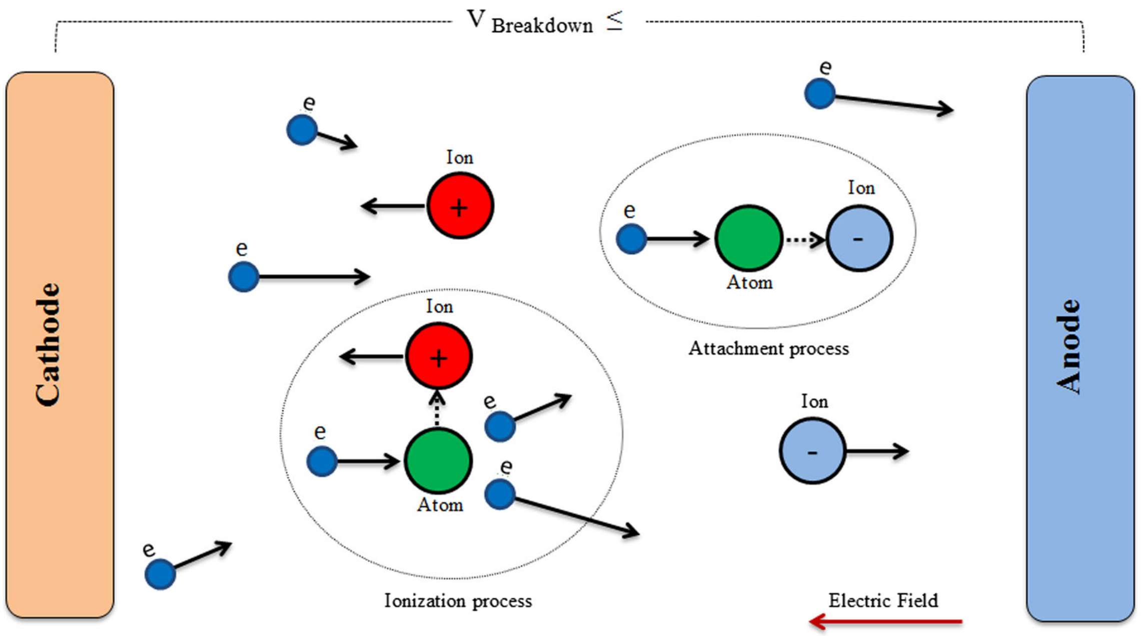


Fig. 1 Schematic representation of the electron emission and medium breakdown process

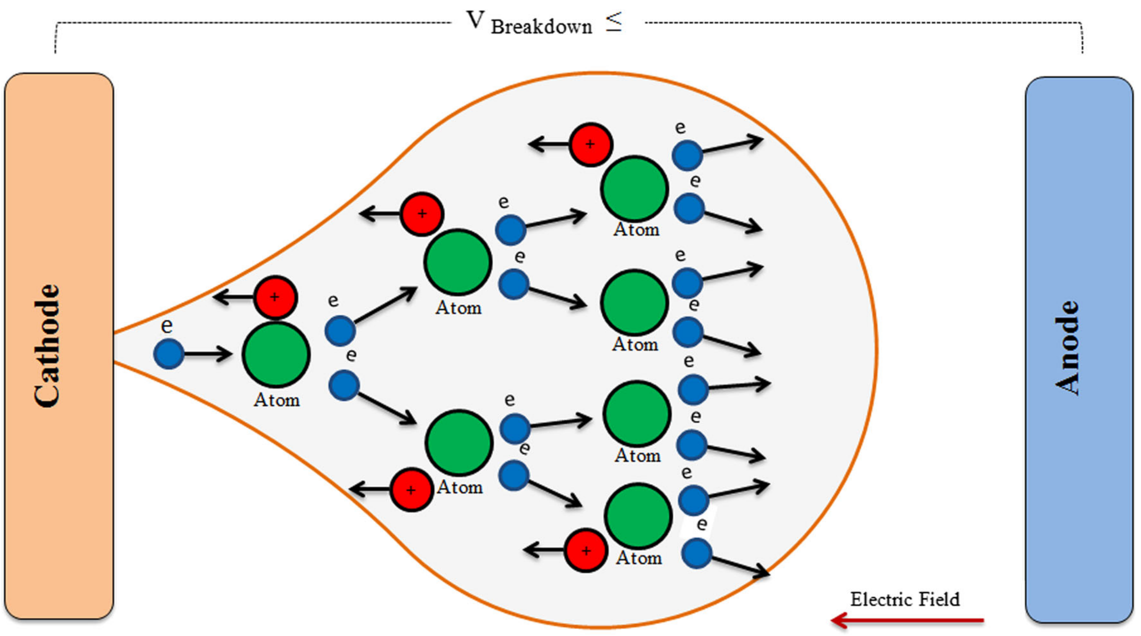


Fig. 2 Schematic view of the electron avalanche process

an avalanche of electrons defused in a cascading ionization process.

2.3 Streamer mechanism

In gas mediums like air, electric discharge phenomena take place based on the Townsend mechanism under low pressures and uniform electric fields (Xiao 2016). In practice, Townsend discharge theory does not consider the effect of positive ions space charge which can cause significant electric field distortion. Additionally, in high pressure and non-uniform electric fields, discharge occurs very fast that cannot be explained by the Townsend mechanism. Describing the defects of Townsend discharge theory, Loeb and Meek (1941) introduced the streamer breakdown theory to clarify the phenomenon of discharge under higher pressure. The streamer is an ionized thin channel emerged from a primary avalanche in an adequately strong electric field. When the avalanche in the gap reaches a critical size, space charge forms a dipole with electrons being at the head and ions being behind. Hence, an additional electric field is created in the space charge of the avalanche. In this area, Meek and Craggs (1978) have indicated that in the streamer propagation, space charge field at the head of the avalanche shall be comparable to the external fields. Moreover, Meek and Craggs (1978) have proposed the criterion for the avalanche to streamer formation. According to this criterion, for the electron avalanche changing into the streamer, the radial electric field intensity of the space charge must be equal or greater than the external field.

$$E_a = \left(\frac{e}{4\pi\varepsilon_0 r^2}\right) \times \exp(\alpha.x) \geq E_0 \tag{3}$$

where E_a and E_0 are the avalanche surface electric field and external field, respectively. Exp shows the exponential function, and additionally, e represents the electron electric charge ($-\ 1.602 \times 10^{-19}$ C) and r defined as avalanche head radius. ε_0 is the electric constant, and $\exp(\alpha.x)$ calculates the primary avalanche space charge. Figure 3 shows streamer propagation and electric field in the electric discharge process.

As shown in Fig. 3, at the tip of the streamer, a strong electric field appears and the electrons are attracted to the streamer head and secondary avalanches are created to forward the streamer for the next step. Moreover, the streamer uses a radar search mechanism to find the best path for ionizing the medium especially the air. Additionally, the paths with the higher ionization and lower attachment probabilities are the candidate for the ionization process. Figure 4 presents a simplified schematic view of the streamer radar search algorithm in an electric discharge process. Furthermore, in each step, electrons find best solution in a spherical search space at the tip of the streamer.

3 Electron radar search algorithm

3.1 Theoretical concepts

The proposed optimization algorithm relies on the spark discharge mechanism and formulates a new novel meta-

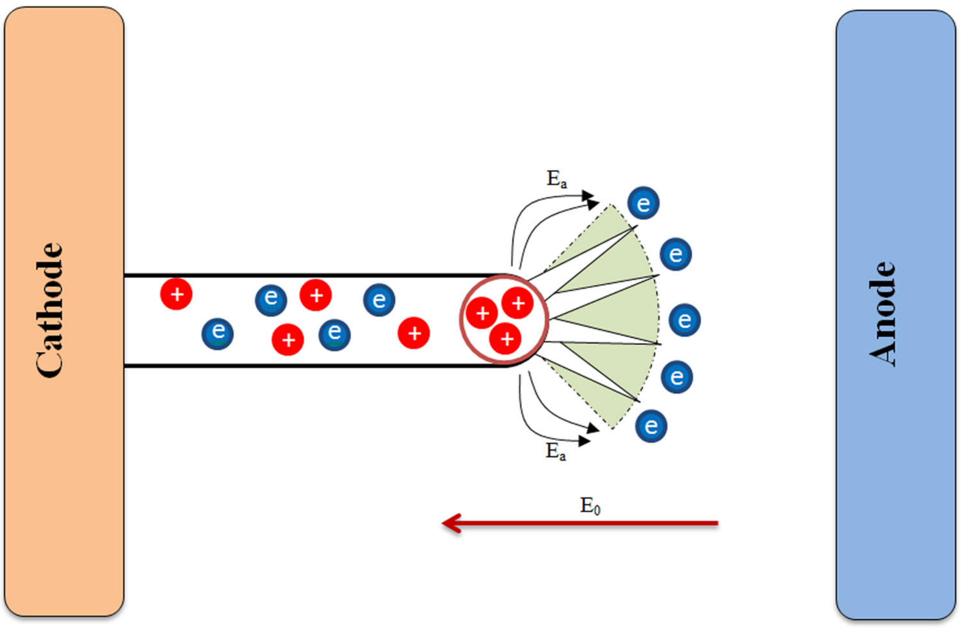


Fig. 3 Streamer propagation in an electric discharge process

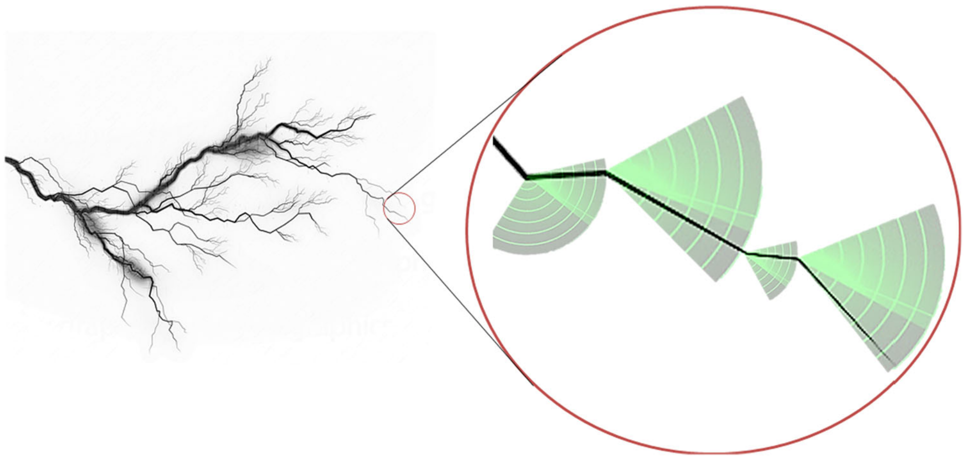


Fig. 4 Streamers achieve new position using a radar search mechanism

heuristic algorithm called ERSA based on the streamer mechanism. Each electron emitted from the cathode plate collides with medium molecules and creates positive ion if the ionization rate is greater than the attachment rate ($\alpha > \beta$). Otherwise, a negative ion is generated and moved to the anode plate. The streamer continues the searching process when the number of electrons is greater than a critical value. The streamers which cannot satisfy the critical value's condition were eliminated from the search space. At each step, the streamer explores the tip spherical space for finding the better objective function, and moreover, the streamer tip position is updated for the next step. Additionally, local optimum solutions are achieved after eliminating all the streamers from the search space. Figure 5 shows ERSA functionality in finding the local optimum solutions.

Additionally, the position of the streamers in two different channels until the 6th iteration is presented in Fig. 5. In each step, streamers explore the best solution in a restricted spherical space. Moreover, according to the particles collision theory (Xiao 2016), streamers forking probability increases with the growth of electrons number at the streamers' tip. In forking point, two different streamers emerge in the search domain. The first streamer moves to the best solution, and the second appeared in a random solution created in the spherical space. Additionally, streamers are eliminated if the number of electrons became smaller than a predefined value. As mentioned before, the ERSA uses three outstanding characteristics, namely (1) improvement and mutation (forking concept) of particles in the local search, (2) more emphasis on the better solutions during the iterations, and (3) an initiated approach that avoids being trapped in local minima that

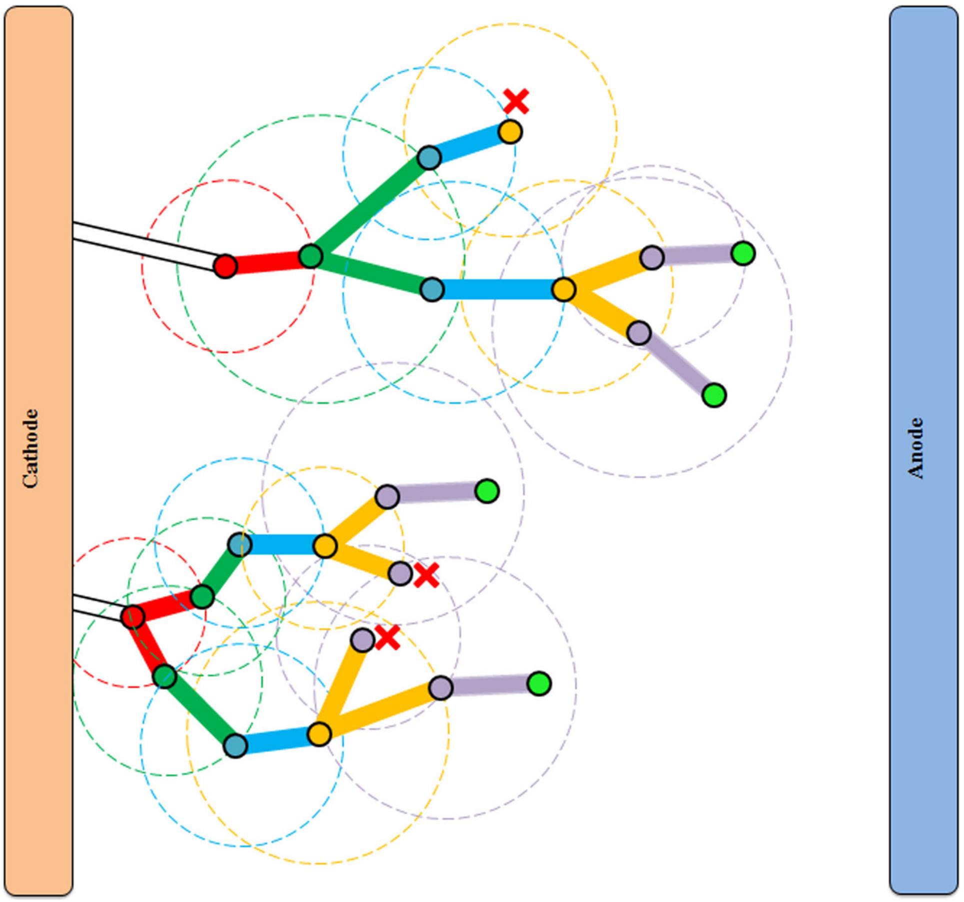


Fig. 5 ERSA functionality in finding the local optimum solutions

improve the algorithm performance. The algorithm is started with an initial population of streamers, and at each iteration, streamers search optimal solution in a local area. Moreover, the streamers forked with a probability value and a new streamer is created when the number of electrons exceeded a predefined value. Therefore, a local mutation is considered in the search space. On the other hand, as described in Sect. 2.2, additional electrons are added to the streamers tip from the cathode plate after each iteration. Furthermore, the streamer with the better solution receives more electrons than the other streamers. For the third characteristic, when the streamer does not find the better solution from its current solution, the streamer increases the local search space by decreasing the number of electrons to a critical value (CV). By development of the local search space, the probability of finding better solutions increased. However, the streamer eliminated when the better solution not be achieved in the new local search space. Generally, according to the above description, the ERSA can briefly be outlined as follows:

Initial population First N initial solutions (initial streamers) are selected randomly in the feasible region for starting the algorithm. Moreover, ERSA parameters including (E_n^0, $n \in \{1, 2\ldots N\}$), search radius r, and CV are determined at the first stage.

Streamer updating Number of electrons in streamer n and at iteration t (E_n^t) is updated after each iteration as:

$$E_n^t = E_n^{t-1} \times \exp\left(\left[\frac{|F_{nt} - F_{nt-1}|}{F_{n0}}\right] \times x\right) + \lambda) \tag{4}$$

$$\lambda = \beta \times (1 - \exp(-|F_{nt} - F_{nt-1}|)) + (1 - \beta) \times \exp(-|F_{nt} - F_{\text{Best}}|) \tag{5}$$

where F_{Best} refers to the best solution of streamers and F_{nt}, F_{nt-1}, and F_{n0} are the objective function of streamer n over the iteration t, iteration $t-1$, and the first iteration, respectively. β is a coefficient that controls the balance between the exploration and the exploitation. $\lambda \times E_n^{t-1}$ represents the number of electrons must be added to the streamers at each iteration. However, the streamers receive additional electrons according to their solution quality. λ consists of two expressions and calculates how additional electrons must be added to the streamers. In the first expression, the importance of function improvement in the local search has been emphasized, and in the second expression, the algorithm tries to allot more additional energy to the better streamers' functions. Furthermore,

x presents the Euclidean distance between the streamer tip position in iteration t and $t-1$. The streamers' positions are updated based on the best solution in a spherical search space in which the streamers' last positions and r are the center and radius of the searching sphere, respectively. To find near-optimum solution in spherical search space, M randomly points are selected and the best solution is considered approximately as the optimum result.

Forking event The streamers can branch into two new streamers with the following probability.

$$P_r = \begin{cases} 1 - \left[\dfrac{2 \times \mathrm{CV}}{E_n^t}\right] & E_n^t > 2 \times \mathrm{CV} \\ 0 & E_n^t \leq 2 \times \mathrm{CV} \end{cases} \tag{6}$$

After the forking process, the electrons are divided into the initial streamer and new streamer equally. Moreover, the number of electrons is updated and set to $\frac{E_n^t}{2}$ for the initial and the new streamer. Therefore, the streamer can be forked when the number of electrons is greater than twice the CV.

Escaping from local minima for finding the better solution, the streamer searches local spherical space and selects the better one. However, the current solution may be the local optimum result in the search space. In this regard, to escape from local optima, the streamer increases the radius of the sphere to the maximum value satisfying CV condition. To reach this purpose, the streamer decreases the number of electrons to the CV. Therefore, the maximum value of r in streamer n and iteration t is calculated as follows:

$$r_{\mathrm{tn}}^{\max} = r \times e^{\left[\frac{E_n^t - \mathrm{CV}}{\mathrm{CV}}\right]} \tag{7}$$

Additionally, the streamer searching process is terminated and the streamer is eliminated when the better solution in the spherical space with radius $r_{\mathrm{tn}}^{\max}$ is not found.

Finally, the ERSA is terminated after eliminating all streamers from the searching process. To find solutions with acceptable accuracy, the algorithm consists of six initial parameters, namely β, N, $E_n^0, \mathrm{CV}, r,$ and M. β is a coefficient that controls the balance between the exploration and the exploitation. Therefore, by increasing the β value ($\beta > 0.5$), the algorithm focuses on the local optimum solutions more than the global one. However, the $\beta = 0.5$ will be appropriate for solving the model generally. N presents the number of initial solutions, and E_n^0 presents the number of electrons considered for each initial solution. When the E_n^0 value increases, the probability of finding better solutions and the algorithm computation time increase too. However, it is supposed that E_n^0 be set in the rage of [500, 1500]. Moreover, the solutions will be eliminated when the number of electrons is less than the CV. Considering CV = 0.1 $\times E_n^0$ is the proper selection. r is considered as the searching radius for each solution, and it suggested that r = 10 for uncountable search spaces and r = 0.2 for the finite search area. M declares the number of random points selected for the local search at each iteration. However, M can be initialized by N^2 (N is the number of initial solutions). Consequently, the input parameters initialization can be summarized as below:

β: $\beta = 0.5$ generally, $\beta = 0.75$ with the local optimum solution considerations, and $\beta = 0.25$ when finding the global optimum solution is the final goal.
N: $N = 200$ for infinite search space, and $N = 50$ for finite search space.
$\boldsymbol{E_n^0}$: $E_n^0 \in [500, 1500]$
CV: CV = 0.1 $\times E_n^0$.
r: r =10 for uncountable search spaces and $r = 0.2$ for the finite search area.
M: $M = N^2$

To present the ERSA briefly, the pseudocode of the algorithm is shown in Table 1.

3.2 Exploration and exploitation

In the last years, many research scholars have studied exploration and exploitation concepts in meta-heuristics algorithms (Junqin and Jihui 2014; Hills et al. 2015; Crepinsek et al. 2013). Exploration and exploitation are important meta-heuristic features interpreted as global search and local search, respectively. Moreover, exploration concept is considered as the algorithm ability to explore global optimum solution within the whole search space. On the other hand, exploitation focuses on searching within the neighborhood of previously visited points. A search algorithm needs to establish a tactical balance between these two sometimes conflicting goals. The ERSA uses the random-based parameter to perform exploration in the search space. Randomly generated initial streamers can search in all directions from the initial position affecting by initial population parameter. Moreover, spherical searching at the tip of the streamers and branching mechanism enable the ERSA to improve exploitation ability throughout the searching process. In this paper, an index is introduced as exploitation coefficient (β) to control exploration and exploitation rate. Moreover, as it is described in Eq. 5, by setting a higher β values, the algorithm allots more additional electrons to the best streamers according to function improvement ($|F_{nt} - F_{nt-1}|$.) and emphasis on the exploitation concept. When the number of electrons increases, the probability of forking and finding the better solution in the local search increased too. However, by reducing the value of β, the additional electrons are

Table 1 The pseudocode of the ERSA algorithm developed in this study

```
Start
        Define the initial population of streamers ($n$=1, 2, …, $N$)
        Determine input parameters $\beta$, $N$, $E_n^0, CV, r, M$ and $t = 0$
        while the number of existing streamers > 0
            for each streamer
                if (Eliminating condition satisfied)
                    Eliminate streamer and go to the next streamer
                else if (Forking condition satisfied)
                    Create new streamer with the random position in the
                    spherical search space and update population of streamers
                end if
                Update streamer position and best objective function
            end for
        end while
        return best objective function
End
```

received by the streamers that achieved better solutions globally. Therefore, the algorithm increases the probability of finding better solutions in the pioneer streamers. Therefore, there is a trade-off between the ERSA exploration and exploitation abilities considering different β values.However, Fig. 6 is prepared to show the algorithm progress steps and display the position of streamers for different iterations.

4 Numerical results and validation

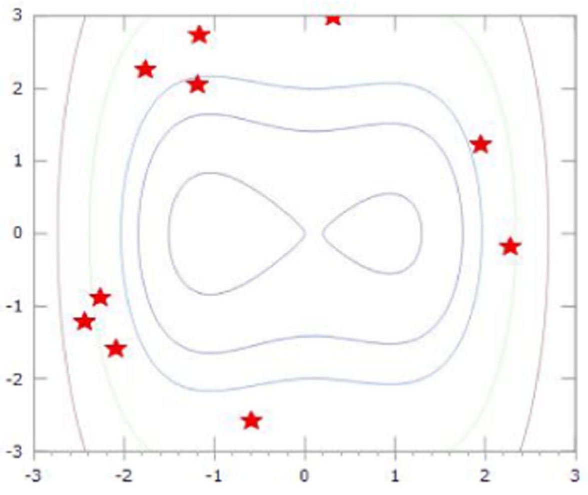
In order to verify the efficiency of ERSA, some numerical examples are considered from the literature. Hence, the proposed new algorithm is tested and validated using a well-utilized set of 20 unimodal and multimodal benchmark functions (Kaveh and Talatahari 2010; Shareef et al. 2015). The modality of a function refers to the number of peaks in the function's surface. However, in multimodal functions, there are at least two vague peaks in the searching space. Moreover, the ERSA is compared with the popular meta-heuristics methods, namely DSA, PSO, FFA, and BSA, based on the function statistics including best, worst, average, and standard deviation of the objective functions acquired in the running process. Additionally, each objective function is optimized 50 times with different initial populations and considering the number of iterations as 100. For more information, the experiments are implemented on a PC with 2.6 GHz CPU and 4 GB RAM.

4.1 Validation using unimodal functions

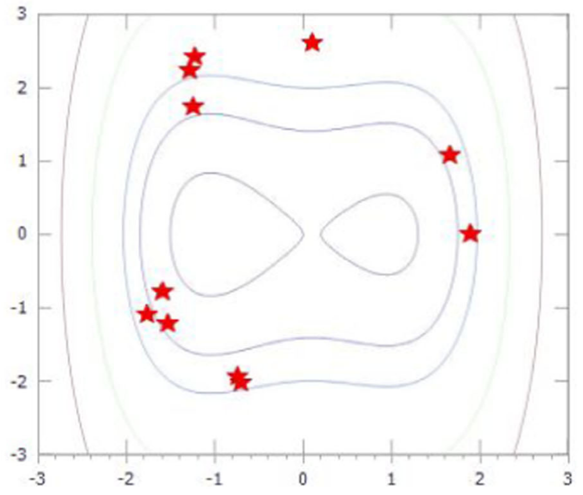
This section is conducted to assess the efficiency of the ERSA in finding the global minimum value of the unimodal benchmark functions. For this purpose, seven unimodal functions, namely Sphere, Step, Quadratic, Rosenbrock, Booth, Schwefel 2.21, and Schwefel 1.2, were selected according to the Shareef et al. (2015), Kaveh and Zolghadr (2016), and Mirjalili et al. (2014) researches. Table 2 presents the list of unimodal functions applied in this paper. Moreover, as the results shown in Table 4, the ERSA performance is acceptable in finding the global optimum solution for all seven unimodal functions. During the optimization process, the performances of the five meta-heuristic algorithms in terms of computation time were almost similar and less than five second. Additionally, the convergence curve of the algorithms and the benchmark functions 2D-dimension surface and contour lines plot are presented in Fig. 7.

4.2 Validation using multimodal functions

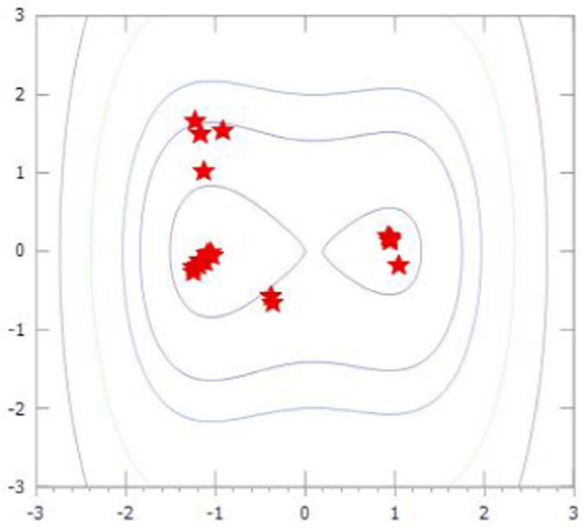
To evaluate the efficiency and reliability of the ERSA in solving multimodal functions, nine benchmark functions, namely Aluffi-Pentini, Schwefel, Becker and Lago, Bohachevsky 2, Griewank, Rastrigin, Cosine mixture, Branin, and Six-Hump Camel Back, were selected for experimental tests (see Table 3). These functions are presented to test the algorithm's ability in avoiding to be trapped in local minima. Additionally, the algorithm best, worst, and average solution, the stand deviation value, and the computation time were calculated for the studied functions from 50 independent runs. The results showed the ERSA average error (best solution − optimum solution) in sixteen popular benchmark functions is approximately 2.34E−4. On the other hand, the algorithm presents a robust solution in which the standard deviation value is 14.75 in the worst case (F_9). As shown in Table 4, the ERSA found the better near-optimum solutions in thirteen benchmark functions in all statistics and acquired the best solutions for all studied multimodal functions except F_{12}. Furthermore, in more detail, the algorithms statistics results from 50 runs are shown in Table 4.



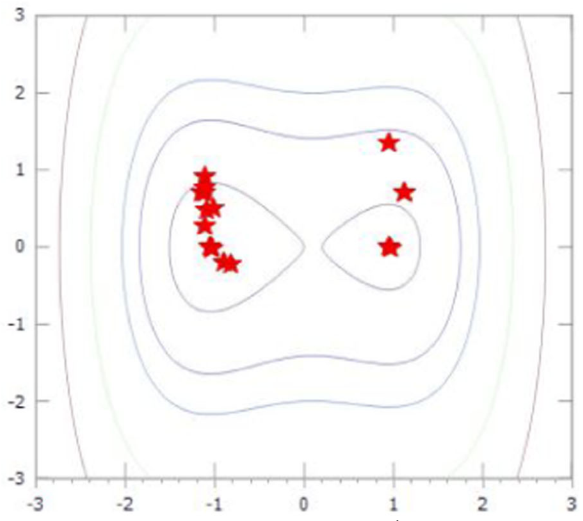
(a) Initial streamers' position



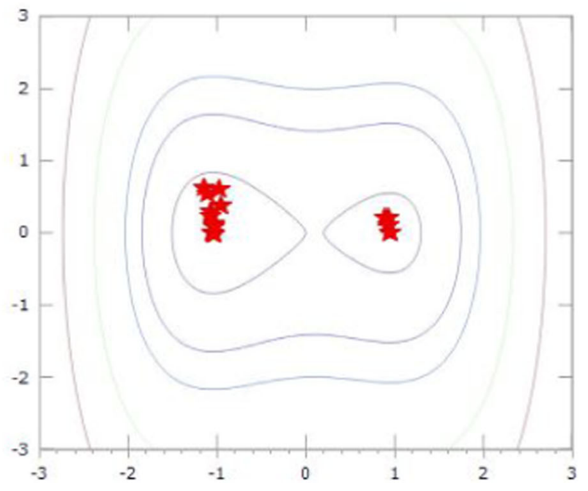
(b) Streamers' position after 10th iteration



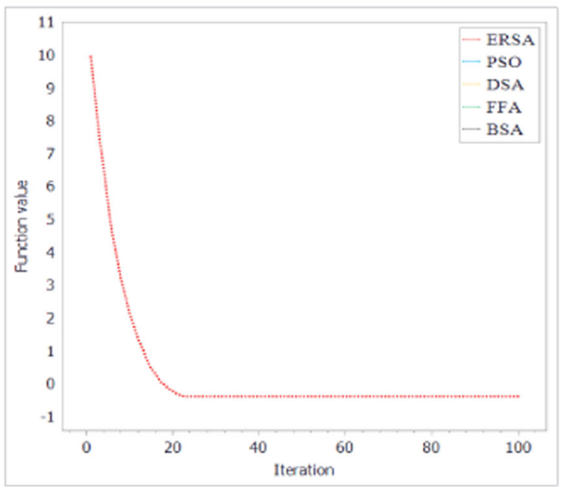
(c) Streamers' position after 30th iteration



(d) Streamers' position after 50th iteration



(e) Streamers' position after 80th iteration



(f) Convergence curve

Fig. 6 Streamers' position at different ERSA iterations (Aluffi-Pentini benchmark function)

Table 2 Unimodal benchmark functions

Functions	Name	Expression	Search space	$f_{\min}$
F_1	Sphere	$F(X)=\sum_{i=1}^{n}x_i^2$	$[-10, 10]^{100}$	0
F_2	Step	$F(X)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	$[-10, 10]^{100}$	0
F_3	Quartic	$F(X)=\sum_{i=1}^{n}ix_i^4$	$[-10, 10]^{100}$	0
F_4	Rosenbrock	$F(X)=100(x_2-x_1^2)^2+(x_1-1)^2$	$[-30, 30]^2$	0
F_5	Booth	$F(X)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	$[-5, 5]^2$	0
F_6	Schwefel 2.21	$F(X)=\max_i(\lvert x_i\rvert, 0\le i\le 10)$	$[-10, 10]^2$	0
F_7	Schwefel 1.2	$F(X)=\sum_{i=1}^{10}\left(\sum_{j=1}^{i}x_j\right)^2$	$[-100, 100]^{100}$	0

4.3 ERSA characteristics analysis

The ERSA benefits three outstanding characteristics including (1) improvement and mutation (forking concept) of particles in the local search, (2) more emphasis on the better solutions during the iterations, and (3) an initiated approach that avoids being trapped in local minima.

To check the improvement and mutation (forking concept) of particles in the local search, the probability of forking increased and the results show that the particles population changed and the quality of solutions improved. In this regard, the *CV* decreased from 0.23 $\times E_n^0$ to 0.03 $\times E_n^0$ and the results showed the streamers population growth and enhancement in quality of solutions considering forking probability in Eq. 6. Furthermore, to clarify the ERSA contribution on using local and global best solutions, the model was run with different β initial values. As the results show, in $\beta = 1$, the streamers focus only to their previous solutions and by decreasing the β value, the effect of global best solution (best solution achieved) on finding the better results increased. Moreover, by setting $\beta = 0$, the streamers search new positions considering global best solution characteristics with the most importance. On the other hand, to declare the ERSA third contribution, the algorithm was tested with different searching radius (r). As the results show, with considering larger values for r parameter, the probability of finding the better solutions increases generally. Briefly, when the r value decreases, the algorithm needs more iterations to find the better solutions. The trends of finding optimum solutions are presented in Fig. 8.

In another experiment, the trend of finding the optimum solutions in different meta-heuristic algorithms including ERSA, PSO, DSA, and FFA has been compared using Branin function. Moreover, the algorithms were run with 10 initial population (*N* = *10*). Figure 9 shows the algorithms' best position over the considered iterations.

4.4 Engineering design problem

In this section to further investigate the efficiency of the proposed meta-heuristic algorithm, two well-studied constrained engineering design problems including tension/compression spring and welded beam design problems are employed. These problems have been solved previously using a wide variety of algorithms in the optimization literature. To simplify the problem, constraints can turn into the penalty functions and the amount of penalty will be zero if the constraints are satisfied. Otherwise, a predetermined value of penalty is calculated as the ratio of violated constraints. Additionally, in order for the comparison to be fair, the results are considered of best methods in the state of the art.

4.4.1 Tension/compression spring design

The aim of this problem is to minimize a tension/compression spring weight subject to the constraints on the shear stress, surge frequency, and minimum deflection as shown in Fig. 10. The model consists of three design variables, namely wire diameter (d), mean coil diameter (D), and the number of active coils (N). The problem can be formulated as follows:

Minimize

$$f(x)=(x_3+2)x_2x_1^2 \tag{8}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 1-\frac{x_3x_2^3}{71785x_1^4}\le 0,\\
g_2(\vec{x}) &= \frac{4x_2^2-x_1x_2}{12566(x_2x_1^3-x_1^4)}+\frac{1}{5108x_1^2}-1\le 0,\\
g_3(\vec{x}) &= 1-\frac{140.45x_1}{x_3x_2^2}\le 0,\\
g_4(\vec{x}) &= \frac{x_1+x_2}{1.5}-1\le 0,
\end{aligned} \tag{9}$$

(a)

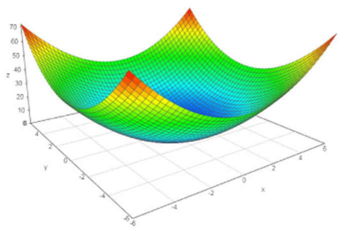
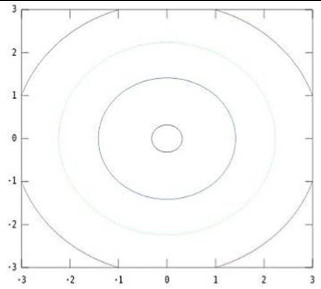
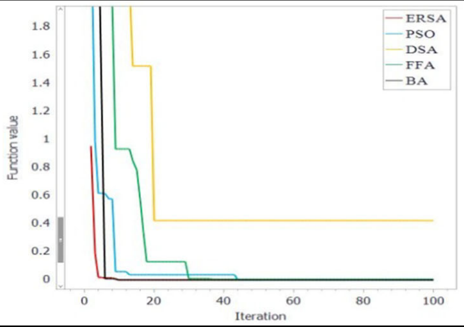
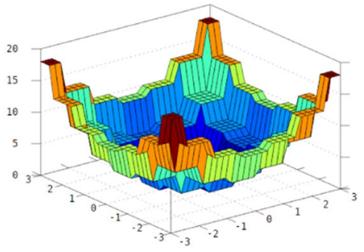
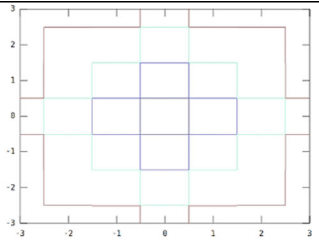
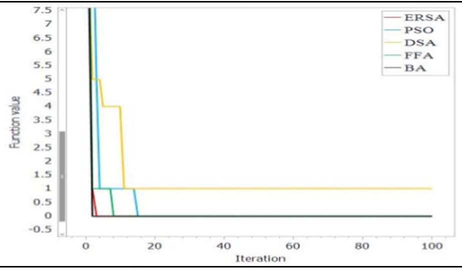
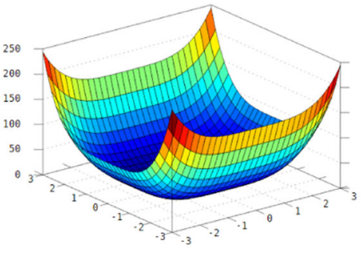
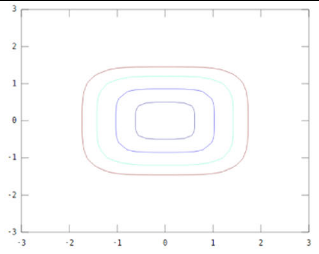
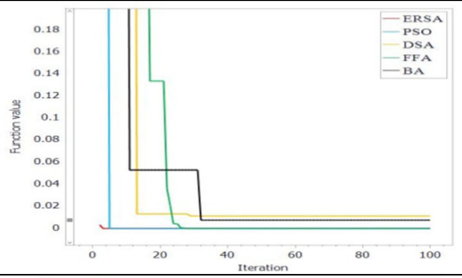
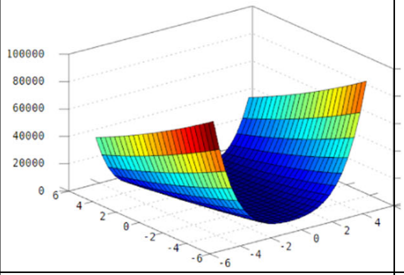
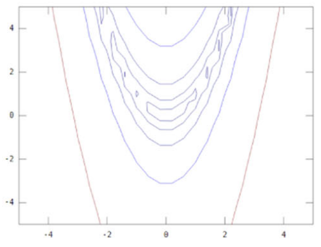
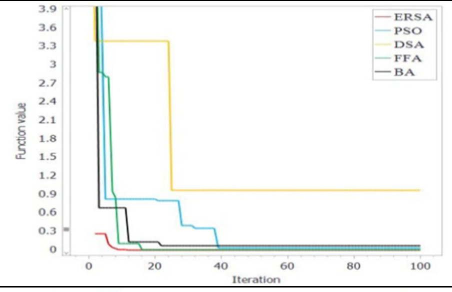
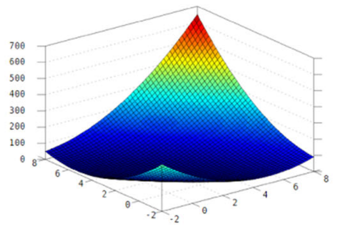
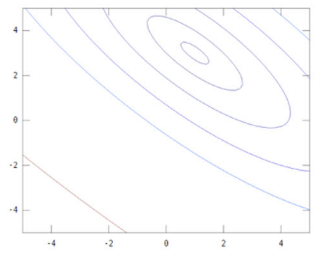
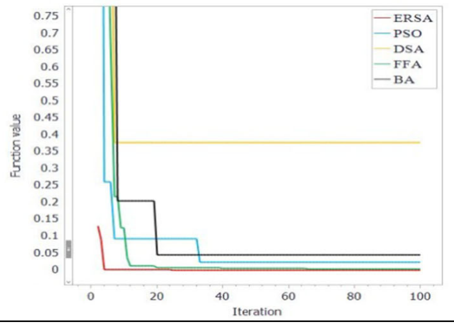
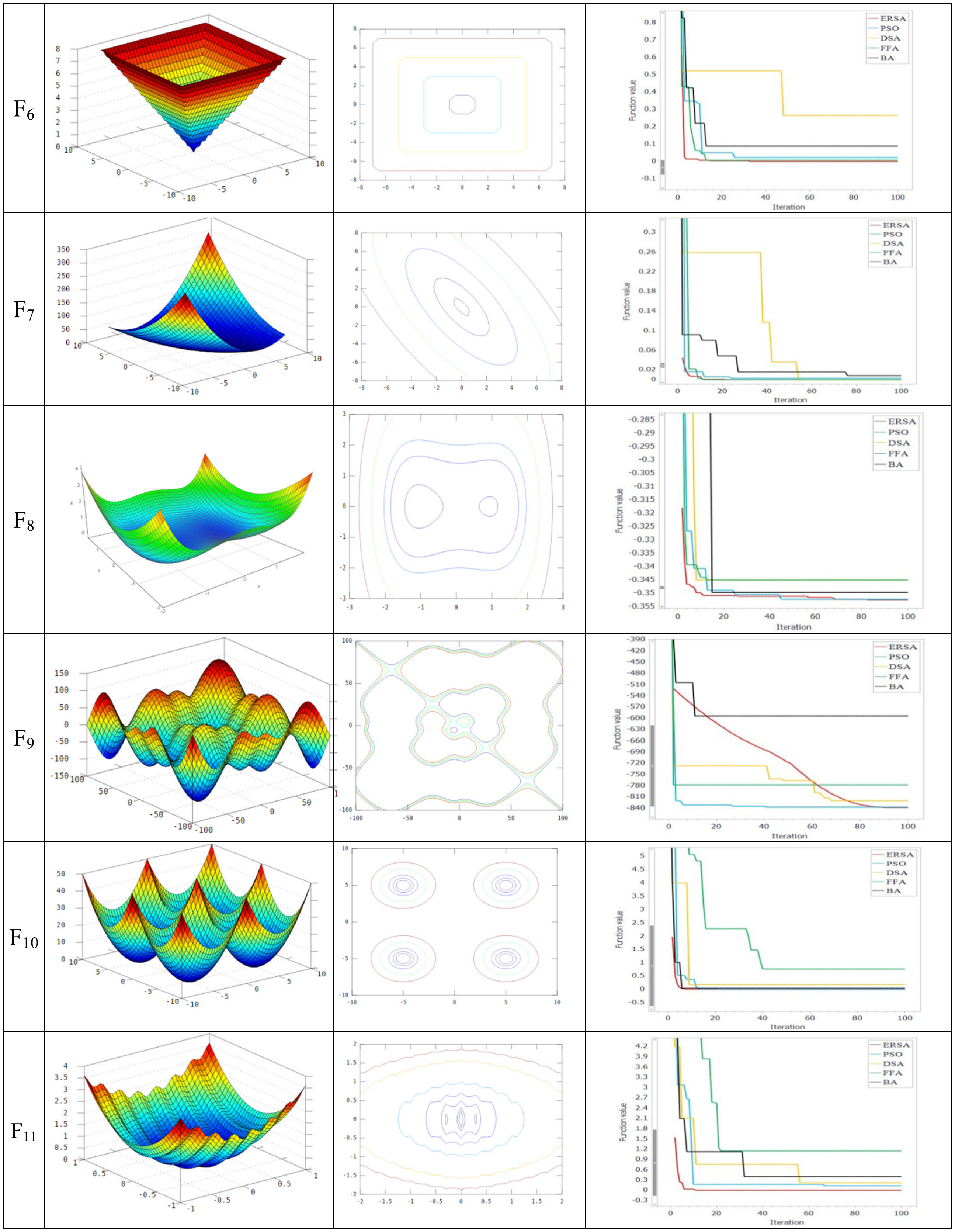
Function ID	2D-dimention surface	Contour lines	Convergence curve
F_1			
F_2			
F_3			
F_4			
F_5			

Fig. 7 Additional information on the benchmark functions

(b)


F_6
F_7
F_8
F_9
F_{10}
F_{11}
ERSA
PSO
DSA
FFA
BA
Function value
Iteration

(c)

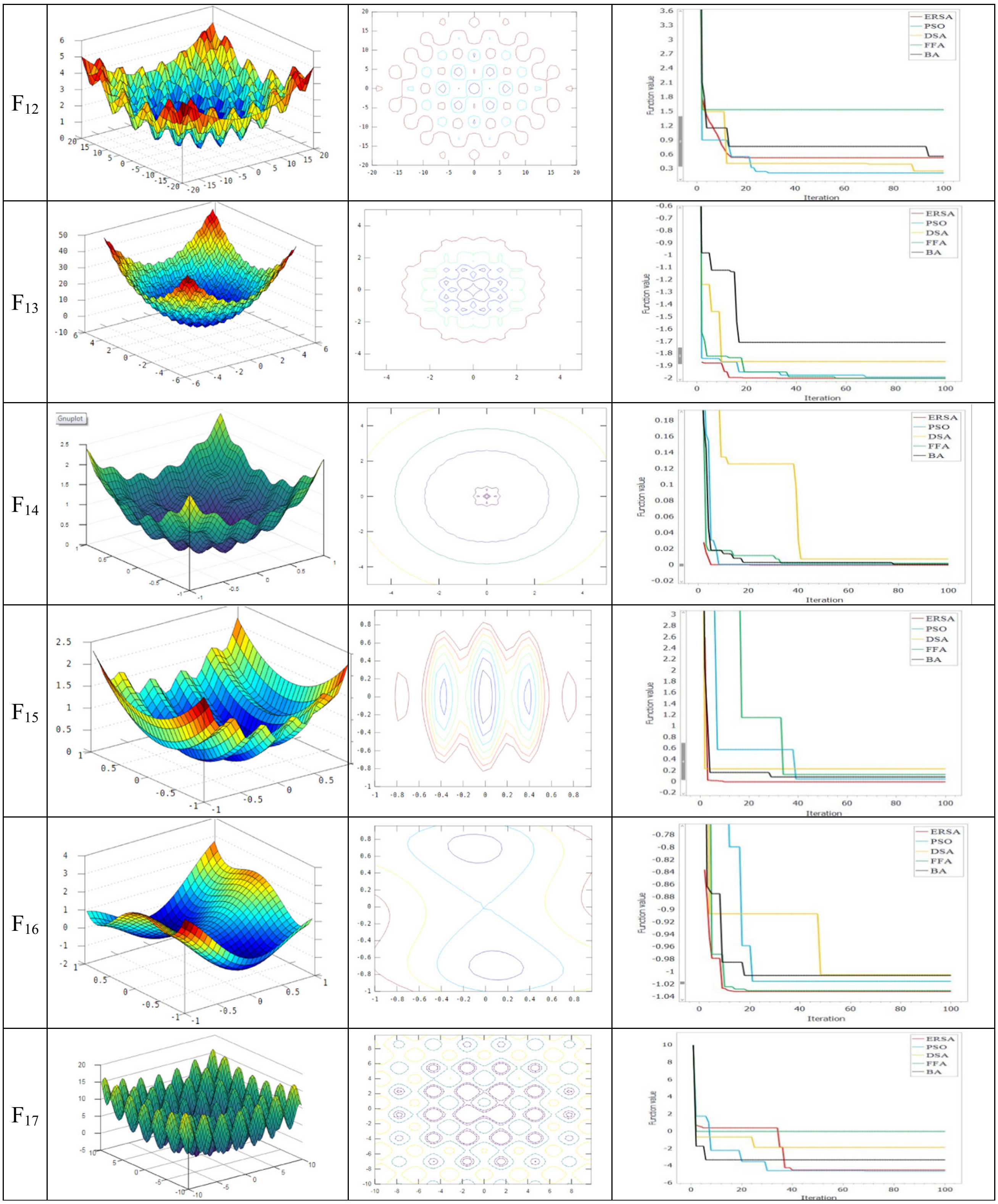


Fig. 7 continued

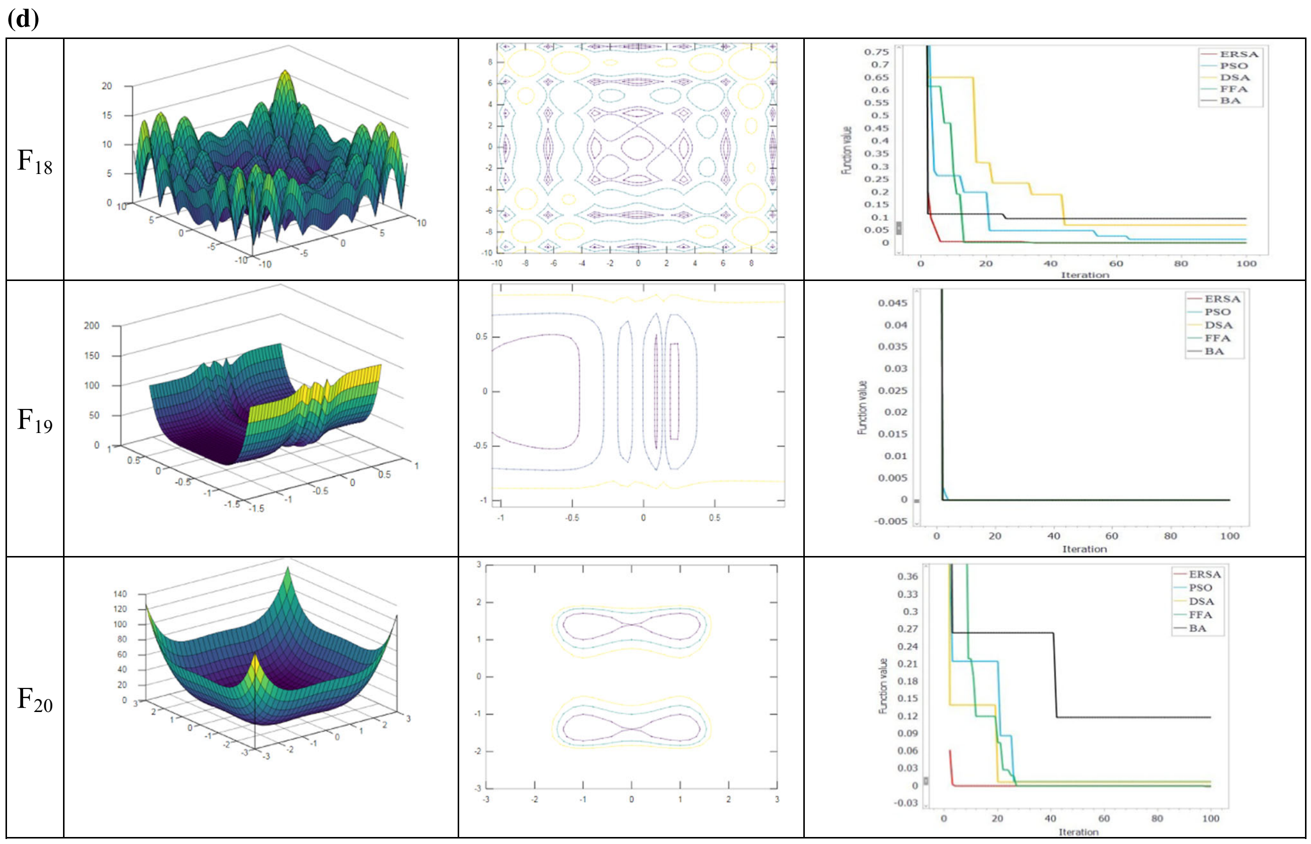


Fig. 7 continued

Variable ranges

$$\begin{aligned} &0.05 \leq x_1 \leq 2.00, \\ &0.25 \leq x_2 \leq 1.30, \\ &2.00 \leq x_3 \leq 15.0 \end{aligned} \tag{10}$$

This problem was solved by many optimization techniques in the last decades. In this regards, He and Wang (2007) solved the model using the co-evolutionary particle swarm optimization algorithm. Moreover, GA (Coello 2000), DE (Huang et al. 2007), thermal exchange optimization (TEO) (Kaveh and Dadras 2017), GWO (Mirjalili et al. 2014), and colliding bodies optimization (CBO) (Kaveh and Mahdavi 2014) are some meta-heuristic algorithms applied to solve this problem. The results obtained by the ERSA and other optimization algorithms results are presented in Table 5. It can be seen from Table 5 that the ERSA and TEO found the best design solution which is lighter than the best-known design quoted in the literature. Furthermore, as shown in Table 6, the ERSA achieved the better solution according to the worst and average design values compared with the other optimization techniques. Additionally, the statistical results showed the coefficient of variation (CV) in 30 independent runs is less than 1.127E–4 for the ERSA.

4.4.2 Welded beam design

The welded beam design problem has been utilized widely in the literature to evaluate the performance of the new developed methods (Mirjalili et al. 2014; Kaveh and Dadras 2017; Kaveh and Zolghadr 2016; He and Wang 2007; Kaveh and Mahdavi 2014). This problem minimizes the total fabrication costs of a welded beam structure subject to shear stress (s), bending stress in the beam (r), buckling load (Pc), end deflection (d), and side constraints. Furthermore, thickness of weld (h), length of the attached part of the bar (l), the height of the bar (t), and thickness of the bar (b) are four problem variables as shown in Fig. 11. The welded beam design mathematical programming can be expressed as below:

Minimize

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{11}$$

Subject to

Table 3 Multimodal benchmark functions

Functions	Name	Expression	Search space	f min
F_8	Aluffi-Pentini	$F(X) = \frac{1}{4}x_1^4 - \frac{1}{2}x_1^2 + \frac{1}{10}x_1 + \frac{1}{2}x_2^2$	$[-10,10]^2$	-0.352386
F_9	Schwefel	$F(X) = \sum_{i=1}^{n} x_i \cdot \sin\left(\sqrt[2]{\lvert x_i\rvert}\right)$	$[-500,500]^2$	-837.9658
F_{10}	Becker and Lago	$F(X) = \sum_{i=1}^{n}(\lvert x_i\rvert - 5)^2$	$[-10,10]^{100}$	0
F_{11}	Bohachevsky 2	$F(X) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	$[-10,10]^2$	0
F_{12}	Griewank	$F(X) = 1 + 0.005\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-100,100]^2$	0
F_{13}	Rastrigin	$F(X) = \sum_{i=1}^{2}(x_i^2 - \cos(18x_i))$	$[-1,1]^2$	-2
F_{14}	Cosine mixture	$F(X) = 0.4 + \sum_{i=1}^{4} x_i^2 - 0.1\sum_{i=1}^{4}\cos(5\pi x_i)$	$[-1,1]^4$	0
F_{15}	Branin	$F(X) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	$[-15,15]^2$	0.398
F_{16}	Six-Hump Camel Back	$F(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	-1.0316285
F_{17}	Ackley 4	$F(X) = \sum_{i=1}^{n}\left[e^{-0.2}.\sqrt{x_i^2 + x_{i+1}^2} + 3(\cos(2x_i) + \sin(2x_{i+1}))\right]$	$[-35, 35]^{100}$	-4.593268
F_{18}	Alpine 1	$F(X) = \sum_{i=1}^{n}\lvert x_i\sin(x_i) + 0.1x_i\rvert$	$[-10, 10]^{100}$	0
F_{19}	Csendes	$F(X) = \sum_{i=1}^{n} x_i^6\left(2 + \sin\frac{1}{x_i}\right)$	$[-1, 1]^{100}$	0
F_{20}	Qing	$F(X) = \sum_{i=1}^{n}\left(x_i^2 - i\right)^2$	$[-10, 10]^{100}$	0

$$\begin{aligned}
&g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0,\\
&g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0,\\
&g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0,\\
&g_4(\vec{x}) = x_1 - x_4 \le 0,\\
&g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,\\
&g_6(\vec{x}) = 0.125 - x_1 \le 0,\\
&g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0,
\end{aligned} \tag{12}$$

Variable ranges

$$\begin{aligned}
&0.1 \le x_1 \le 2,\\
&0.1 \le x_2 \le 10,\\
&0.1 \le x_3 \le 10\\
&0.1 \le x_4 \le 2
\end{aligned} \tag{13}$$

where

$$\begin{aligned}
&\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}\\
&\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right),\\
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\\
&J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\
&\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}
\end{aligned} \tag{14}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\,\text{lb}, L = 14\,\text{in.}, \delta_{\max} = 0.25\,\text{in.}, E = 30 \times 10^6\,\text{psi}, G = 12 \times 10^6\,\text{psi}$$

$$\tau_{\max} = 13,600\,\text{psi}, \sigma_{\max} = 30,000\,\text{psi}$$

In this regard, many researches have solved this problem using developed methods. Coello (2000) and Deb (1991) employed GA, and also Lee and Geem (2005) have utilized an improved HS algorithm to solve this model. Moreover, CSS (Kaveh and Talatahari 2010), GWO (Mirjalili et al. 2014), TWO (Kaveh and Zolghadr 2016), and BA (Gandomi et al. 2013) are some algorithms applied to this problem. Table 7 presents the ERSA optimization results and compares it to the other developed methods. It can be seen that solution obtained by the ERSA is better than the other algorithms solution. As the results showed, the ERSA decreases fabrication costs more than 0.3% with determining better designing parameters. Moreover, statistical results for 30 independent runs are shown in Table 8. The statistical data reported in Table 8 show that the ERSA results in the best, worst, average, and the standard deviation of solutions are fully consistent with the literature.

5 Conclusions

This paper proposed a novel meta-heuristic algorithm called ERSA that was inspired by the electron discharge mechanism in an electric field. The ERSA uses electrons avalanche and streamer mechanism concept to find the optimal solution in the search space. Electrons search the surrounding environment with a radar mechanism and select the least resistance path for the next move, and this

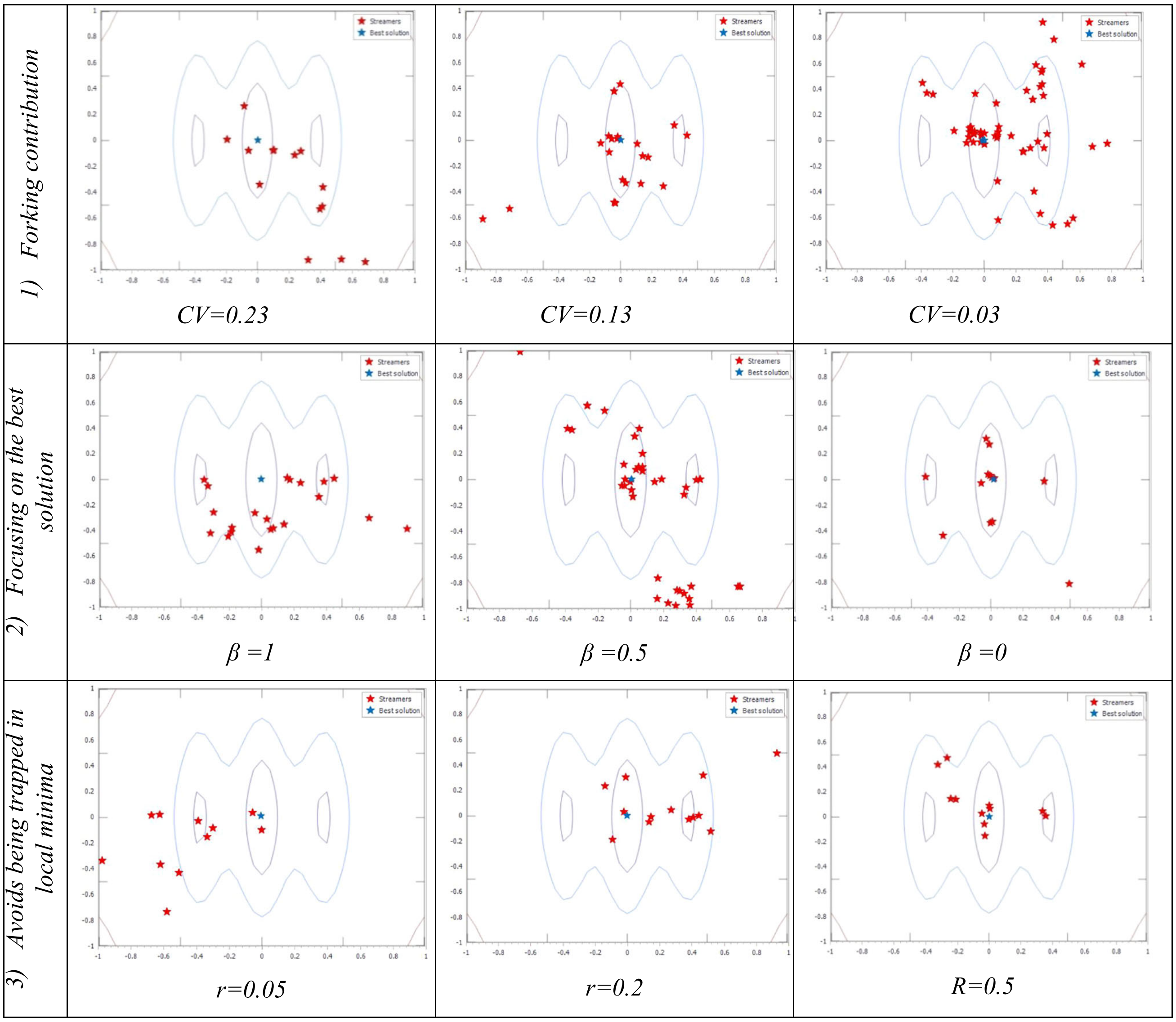


Fig. 8 The ERSA performance in different input parameters

Table 4 Optimization results for the studied benchmark functions

Function	Statistics	ERSA	DSA	PSO	FFA	BSA
F_1 Sphere	Best	**0.0000126935**	0.0543621447	0.0912455476	0.0841263598	0.0196910416
	Worst	**0.0044875569**	2.2100326589	0.1551412536	4.4125656598	0.1359831075
	Average	**0.0008001459**	0.4125336532	0.0861240036	0.9010234586	0.0208017645
	Standard deviation	**0.0000112369**	0.0981745441	0.0424171563	0.1301474456	0.0052270567
	Running time (s)	14.8735189	**12.7155823**	13.8096385	14.9158476	14.0197398
F_2 Step	Best	**0**	**0**	**0**	**0**	**0**
	Worst	**0**	12.365488974	**0**	24	2.3659856324
	Average	**0**	2.7166666666	**0**	4.5666666666	0.8666666666
	Standard deviation	**0**	0.7236598541	**0**	0.9811239576	0.0871124173
	Running time (s)	15.8273174	12.9190967	**11.9898423**	14.1294992	14.1506888
F_3 Quartic	Best	**2.195685E−12**	0.0132659874	0.0000003123	0.0000002365	0.0000060635
	Worst	**0.0000043265**	3.4123186594	1.0214589647	112.21364475	0.2236584954
	Average	**0.0000000336**	0.4789546329	0.0121459666	12.712569399	0.0728569002
	Standard deviation	**0.0000000111**	0.0661247889	0.0274598663	7.1240036984	0.0080385694
	Running time (s)	16.9328894	**13.8351907**	14.9591790	15.1216641	16.1857506
F_4 Rosenbrock	Best	**0.0000196413**	0.5516708449	0.0017527144	0.0008096135	0.0190446276
	Worst	**0.2483695372**	18.562865911	5.7486422839	3.7794864566	27.451636884
	Average	**0.0192826503**	1.5527464681	0.8386140649	0.6280125726	3.5790772145
	Standard deviation	**0.0087229665**	0.5968459801	0.2509893620	0.1766286376	1.1987155907
	Running time (s)	4.3703829	**3.5793722**	3.6397968	3.8184892	3.8325022
F_5 Booth	Best	**0.0000001284**	0.0054970804	0.0000078864	0.0000208229	0.0012605889
	Worst	**0.0001584672**	1.8874344394	0.0868159321	0.3925022203	1.8764964764
	Average	**0.0000205843**	0.5842982423	0.0134541223	0.0264878274	0.1748830717
	Standard deviation	**0.0000061540**	0.1335194706	0.0032217584	0.0164589382	0.0849026480
	Running time (s)	4.9924321	3.7953886	**3.7775033**	3.9802230	3.9568766
F_6 Schwefel 2.21	Best	**0.0003291847**	0.0253474572	0.0084307925	0.0025731912	0.0337243757
	Worst	**0.0096903449**	2.1708378066	0.4387514521	3.3844486607	0.6339667223
	Average	**0.0028549044**	0.3745238903	0.1125122017	0.6554381027	0.1914518284
	Standard deviation	**0.0004557241**	0.0751425240	0.0176598064	0.1654317921	0.0224672350
	Running time (s)	4.3967716	**3.4957970**	3.5837364	3.7164207	3.7535927
F_7 Schwefel 1.2	Best	**0.0000034236**	0.0096853214	0.0003142593	0.0007889856	0.0048965321
	Worst	**0.0004123655**	5.1253695874	0.1258963654	26.478965321	1.6359854756
	Average	**0.0001235688**	0.7124596859	0.0224578621	3.4789001256	0.4158659452
	Standard deviation	**0.0004420032**	0.6354694752	0.0087756966	0.9653854212	0.3147965532
	Running time (s)	15.6454450	12.5536317	**11.6132316**	14.7358925	15.8033551
F_8 Aluffi-Pentini	Best	**− 0.3523860738**	− 0.3517789235	− 0.3521813077	− 0.3523846324	− 0.3523787581
	Worst	− 0.1526394417	0.4655852417	**− 0.3299562369**	7.9574540098	− 0.2312091958
	Average	− 0.3324114105	− 0.2759004207	**− 0.3480691936**	0.0544586336	− 0.3249396361

Table 4 (continued)

Function	Statistics	ERSA	DSA	PSO	FFA	BSA
	Standard deviation	0.02105514373	0.0289992874	**0.0009185084**	0.2860006576	0.0064898757
	Running time (s)	4.2614402	**3.5553871**	3.6396053	3.8303477	3.7251277
***F*₉** Schwefel	Best	− **837.96575063**	− 837.49782822	− 837.93498191	− 758.31077164	− 837.85755615
	Worst	− **601.08816243**	− 580.27633043	− 592.31216879	− 502.26141739	− 438.11491110
	Average	− **756.98841299**	− 745.66059199	− 745.96905730	− 614.98424507	− 664.53105143
	Standard deviation	14.759803037	13.314550917	13.7449277525	**12.7598293368**	20.420482023
	Running time (s)	4.2444659	**3.3212525**	3.3775655	3.6406592	3.5499804
***F*₁₀** Becker and Lago	Best	**0.0000311487**	0.0014785632	0.0011245896	0.0003874536	0.0042542569
	Worst	**0.0214856231**	3.8745236220	1.3659856422	8.3265985654	4.2153965884
	Average	**0.0007444785**	0.9874521036	0.0887956421	1.2214586339	0.0965325866
	Standard deviation	**0.0005214586**	0.4102547856	0.0296548563	0.9743652311	0.2365985422
	Running time (s)	19.6688286	16.5670846	**14.6546794**	18.8791853	23.9087196
***F*₁₁** Bohachevsky 2	Best	**0.0000010263**	0.0232432639	0.0019110428	0.0002596456	0.0145451127
	Worst	0.3256215497	3.2823906232	**0.2845259632**	7.9995851987	0.5592994636
	Average	**0.0535298959**	0.6404569372	0.1272920554	0.7186914386	0.2974215299
	Standard deviation	0.0194256253	0.1182460750	**0.0184906136**	0.3355269663	0.0280798426
	Running time (s)	4.6179668	**3.4984425**	3.6465960	3.7673878	3.8432397
***F*₁₂** Griewank	Best	0.0007562427	0.0033997271	0.0008827278	0.1996770138	**0.0002395357**
	Worst	1.3182233452	0.5671686649	**0.2057269279**	2.0105133449	0.3623249103
	Average	0.3846983230	0.2554430367	**0.0645399263**	0.8207017011	0.1363360376
	Standard deviation	0.0710743410	0.0255468062	**0.0123748241**	0.0794221781	0.0175142970
	Running time (s)	4.7787636	**3.7192371**	3.7809757	4.0971575	3.9974251
***F*₁₃** Rastrigin	Best	− **1.9999919133**	− 1.9984801871	− 1.9991616546	− 1.9987823399	− 1.9993632459
	Worst	− 1.7577973189	− 1.6266095265	− 1.8631471564	− **1.9778845735**	− 1.3306392285
	Average	− 1.9578390243	− 1.8486462072	− 1.9618592136	− **1.9934292748**	− 1.8813336045
	Standard deviation	0.01215424658	0.0226395346	0.0088850785	**0.00094023070**	0.0256581568
	Running time (s)	4.8005912	**3.8410008**	3.9927653	4.2181285	4.1336885
***F*₁₄** Cosine mixture	Best	**0.0000001094**	0.0000226890	0.0000042803	0.0000013503	0.0001071184
	Worst	0.1477897047	0.2071301280	0.0089628233	**0.0018160461**	0.0418751522
	Average	0.0279586764	0.0376795294	0.0014210015	**0.0007062095**	0.0121046348
	Standard deviation	0.0106606629	0.0100023674	0.0003643068	**0.0001037312**	0.0026699677
	Running time (s)	5.4834995	**4.3976854**	4.4514192	4.6960075	4.6475623
***F*₁₅** Branin	Best	**0.3978873695**	0.3978970193	0.3980311829	0.3978926357	0.3990537731
	Worst	**0.3984309096**	1.7352295028	0.5117292713	0.4624070675	0.7637011218
	Average	**0.3979797875**	0.6141891609	0.4076302282	0.4048996144	0.4357680962
	Standard deviation	**0.0000289217**	0.0521291678	0.0039562460	0.0027360880	0.0124269535
		4.5222256	**3.4724079**	3.5957343	3.7763266	3.8108006

Table 4 (continued)

Function	Statistics Running time (s)	ERSA	DSA	PSO	FFA	BSA
F_{16} Six-Hump Camel Back	Best	− **1.0316284401**	− 1.0261714195	− 1.0312991120	− 1.0316193437	− 1.0306465223
	Worst	− 0.2154630537	0.8115349221	− 0.9498571204	− **1.0171651067**	− 0.6647976442
	Average	− 1.0043308035	− 0.7641816719	− 1.0159237714	− **1.0296404417**	− 0.9824152765
	Standard deviation	0.0276674093	0.0797801354	0.0034526918	**0.00063329433**	0.0136078439
	Running time (s)	4.6590905	**3.6718246**	3.6725383	3.8557413	3.8437935
F_{17} Ackley 4	Best	− **4.5900655072**	− 4.4770388808	− 4.5864506849	− 4.0971845659	− 4.4308436848
	Worst	− **0.0767032183**	1.2557259888	− 3.0591063564	10.1043746530	− 0.2282177150
	Average	− **4.1239429505**	− 2.4971182991	− 4.0812498260	3.71274154310	− 2.7979610739
	Standard deviation	**0.2057455765**	0.2870984814	0.0836275789	0.72347784819	0.19552012914
	Running time (s)	15.5738182	13.4812846	14.5179029	14.7676127	13.7339640
F_{18} Alpine 1	Best	**0.0001748201**	0.0027872125	0.0008057404	0.0003009481	0.0008255522
	Worst	**0.0029643928**	0.6594990448	0.3776791574	0.0730130045	1.0165180160
	Average	**0.0009180795**	0.1151773271	0.0300212391	0.0124206805	0.1206337721
	Standard deviation	**0.0001250802**	0.0252016346	0.0135394924	0.0032959566	0.0371929668
	Running time (s)	18.0411755	15.7892339	15.8412342	14.0686711	14.0173556
F_{19} Csendes	Best	**1.094219E−28**	2.801765E−17	7.448558E−17	3.090859E−18	1.009460E−19
	Worst	**1.862499E−14**	0.0000024563	4.529905E−10	2.558006E−13	0.0000000375
	Average	**1.375928E−15**	0.0000001491	3.177963E−11	4.404651E−14	0.0000000022
	Standard deviation	**7.492856E−16**	0.0000000884	1.656706E−11	1.420352E−14	0.0000000013
	Running time (s)	20.6906020	16.4694485	19.5962155	16.8466520	18.8265655
F_{20} Qing	Best	**0.0000004152**	0.0001762621	0.0011813659	0.0000470379	0.0000169221
	Worst	**0.0011991912**	1.2054775340	0.1683650380	43.701354081	0.6895830982
	Average	**0.0001074122**	0.3479357902	0.0291718465	1.8008669187	0.1035906398
	Standard deviation	**0.0000418669**	0.0720063675	0.0080807461	1.4878602864	0.0256846011
	Running time (s)	16.8362021	13.2536219	13.2969172	15.4984034	15.6627949

The bold numbers are considered as best results

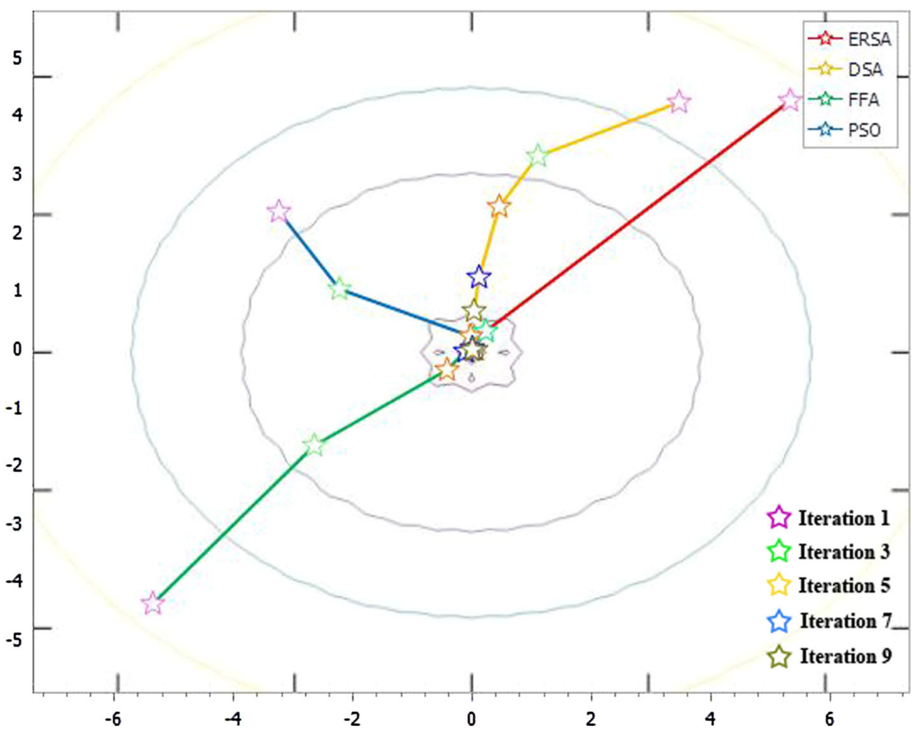


Fig. 9 The trend of investigated algorithms in finding optimum solution

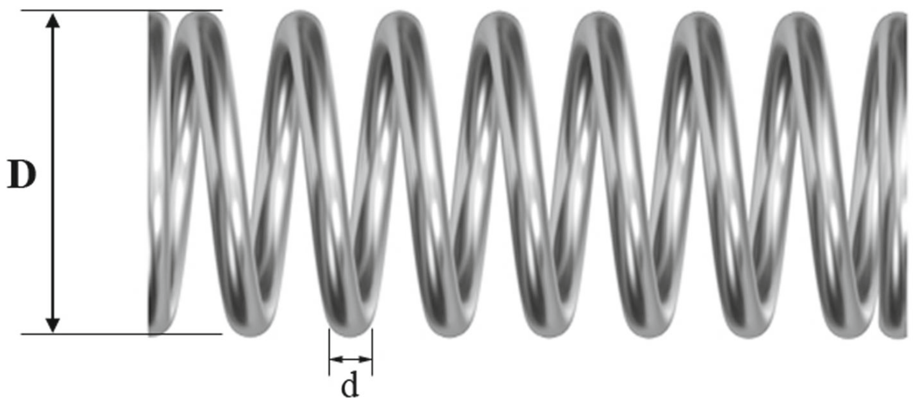


Fig. 10 Schematic view of the tension/expression spring

Table 5 Comparison of the optimization results obtained in the tension/compression spring problem

Algorithms	Optimization variables			Weight
	$x_1(d)$	$x_2(D)$	$x_3(N)$	
CPSO (He and Wang 2007)	0.051728	0.357644	11.24454	0.012674
GA (Coello 2000)	0.051480	0.351661	11.63220	0.012704
GWO (Mirjalili et al. 2014)	0.051690	0.356737	11.28885	0.012666
CBO (Kaveh and Mahdavi 2014)	0.051894	0.361674	11.00784	0.012669
TEO (Kaveh and Dadras 2017)	0.051775	0.358791	11.16839	0.012665
DE (Huang et al. 2007)	0.051609	0.354714	11.41083	0.012670
Mathematical optimization (Belegundu and Arora 1985)	0.050000	0.315900	14.25000	0.012833
Constraint correction (Arora 2011)	0.053396	0.399180	9.185400	0.012730
Present work	0.051814	0.359711	11.11467	0.012665

Table 6 Statistical results of different methods in the tension/compression spring problem

Algorithms	Best	Worst	Average	Std dev.
CPSO (He and Wang 2007)	0.012674	0.012924	0.012730	5.1985E−5
GA (Coello 2000)	0.012704	0.012822	0.012769	3.9390E−5
GWO (Mirjalili et al. 2014)	0.012666	N/A	N/A	N/A
CBO (Kaveh and Mahdavi 2014)	0.012669	0.128808	0.127296	5.0037E−5
TEO (Kaveh and Dadras 2017)	0.012665	0.012715	0.012685	4.4079E−6
DE (Huang et al. 2007)	0.012670	0.012793	0.012703	2.7020E−5
Mathematical optimization (Belegundu and Arora 1985)	0.012833	N/A	N/A	N/A
Constraint correction (Arora 2011)	0.012730	N/A	N/A	N/A
Present work	0.012665	0.012701	0.012684	1.6101E−6

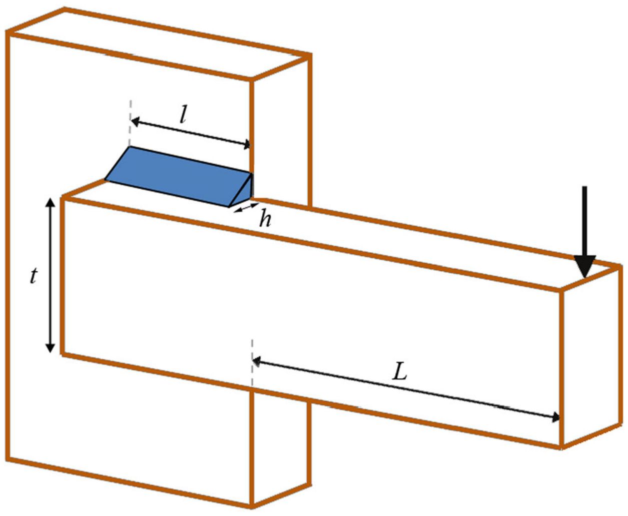


Fig. 11 Schematic view of the welded beam structure

procedure is continued until satisfying the algorithm finishing condition. Additionally, this algorithm utilizes streamers' branching and eliminating concept for further investigation. To test the efficiency of the algorithm, sixteen well-known unimodal and multimodal benchmark functions were employed. The proposed ERSA determines satisfactory exploration, exploitation, and convergence characteristics. The results showed that the ERSA produces competitive solutions compared to the most popular algorithms including PSO, DSA, FFA, and BSA. Additionally, the proposed algorithm found the better solution in more than 93% of the reviewed benchmark functions. Moreover, the proposed algorithm was tested in the engineering design problem and the results showed the ERSA is able to present high-performance solution in the search space.

For future researches, it is recommended to develop binary and multi-objective forms of the ERSA. Furthermore, sensitivity study and true adjustment of the

Table 7 Comparison of the optimization results obtained in the welded beam design

Algorithms	Optimization variables				Cost
	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	
CPSO (He and Wang 2007)	0.202369	3.544214	9.04821	0.205723	1.728024
GA (Deb 1991)	0.2489	6.1730	8.1789	0.2533	2.433116
GWO (Mirjalili et al. 2014)	0.205676	3.478377	9.03681	0.205778	1.72624
CBO (Kaveh and Mahdavi 2014)	0.205722	3.47041	9.037276	0.205735	1.724663
TEO (Kaveh and Dadras 2017)	0.205681	3.472305	9.035133	0.205796	1.725284
BA(Huang et al. 2007)	0.2015	3.562	9.0414	0.2057	1.7312065
HS (Lee and Geem 2005)	0.2442	6.2231	8.2915	0.2443	2.3807
Geometric Programming (Ragsdell and Phillips 1976)	0.2444	6.2189	8.2915	0.2444	2.3815
Present work	0.202674	3.516789	9.101871	0.203189	1.7181387

Table 8 Statistical results of different methods in the welded beam design

Algorithms	Best	Worst	Average	Std dev.
CPSO (He and Wang 2007)	1.728024	1.782143	1.748831	0.012926
GA (Deb 1991)	2.433116	N/A	N/A	N/A
GWO (Mirjalili et al. 2014)	1.72624	N/A	N/A	N/A
CBO (Kaveh and Mahdavi 2014)	1.724662	1.725707	1.725059	0.0002437
TEO (Kaveh and Dadras 2017)	1.725284	1.931161	1.768040	0.0581661
BA(Huang et al. 2007)	1.7312065	2.3455793	1.8786560	0.2677989
HS (Lee and Geem 2005)	2.3807	N/A	N/A	N/A
Geometric Programming (Ragsdell and Phillips 1976)	2.3815	N/A	N/A	N/A
Present work	1.7181387	1.8653583	1.7716815	0.0066584

parameters, especially r and β, can improve the convergence rate of the ERSA.

Compliance with ethical standards

Conflict of interest Authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abbas NM, Solomon DG, Bahari MF (2007) A review on current research trends in electrical discharge machining (EDM). Int J Mach Tools Manuf 47:1214–1228

Abraham A, Das S, Roy S (2008) Swarm intelligence algorithms for data clustering. Springer, Boston, MA, Soft Comput. Knowl. Discov. Data Min., pp 279–313

Arora J (2011) Introduction to optimum design, 3rd Editio edn. Elsevier, Amsterdam

Bäck T, Schwefel H-P (1995) An overview of evolutionary algorithms for parameter optimization. Evol Comput 1:1–23

Banzhaf W, Nordin P, Keller RE, Francone FD (1997) Genetic programming: an introduction, 1st edn. Morgan Kaufmann, Burlington

Beauchamp K (2001) History of telegraphy, 1st edn. The Institution of Engineering and Technology, London

Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part I: theory. Int J Numer Methods Eng 21:1561–1748

Byrne C, Tainsky M, Fuchs E (1994) Programming gene expression in developing epidermis. Development 120:2369–2383

Cacchiani V, D'Ambrosio C (2017) A branch-and-bound based heuristic algorithm for convex multi-objective MINLPs. Eur J Oper Res 260:920–933

Cantarella GE, de Luca S, di Pace R, Memoli S (2015) Network signal setting design: meta-heuristic optimisation methods. Transp Res Part C Emerg Technol 55:24–45

Chen T, Tsao HL (2009) Using a hybrid meta-evolutionary rule mining approach as a classification response model. Expert Syst Appl 36:1999–2007

Christensen J, Bastien C (2015) Seven: heuristic and meta-heuristic optimization algorithms. In: Christensen J (ed) Nonlinear optimization of vehicle safety structures, 1st edn. Butterworth-Heinemann, pp 277–314

Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127

Crepinsek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45:1–33

Daneshmand A, Facchinei F, Kungurtsev V, Scutari G (2015) Hybrid random/deterministic parallel algorithms for convex and non-convex big data optimization. IEEE Trans Signal Process 63:3914–3929. https://doi.org/10.1109/TSP.2015.2436357

Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15:4–31

Deb K (1991) Optimal design of a welded beam via genetic algorithms Read More. AIAA J 29:2013–2015. https://doi.org/10.2514/3.10834

Deepa O (2016) Swarm intelligence from natural to artificial systems: ant colony optimization. Int J Appl Graph Theory Wirel Ad Hoc Netw Sens Netw 8:9–17

Du D-Z, Pardalos PM (1999) Handbook of Combinatorial Optimization. Springer, Boston. https://doi.org/10.1007/978-1-4613-0303-9

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. Micro Mach. Hum. Sci, Nagoya

Ebrahimi M, ShafieiBavani E, Wong RK, Fong S, Fiaidhi J (2017) An adaptive meta-heuristic search for the internet of things. Futur Gener Comput Syst 76:486–494

Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37:106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005

Floudas CA (2000) Deterministic global optimization: theory, methods and applications, 1st edn. Springer, US

Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 3:1–16

Fridman A, Chirokov A, Gutsol A (2005) Non-thermal atmospheric pressure. J Phys D Appl Phys 38:1–24. https://doi.org/10.1088/0022-3727/38/2/R01

Gandhi KR, Uma SM, Karnan M (2012) A hybrid meta heuristic algorithm for discovering classification rule in data mining. Int J Comput Sci Netw Secur 12:116–122

Gandomi AH, Yang X-S, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural Comput Appl 22:1239–1255. https://doi.org/10.1007/s00521-012-1028-9

Glover F (1989) Tabu search—part I. ORSA J Comput 1:190–206. https://doi.org/10.1287/ijoc.1.3.190

Goldenberg M (2017) The heuristic search research framework. Knowledge-Based Syst 129:1–3

Griffis SE, Bell JE, Closs DJ (2012) Metaheuristics in logistics and supply chain management. J Bus Logist 33:90–106

Gutjahr WJ (2010) Stochastic search in metaheuristics. In: Price CC, Zhu J (eds) International series in operations research and management science, 1st edn. Springer, Boston, pp 573–97

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

Hills TT, Todd PM, Lazer D, Redish AD, Couzin ID (2015) Exploration versus exploitation in space, mind, and society. Trends Cogn Sci. https://doi.org/10.1016/j.tics.2014.10.004

Ho K, Newman S (2003) State of the art electrical discharge machining (EDM). Int J Mach Tools Manuf 43:1287–1300

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor

Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. MIT Press Cambridge, MA

Horst R, Tuy H (1996) Global optimization: Deterministic approaches, 3rd edn. Springer-Verlag, Berlin Heidelberg

Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356

Junqin XU, Jihui Z (2014) Exploration-exploitation tradeoffs in metaheuristics: survey and analysis. In: Proc. 33rd Chinese control conf, pp 8633–8

Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84. https://doi.org/10.1016/j.advengsoft.2017.03.014

Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27. https://doi.org/10.1016/j.compstruc.2014.04.005

Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system. Acta Mech 213:267–289. https://doi.org/10.1007/s00707-009-0270-4

Kaveh A, Zolghadr A (2016) A novel meta-heuristic algorithm: tug of war optimization. Int J Optim Civ Eng 6:469–492

Keidar M, Beilis I (2013) Plasma engineering: applications from aerospace to bio and nanotechnology. Elsevier Inc., Amsterdam

Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. Science 220:671–680

Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933

Loeb LB, Meek JM (1941) The mechanism of the electric spark. Stanford University Press, Palo Alto

Maniezzo V, Carbonaro A (2002) Ant colony optimization: an overview. Springer, Boston, MA, Oper. Res. Sci. Interfaces Ser., pp 469–492

Meek JM, Craggs JD (1978) Electrical breakdown of gases. Wiley, Hoboken

Mirjalili S, Mohammad S, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Pinebrook WE (1987) The evolution strategy. Int J Model Simul 7:81–84

Pishvaee MS, Farahani RZ, Dullaert W (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. Comput Oper Res 37:1100–1112

Qureshi AS, Khan A, Zameer A, Usman A (2017) Wind power prediction using deep neural network based meta regression and transfer learning. Appl Soft Comput 58:742–755

Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind 98:1021–1025

Ranaboldo M, García-Villoria A, Ferrer-Martí L, Moreno RP (2015) A meta-heuristic method to design off-grid community electrification projects with renewable energies. Energy 93:2467–2482

Rashedi E, Nezamabadi-pour H, Saryazdi SGSA (2009) A gravitational search algorithm. Inf Sci 179:2232–2248

Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput J 36:315–333. https://doi.org/10.1016/j.asoc.2015.07.028

Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proc. 1999 congr. evol. comput. (Cat. No. 99TH8406), IEEE, Washington, pp 1945–50

Sorensen K (2015) Metaheuristics—the metaphor exposed. Int Trans Oper Res 22:3–18. https://doi.org/10.1111/itor.12001

Vidal T, Battarra M, Subramanian A, Erdogan G (2015) Hybrid metaheuristics for the clustered vehicle routing problem. Comput Oper Res 58:87–99

Walters JP (1969) Historical advances in spark emission spectroscopy. Appl Spectrosc 23:317–331

Xhafa F, Abraham A (2008) Metaheuristics for scheduling in industrial and manufacturing applications. Springer-Verlag, Berlin Heidelberg

Xiao D (2016) Gas discharge and gas insulation. Springer-Verlag, Berlin Heidelberg

Yang X-S (2009) Firefly algorithms for multimodal optimization. Springer, Berlin, Heidelberg, Stoch. Algorithms Found. Appl., pp 169–178

Yang X-S (2010) A new metaheuristic bat-inspired algorithm. Springer, Berlin, Heidelberg, Nat. Inspired Coop. Strateg. Optim., pp 65–74

Yao X, Liu Y (1996) Fast evolutionary programming. In: Proceedings of the fifth annual conference on evolutionary programming, pp 451–60

Zaepffel C, Hong D, Bauchire J-M (2007) Experimental study of an electrical discharge used in reactive media ignition. J Phys D Appl Phys 40:1052–1058

Zheng Y-J (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11