Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 2, April 2025, pp. 1486~1496
ISSN: 2302-9285, DOI: 10.11591/eei.v14i2.8665

A stochastic optimization called split-compete optimization and its utilization on economic load dispatch problem

Purba Daru Kusuma, Faisal Candrasyah Hasibuan
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received May 6, 2024
Revised Oct 23, 2024
Accepted Nov 19, 2024

Keywords:

Constrained optimization
Economic load dispatch
High-dimensional optimization
Metaheuristic
Random motion
Split-compete optimization
Stochastic process

ABSTRACT

The economic load dispatch (ELD) problem is one crucial optimization problem in a power system. Metaheuristics has become a standard method to tackle this problem. Ironically, ELD is not popular enough to become a constrained use case in most studies introducing new metaheuristics, where four designs in mechanical engineering have become the most popular ones. This work introduced a new metaheuristic called split-compete optimization (SCO). It contains three serial steps where two directed searches are employed and competed against each other in each step. SCO is then assessed to tackle both unconstrained and constrained problems where 23 classic functions represent the unconstrained problems and two cases in ELD problems represent the constrained ones. Five new metaheuristics are chosen as competitors, including the coati optimization algorithm (COA), language education optimization (LEO), northern goshawk optimization (NGO), Kookaburra optimization algorithm (KOA), and walrus optimization algorithm (WaOA). In the first evaluation, SCO is superior enough by being better than COA, LEO, NGO, KOA, and WaOA in 14, 13, 20, 15, and 10 functions, whereas SCO is dominant in high-dimensional functions. Meanwhile, SCO is competitive in solving both cases of ELD problems.

Corresponding Author:

Purba Daru Kusuma
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: purbodaru@telkomuniversity.ac.id

1. INTRODUCTION

The economic load dispatch (ELD) problem is a widespread optimization problem in power systems. This system contains several generators that produce electric power inside their range. The produced power should meet the demand which is known in advance. On the other hand, there is a fixed and variable cost to operate these generators. The cost function among generators may be different as this system contains several types of generating units [1]. As an optimization problem, it mainly aims to minimize the operational or fuel cost [2]. On the other hand, Metaheuristics are often utilized as an optimization technique to tackle ELD problems. Many studies used various metaheuristics, such as particle swarm optimization (PSO) [3], artificial bee colony (ABC) [4], simulated annealing (SA) [5], social group optimization (SGO) [6], bat algorithm (BA) [7], and teaching learning-based optimization (TLBO) [8].

Plenty of metaheuristics have been introduced in recent years. Many of them were inspired by natural phenomena, especially animal behavior during hunting or searching for food and mating. These metaheuristics include coati optimization algorithm (COA) [9], walrus optimization algorithm (WaOA) [10], kookaburra optimization algorithm (KOA) [11], northern goshawk optimization (NGO) [12], golden jackal optimization (GJO) [13], clouded leopard optimization (CLO) [14], cheetah optimization (CO) [15], Komodo mlipir algorithm (KMA) [16], chameleon swarm algorithm (CSA) [17], squirrel search optimization (SSO)

[18], white shark optimization (WSA) [19], crayfish optimization algorithm (COA) [20], and red fox optimization (RFO) [21]. Several metaheuristics use human or social behavior as a metaphor, such as language education optimization (LEO) [22], deep sleep optimization (DSO) [23], preschool education optimization algorithm (PEOA) [24], and migration algorithm (MA) [25]. Meanwhile, several metaheuristics are free from metaphor, such as total interaction algorithm (TIA) [26], fully informed search algorithm (FISA) [27], golden search optimization (GSO) [28], attack leave optimization (ALO) [29], and three on three optimizers [30].

Despite plenty of studies on the ELD problem and metaheuristics, several unresolved considerations and issues remain. First, many recent metaheuristics use metaphors and promote this imitation as a novelty rather than going directly to its novel mechanism. Second, most of the studies introducing new metaheuristics focused on assessing its performance to tackle various unconstrained problems, such as 23 classic functions, such as in KMA [16] or standard constrained functions like in CEC series like in COA [9] or KOA [11]. Third, four design optimization problems (speed reducer, pressure vessel, spring, and welded beam) in mechanical engineering have become widespread constrained problems in many studies, introducing new metaheuristics. In contrast, many other constrained issues, such as the ELD problem, are far less popular and rare in these studies. Fourth, many studies on the ELD problem still used old metaheuristics as their proposed technique and far older metaheuristics as their competitors. Only a few of these studies used new metaheuristics, such as GJO [31], and dandelion optimization (DO) [32]. The fifth, mere investigation, is commonly found in many studies introducing new metaheuristics by promoting their superiority against their competitors but lacking the drawback to their concept, techniques, and the deep exploration of the nature of their use cases.

This work aims to introduce a new stochastic optimization called split-compete optimization (SCO). SCO is built based on a swarm intelligence baseline, consisting of autonomous entities actively searching for optimal solutions in every iteration without central coordination. SCO is also a metaphor-free metaheuristic, so any natural or social behavior does not inspire its development. Then, SCO is employed to tackle both unconstrained and constrained optimization problems.

The scientific contributions of this work can be categorized as follows:

- A new metaphor-free swarm-based metaheuristic called SCO is introduced, employing multiple searches performed in serial steps.
- A new concept of competition among searches is introduced, compared to single search, iteration-controlled search, and population split-based search.
- The performance of SCO is assessed by challenging it to tackle 23 classic functions that cover a broad scope of problems, including unimodal and multimodal functions.
- The performance of SCO is also assessed by employing it to tackle two cases in ELD problems representing constrained optimization problems.

The remainder of this paper is provided as follows. The research method of this work is provided in section 2, which contains the model of the proposed SCO and classic ELD problem. Section 3 provides the simulation scenario performed to investigate the performance of SCO, the result, and the in-depth discussion due to the result, findings, and limitations. In the end, the presentation of this study is summarized in section 4 as a conclusion and path for future studies.

2. METHOD

2.1. Proposed metaheuristic model

The proposed SCO framework comprises three sequential steps, each composed of two directed searches. In each step, the initial search is directed towards the finest member. Subsequently, this search is juxtaposed with various other-directed searches as the second search. In the first step (Figure 1(a)), the second search entails directing towards the average of all finer members combined with the finest member. In the second step (Figure 1(b)), the second search involves directing relative to a randomly selected entity inside the swarm. Lastly, the second search is produced relative to a randomized entity inside the space in the third step (Figure 1(c)). Two seeds are generated in each step, competing against each other. The superior seed is chosen as the candidate for replacing the current entity (Figure 1(d)). If this candidate outperforms the current entity, its value is replaced. Additionally, the value of the finest entity is updated after each step. These four motions are illustrated in Figure 1.

This multiple-search approach is adopted based on the premise that no superior technique exists, as stated in the NFL theorem [11]. Moreover, the multiple search approach is common in recent metaheuristics. These multiple searches are employed through serial steps, role split, or both. Meanwhile, these four searches are chosen among many other searches based on several reasons. The directed search toward the finest entity has been widely employed in many metaheuristics, such as PSO [33], COA [9], zebra optimization algorithm

(ZOA) [34], and KMA [16]. This search has been proven to tackle unimodal problems quickly. This is also why this search becomes the backbone of SCO, as it is employed in all steps. The directed search toward the average of all finer entities is a derivative of the motion toward a set of better entities as used in grey wolf optimization (GWO) [35], and KMA [16]. It is designed to become an alternative to exploitation rather than relying only on the finest entity. The search relative to a randomly selected entity inside the swarm is intended for exploration based on the interaction among the swarm members. Ultimately, the search relative to a random entity inside space is designed to expand exploration, especially to trace a broader area inside space, as the swarm tends to converge as iteration increases.

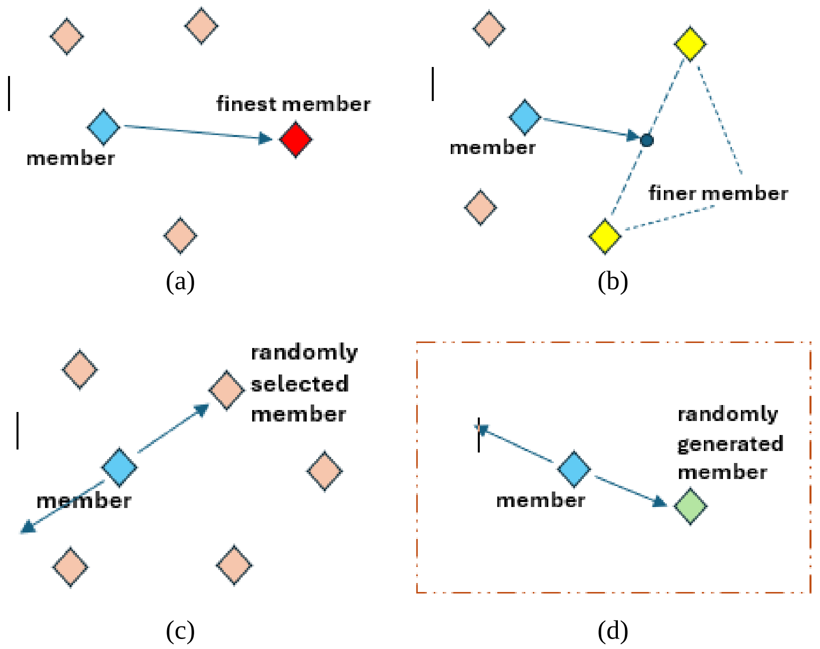

Figure 1. Illustration of search: (a) first search, (b) second search, (c) third search, and (d) fourth search

The algorithm of SCO is provided in Algorithm 1 in pseudocode. The flowchart of SCO is provided in Figure 2. As a metaheuristic, SCO contains two stages: initialization and iteration. The initialization is given in lines 2 to 4. The iteration is provided from lines 5 to 11. The annotations used in this paper are provided in Table 1. The mathematical formulation to support the algorithm is given from (1) to (11).

Algorithm 1. Split-compete optimization

```
1     start
2       for all X
3         initialize x_i, then update x_finest
4       end
5       for t=1 to t_max
6         for all X
7           the first step is performing and competing the 1st and 2nd searches, update x_i,
      then update x_finest
8           the second step is performing and by competing the 1st and 3rd searches, update
      x_i, then update x_finest
9           the third step by performing and competing the 1st and 4th searches, update x_i,
      then update x_finest
10        end
11      end
12    end
13    return x_finest
```

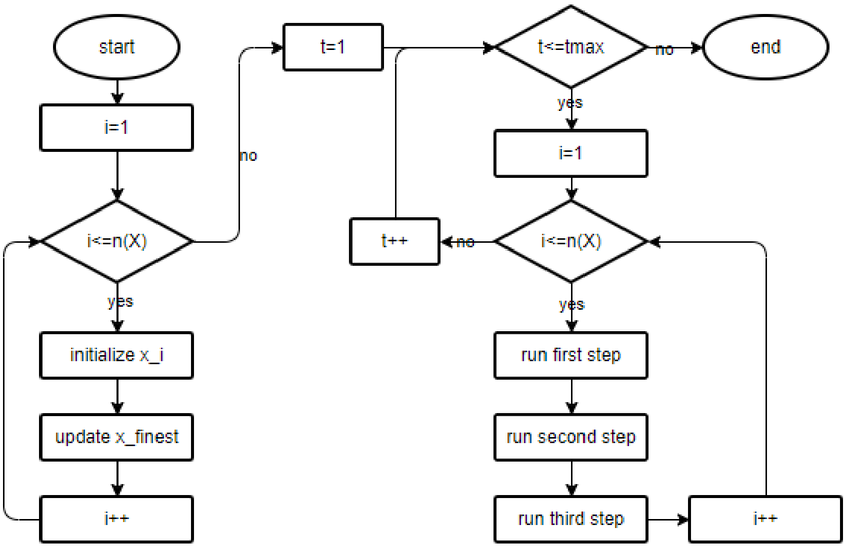

Figure 2. Flowchart of SCO

Table 1. Annotations

Variable	Description
lb, *ub*	Lower and upper boundaries of space
o	Objective function
s_1, s_2, s_3, s_4	The first to fourth seeds
s_{f1}, s_{f2}, s_{f3}	The first to third final seeds
t	Iteration
t_{max}	Maximum iteration
U_f, U_d, U_p	Floating point, integer, and population uniform random
x	Entity
X	Set of entities (swarm)
x_{finer}	Finer entity
X_{finer}	Set of finer entities
x_{finest}	The finest entity
x_{sel1}, x_{sel2}	The first and second selected entity

Two processes are conducted in the initialization stage. The first process generates an initial solution uniformly inside the space, as formalized in (1). The second process is updating the finest entity as formalized in (2). Each entity inside the swarm performs these two serial processes.

$$x_{i,j} = U_f\left(lb_j, ub_j\right) \tag{1}$$

$$x'_{finest} = \begin{cases} x_i, o(x_i) < o(x_{finest}) \\ x_{finest}, otherwise \end{cases} \tag{2}$$

Each entity performs three serial steps inside every iteration. Each step contains four processes: generating two seeds, competing these two seeds to find the final seed, updating the entity based on comparing the final seed and the current value, and updating the finest entity. In general, there are four distinct searches during the iteration. Each search generates a seed. The first search is provided using (3). The second search is given in (4). The third search is shown in (5) and (6). The fourth search is provided in (7) and (8).

$$s_{1,i,j} = x_{i,j} + U_f(0,1)(x_{finest,j} - U_d(1,2)x_{i,j}) \tag{3}$$

$$s_{2,i,j} = x_{i,j} + U_f(0,1)\left(\frac{\sum_{k=1}^{n(X_{finer,i})} x_{finer,i,k,j} + x_{finest,j}}{n(X_{finer,i})+1} - U_d(1,2)x_{i,j}\right) \tag{4}$$

$$x_{sel1} = U_p(X) \tag{5}$$

$$s_{3,i,j} = \begin{cases} x_{i,j} + U_f(0,1)\left(x_{sel1,j} - U_d(1,2)x_{i,j}\right), o(x_{sel1}) < o(x_i) \\ x_{i,j} + U_f(0,1)\left(x_{i,j} - U_d(1,2)x_{sel1,j}\right), otherwise \end{cases} \quad (6)$$

$$x_{sel2,j} = U_f(lb_j, ub_j) \quad (7)$$

$$s_{4,i,j} = \begin{cases} x_{i,j} + U_f(0,1)\left(x_{sel2,j} - U_d(1,2)x_{i,j}\right), o(x_{sel2}) < o(x_i) \\ x_{i,j} + U_f(0,1)\left(x_{i,j} - U_d(1,2)x_{sel2,j}\right), otherwise \end{cases} \quad (8)$$

There are three final seeds produced by every entity in every iteration. These final seeds are produced serially due to competing two searches in each step. The first, second, and third seeds are formalized using (9) to (11) respectively. Then, the value updates if the entity is based on the final seed and formalized using (12).

$$s_{f1,i} = \begin{cases} s_{1,i}, o\left(s_{1,i}\right) < o(s_{2,i}) \\ s_{2,i}, otherwise \end{cases} \quad (9)$$

$$s_{f2,i} = \begin{cases} s_{1,i}, o\left(s_{1,i}\right) < o(s_{3,i}) \\ s_{3,i}, otherwise \end{cases} \quad (10)$$

$$s_{f3,i} = \begin{cases} s_{1,i}, o\left(s_{1,i}\right) < o(s_{4,i}) \\ s_{4,i}, otherwise \end{cases} \quad (11)$$

$$x_i' = \begin{cases} s_{f,i}, o\left(s_{f,i}\right) < o(x_i) \\ x_i, otherwise \end{cases} \quad (12)$$

The computational complexity of SCO can be investigated by drawing back to the algorithm presentation. The computational complexity of any algorithm is studied by observing the number of loops employed in the related algorithm. In SCO, as in any metaheuristics, complexity differences exist between the initialization and iteration stages. In the initialization stage, the complexity of SCO is provided as $O(n(X)d)$. Meanwhile, the complexity during the iteration stage is provided as $O(n(X)^2.d.t_{max})$. There are two loops in the initialization stage: the outer loop is the loop for entire swarm entities, and the inner loop is the whole dimension. On the other hand, there are three loops during the iteration stage. The outer loop is the loop from the first iteration to the maximum iteration. The intermediate loop is the loop for entire swarm entities. The inner loop is the loop for the whole dimension. Meanwhile, the loop for entire swarm entities is employed twice as a nested loop, as the second one is used to find the finer entities during the first step.

2.2. Economic load dispatch problem

In ELD, the power system is characterized by a collection of generators or generating units. These units exhibit either homogeneity or diversity depending on their power sources. While some generators rely on non-renewable energy sources like coal and fuel, others utilize renewable sources such as ocean waves, water, solar, and wind. Considering this diversity, the fundamental model of these generating units is outlined in (13). For a comprehensive mathematical representation of the ELD problem, please refer to the study by [36].

$$G = \{g_1, g_2, g_3, \dots, g_n\} \quad (13)$$

This power system operates by producing electricity or power to meet the power demand. Each generating unit produces power inside the range of its minimum and maximum power, representing the search space provided in (14). Then, the power output from all generating units will be accumulated as total power output, as explained in (15). This total power output should meet the power demand, which can be seen as an equality constraint as provided in (16). The power is usually given in watts.

$$p_{min,i} \le p_i \le p_{max,i} \quad (14)$$

$$p_{total} = \sum_{i=1}^{n(G)} p_i \quad (15)$$

$$p_{demand} = p_{total} \quad (16)$$

As an optimization problem, ELD has an objective or soft constraint. In its basic form, its objective is to minimize the total operating cost, as provided in (17). This total operating cost is obtained by accumulating the operating cost of all active generating units as formalized in (18). Meanwhile, the introductory presentation of this operating cost can be provided in (19) as quadratic functions of output power with three constants. In (19), α represents the fixed cost while β and γ represent the constants for the variable cost. In general, the operating cost is provided in USD/hour.

$$o = \min\left(oc_{total}\right) \tag{17}$$

$$oc_{total} = \sum_{i=1}^{n(G)} oc(g_i) \tag{18}$$

$$oc(g_i) = \alpha_i + \beta_i p_i + \gamma_i p_i^2 \tag{19}$$

3. RESULTS AND DISCUSSION

3.1. Simulation and result

There are two evaluations of SCO in this paper. The first is the unconstrained case evaluation, and the second is the constrained case evaluation. In the unconstrained case evaluation, SCO is challenged to optimize the set of classic 23 functions. These 23 functions are split into three groups: seven high-dimension unimodal functions, six high-dimension multimodal functions, and ten fixed-dimension multimodal functions. A detailed description of these functions can be found in [37]. In the constrained case evaluation, SCO is challenged to optimize two ELD cases. The system contains ten generating units; in the second case, the system contains thirteen generating units. The specification, including the power demand, power range of the generating units, and the constants of these generating units of the first case, can be seen in [36], while the second case can be seen in [38].

In both evaluations, SCO competes with five recent metaheuristics. These competitors include COA, LEO, NGO, KOA, and WaOA. All these competitors are built based on swarm intelligence and consist of multiple searches implemented in serial steps. In both evaluations, the swarm size is 10, while the maximum iteration is 20. The dimensions for the high-dimension functions are set to 30. The unconstrained case evaluation results are provided in Tables 2 to 5. Meanwhile, the outcome of the constrained case evaluation is provided in Tables 6 and 7.

Table 2 shows that SCO can handle the high dimension unimodal functions. In this group, SCO becomes the first best of five functions (f_1, f_2, f_3, f_5, and f_7) and the second best of two functions (f_4 and f_6). SCO also achieves the global optimal solution in two functions (f_1 and f_2). On the other hand, NGO becomes the worst performer in this first group of functions. Besides, WaOA becomes the second-best performer as it achieves the second-best in four functions (f_1, f_3, f_5, and f_7). The performance gap between the best performer and worst performer is wide in four functions (f_1, f_3, f_4, and f_5).

Table 2. Simulation result for high-dimension unimodal functions

F	Parameter	COA [9]	LEO [22]	NGO [12]	KOA [11]	WaOA [10]	SCO
1	Average	0.5286	0.0003	3.1162×10^{1}	0.1037	0.0002	0.0000
	Position	5	3	6	4	2	1
2	Average	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Position	1	1	1	1	1	1
3	Average	7.8936×10^{2}	5.0442×10^{1}	1.0725×10^{4}	1.1016×10^{3}	2.1735×10^{1}	1.3041×10^{1}
	Position	4	3	6	5	2	1
4	Average	1.4435	0.0265	5.2258	0.3267	0.0139	0.0152
	Position	4	3	6	5	1	2
5	Average	3.9206×10^{1}	2.8927×10^{1}	1.2104×10^{3}	3.0230×10^{1}	2.8896×10^{1}	2.8872×10^{1}
	Position	5	3	6	4	2	1
6	Average	6.3273	5.5436	4.8029×10^{1}	5.8291	4.8619	5.1044
	Position	5	3	6	4	1	2
7	Average	0.0370	0.0121	0.0564	0.0243	0.0118	0.0108
	Position	5	3	6	4	2	1

The result in Table 3 indicates fiercer competition in solving the high-dimension multimodal functions than the high-dimension unimodal ones. SCO achieves the best performer in one function (f_{10}). On the other hand, WaOA becomes the best performer in two functions (f_9 and f_{13}). In this group, NGO still becomes the worst performer as it achieves the worst result in all six functions. The performance gap between the best and worst performers is wide in only three functions (f_9, f_{10}, and f_{11}).

Table 3. Simulation result for high-dimension multimodal functions

F	Parameter	COA [9]	LEO [22]	NGO [12]	KOA [11]	WaOA [10]	SCO
8	Average	-3.8152×10^3	-3.4797×10^3	-2.8095×10^3	-3.3695×10^3	-3.0902×10^3	-3.2750×10^3
	Position	1	2	6	3	5	4
9	Average	6.3555	2.0457×10^1	1.5992×10^2	1.7689×10^1	0.1008	6.1560
	Position	3	5	6	4	1	2
10	Average	0.1696	0.0030	2.8250	0.1027	0.0028	0.0004
	Position	5	3	6	4	2	1
11	Average	0.3423	0.0010	1.4465	0.0870	0.0102	0.0188
	Position	5	1	6	4	2	3
12	Average	0.8170	0.6924	1.8743	0.7932	0.6830	0.6532
	Position	5	3	6	4	2	1
13	Average	3.4450	2.9683	7.2517	3.2284	1.7247	2.8794
	Position	5	3	6	4	1	2

The result in Table 4 indicates that this group of functions is very competitive. The performance gap between the best and worst performers is narrow in all functions. SCO becomes the best performer in only three functions (f_{15}, f_{16}, and f_{19}). In these three functions, SCO never becomes the sole best performer. Moreover, the fierce competition in this group is also shown by the lack of function, where the quality of the best performer is lower than half of the worst performer except in f_{21}.

Table 4. Simulation result for fixed-dimension multimodal functions

F	Parameter	COA [9]	LEO [22]	NGO [12]	KOA [11]	WaOA [10]	SCO
14	Average	4.0210	4.7036	7.1144	6.9920	6.7870	4.3715
	Position	1	3	6	5	4	2
15	Average	0.0056	0.0006	0.0053	0.0034	0.0019	0.0006
	Position	6	1	5	4	3	1
16	Average	-1.0311	-1.0314	-1.0302	-1.0277	-1.0316	-1.0316
	Position	4	3	5	6	1	1
17	Average	0.3985	0.3982	0.4098	0.4005	0.3981	0.3982
	Position	4	2	6	5	1	2
18	Average	3.0088	3.0002	4.0673	3.0241	$1.2000\text{x}10^1$	4.3536
	Position	2	1	4	3	6	5
19	Average	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	Position	1	1	1	1	1	1
20	Average	-3.1063	-3.1990	-2.6961	-3.1429	-3.1667	-2.9713
	Position	4	1	6	3	2	5
21	Average	-6.5261	-3.5110	-2.5342	-5.6825	-4.1688	-3.7610
	Position	1	5	6	2	3	4
22	Average	-5.7265	-4.3010	-2.7945	-5.0157	-4.5013	-3.2486
	Position	1	4	6	2	3	5
23	Average	-4.6856	-4.1985	-2.4593	-4.8817	-4.0884	-3.6328
	Position	2	3	6	1	4	5

The result in Tables 2 and 4 is then summarized in Table 5. This summary presents the proposed SCO's superiority over its competitors. Data in Table 5 presents the number of functions where SCO outperforms its competitors in every group of functions. Table 5 shows the superiority of SCO over COA, LEO, NGO, and KOA. Meanwhile, equal performance occurs between SCO and WaOA as SCO outperforms WaOA in ten functions and is equal in three functions. In general, SCO is very superior in solving high-dimensional unimodal functions. Meanwhile, SCO is not exceptional in solving fixed dimension multimodal functions except compared to NGOs.

Table 5. Supremacy comparison

Cluster	COA [9]	LEO [22]	NGO [12]	KOA [11]	WaOA [10]
1	6	6	6	6	4
2	5	4	6	5	3
3	3	3	8	4	3
Total	14	13	20	15	10

The results in Tables 6 and 7 indicate the fierce competition in solving the ELD problem in both cases. The range among these six metaheuristics is minimal compared to their median. The range to the median is 0.29% in the first case and 0.067% in the second case.

Table 6. Simulation results in solving the ELD problem consisting of ten generating units

No	Metaheuristic	Total fuel cost (USD/hour)		
		Average	Min	Max
1	SCO	106,103	106,018	106,314
2	WaOA	106,074	105,991	106,206
3	KOA	106,156	106,015	106,346
4	NGO	106,391	106,000	107,402
5	LEO	106,155	106,033	106,377
6	COA	106,237	106,004	106,647

Table 7. Simulation results in solving the ELD problem consisting of thirteen generating units

No	Metaheuristic	Total fuel cost (USD/hour)		
		Average	Min	Max
1	SCO	17,941	17,938	17,951
2	WaOA	17,938	17,935	17,942
3	KOA	17,938	17,934	17,942
4	NGO	17,950	17,943	17,963
5	LEO	17,938	17,935	17,941
6	COA	17,942	17,937	17,950

3.2. Discussion

This section delves into a comprehensive examination of the results, findings, limitations, and future research and development prospects. Specifically, this study scrutinizes the strengths and weaknesses of the proposed SCO method compared to five other competing approaches. Unlike many studies that emphasize the performance evaluation of newly introduced metaheuristic techniques, this discussion delves into deeper unconstrained aspects. It provides a nuanced analysis of how SCO fares against its competitors, shedding light on its distinctive features, potential advantages, and areas where improvements may be warranted. By delving into this detailed comparative analysis, the study contributes to a more holistic understanding of the efficacy and applicability of SCO in optimization tasks, paving the way for further advancements in metaheuristic research and development.

In general, several key findings were found in this study. The first finding is that the proposed SCO is superior in solving high-dimensional functions, whether unimodal or multimodal. In contrast, SCO is not exceptional but still competitive in solving fixed-dimension functions. The second finding is that the proposed SCO is also competitive in solving the ELD problem in both cases. The third finding is that the ELD problem is highly competitive, with no metaheuristics that can be superior to others. A more comprehensive analysis of these findings and the drawbacks to the technical and unconstrained aspects will be discussed in the following discussion.

The proposed SCO is superior at solving high-dimension functions. Its dominance is higher in solving the unimodal functions than the multimodal ones, as provided in Table 5. This circumstance can be linked to the directed search toward the finest entity. This search is also performed by COA [9], LEO [22], KOA [11], and WaOA [10]. In COA [9], half of swarm members perform this search in the first step. In LEO [22] and WaOA [10], all swarm members perform this search in the first step. Meanwhile, in KOA [11], the reference in the directed search is a randomly selected finer member. Meanwhile, in SCO, all entities perform this search in all its three steps. Moreover, the superiority of SCO in these functions is also supported by the directed search toward the means of finer entities plus the finest entity. NGOs endorsed this circumstance, becoming the worst performer in these two groups. NGO is the only competitor that does not perform the directed search toward the finest entity, as it performs only the directed search relative to a randomly chosen entity inside the swarm [12]. Meanwhile, this search is also employed in SCO, WaOA, LEO, and NGOs.

In contrast, the directed search toward the finest entity does not significantly affect solving the fixed dimension multimodal functions. This finding can be traced from the tight competition in this third group of functions. Although NGO is still the worst performer in this group, its gap to the best performer is not so far as in the first and second groups of functions. On the other hand, although SCO performed this search three times, it still failed to become superior.

The very tight competition in solving the ELD problem shows that, in general, the competitiveness of many metaheuristics in solving this problem is almost equal. Data in Tables 6 and 7 strengthens this finding, as no metaheuristic can generate a final solution with a significant quality gap. This tight competition is also found in many existing studies in ELD where the proposed method cannot improve its competitors with substantial gaps. This tight competition can be investigated based on the specifications of the system. In the first case, the demand is 2,000 watt/hour [36]. Meanwhile, the maximum capacity is 2,365 watts/hour [36]. This means that the market has 84,5% of the maximum capacity. On the other hand, the maximum power of g_9 and g_{10} is 470 watts/hour each [36]. It means that both generators play a significant role in meeting the demand. Shutting down one of these generators makes the system fail to meet the demand. On the other hand, the fixed cost of each generator is much higher than the variable cost, as observed from the constants of the cost function. In contrast, the circumstances in the second ELD problem are slightly different. Based on [38], the minimum power is 550 watts/hour, and the maximum is 2,960 watts/hour. Unfortunately, many of these generators have the exact specifications for the power range and constants in the cost function [9]. This circumstance makes the difference in output power in generators does not affect the significant difference in total cost. The equality constraint in meeting the demand makes the ELD problem less maneuverable.

The results from both evaluations also highlight the limited or trivial impact of neighborhood search with a reduced search space, a technique utilized by all five competitors. In contrast, SCO does not incorporate this search strategy. However, the absence of neighborhood search does not diminish SCO's efficacy as a metaheuristic in either evaluation. Interestingly, the presence of this search strategy in all five competitors fails to improve their performance significantly. This observation underscores the greater significance of directed search over neighborhood search in swarm-based metaheuristics.

Several limitations in this study can be used as a baseline for future studies. First, there are a lot of ELD problems, from the classic ones, such as IEEE bus systems, to the more specific ones, such as the power grid system in India and Indonesia. On the other hand, assessing any proposed technique in all these cases is impossible. This means that future studies can use these different use cases. This current study has not included other considerations, such as emissions, and power loss. So, these parameters can also be considered in future studies.

4. CONCLUSION

This study introduced a new metaheuristic called SCO, and its implementation was used to tackle unconstrained and constrained problems. The formal model of SCO has been provided through pseudocode and mathematical formulation. Two evaluations were performed to investigate the performance of SCO, and five new metaheuristics were chosen as competitors. The unconstrained case evaluation shows that SCO is better than COA, LEO, NGO, KOA, and WaOA in 14, 13, 20, 15, and 10 functions out of 23. In this evaluation, the superiority of SCO occurs mainly in solving high-dimensional functions where a directed search toward the finest entity provides a significant contribution. On the other hand, competition in fixed-dimension multimodal functions is tight, and the range between the best and worst performers is narrow. Another finding in this evaluation is that the neighborhood search with declining local space is not critical as its contribution is far less than the directed search. The very tight competition also occurs in solving the ELD problem where the ratio between the range of the best and worst performers to the median is minimal so that no metaheuristic can produce results with a significant gap to others. This circumstance comes from the nature of the ELD problem, which is less maneuverable. Several factors contribute to the result, such as equality constraint where the demand is close to the maximum power, the significance of the fixed cost, and the uniformity of generating units. In the future, this study can be expanded by employing various use cases of ELD problems, both classical and contemporary, and adopting additional considerations, such as emission, and power loss.

ACKNOWLEDGEMENTS

The publication of this paper is financially supported by Telkom University, Indonesia, without specific grant.

REFERENCES

[1] H. S. Maharana and S. K. Dash, “Dual objective multiconstraint swarm optimization based advanced economic load dispatch,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 1924–1932, Jun. 2021, doi: 10.11591/ijece.v11i3.pp1924-1932.

[2] A. T. Doan, D. T. Viet, and M. Q. Duong, “Economic load dispatch solutions considering multiple fuels for thermal units and generation cost of wind turbines,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 3718–3726, Oct. 2021, doi: 10.11591/ijece.v11i5.pp3718-3726.

[3] N. Singh *et al.*, “Novel Heuristic Optimization Technique to Solve Economic Load Dispatch and Economic Emission Load Dispatch Problems,” *Electronics*, vol. 12, no. 13, pp. 1–16, Jul. 2023, doi: 10.3390/electronics12132921.

[4] W. Khamsen, C. Takeang, and P. Aunban, “Hybrid method for solving the non smooth cost function economic dispatch problem,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 609–616, Feb. 2020, doi: 10.11591/ijece.v10i1.pp609-616.

[5] K. M. D. Puspitasari, J. Raharjo, A. S. Sastrosubroto, and B. Rahmat, “Generator Scheduling Optimization Involving Emission to Determine Emission Reduction Costs,” *International Journal of Engineering*, vol. 35, no. 8, pp. 1468–1478, 2022, doi: 10.5829/IJE.2022.35.08B.02.

[6] D. C. Secui, G. Bendea, M. L. Secui, C. Hora, and C. Bendea, "The Chaotic Social Group Optimization for the Economic Dispatch Problem," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 666–677, Dec. 2021, doi: 10.22266/ijies2021.1231.59.

[7] F. Tariq, S. Alelyani, G. Abbas, A. Qahmash, and M. R. Hussain, "Solving renewables-integrated economic load dispatch problem by variant of metaheuristic bat-inspired algorithm," *Energies*, vol. 13, no. 23, pp. 1–36, Nov. 2020, doi: 10.3390/en13236225.

[8] T. Khobaragade and K. T. Chaturvedi, "Enhanced Economic Load Dispatch by Teaching–Learning-Based Optimization (TLBO) on Thermal Units: A Comparative Study with Different Plug-in Electric Vehicle (PEV) Charging Strategies," *Energies*, vol. 16, no. 19, pp. 1–18, Oct. 2023, doi: 10.3390/en16196933.

[9] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, pp. 1–43, Jan. 2023, doi: 10.1016/j.knosys.2022.110011.

[10] P. Trojovský and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior," *Scientific Reports*, vol. 13, no. 1, pp. 1–32, May 2023, doi: 10.1038/s41598-023-35863-5.

[11] M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. M. Ahmed, "Kookaburra Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics*, vol. 8, no. 6, pp. 1–54, Oct. 2023, doi: 10.3390/biomimetics8060470.

[12] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021, doi: 10.1109/ACCESS.2021.3133286.

[13] N. Chopra and M. Mohsin Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Expert Systems with Applications*, vol. 198, p. 116924, Jul. 2022, doi: 10.1016/j.eswa.2022.116924.

[14] E. Trojovska and M. Dehghani, "Clouded Leopard Optimization: A New Nature-Inspired Optimization Algorithm," *IEEE Access*, vol. 10, pp. 102876–102906, 2022, doi: 10.1109/ACCESS.2022.3208700.

[15] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, "The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems," *Scientific Reports*, vol. 12, no. 1, pp. 1–20, Jun. 2022, doi: 10.1038/s41598-022-14338-z.

[16] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm," *Applied Soft Computing*, vol. 114, pp. 1–17, Jan. 2022, doi: 10.1016/j.asoc.2021.108043.

[17] M. S. Braik, "Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems," *Expert Systems with Applications*, vol. 174, pp. 1–25, Jul. 2021, doi: 10.1016/j.eswa.2021.114685.

[18] M. Suman, V. P. Sakthivel, and P. D. Sathya, "Squirrel search optimizer: Nature inspired metaheuristic strategy for solving disparate economic dispatch problems," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 111–121, Oct. 2020, doi: 10.22266/ijies2020.1031.11.

[19] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowledge-Based Systems*, vol. 243, p. 108457, May 2022, doi: 10.1016/j.knosys.2022.108457.

[20] H. Jia, H. Rao, C. Wen, and S. Mirjalili, "Crayfish optimization algorithm," *Artificial Intelligence Review*, vol. 56, no. S2, pp. 1919–1979, Nov. 2023, doi: 10.1007/s10462-023-10567-4.

[21] D. Połap and M. Woźniak, "Red fox optimization algorithm," *Expert Systems with Applications*, vol. 166, pp. 1–21, Mar. 2021, doi: 10.1016/j.eswa.2020.114107.

[22] P. Trojovský, M. Dehghani, E. Trojovská, and E. Milkova, "Language Education Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems," *CMES - Computer Modeling in Engineering and Sciences*, vol. 136, no. 2, pp. 1527–1573, 2023, doi: 10.32604/cmes.2023.025908.

[23] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The Deep Sleep Optimizer: A Human-Based Metaheuristic Approach," *IEEE Access*, vol. 11, pp. 83639–83665, 2023, doi: 10.1109/ACCESS.2023.3298105.

[24] P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems based on preschool education," *Scientific Reports*, vol. 13, no. 1, pp. 1–31, Dec. 2023, doi: 10.1038/s41598-023-48462-1.

[25] P. Trojovský and M. Dehghani, "Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems," *CMES - Computer Modeling in Engineering and Sciences*, vol. 137, no. 2, pp. 1695–1730, 2023, doi: 10.32604/cmes.2023.028314.

[26] P. D. Kusuma and A. Novianty, "Total Interaction Algorithm: A Metaheuristic in which Each Agent Interacts with All Other Agents," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 1, pp. 224–234, Feb. 2023, doi: 10.22266/ijies2023.0228.20.

[27] M. Ghasemi *et al.*, "A new metaphor-less simple algorithm based on Rao algorithms: a Fully Informed Search Algorithm (FISA)," *PeerJ Computer Science*, vol. 9, pp. 1–24, Aug. 2023, doi: 10.7717/peerj-cs.1431.

[28] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden Search Optimization Algorithm," *IEEE Access*, vol. 10, pp. 37515–37532, 2022, doi: 10.1109/ACCESS.2022.3162853.

[29] P. D. Kusuma and F. C. Hasibuan, "Attack-Leave Optimizer: A New Metaheuristic that Focuses on The Guided Search and Performs Random Search as Alternative," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 3, pp. 244–257, Jun. 2023, doi: 10.22266/ijies2023.0630.19.

[30] P. D. Kusuma and A. Dinimaharawati, "Three on Three Optimizer: A New Metaheuristic with Three Guided Searches and Three Random Searches," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, pp. 420–429, 2023, doi: 10.14569/IJACSA.2023.0140145.

[31] R. Ragunathan and B. Ramadoss, "Golden jackal optimization for economic load dispatch problems with complex constraints," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 2, pp. 781–793, Apr. 2024, doi: 10.11591/eei.v13i2.6572.

[32] H. D. Nguyen and L. H. Pham, "Solutions of economic load dispatch problems for hybrid power plants using Dandelion optimizer," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2569–2576, Oct. 2023, doi: 10.11591/eei.v12i5.5245.

[33] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[34] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm," *IEEE Access*, vol. 10, pp. 49445–49473, 2022, doi: 10.1109/ACCESS.2022.3172789.

[35] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[36] S. R. Spea, "Solving practical economic load dispatch problem using crow search algorithm," *International Journal of Electrical*

and Computer Engineering (IJECE), vol. 10, no. 4, pp. 3431–3440, Aug. 2020, doi: 10.11591/IJECE.V10I4.PP3431-3440.
[37] F. A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva, and G. Dhiman, "Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems," *Computers, Materials and Continua*, vol. 72, no. 1, pp. 399–416, 2022, doi: 10.32604/cmc.2022.024736.
[38] H. Zein, J. Raharjo, and I. R. Mardiyanto, "A Method for Completing Economic Load Dispatch Using the Technique of Narrowing Down Area," *IEEE Access*, vol. 10, pp. 30822–30831, 2022, doi: 10.1109/ACCESS.2022.3158928.

BIOGRAPHIES OF AUTHORS

Purba Daru Kusuma received his bachelor's and master's degrees in electrical engineering from Bandung Institute of Technology, Indonesia, in 2002 and 2009, respectively. He received his doctoral degree in computer science from Gadjah Mada University, Indonesia, in 2017. Currently, he is an assistant professor at Telkom University, Indonesia. His research area is artificial intelligence, machine learning, and operational research. He can be contacted at email: purbodaru@telkomuniversity.ac.id.

Faisal Candrasyah Hasibuan is a Computer Engineering lecturer at Telkom University, Bandung, West Java, Indonesia. Status as an alma mater, he completed a bachelor's program at the place where he studies now. He was granted a master's degree in electrical engineering with a Computer Engineering specialization at Bandung Institute of Technology (ITB). The field of research interests are but are not limited to embedded systems, the internet of things, and intelligent control systems. He can be contacted at email: faicanhasfcb@telkomuniversity.ac.id.