Complex & Intelligent Systems (2022) 8:2791–2808
https://doi.org/10.1007/s40747-021-00510-x

ORIGINAL ARTICLE

A hybrid ant lion optimization chicken swarm optimization algorithm for charger placement problem

Sanchari Deb[1] · **Xiao-Zhi Gao**[2]

Received: 22 December 2020 / Accepted: 17 August 2021 / Published online: 6 September 2021

Abstract

Transportation electrification is known to be a viable alternative to deal with the alarming issues of global warming, air pollution, and energy crisis. Public acceptance of Electric Vehicles (EVs) requires the availability of charging infrastructure. However, the optimal placement of chargers is indeed a complex problem with multiple design variables, objective functions, and constraints. Chargers must be placed with the EV drivers' convenience and security of the power distribution network being taken into account. The solutions to such an emerging optimization problem are mostly based on metaheuristics. This work proposes a novel metaheuristic considering the hybridization of Chicken Swarm Optimization (CSO) with Ant Lion Optimization (ALO) for effectively and efficiently coping with the charger placement problem. The amalgamation of CSO with ALO can enhance the performance of ALO, thereby preventing it from getting stuck in the local optima. Our hybrid algorithm has the strengths from both CSO and ALO, which is tested on the standard benchmark functions as well as the above charger placement problem. Simulation results demonstrate that it performs moderately better than the counterpart methods.

Keywords Swarm intelligence · Ant lion optimization · Chicken swarm optimization · Charger · Electric vehicle · Optimization · Metaheuristics

Abbreviations

ALO	Ant lion optimization
ACO	Ant colony optimization
BA	Bat algorithm
BSA	Binary lighting search algorithm
DE	Differential evolution
EV	Electric vehicle
CSO	Chicken swarm optimization
CMA-ES	Covariance matrix adaptation evolution strategy
GA	Genetic algorithm
PSO	Particle swarm optimization
NFL	No free lunch
RCCRO	Real coded chemical reaction optimization
SPC-PNX	Real parameter genetic algorithm
SAIFI	System average interruption frequency index
SAIDI	System average interruption duration index
TLBO	Teaching learning based optimization
V2G	Vehicle to grid

* Sanchari Deb
sancharideb@yahoo.co.in

1 VTT Technical Research Centre, Espoo, Finland

2 School of Computing, University of Eastern Finland, Kuopio, Finland

Introduction

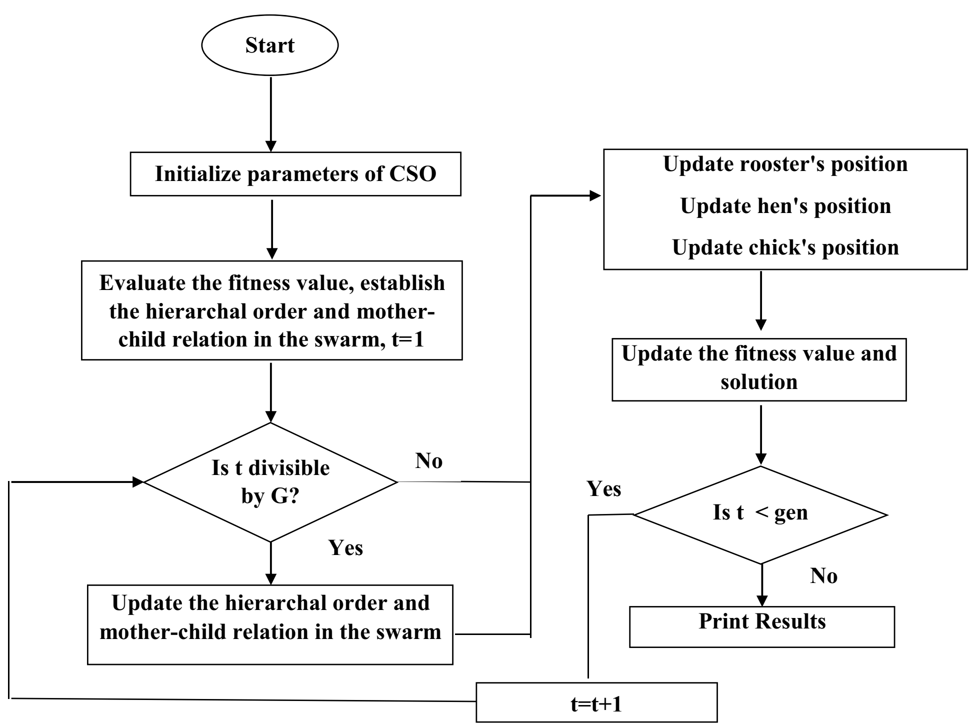
Energy crisis, poor air quality index, and global warming have been some of the major concerns during the past decade. Replacement of the conventional mode of transport powered by fuel with Electric Vehicles (EVs) is a feasible alternate to handle these issues. Adoption of EVs needs the availability of charging facilities. Charging infrastructure needs to be placed according to the charging needs of drivers. Moreover, the addition of EV charger load can increase the load of the power grid. Unfortunately, placement of chargers at the weak points of the power network and uncoordinated charging may lead to voltage instability, spikes in load curve, degradation of reliability indices, power losses, and harmonics [1–9]. Thus, the placement of EV charger must consider both the convolution of transport and distribution network [10]. The conventional algorithms based on differentiation, such as steepest descent and Newton method, have their limitations in coping with the charger placement problem, due to the involvement of

Table 1 Variants of CSO algorithm

References	Year	Descriptions
[24]	2016	Modification of update mechanism of chicks and development of Markov model for convergence analysis of CSO
[25]	2017	Development of opposition learning based CSO
[26]	2017	Introduction of mutation strategy in update of hens in CSO
[27]	2017	Development of chaotic CSO
[28]	2016	Hybridization of CSO with Bat Algorithm (BA)
[29]	2019	Modified update of rooster, cock, hens, and population update strategy in CSO
[30]	2020	Development of clustering algorithm based CSO
[31]	2020	Introduction of improved search strategy with Levy flight in the hen's location update in CSO
[32]	2020	Hybridization of CSO with Tabu search
[33]	2020	Modified update of chicks by adding inertia weights in CSO
[34]	2020	Development of quantum inspired CSO
[35]	2020	Modified update of rooster and introduction of novel constraint handling mechanism in CSO

multiple decision variables, non-linear objective functions, and constraints. In addition, the charger placement problem involves the non-linearity constraints associated with load flow. Most of the existing methods fail to effectively and efficiently handling these constraints. Indeed, metaheuristics has been widely used in engineering optimization. Deb et al. (2019) presented a comprehensive review and comparison of how the metaheuristics performs in attacking the placement problem [11]. Aljanad et al. (2018) applied the improved Binary lighting Search Algorithm (BSA) for the same problem with the Vehicle to Grid (V2G) functionality [12]. Awasthi et al. (2017) formulated the charger placement problem under the multi-objective framework considering the cost as well as operating parameters of power grid as objective functions and utilized hybrid Genetic Algorithm (GA) and Particle Swarm Optimization (GA PSO) [13]. In [14–16], the authors proposed a novel algorithm based on the hybridization of Chicken Swarm Optimization (CSO) and Teaching Learning Based Optimization (TLBO). Zhang et al. (2019) developed a multi-objective PSO with the economic factors and service abilities of the charging stations being taken into account [17]. Zeb et al. (2020) formulated the charger placement problem as a nonlinear stochastic constrained optimization problem and used PSO as an appropriate solution [18]. Mohanty et al. (2021) used Jaya algorithm with the cost as the objective function [19]. Reddy

Fig. 1 Flowchart of CSO



et al. (2020) presented a new PSO for the optimal placement of charging stations in unbalanced radial distribution network with the power loss as the objective function [20]. Amini et al. (2017) proposed an optimal placement strategy for chargers in parking lots using GA [21]. From [11–21], it can be noticed that authors have used a large variety of metaheuristics for solving the charger placement problem, and more efficient metaheuristics are attracting growing research interest. This work focuses on developing a novel hybrid algorithm considering the amalgamation of CSO with ALO. CSO is a metaheuristic mimicking the food searching mechanism of chicken in a swarm [22, 23]. It has a good utilization rate of population, but sometimes gets stuck in local optima. Several variants of CSO, as given in Table 1, have been developed to overcome the inherent shortcomings. For example, in [24], the authors have modified the update mechanism of chicks as the chicks have lowest fitness and more prone to get stuck in local optima. Additionally, they have analyzed the convergence characteristics of the modified CSO by a Markov model-based approach. Similarly, in [25], Qu et al. proposed an improved version of CSO by introducing elite opposition-based learning to promote diversity in the population. In [26], Wang et al. introduced a mutation strategy in the update mechanism of hens. In [27], Ahmed et al. discussed a chaotic map based CSO to improve the exploration capacity of basic CSO. The chaotic CSO performed better than the basic CSO on feature selection problem. In [28], Liang et al. hybridized BA with CSO and observed that the hybrid algorithm outperformed the standalone algorithms on the standard benchmark problems. In [29], Fu et al. developed an improved version of CSO with modified update equation of rooster, cock, and hen, and utilized it for solving the trajectory optimization problem. Moreover, in the same work, the authors introduced a novel constraint handling mechanism driven by assigning adaptive penalties. In [30], Osamy et al. designed a modified version of CSO based on clustering and hybridization with GA in dealing with the wireless sensor network optimization problem. In [31], Liang et al. constructed an improved version of CSO by introduction of the improved search strategy with Levy flight in the hen's location update in CSO, and used it for the path planning of robots. In [32], Niazy et al. hybridized Tabu search with CSO for the vehicle routing problem. In [33], an improved version of CSO was proposed with modified update equation of chicks by introducing inertia constants and used the improved CSO for handling the load scheduling problem. In [34], a quantum inspired CSO was discussed, which converged faster than the basic CSO. In [35], Deb et al. proposed a variation of CSO with a modified update equation of rooster and a novel constraint handling mechanism.

The optimization performance of the basic CSO can be further improved by the hybridization or modification of some of its algorithmic components. The prime motivation is the well known No Free Lunch (NFL) theorem [36], which states that any single algorithm cannot perform equally well on all the optimization problems. Thus, our work targets at enhancing the CSO by hybridizing it with ALO, which is a metaheuristic technique mimicking the hunting process of antlions [37–39]. Numerical simulations demonstrate that fine-tuning of the solutions obtained by ALO with CSO can significantly reduce the chances of getting stuck in local optima, thus leading to an enhanced convergence of the hybrid algorithm.

The rest of the paper is organized as follows. Section 2 and Sect. 3 explain the working principles of the basic CSO and ALO, respectively. Section 4 elaborates the hybrid ALO CSO. Section 5 demonstrates the performances of this new algorithm on the standard benchmark functions. Section 6 presents how ALO CSO performs on real-world complex problems. Section 7 discusses its applications in coping with the charger placement problem. Finally, Sect. 8 concludes our work with some remarks and conclusions.

CSO

CSO mimics the intelligence of swarm, and is developed by Meng et al. in 2014 [22]. It is inspired by the behaviors of chicken swarm, where the intelligence of chicken swarm is effectively utilized to obtain the optimal solution. The CSO imitates the hierarchal order in a chicken swarm and the food searching process of the swarm. More precisely, the population of chicken in the group is subdivided into dominant rooster, hens, and chicks, depending on the fitness values of the chickens. Those chickens with the highest fitness value are assigned as roosters, chickens with the least fitness value are assigned as chicks, and the chickens with the intermediate fitness value are assigned as hens. Establishment of mother–child relationship in a random manner is another salient feature of this algorithm. After every G time steps, the hierarchal order and mother–child relationship are updated. In addition, CSO utilizes the natal behaviors of hens to follow their group mate rooster and chicks to follow their mother. As a matter of fact, chickens always try to steal the food found by others, which gives rise to a competition for food in the group.

The flowchart of CSO is shown in Fig. 1, and the pseudo codes are given in Algorithm 1.

Algorithm 1 Pseudo codes of CSO [23], [24]

Initialize the population of chicken with size N, and define other algorithm specific parameters including G, size of RN, HN,CN, and MN;
Evaluate the fitness value of N chicken, t=0 , and establish the hierarchal order in the swarm and mother-child relationship;
While (t<gen)
t=t+1;
If(t%G==0)
Establish the hierarchal order in the swarm and mother-child relationship;
Else
For i=1:PN
If i==rooster
Update its solution by

$$x_{i,j}^{t+1}=x_{i,j}^{t}\times(1+\mathrm{randn}(0,\sigma^2))$$

If $f_i \le f_k$

$$\sigma^2 = 1$$

Else

$$\sigma^2 = \exp\left(\frac{(f_k - f_i)}{|f_i| + \varepsilon}\right)$$

where randn(0,σ^2) is a Gaussian distribution function with mean 0 and standard deviation σ^2. f is the fitness value of corresponding x, k is randomly selected rooster's index.ϵ is a small constant value, which is used to avoid zero division error.

End if
If i==hen
Update its solution by

$$x_{i,j}^{t+1}=x_{i,j}^{t}+S1\times\mathrm{rand}\times(x_{r1,j}^{t}-x_{i,j}^{t})+S2\times\mathrm{rand}(x_{r2,j}^{t}-x_{i,j}^{t})$$

$$S1 = \exp\left(\frac{f_i - f_{r1}}{abs(f_i) + \varepsilon}\right)$$

$$S2 = \exp(f_{r2} - f_i)$$

where rand is a randomly generated number between 0 and 1. $r1 \in [1, N]$ *is an index of rooster which is i^{th} hen's group mate.* $r2 \in [1, N]$ *is an index of rooster or hen, which is randomly chosen so that r1 is not equal to r2.*

End if
If i==chick
Update its solution by

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL \times (x_{m,j}^{t} - x_{i,j}^{t})$$

where $x_{m,j}^{t}$ represents the position of i^{th} chick's mother. FL is a parameter, which signifies that the chick would follow its mother. FL is generally chosen in between 0 and 2.

End if
Evaluate the new solutions
Update the new solutions, if they are better than the previous one
End for
End if else
End while

ALO

ALO is a novel metaheuristic algorithm mimicking the hunting process of antlions. It mathematically models the interaction of ants and antlions in nature, in which the random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps are all considered and implemented. The flowchart of ALO is shown in Fig. 2, and the pseudo codes are given in Algorithm 2.

Algorithm 2 Pseudo codes of ALO [38], [39]

Initialize randomly the first population of ants and antlions

Calculate the fitness of ants and antlions

Find the antlions with the best fitness, and consider them as elite

while the end criterion is not satisfied

for every ant

Select an antlion using roulette wheel

Update c, d by:

$$c^t = \frac{c^t}{I}$$

$$d^t = \frac{d^t}{I}$$

where c^t *is the minimum of variable at the iteration t,* d^t *is the vector containing the maximum of all variables at the iteration t, and I is a ratio dependent on the current iteration and the maximum number of iterations*

Create a random walk, and normalize it by:

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \ldots\ cumsum(2r(t_n) - 1)]$$

where cumsum calculates cumulative sum, 2r is a stochastic function, n is the maximum number of iteration, and t is the random walk step

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i$$

where a_i *is the minimum of random walk for variable i,* d_i *is the maximum of random walk for variable i,* c_i^t *is the minimum of variable i in iteration t, and* d_i^t *is the maximum of variable i at iteration t*

Update the position of ant by:

$$Ant_i^t = \frac{R_A^t}{2} + \frac{R_E^t}{2}$$

end for

Calculate the fitness of all ants

Replace an antlion with its corresponding ant, if it becomes fitter by:
$Antlion_j^t = Ant_i^t$ *if* $f(Ant_i^t) > f(Antlion^t)$
Update elite if an antlion becomes fitter than the elite
end while
Return elite

Hybrid ALO CSO

As we know that standalone algorithms are sometimes not efficient enough to manipulate with the uncertainty of real-world optimization problems. Hybridization of algorithms provides improved solutions to such emerging problems as economic load dispatch [47–49], unit commitment [50, 51],

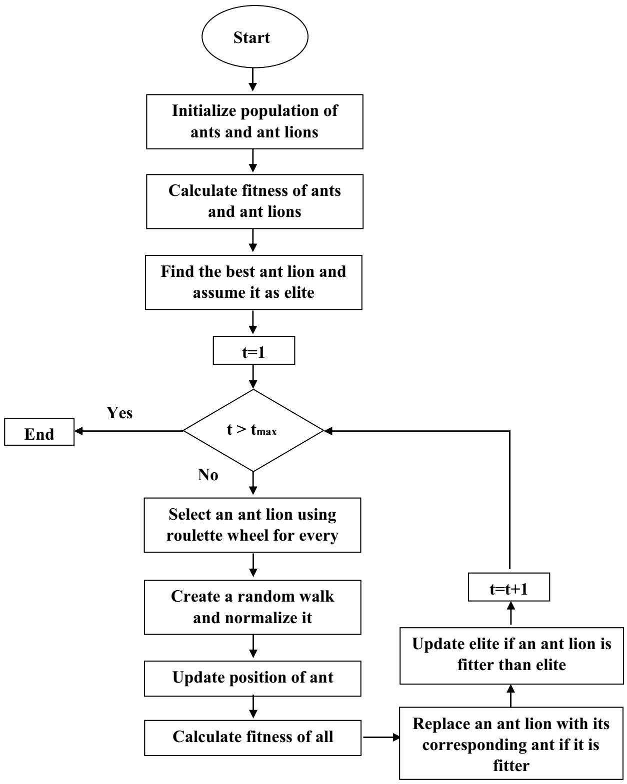


Fig. 2 Flowchart of ALO

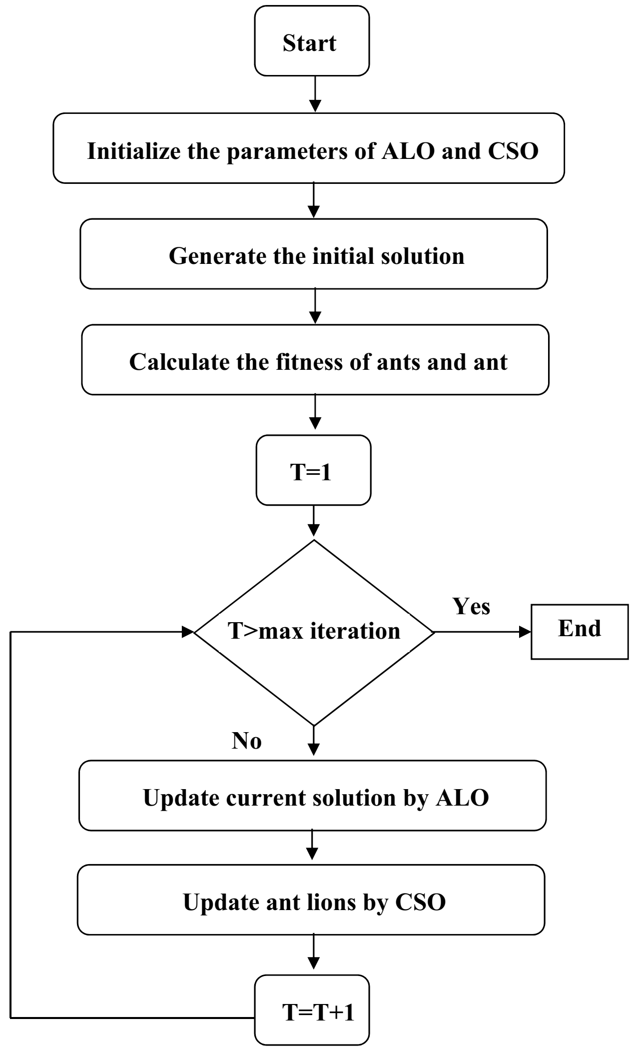


Fig. 3 Hybridization of ALO and CSO

hydrothermal scheduling [52, 53], and route planning [54]. Additionally, hybrid algorithms possess the merits of multiple individual algorithms, and can avoid their shortcomings. Therefore, the hybridization of ALO and CSO is developed in our paper. It is expected that the grading mechanism of CSO when embedded in ALO will increase the utilization rate of population. The refinement of the solutions obtained by ALO with CSO can further reduce the chances of getting stuck in local optima, thus leading to a faster convergence. This hybridization scheme of ALO and CSO is shown in Fig. 3.

Performance of on standard benchmark functions

The performance of ALO CSO is first tested on several standard benchmark functions including unimodal, multimodal as well as composite functions, as given in Table 2. The algorithm-specific parameters are the same as in [14, 39] (Table 3), and the general parameters are set as in [39]. Its performance is further compared with that of CSO, ALO, TLBO, CSO, and TLBO, as shown in Table 4. From Table 4, it is clear that ALO CSO is better than the standalone algorithms, such as CSO, TLBO, and ALO, for all the benchmark functions. ALO CSO performs equivalently to CSO TLBO for f1 and f2, and better than CSO TLBO for the other benchmark functions. Furthermore, Friedman rank test is performed in the simulations, and the results are shown in Fig. 4. It can be discovered that ALO CSO has obtained the best rank. The convergence curves of these algorithms for benchmark function f1, f4, f7, f9 are shown in Figs. 5, 6, 7 and 8, respectively. Particularly, in case of f1 and f4, the proposed hybrid algorithm converges faster than ALO, because the solutions obtained by ALO can be fine-tuned by CSO. Moreover, t test was conducted at a significance level of 0.005. For fair comparison all the algorithms are run 20 times. The goal of performing t test is to compare the average values of the two data sets and determine if they came from the same population. Figures 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 show the t test results for the objective functions in Table 2, from which we find out that there are differences in the mean values of objective functions of all the pairs. In the aforementioned figures, the x axis represents the corresponding algorithm and the y axis represents the t value. In addition, the positive t-value indicates that the mean value of the objective function of ALO CSO is much better than that of the other algorithms.

Table 2 Standard benchmark functions

Nature	Function	Range	Dim	$f_{\min}$
Unimodal	Sphere $f_1(x) = \sum_{i=1}^{n} x_i^2$	− 100, 100	10	0
	Schwefel 2.22 $f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	− 10, 10	10	0
	Rosenbrock $f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2$	− 30, 30	10	0
	Step $f_4(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	− 100, 100	10	0
Multimodal	Schwefel $f_5(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	− 500, 500	10	− 418.9829 × Dim
	Rastrigin $f_6(x) = \sum_{i=1}^{n} (x_i^2 - 10cos2\pi x_i + 10)$	− 5.12, 5.12	10	0
	Ackley $f_7(x) = -20\exp\left(0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)$	− 32, 32	10	0
	Griewank $f_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	− 600, 600	10	0
Composite	f_9 $f_1,f_2,\ldots f_9,f_{10}$ Sphere function $[O_1,O_2,\ldots O_9,O_{10}]=[1;1;1;\ldots;1]$ $[\lambda_1, \lambda_2,\ldots \lambda_9, \lambda_{10}]=[5/100, 5/100\ldots5/100]$	− 5, 5	4	0
	f_{10} $f_1,f_2,\ldots f_9,f_{10}$ Griewank's Function $[O_1,O_2,\ldots O_9,O_{10}]=[1;1;1;\ldots;1]$ $[\lambda_1, \lambda_2,\ldots \lambda_9, \lambda_{10}]=[5/100,5/100\ldots5/100]$	− 5, 5	2	0

Performances of ALO CSO on real-world optimization problems

Table 3 Algorithm-specific parameter settings

Algorithm	Parameter
ALO	$w=2$ when $t>0.1\ T$, $w=3$ when $t>0.5\ T$, $w=4$ when $t>0.75\ T$, $w=5$ when $t>0.9\ T$, and $w=6$ when $t>0.95\ T$ t is current iteration, and T is maximum iteration
CSO	RN = 0.2PN, HN = 0.5PN, CN = PN-RN-HN, MN = 0.3PN, G = 5
ALO CSO	RN = 0.2PN, HN = 0.5PN, CN = PN-RN-HN, MN = 0.3PN, G = 5 $w=2$ when $t>0.1\ T$, $w=3$ when $t>0.5\ T$, $w=4$ when $t>0.75\ T$, $w=5$ when $t>0.9\ T$, and $w=6$ when $t>0.95\ T$
CSO TLBO	RN = 0.3PN, HN = 0.4PN, CN = PN-RN-HN, MN = 0.3PN, G = 3, INV = 5

In this section, ALO CSO is validated on complex optimization problems, such as economic load dispatch [44–46] and speed reducer design [22]. The economic load dispatch problem is attacked for 38 generator test system [44] by ALO CSO. The general parameter settings are the same as in [35]. The performance of ALO CSO algorithm in dealing with the economic load dispatch problem is compared with that of the other algorithms like RCCRO, CSO TLBO, and DE. The results of RCCRO, DE, CSO TLBO, and TLBO are taken from [35]. The mean fitness values over 50 independent trials obtained by these algorithms are presented in Table 5, from which the superiority of ALO CSO over TLBO, RRCRO, CSO TLBO, and DE in this case study is clearly demonstrated.

The proposed algorithm is also used for handling the speed reduced design problem, and its performance is compared with that of CSO, BFA, ABC, and CSO TLBO. The setting of general and algorithm-specific parameter are the same as in [22]. Table 6 illustrates the superior performance of ALO CSO as compared to the aforementioned benchmark algorithms in this problem. It should be noted that both the economic load dispatch and speed reducer design are high dimensional problems. From the results in Tables 5 and 6, it can be concluded that our ALO CSO performs comparatively well on these two demanding testbeds.

Table 4 Performances of ALO CSO on standard benchmark functions

Function	ALO		CSO		ALOCSO		TLBO		CSOTLBO	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
f_1	1.26e−10	2.59e−10	0	5.4321e−8	0	0	0	1.978e−45	0	0
f_2	1.017e−06	1.84241e−06	0	0	0	0	1.2118e−45	1.9018e−45	0	0
f_3	0.032765	0.346772393	0.01257644	1.438652	3.9577e−05	4.6939e−05	2.11653	4.32152	0.02169	0.254349
f_4	1.6743e−10	2.56183e−10	2.8356e−06	1.4321e−05	1.5543e−10	2.8721e−10	0.0305	0.2484	9.9456e−07	2.3586e−06
f_5	− 2551.3857	− 1606.27643	− 2578.8763	− 1674.0982	− 4189.8289	− 2285.4321	− 2569.7	− 2983.5	− 4188.5	− 4189.5
f_6	6.6543e−06	7.71411e−06	0	1.1258e−12	0	1.4211e−14	0	1.8754e−10	0	2.3412e−15
f_7	2.2167e−15	3.73035e−15	8.4328e−16	8.8818e−16	4.5438e−16	6.76548e−16	4.4409e−15	4.4409e−15	8.6719e−16	8.8818e−16
f_8	0.01632441	0.018604494	0	3.9042e−12	0	5.8382e−15	0	1.0986e−12	0	1.0125e−15
f_9	1.1281	1.2531	0.8431	0.9980	0.8271	0.9980	0.9980	0.9980	0.9980	0.9980
f_{10}	0.020365	14.56498	3.9615e−04	6.2329e−04	2.12143e−4	0.00031192	3.0749e−04	0.000653	3.0749e−04	0.000753

Performance of ALO CSO on charger placement problem

The performance of the proposed ALO CSO is validated in attacking the complex optimal design problem of placing chargers. These chargers increase the net load demand of the power grid [1]. Thus, the charger placement must take into account in the security of the power grid and for convenience of the EV drivers. Several formulations of charger placement are reported in the existing literature [10]. In this work, the ALO CSO algorithm is validated on the single-objective formulation of the charger placement problem in [3] with the only objective function as the cost.

The position and size of charging stations are considered as the decision variables. Symbolically, the decision variables are given as follows.

- *nb*, Superimposed nodes of the road and distribution network, where charging stations are placed
- N_{fastnb}, Number of fast charging stations placed at *nb*
- N_{slownb}, Number of slow charging stations placed at *nb*

The objective function under consideration is the minimization of the cost. Mathematically, the objective function is represented as in (1)

$$f = \min(C_i + C_o + C_t + C_p) \tag{1}$$

where C_i is the investment cost, C_o is the operating cost, C_t is the travel time cost, and C_p is the cost in terms of net penalty paid.

The mathematical representation of C_i, C_o, C_t and C_p are given by (2) to (10).

$$C_i = C_o = f(N_{\text{fastnb}}, N_{\text{slownb}}) \tag{2}$$

$$C_i = \sum N_{\text{fastnb}} \times C_{\text{fast}} + \sum N_{\text{slownb}} \times C_{\text{slow}} \tag{3}$$

$$C_o = \left(\sum N_{\text{fastnb}} \times P_{\text{fast}} + \sum N_{\text{slownb}} \times P_{\text{slow}}\right) \times C_{\text{electricity}} \tag{4}$$

where C_{fast} is the installation cost of fast charging station, C_{slow} is the installation cost of slow charging station, P_{fast} is the capacity of fast charging station, P_{slow} is the capacity of slow charging station, and $C_{\text{electricity}}$ is the per unit cost of electricity.

$$C_t = f(nb) \tag{5}$$

$$C_t = d_{CS} \times P_{CS} \tag{6}$$

Fig. 4 Friedman ranks of the algorithms

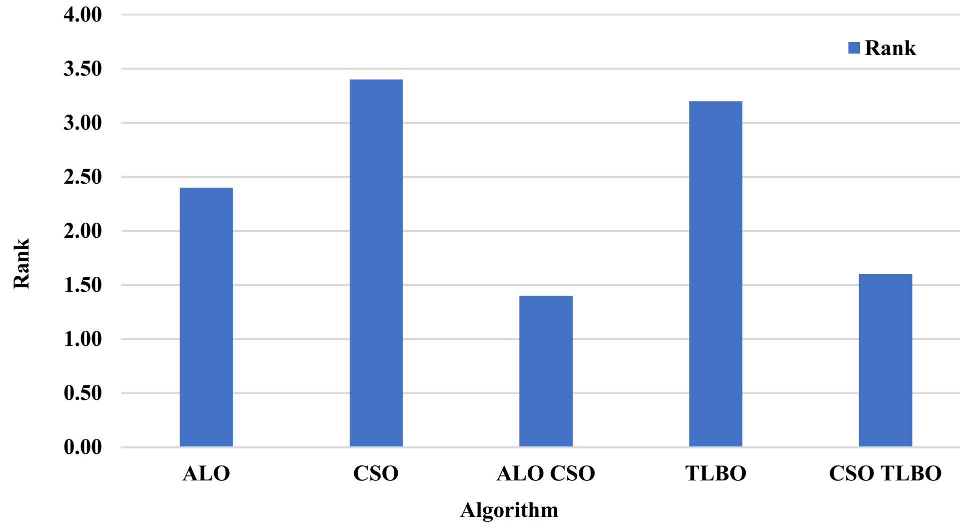
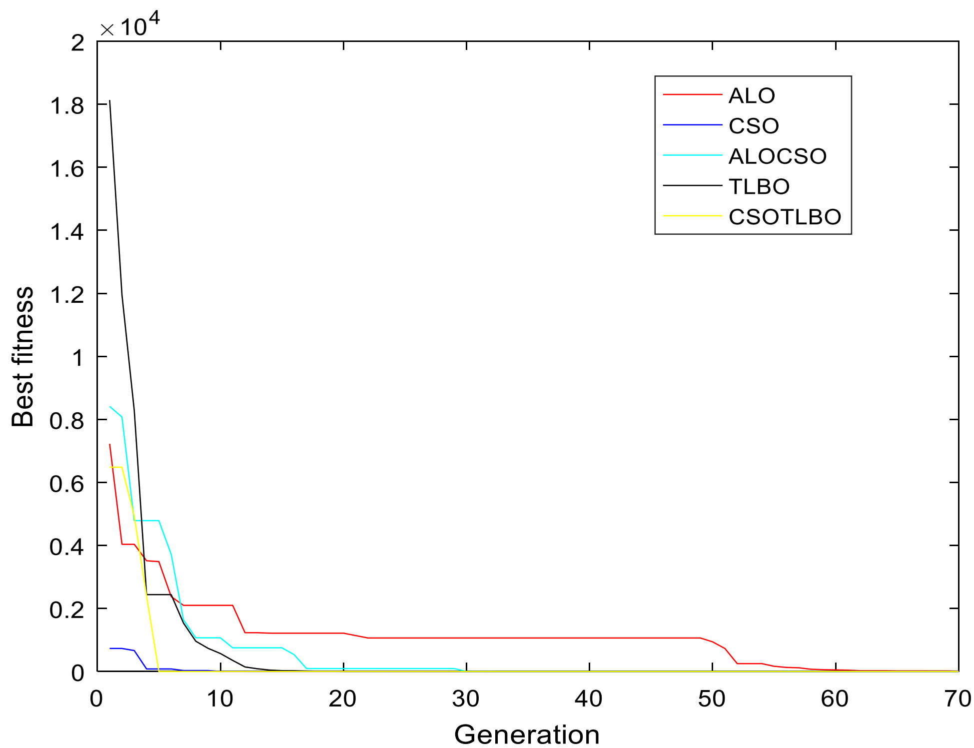


Fig. 5 Convergence curve for f_1



where d_{CS} is the distance between the charging station and the point of charging demand, and P_{CS} is the cost incurred by EV for travelling per km.

$$C_p = AENS_p + VD_p \tag{7}$$

$$AENS_p = C_{AENS} \times AENS_{CS} \tag{8}$$

$$VD_p = VD \times C_{VD} \tag{9}$$

$$VD = \left|V_{base} - V_{CS}\right| \tag{10}$$

where $AENS_p$ is the penalty paid for AENS, VD_p is the penalty paid for voltage deviation, C_{AENS} is the penalty for per unit of energy not served, $AENS_{CS}$ is the AENS after placement of charging station, VD is the voltage deviation, C_{VD} is the penalty paid for per unit of voltage deviation, V_{base} is the base value of bus voltage, and V_{CS} is the bus voltage after the placement of charging station.

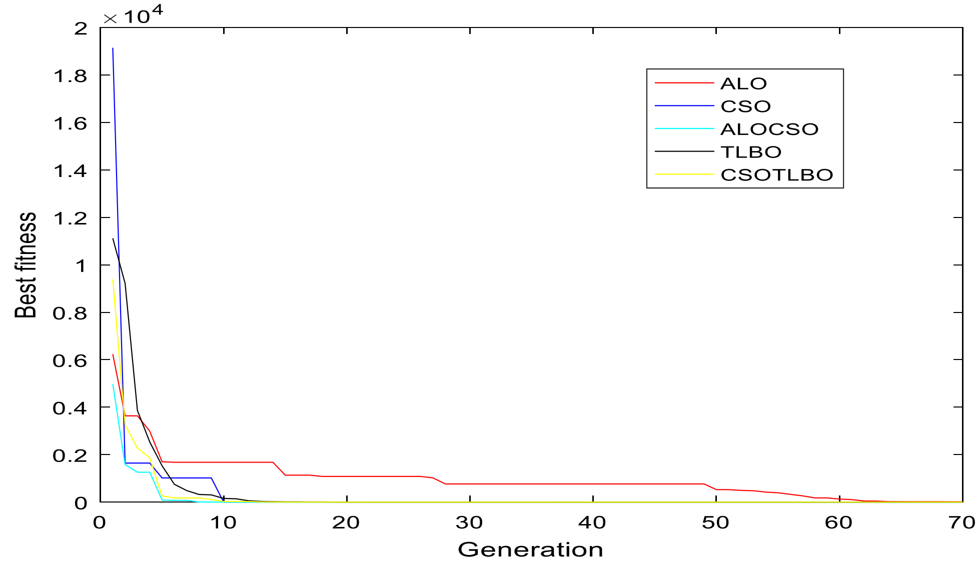


Fig. 6 Convergence curve for f_4

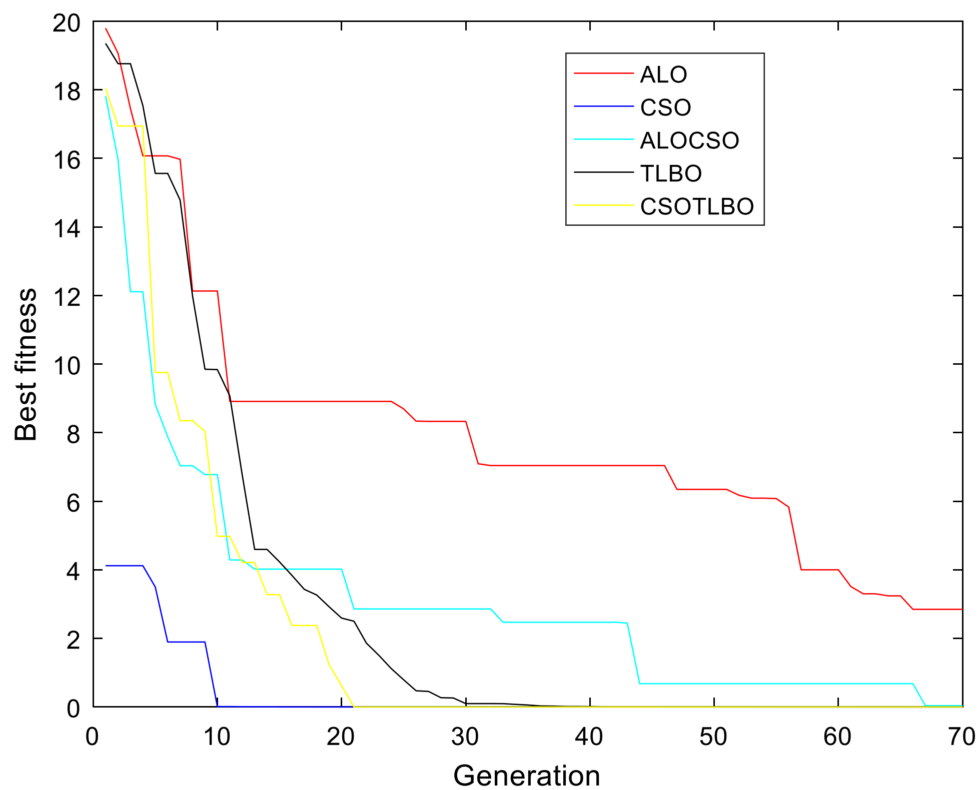


Fig. 7 Convergence curve for f_7

The charging station placement problem is always subject to a number of equality as well as inequality constraints. The constraints are given in (11) to (14).

$$0 < N_{fastnb} \leq n_{fastnb} \tag{11}$$

$$0 < N_{slownb} \leq n_{slownb} \tag{12}$$

$$L_{network} \leq L_{\max} \tag{13}$$

$$S_{\min} \leq S_i \leq S_{\max} \tag{14}$$

The above formulation of the charger placement problem is examined on standard superimposed 33 bus distribution and 25 node road network. The algorithm-specific and general parameter settings are the same as in [35] and given in

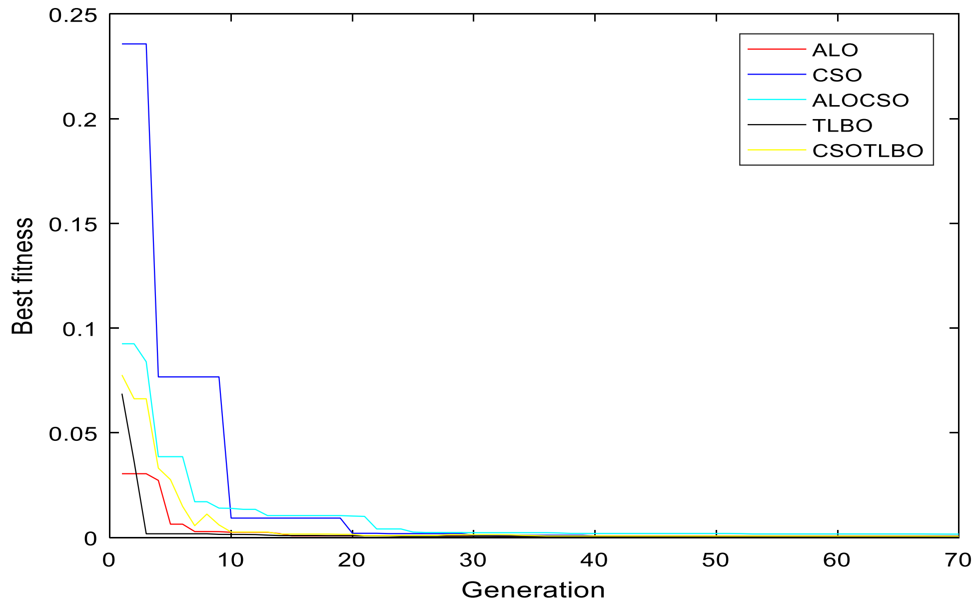


Fig. 8 Convergence curve for f_9

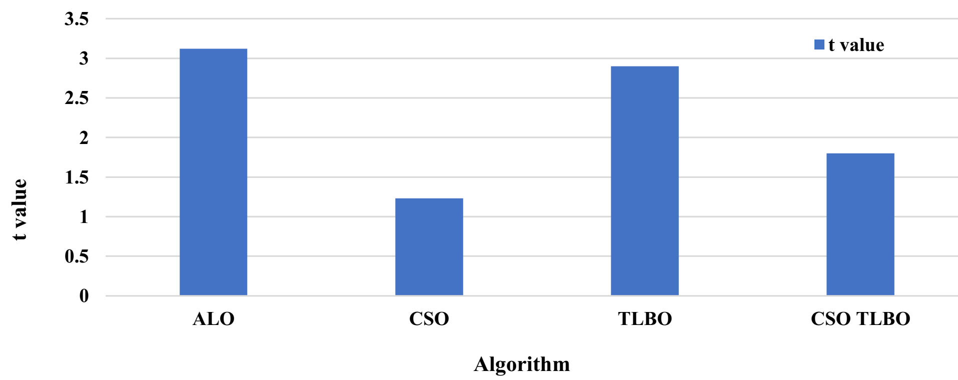


Fig. 9 T test result for f_1

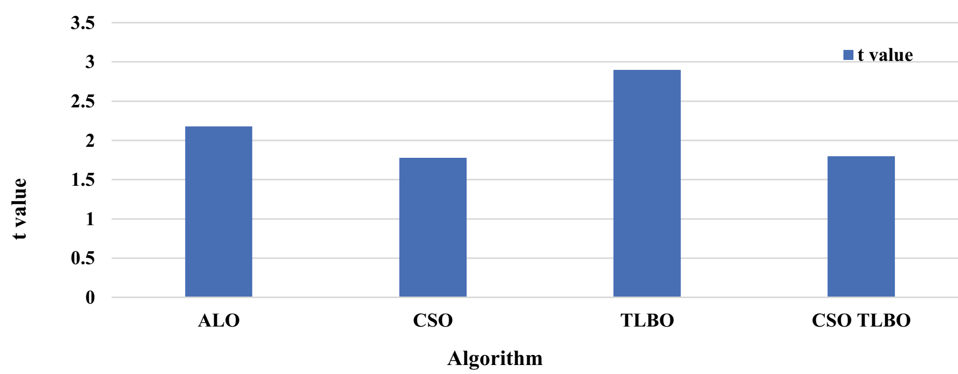


Fig. 10 T test result for f_2

Table 3. The performance of ALO CSO is compared with that of the other benchmark algorithms, e.g., CSO, TLBO, CSO TLBO, PSO, DE, and GA. Table 7 demonstrates the mean fitness values obtained by these algorithms, where the superior performance of ALO CSO is clearly visible. Moreover, the convergence curves of all the algorithms for the charger placement problem are shown in Fig. 19.

The impact of charger placement on different operating parameters of power system, such as power loss, SAIFI, and SAIDI are shown in Figs. 20, 21, and 22, respectively. It is observed that the operating parameters are within the prescribed limit. Furthermore, the impact of G that is an algorithm-specific parameter on the performance of ALO

Fig. 11 f test result for f_3

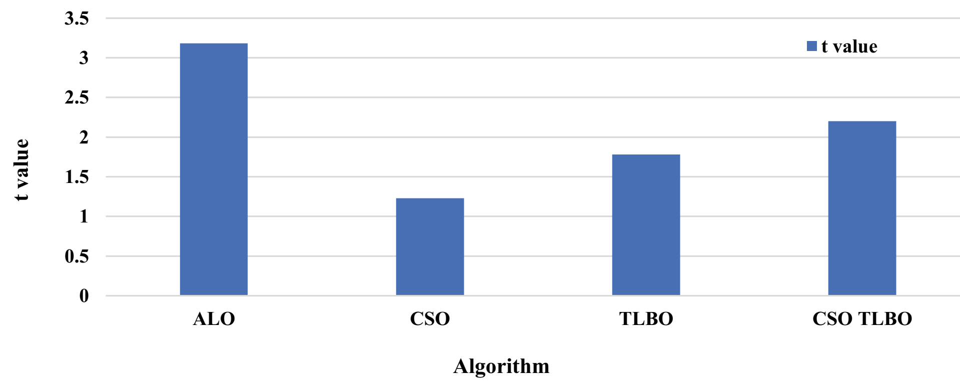


Fig. 12 f test result for f_4

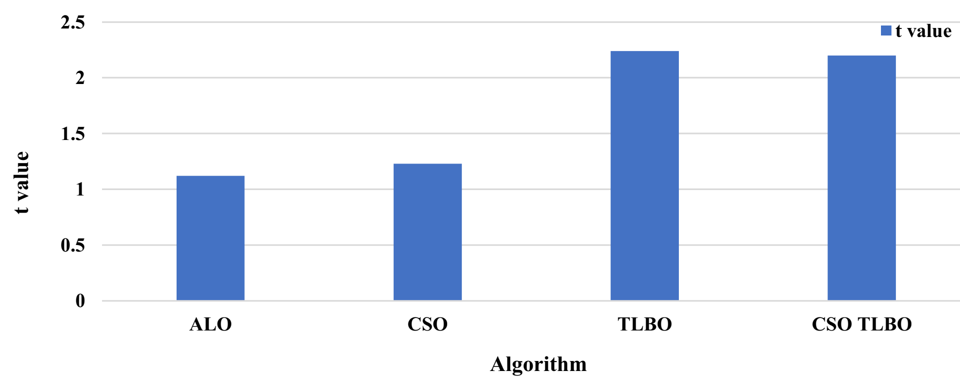


Fig. 13 f test result for f_5

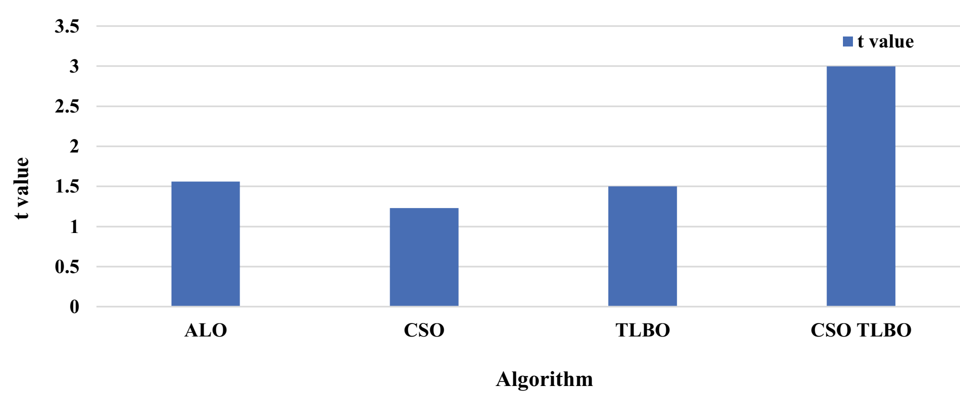


Fig. 14 f test result for f_6

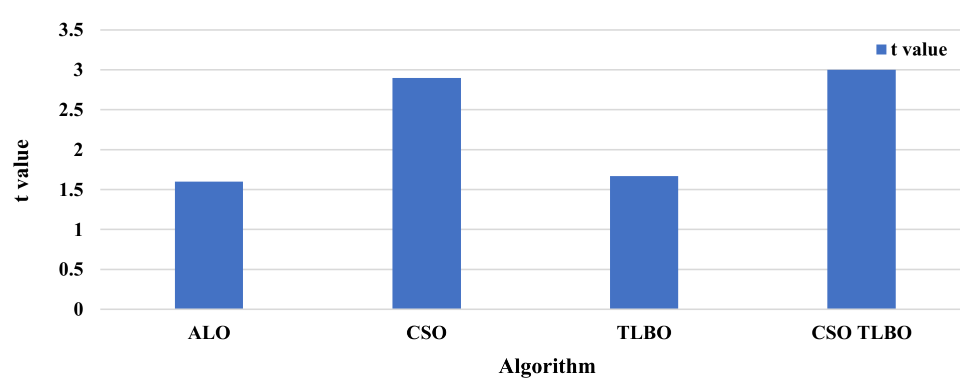


Fig. 15 f test result for f_7

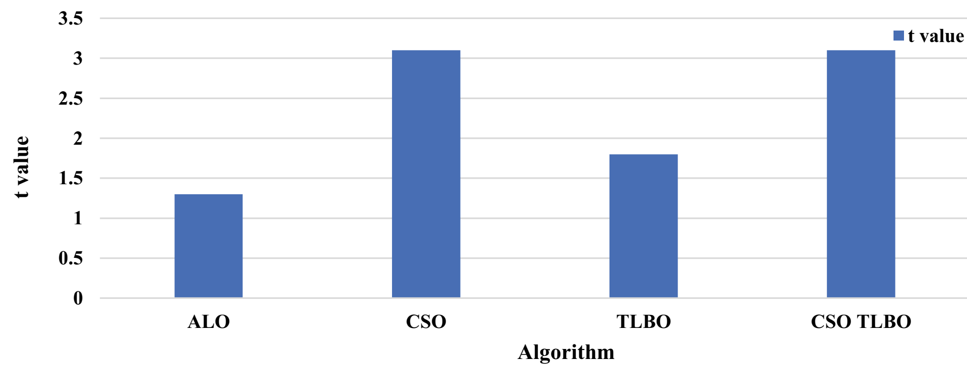


Fig. 16 f test result for f_8

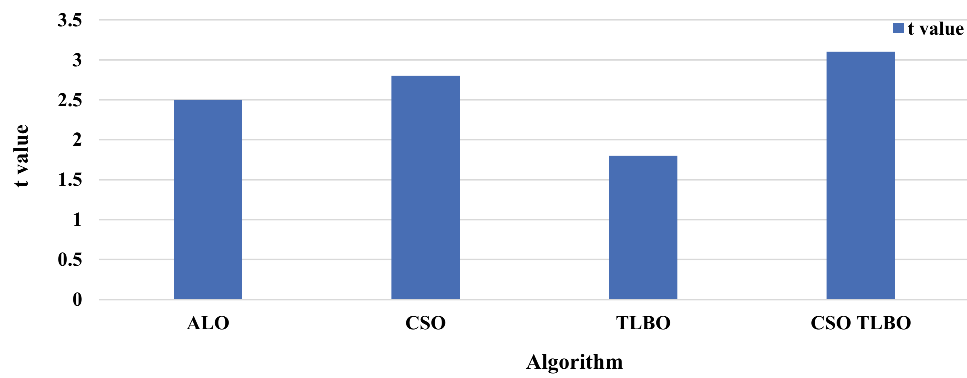


Fig. 17 f test result for f_9

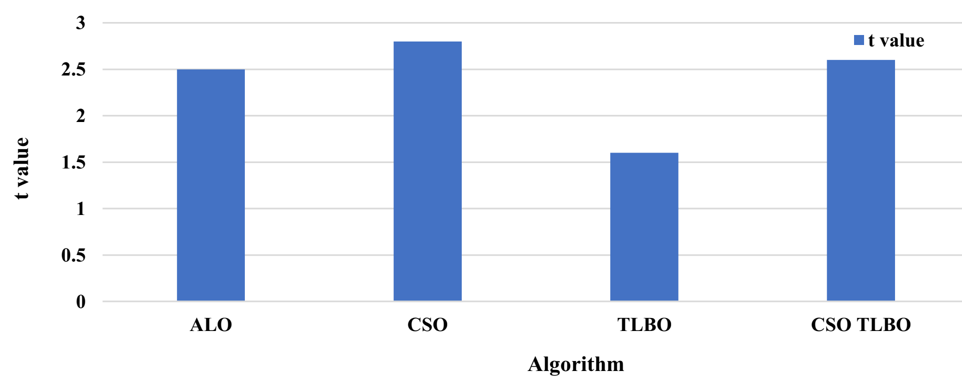


Fig. 18 f test result for f_{10}

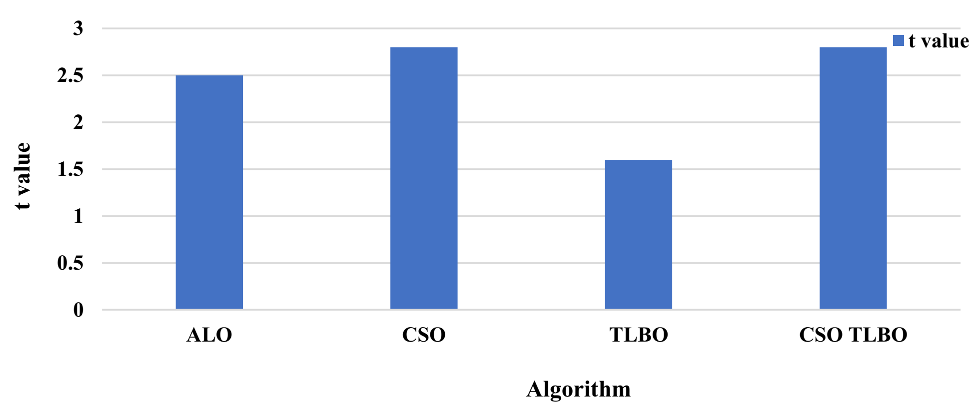


Table 5 Statistical comparison of ALO CSO with other algorithms in handling economic load dispatch problem

Algorithm	Mean fitness ($/hr)
TLBO	9,411,938.55723
RCCRO	9,412,404.277425
DE	9,417,237.290970
ICSOTLBO	9,411,938.54700
ALO CSO	9,411,927.24700
GA PSO	9,411,938.2687
ACO PSO	9,411,927.3467

Table 6 Statistical comparison of ALO CSO with other algorithms in handling speed reducer design problem

Algorithm	Mean Fitness
CSO	2997.764
BFA	3014.759
ABC	2997.05841
CSO TLBO	2997.0391
ALO CSO	2996.605
GA PSO	2997.0365
ACO PSO	2996.605

Table 7 Statistical comparison of ALO CSO with other algorithms in handling charger placement problem

Algorithm	Mean fitness
ALO CSO	1.4968
CSO TLBO	1.5241
CSO	1.5430
TLBO	1.5413
PSO	1.5413
DE	1.5497
GA	1.5584

CSO is tested, and the analysis results are given in Table 8, in which this algorithm performs the best when G equals 10.

The average execution time of all the algorithms involved is given in Table 9. These algorithms are tested using MATLAB 2016a software installed on a computer with the processor of Intel i7 CPU. From Table 9, we can discover that the execution time of ALO CSO is longer than that of the standalone algorithms, due to execution of the two algorithms used.

Conclusions

As we know that public acceptance of EVs needs the availability of charging infrastructure. This research work proposes a novel ALO CSO algorithm for dealing with the optimal charger placement problem. The developed ALO CSO is validated on the standard benchmark functions and complex real-world problems. Simulation results show and verify its competitive performances compared to the standalone algorithms. Moreover, in the ALO CSO, the chance of getting stuck in the local optima is effectively avoided by fine-tuning the solutions obtained by ALO with CSO. The new algorithm is also examined with the charger placement problem, in which it can outperform both the standalone and other benchmark algorithms. The ALO CSO is well capable of allocating the chargers without compromising with the safety and security of the power system. Our future work will focus on the further enhancement of this new algorithm, such as,

- Development of an adaptive ALO CSO,
- Hybridization of CSO with other metaheuristics techniques,
- Use it to cope with other practical problems, e.g., route planning, optimal load flow, and unit commitment.

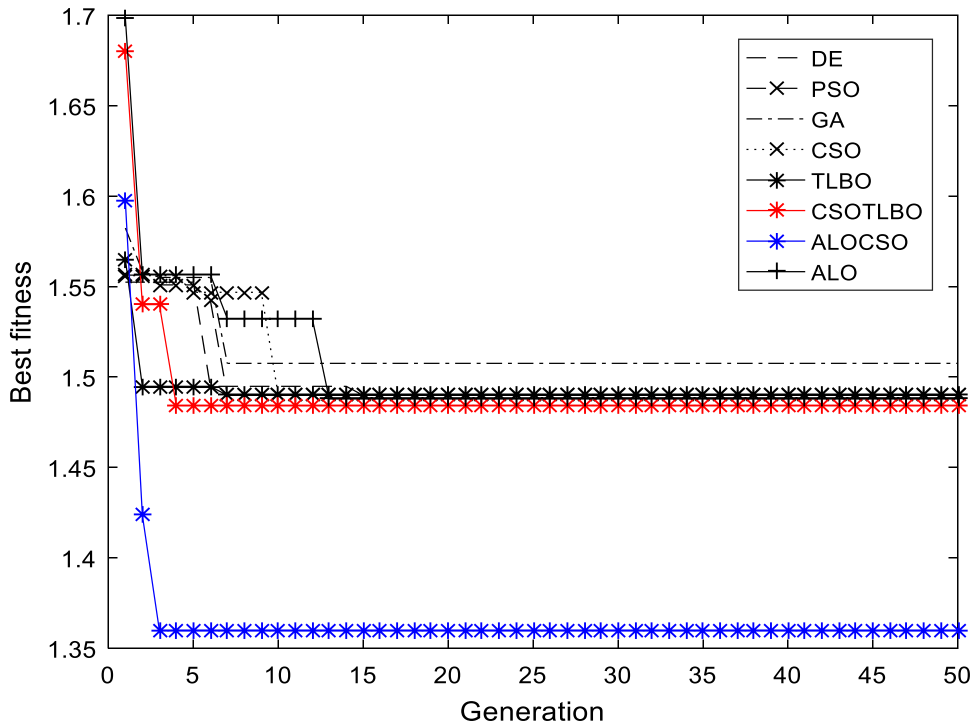


Fig. 19 Convergence curve for charger placement problem

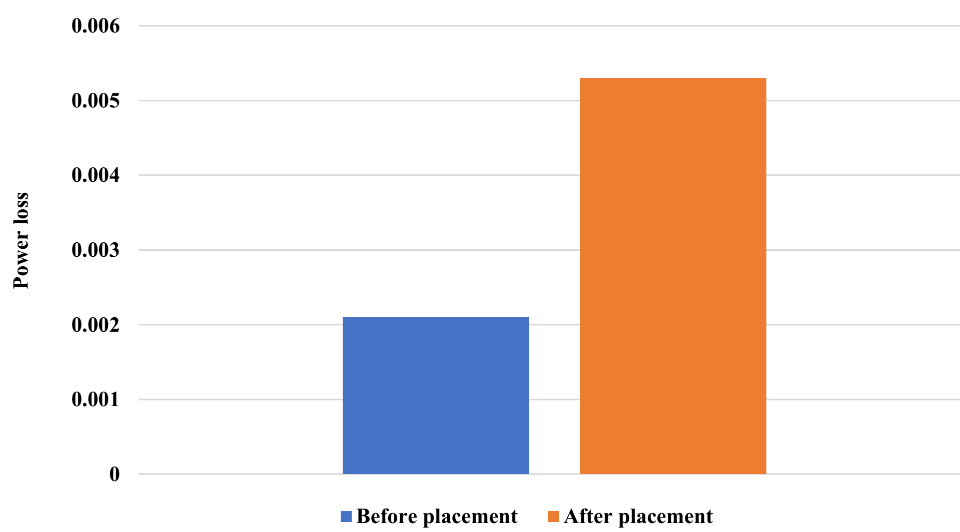


Fig. 20 Impact of charger placement on power loss in p.u

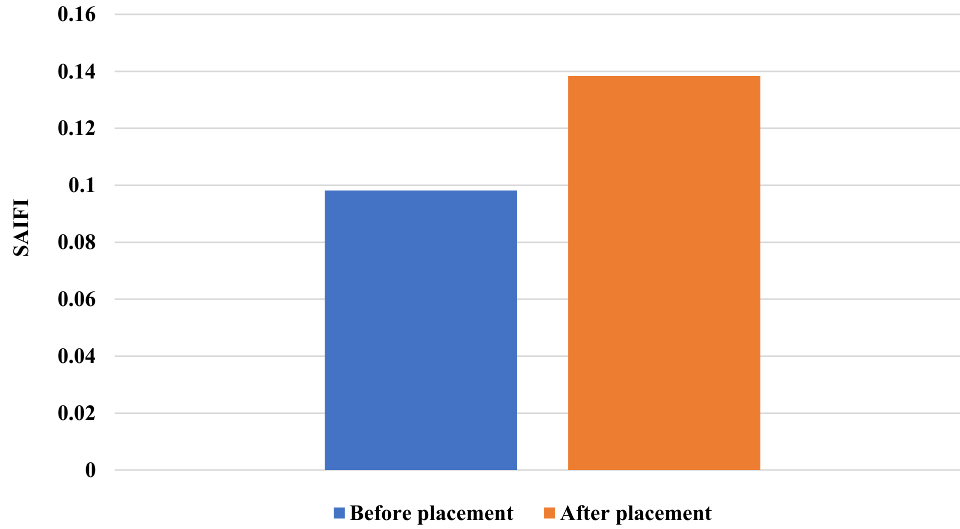


Fig. 21 Impact of charger placement on SAIFI

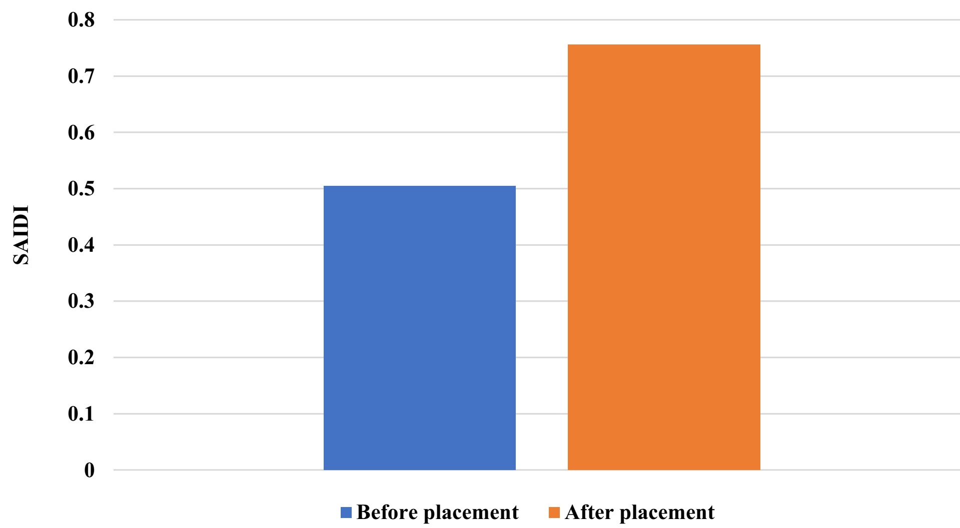


Fig. 22 Impact of charger placement on SAIDI

Table 8 Impact of G on the performance of ALO CSO

G	Mean fitness
5	1.5016
10	1.4968
15	1.5413

Table 9 Comparison of computational time of ALO CSO with other algorithms in handling charger placement problem

Algorithm	Average execution time (s)
ALO CSO	17.5
ALO	10.6
CSO	7.87
TLBO	25.56
CSO TLBO	18.63
DE	10.99
PSO	13.8
GA	30.9

Declarations

Conflict of interest We have no conflict of interest with this research article.

Human and animal rights We use no animal in this research.

References

1. Deb S, Tammi K, Kalita K, Mahanta P (2018) Impact of electric vehicle charging station load on distribution network. Energies 11(1):178
2. González LG, Siavichay E, Espinoza JL (2019) Impact of EV fast charging stations on the power distribution network of a Latin American intermediate city. Renew Sustain Energy Rev 107:309–318
3. Deb S, Kalita K, Mahanta P (2017) Review of impact of electric vehicle charging station on the power grid. In: 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy). IEEE, pp 1–6
4. Deb, S., Kalita, K., & Mahanta, P. (2017, December). Impact of electric vehicle charging stations on reliability of distribution network. In *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)* (pp. 1–6). IEEE.
5. Deb S, Kalita K, Mahanta P (2019) Distribution network planning considering the impact of electric vehicle charging station load. In: Smart Power Distribution Systems. Academic Press, pp 529–553
6. Jiang Z, Tian H, Beshir MJ, Sibagatullin R, Mazloomzadeh A (2016) Statistical analysis of Electric Vehicles charging, station usage and impact on the grid. In: 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, pp 1–5
7. Alshareef SM, Morsi WG (2017) Impact of fast charging stations on the voltage flicker in the electric power distribution systems. In: 2017 IEEE Electrical Power and Energy Conference (EPEC). IEEE, pp 1–6
8. Nour M, Ramadan H, Ali A, Farkas C (2018) Impacts of plug-in electric vehicles charging on low voltage distribution network. In: 2018 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, pp 357–362
9. Mastny P, Moravek J, Vrana M, Wasserbauer V, Vojtek M (2017) Mathematical model of fast charging station with integrated accumulators impact on low voltage distribution network. In: 2017 18th International Scientific Conference on Electric Power Engineering (EPE). IEEE, pp 1–6
10. Deb S, Tammi K, Kalita K, Mahanta P (2018) Review of recent trends in charging infrastructure planning for electric vehicles. Wiley Interdisciplinary Reviews: Energy and Environment 7(6):e306
11. Deb S, Gao XZ, Tammi K, Kalita K, Mahanta P (2019) Nature-inspired optimization algorithms applied for solving charging station placement problem: overview and comparison. Arch ComputMethods Eng 1–16
12. Aljanad A, Mohamed A, Shareef H, Khatib T (2018) A novel method for optimal placement of vehicle-to-grid charging stations in distribution power system using a quantum binary lightning search algorithm. Sustain Cities Soc 38:174–183
13. Awasthi A, Venkitusamy K, Padmanaban S, Selvamuthukumaran R, Blaabjerg F, Singh AK (2017) Optimal planning of electric vehicle charging station at the distribution system using hybrid optimization algorithm. Energy 133:70–78
14. Deb S, Kalita K, Gao XZ, Tammi K, Mahanta P (2017) Optimal placement of charging stations using CSO-TLBO algorithm. In: 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE, pp 84–89
15. Deb S, Tammi K, Gao XZ, Kalita K, Mahanta P (2020) A hybrid multi-objective chicken swarm optimization and teaching learning based algorithm for charging station placement problem. IEEE Access
16. Deb S, Tammi K, Kalita K, Mahanta P (2019) Charging station placement for electric vehicles: a case study of Guwahati city, India. IEEE Access 7:100270–100282
17. Zhang Y, Zhang Q, Farnoosh A, Chen S, Li Y (2019) GIS-based multi-objective particle swarm optimization of charging stations for electric vehicles. Energy 169:844–853
18. Zeb MZ, Imran K, Khattak A, Janjua AK, Pal A, Nadeem M, Khan S (2020) Optimal placement of electric vehicle charging stations in the active distribution network. IEEE Access 8:68124–68134
19. Mohanty AK, Babu PS Optimal placement of electric vehicle charging stations using JAYA algorithm. In: Recent Advances in Power Systems. Springer, Singapore, pp 259–266
20. Reddy MSK, Selvajyothi K (2020) Optimal placement of electric vehicle charging station for unbalanced radial distribution systems. Energy Sour Part A Recovery Util Environ Effects 1–15
21. Amini MH, Moghaddam MP, Karabasoglu O (2017) Simultaneous allocation of electric vehicles' parking lots and distributed renewable resources in smart power distribution networks. Sustain Cities Soc 28:332–342
22. Meng X, Liu Y, Gao XZ, Zhang H (2014) A new bio-inspired algorithm: chicken swarm optimization. In: International conference in swarm intelligence. Springer, Cham, pp 86–94
23. Deb S, Gao XZ, Tammi K, Kalita K, Mahanta P (2019) Recent studies on chicken swarm optimization algorithm: a review (2014–2018). Artif Intell Rev 1–29
24. Wu D, Xu S, Kong F (2016) Convergence analysis and improvement of the chicken swarm optimization algorithm. IEEE Access 4:9400–9412
25. Qu C, Zhao SA, Fu Y, He W (2017) Chicken swarm optimization based on elite opposition-based learning. Math Probl Eng
26. Wang K, Li Z, Cheng H, Zhang K (2017) Mutation chicken swarm optimization based on nonlinear inertia weight. In: Computer and Communications (ICCC), 2017 3rd IEEE International Conference on. IEEE, pp 2206–2211
27. Ahmed K, Hassanien AE, Bhattacharyya S (2017) A novel chaotic chicken swarm optimization algorithm for feature selection. In: Research in Computational Intelligence and Communication Networks (ICRCICN), 2017 Third International Conference on. IEEE, pp 259–264
28. Liang S, Feng T, Sun G, Zhang J, Zhang H (2016) Transmission power optimization for reducing sidelobe via bat-chicken swarm optimization in distributed collaborative beamforming. In:

Computer and Communications (ICCC), 2016 2nd IEEE International Conference on. IEEE, pp 2164–2168

29. Fu W, Wang B, Li X, Liu L, Wang Y (2019) Ascent trajectory optimization for hypersonic vehicle based on improved chicken swarm optimization. IEEE Access 7:151836–151850
30. Osamy W, El-Sawy AA, Salim A (2020) CSOCA: chicken swarm optimization based clustering algorithm for wireless sensor networks. IEEE Access 8:60676–60688
31. Liang X, Kou D, Wen L (2020) An improved chicken swarm optimization algorithm and its application in robot path planning. IEEE Access 8:49543–49550
32. Niazy N, El-Sawy A, Gadallah M (2020) A hybrid chicken swarm optimization with tabu search algorithm for solving capacitated vehicle routing problem. Int J Intell Eng Syst 13(4):237–247
33. Wang J, Zhang F, Liu H, Ding J, Gao C (2020) A novel interruptible load scheduling model based on the improved chicken swarm optimization algorithm. CSEE J Power Energy Syst
34. Qiuqiao Z, Wang B, Wei L, Haishan W (2020) Chicken swarm optimization algorithm based on quantum behavior and its convergence analysis. In: 2020 39th Chinese Control Conference (CCC). IEEE, pp 2107–2112
35. Deb S, Gao XZ, Tammi K, Kalita K, Mahanta P (2020) A new teaching–learning-based chicken swarm optimization algorithm. Soft Comput 24(7):5313–5331
36. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
37. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
38. Abualigah L, Shehab M, Alshinwan M, Mirjalili S, Abd Elaziz M (2020) Ant Lion Optimizer: A Comprehensive Survey of Its Variants and Applications. Arch. Comput. Methods Eng
39. Mani M, Bozorg-Haddad O, Chu X (2018) Ant lion optimizer (ALO) algorithm. In: Advanced Optimization by Nature-Inspired Algorithms. Springer, Singapore, pp 105–116
40. Zhai Z, Li S, Liu Y, Li Z (2015) Teaching-learning-based optimization with a fuzzy grouping learning strategy for global numerical optimization. J Intell Fuzzy Syst 29(6):2345–2356
41. Rao RV, Waghmare GG (2013) Solving composite test functions using teaching-learning-based optimization algorithm. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). Springer, Berlin, Heidelberg, pp 395–403
42. Satapathy SC, Naik A (2014) Modified teaching–learning-based optimization algorithm for global numerical optimization—a comparative study. Swarm Evol Comput 16:28–37
43. Ballester PJ, Stephenson J, Carter JN, Gallagher K (2005) Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In: The 2005 IEEE Congress on Evolutionary Computation. IEEE, pp 498–505
44. Bhattacharjee K, Bhattacharya A, Dey SHN (2014) Oppositional real coded chemical reaction optimization for different economic dispatch problems. Int J Electr Power Energy Syst 55:378–391
45. Bhattacharjee K, Bhattacharya A, Dey SHN (2014) Teaching-learning-based optimization for different economic dispatch problems. Scientia Iranica. Trans D Comput Sci Eng Electr 21(3):870
46. Bhattacharjee K, Bhattacharya A, Dey SHN (2014) Chemical reaction optimisation for different economic dispatch problems. IET Gener Transm Distrib 8(3):530–541
47. Dhiman G (2020) MOSHEPO: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems. Appl Intell 50(1):119–137
48. Kumar M, Dhillon JS (2018) Hybrid artificial algae algorithm for economic load dispatch. Appl Soft Comput 71:89–109
49. Al-Betar MA, Awadallah MA, Krishan MM (2019) A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer. Neural Comput Appl 1–28
50. Boqtob O, El Moussaoui H, El Markhi H, Lamhamdi T (2019) Optimal robust unit commitment of microgrid using hybrid particle swarm optimization with sine cosine acceleration coefficients. Int J Renew Energy Res 9(3):1125–1134
51. Yang Z, Li K, Guo Y, Feng S, Niu Q, Xue Y, Foley A (2019) A binary symmetric based hybrid meta-heuristic method for solving mixed integer unit commitment problem integrating with significant plug-in electric vehicles. Energy 170:889–905
52. Ghosh S, Kaur M, Bhullar S, Karar V (2019) Hybrid abc-bat for solving short-term hydrothermal scheduling problems. Energies 12(3):551
53. Chen G, Gao M, Zhang Z, Li S (2020) Hybridization of chaotic grey wolf optimizer and dragonfly algorithm for short-term hydrothermal scheduling. IEEE Access 8:142996–143020
54. Qu C, Gai W, Zhang J, Zhong M (2020) A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. Knowl-Based Syst 194:105530