Received: January 29, 2024. Revised: March 11, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Stochastic Shaking Algorithm: A New Swarm-Based Metaheuristic and Its Implementation in Economic Load Dispatch Problem

Purba Daru Kusuma[1]* **Anggunmeka Luhur Prasasti[1]**

[1]*Computer Engineering, Telkom University, Indonesia*
* Corresponding author's Email: purbodaru@telkomuniversity.ac.id

Abstract: This paper introduces a novel metaheuristic named the stochastic shaking algorithm (SSA), which is rooted in swarm intelligence principles. The innovation lies in its unique utilization of iteration for selecting references during guided searches through a stochastic approach. The optimization process involves two sequential steps: the primary reference in the first step is the finest swarm member, while in the second step, it is the mean of all finer swarm members plus the finest one. This primary reference is then combined with a randomly chosen solution within the space, serving as the secondary reference. SSA undergoes evaluation in two contexts. The first involves assessing its performance using a set of 23 classic functions as a theoretical use case. The second involves tackling the economic load dispatch problem (ELD), a practical use case featuring a system with 13 generators of various energy resources. The study compares SSA against five other metaheuristics—One to One Based Optimization (OOBO), Kookaburra Optimization Algorithm (KOA), Language Education Optimization (LEO), Total Interaction Algorithm (TIA), and Walrus Optimization Algorithm (WaOA). Results indicate SSA's superiority over OOBO, KOA, LEO, TIA, and WaOA in 21, 13, 11, 16, and 14 functions out of 23 functions, respectively. Additionally, the evaluation of the economic load dispatch problem reveals intense competition among the six metaheuristics.

Keywords: Optimization, Metaheuristic, Swam intelligence, Economic load dispatch, Energy.

1. Introduction

The economic load dispatch problem is a well-known optimization challenge within the energy sector. This problem, a variant of the economic dispatch problem with a differentiable form [1], has been the subject of numerous studies aiming to leverage metaheuristic approaches for optimization. Unfortunately, many of these investigations continue to rely on, modify, or integrate traditional metaheuristics to form hybrid methods. Examples include simulated annealing (SA) [2], modified particle swarm optimization (MPSO) [3], modified directional bat algorithm (DBA) [4], teaching learning-based optimization (TLBO) [5], particle swarm optimization (PSO) [6], slime mold algorithm (SMO) [7], chaotic social group optimization (CSGO) [8], multi-verse optimization (MVO) [9], and others.

In recent years, a plethora of novel metaheuristics have emerged, with the majority being rooted in swarm intelligence principles. Many of these swarm-based metaheuristics draw inspiration from animal behavior, adopting metaphors in their designs. Examples include the Kookaburra Optimization Algorithm (KOA) [10], Lyrebird Optimization Algorithm (LOA) [11], Stochastic Komodo Algorithm (SKA) [12], Green Anaconda Optimization (GAO) [13], Walrus Optimization Algorithm (WaOA) [14], Coati Optimization Algorithm (COA) [15], White Shark Algorithm (WSA) [16], Squirrel Search Optimization (SSO) [17], Tasmanian Devil Optimization (TDO) [18], Northern Goshawk Optimization (NGO) [19], Osprey Optimization Algorithm (OOA) [20], and others. Some swarm-based metaheuristics incorporate social behavior elements, as seen in the Migration Algorithm (MA) [21], Language Education Optimization (LEO) [22], Mother

Optimization Algorithm (MOA) [23], and so forth. Conversely, several metaheuristics eschew metaphors altogether, such as One-to-One Based Optimization (OOBO) [24], Total Interaction Algorithm (TIA) [25], Attack-Leave Optimization (ALO) [26], Subtraction-Average Based Optimization (SABO) [27], Fully Informed Search Algorithm (FISA) [28], and so on.

Numerous research articles introducing innovative metaheuristic approaches predominantly focus on four mechanical engineering design problems: pressure vessel design, welded beam design, tension/compression spring design, and speed reducer design. In contrast, there is a notable scarcity of research articles exploring the application of these metaheuristics within use cases in the energy sector. SSO is an example of a new metaheuristic that is utilized to solve disparate economic dispatch problems in several power systems [17]. The Technique of Narrowing Down Area (ToNDA) is another example of a new metaheuristic developed based on the neighborhood search designed especially for economic load dispatch problems [1]. In ToNDA, the local search space, which is the power range in each generator is reduced through iteration [1]. This concept is like neighborhood search which becomes a secondary search in many swarm-based metaheuristics, such as NGO [19], COA [15], OOA, and so on.

Meanwhile, there is a challenge to explore the use of iteration not only for a counter but also to decide the strategy. In some metaheuristics, such as TIA, FISA, and OOBO, iteration is used only for the counter. In some other metaheuristics like NGO, COA [15], and WaOA [14], iteration is used to determine the local search space in its neighborhood search.

The present research introduces an innovative metaheuristic termed the Stochastic Shaking Algorithm (SSA). SSA is developed within the framework of swarm intelligence and introduces a unique approach that employs iteration as a crucial factor in determining the strategy. Subsequently, the effectiveness of SSA is evaluated by applying it to the economic load dispatch problem, serving as a practical use case, in addition to assessing its performance on 23 classic functions representing the theoretical use case.

The primary scientific contributions of this study are outlined as follows.

- A new swarm-based metaheuristic that is developed based on swarm intelligence and utilizes iteration to determine the strategy is introduced.
- The presentation of the SSA consists of the concept, algorithm, and mathematical formulation.
- The performance of SSA is assessed on 23 classic functions as theoretical use cases and economic load dispatch problem as practical use case.
- The competition of SSA with five recent metaheuristics is performed to assess the improvement of the proposed algorithm with the existing methods.

The subsequent sections of this research article are organized as follows. Section 2 provides a comprehensive review of recent studies that have introduced new metaheuristics, encompassing details on the strategy employed and the use cases considered for assessment. In Section 3, the model of the proposed Stochastic Shaking Algorithm (SSA) and its application to the economic load dispatch problem are presented. The SSA model includes an explanation of the concept, the algorithm, and the mathematical formulation. Simultaneously, the economic load dispatch problem model comprises the concept and its corresponding mathematical formulation. Moving on to Section 4, the experimental setup for the assessment and the obtained results are elucidated. Section 5 delves into an in-depth investigation of the results, highlights key findings, and discusses any limitations to the theoretical framework. Finally, Section 6 concludes the study and outlines potential avenues for future research.

2. Related works

Swarm intelligence has been utilized as a baseline or framework for the development of many recent metaheuristics. Nowadays, more metaheuristics are developed based on swarm intelligence rather than other approaches, such as evolutionary systems, neighborhood search, and so on. The swarm intelligence can be easily acknowledged by the existence of a population where each member is active and autonomous in performing searches. Swarm intelligence is also easily acknowledged by the existence of the guided search where the searching is represented by the motion toward or away from certain references. These references can be the finest member, any other member, the worst member, the mean of some finer members, and so on. This guided search becomes the sole search in some swarm-based metaheuristics or the primary search in some other swarm-based metaheuristics which are also enriched with other searches like the neighborhood search.

Table 1. The mechanics of shortcoming metaheuristics and the theoretical test used in their first introduction

No	Metaheuristics	References	Sequential Steps	Use Case
1	OOBO [24]	a random permutation member	1	23 classic functions, CEC 2017, 4 design problems
2	KOA [10]	a randomly chosen better member	2	CEC 2011, CEC 2017, 4 design problems
3	LEO [22]	a randomly chosen member from a group consisting of all finer members and the finest member; a randomly chosen other member	3	23 classic functions; CEC 2017; 4 design problems
4	TIA [25]	all other members	1	23 classic functions
5	WaOA [14]	the finest member, a randomly chosen another member	3	23 classic functions, CEC 2015, CEC 2017, 4 design problems
6	LOA [11]	a randomly chosen another member	1	CEC 2017, CEC 2011, 4 design problems
7	GAO [13]	a randomly chosen better member using a normal distribution	2	CEC 2011, CEC 2017, CEC 2019
8	ALO [26]	the finest member, balance mixture between the finest member and a randomly chosen member, balance mixture between two randomly chosen member	3	23 classic functions
9	SABO [27]	all members	1	23 classic functions, CEC 2017,
10	FISA [28]	mean of all finer members plus the finest member, mean of all worse members plus the worst member	1	CEC 2005, CEC 2014, 4 design problems
11	this work	the finest member, balance mixture between the finest member and a random solution within space, means of all finer members plus the finest member, balance mixture between the mean of all finer members plus the finest member and a random solution within space	2	23 classic functions, economic load dispatch problem

In recent years, many swarm-based metaheuristics have been developed by multiple searches rather than relying on only a single search. This multiple search approach is adopted due to the imperfect of any search so that the weakness of a search should be covered by another search. Moreover, a search may be suitable for certain conditions but performs mediocre in other circumstances. For example, Komodo mlipir algorithm (KMA) enriches the guided search with the crossover with the finest quality swarm member for the moderate quality swarm members [29]. This multiple-search approach can be conducted sequentially and performed by all swarm members, or it can be performed stochastically. Another option is to split the population so that the same search is performed by swarm members within the same group while the other swarm members perform different searches as seen in KMA [29] or COA [15].

In many studies proposing new metaheuristics, certain assessment is performed to investigate the performance of the proposed method. Several sets of mathematical functions like 23 classic functions or IEEE CEC series are often utilized to represent the theoretical use cases. This theoretical optimization use case has become the primary use case in many studies proposing new metaheuristics. Meanwhile, the practical use case is also found in some studies where four design problems in mechanical engineering can be found in many studies associated with Dehghani, Mirjalili, or Braik, such as chameleon swarm algorithm (CSA) [30], OOBO [24], geometric mean optimizer (GMO) [31], Geyser Inspired Algorithm (GEA) [32], and so on.

A summary of some recent metaheuristics is exhibited in Table 1. All metaheuristics in Table 1 were first introduced in 2023. All of them are swarm-based metaheuristics. The information in Table 1 includes the references used during the guided search, the number of sequential steps, and the use case utilized during the assessment. The information on the proposed method is written in the last row to provide a clear positioning of the proposed method compared to the existing ones.

Based on this explanation, there are some spaces to conduct a study in proposing a new metaheuristic. First, iteration can be utilized to determine the strategy for searching rather than to determine the acceptance of worse solutions as in SA or the local search space as in NGO [19], WaOA [14], COA [15], and so on. Second, there are various practical problems besides the common four design problems in mechanical engineering that can be used to assess the performance of a new metaheuristic in its first introduction. In this case, the economic load dispatch problem as an optimization problem in the energy sector is rare to find in many studies introducing a new metaheuristic.

Table 2. Notations used in stochastic shaking algorithm

Notation	Description
a	index of the swarm member
b	index for the dimension
f	objective function
r_u	real uniform random [0, 1]
r_i	integer uniform random [1, 2]
t	iteration
t_{max}	maximum iteration
x	swarm member
X	swarm
x_l	lower boundary of the space
x_u	upper boundary of the space
x_{mean}	mean of finer members plus the finest member
X_{pool}	a set consisting of all finer members plus the finest member
x_{fst}	the finest member
x_r	the reference
x_c	the seed

3. Model

3.1 Stochastic shaking algorithm

In a swarm intelligence-based metaheuristic, each member of the swarm functions independently as an autonomous tracker, autonomously pursuing improvements. This paradigm is also implemented in the Stochastic Shaking Algorithm (SSA). While SSA is fundamentally a swarm intelligence approach, it incorporates controlled mutation influenced by the iteration. The likelihood of mutation is initially high during the early iterations, gradually decreasing linearly over subsequent iterations. Notably, the mutation is applied to the reference used in the guided search rather than to the individual swarm members.

SSA employs two references in general. The first reference is the finest member of the swarm, a common practice in existing metaheuristics for effective exploitation. The second reference is the mean or average of a selection of superior members, combined with the finest member. This dual-reference strategy is designed to guide swarm members towards areas where superior members are concentrated, enhancing exploration and convergence capabilities.

The novel strategy is introduced by mixing the two references with a random solution within space. This strategy is designed to improve the exploration capability as moving toward the finest member or the mean of better members plus the finest member cannot guarantee improvement escaping from the local optimal. In this context, the main reference is mixed with a randomly generated solution within the space in a balanced portion.

This fundamental concept is then transformed into a searching strategy of SSA. In SSA, there are two sequential steps performed by each swarm member in every iteration. The finest swarm member becomes the primary reference in the first step while the mean of finer members plus the finest member becomes the primary reference in the second step.

The secondary reference is determined stochastically. The balance mixture between the finest member and a random solution may become the secondary reference in the first step while the balance mixture between the mean of better members plus the finest member and a randomly generated solution may become the secondary reference in the second step. The determination of choosing the secondary reference is based on the threshold constructed from the iteration divided by the maximum iteration. The secondary reference is chosen whether the primary reference or the balance mixed reference.

The stringent acceptance rule is applied in SSA. There are two offspring generated in every step. The first offspring is produced by the guided search toward the primary reference while the second offspring is produced by the guided search toward the secondary reference. The better offspring becomes the final seed to be compared with the current value of the swarm member.

The formalization of SSA is presented in algorithm 1 and Eqs. (1)-(16). Algorithm 1 encapsulates the comprehensive formalization of the entire SSA process. Meanwhile, Eqs. (1)-(16) explicates the formulation for each distinct step. Table 2 provides a detailed presentation of the notations employed in the formalization of SSA, ensuring clarity and precision in understanding the algorithmic processes.

Algorithm 1: stochastic shaking algorithm

```
begin
 for a=1 : n(X)
  initialize x_a using Eq. (1)
```

Received: January 29, 2024. Revised: March 11, 2024.

```
4       update x_finest using Eq. (2)
5     end
6     for t=1 : t_max
7       for a=1 : n(X)
8         perform 1st search using Eqs. (3)-(8)
9         update x_finest using Eq. (2)
10        perform 2nd search using Eqs. (9)-(16)
11        update x_finest using Eq. (2)
12      end
13    end
14  End
15  return x_finest
```

As with any metaheuristic, there are two stages in SSA: initialization and iteration. In algorithm 1, initialization is presented from line 2 to line 5. Meanwhile, iteration is presented from line 6 to line 13. There are two processes conducted in the initialization which are generating an initial solution for each swarm member and the updating of the finest member. Meanwhile, the two guided searches are conducted during iteration. The finest member is updated each time a search is performed. The finest member becomes the final solution, and it becomes the output of the algorithm.

The mathematical formulation in the iteration is presented in Eqs. (1) and (2). Eq. (1) shows that the initial solution for each member is uniformly distributed within the space. Eq. (2) presents the stringent acceptance rule in the updating of the finest member.

$$x_{a,b} = x_{l,b} + r_u\left(x_{u,b} - x_{l,b}\right) \quad (1)$$

$$x_{finest}' = \begin{cases} x_a, f(x_a) < f(x_b) \\ x_{fst}, else \end{cases} \quad (2)$$

The first search is formalized using Eqs. (3)-(8). Eq. (3) shows that the finest member becomes the primary reference in the first search. Eq. (4) formulates the first seed in the first search from the motion toward the finest member. Eq. (5) formulates the secondary reference in the second search which may be the finest member or the balanced mixture between the finest member and a random solution within space. Eq. (6) formulates the second seed in the first search. Eq. (7) formulates the selection for the final seed in the first search. Eq. (8) formulates the stringent acceptance rule in the first search.

$$x_{r11,a,b} = x_{fst,b} \quad (3)$$

$$x_{c11,a,b} = x_{a,b} + r_u(x_{r11,a,b} - r_i x_{a,b}) \quad (4)$$

$$x_{r12,a,b} = \begin{cases} x_{fst,b}, r_u > \frac{t}{t_{max}} \\ \frac{x_{fst,b} + x_{l,b} + r_u(x_{u,b} - x_{l,b})}{2}, else \end{cases} \quad (5)$$

$$x_{c12,a,b} = x_{a,b} + r_u(x_{r12,a,b} - r_i x_{a,b}) \quad (6)$$

$$x_{c1,a} = \begin{cases} x_{c11,a}, f\left(x_{c11,a}\right) < f(x_{c12,a}) \\ x_{c12,a}, else \end{cases} \quad (7)$$

$$x_a' = \begin{cases} x_{c1,a}, f\left(x_{c1,a}\right) < f(x_a) \\ x_a, else \end{cases} \quad (8)$$

The second search is formulated using Eqs. (9)-(16). Eq. (9) formulates the pool consisting of all finer members plus the finest member. Eq. (10) formulates the mean member based on the average of all members within the pool. Eq. (11) shows that the mean member becomes the primary reference in the second search. Eq. (12) shows that the motion toward this primary reference generates the first seed for the second search. Eq. (13) shows that the secondary reference can be the mean member or the balanced mixture between the mean member and a random solution within space. Eq. (14) states that the second seed is generated through the motion toward the secondary reference. Eq. (15) shows the selection for the final seed in the second search. Eq. (16) shows the stringent acceptance rule in the second search.

$$X_{pool,a} = \{x \in X, f(x) < f(x_a) \cup x_{fst}\} \quad (9)$$

$$x_{mean,a,b} = \frac{\sum x_{pool,a,b}}{n(X_{pool,a})} \quad (10)$$

$$x_{r21,a,b} = x_{mean,b} \quad (11)$$

$$x_{c21,a,b} = x_{a,b} + r_u(x_{r21,a,b} - r_i x_{a,b}) \quad (12)$$

$$x_{r22,a,b} = \begin{cases} x_{mean,b}, r_u > \frac{t}{t_{max}} \\ \frac{x_{mean,b} + x_{l,b} + r_u(x_{u,b} - x_{l,b})}{2}, else \end{cases} \quad (13)$$

$$x_{c22,a,b} = x_{a,b} + r_u(x_{r22,a,b} - r_i x_{a,b}) \quad (14)$$

$$x_{c2,a} = \begin{cases} x_{c21,a}, f\left(x_{c21,a}\right) < f(x_{c22,a}) \\ x_{c22,a}, else \end{cases} \quad (15)$$

$$x_a' = \begin{cases} x_{c2,a}, f\left(x_{c2,a}\right) < f(x_a) \\ x_a, else \end{cases} \quad (16)$$

The computational complexity of SSA can be investigated through the number of loops involved in the process. Based on algorithm1, the computational complexity during the initialization and iteration is different. Meanwhile, in all processes, there is an additional loop which is the loop for the whole dimension in generating a new solution or seed. The computational complexity during the initialization can be presented as $O(n(X).d)$. Meanwhile, the computational complexity during the iteration can be presented as $O(4t_{max}.n(X).d)$. Term four represents two searches where each search consists of two sub-searches: the primary one and the secondary one.

3.2 Economic load dispatch problem

The economic load dispatch problem is conceptualized as an optimization challenge, involving a defined set of generators. Each generator can generate power within a specified minimum to maximum power range. The overall power output is the aggregate of the power generated by individual generators, and this output must satisfy the prevailing power demand. During operation, each generator incurs a cost, typically represented by a quadratic function. The primary objective of the economic load dispatch problem is to minimize the total fuel cost by appropriately adjusting the power output of each generator within its specified range [17]. In this context, the cost is treated as a soft constraint, while the power range serves as a hard constraint. The mathematical formulation of the economic load dispatch problem is detailed in Eqs. (17)-(23), as outlined in [17].

$$G = \{g_1, g_2, g_3, \dots, g_{n(G)}\} \quad (17)$$

$$p_{min,i} \leq p_i \leq p_{max,i} \quad (18)$$

$$p_{tot} = \sum_{n(G)} p_i \quad (19)$$

$$p_{tot} = p_d \quad (20)$$

$$c_{tot} = \sum_{n(G)} c_i \quad (21)$$

$$c_i = \alpha + \beta p_i + \gamma p_i^2 \quad (22)$$

The explanation of Eqs. (17)-(22) is as follows. Eq. (17) states that the system consists of a certain number of generators where g represents the generator. Eq. (18) states that the power of each generator (p_i) should be within the range of its minimum p_{min} and maximum p_{max} values. The total power of the system p_{tot} is the accumulation of the power of all generators as stated in Eq. (19). This total power should meet the demand p_d as stated in Eq. (20). Meanwhile, the total cost c_{tot} is the accumulation of cost produced by all generators as stated in Eq. (21) where c_i is the cost produced by generator i. Eq. (22) presents the cost function of each generator where α, β, and γ are the constants.

4. Simulation and result

This research undertakes two distinct assessments. In the initial assessment, the Stochastic Shaking Algorithm (SSA) is tasked with addressing theoretical problems, utilizing a set of 23 classic functions as representative examples. This segment aims to evaluate SSA's performance in theoretical scenarios. In the subsequent assessment, SSA is tested in addressing the economic load dispatch problem, a practical and real-world challenge within the energy sector. This second assessment seeks to assess SSA's efficacy in tackling practical problems, showcasing its applicability beyond theoretical scenarios.

Five recent swarm-based metaheuristics are chosen as contenders for SSA. These contenders are OOBO, KOA, LEO, TIA, and WaOA. All the metaheuristics under consideration were introduced for the first time in 2023, marking them as novel methodologies. These contenders are consistently evaluated across both assessments conducted in this study. In each assessment, a uniform swarm size of 5 and a maximum iteration limit of 20 are applied.

The selection of the 23 classic functions for evaluation is deliberate, aiming to encompass diverse scenarios. This set comprises seven high-dimension unimodal functions (HDU), six high-dimension multimodal functions (HDM), and ten fixed-dimension multimodal functions (FDM). The functions present a spectrum of spatial characteristics, ranging from narrow to expansive, and terrain features, varying from smooth to undulating. Specific functions within this set exhibit flat terrains interspersed with small, steep holes, posing challenges for locating the global optimal solution. A detailed description of these 23 classic functions is available in Table 3.

The result of the first assessment is presented in Tables 4 to 7. Table 4 exhibits the assessment result for HDU. Table 5 exhibits the assessment result for HDM. Table 6 exhibits the assessment result for FDM. Table 7 summarizes the superiority of SSA compared to its contenders. Meanwhile, the convergence chart is presented in Fig. 1.

Table 4 illustrates that the Stochastic Shaking Algorithm (SSA) excels particularly in addressing high-dimensional functions, emerging as the top performer in five instances (f_1, f_2, f_3, f_4, and f_7).

Received: January 29, 2024. Revised: March 11, 2024.

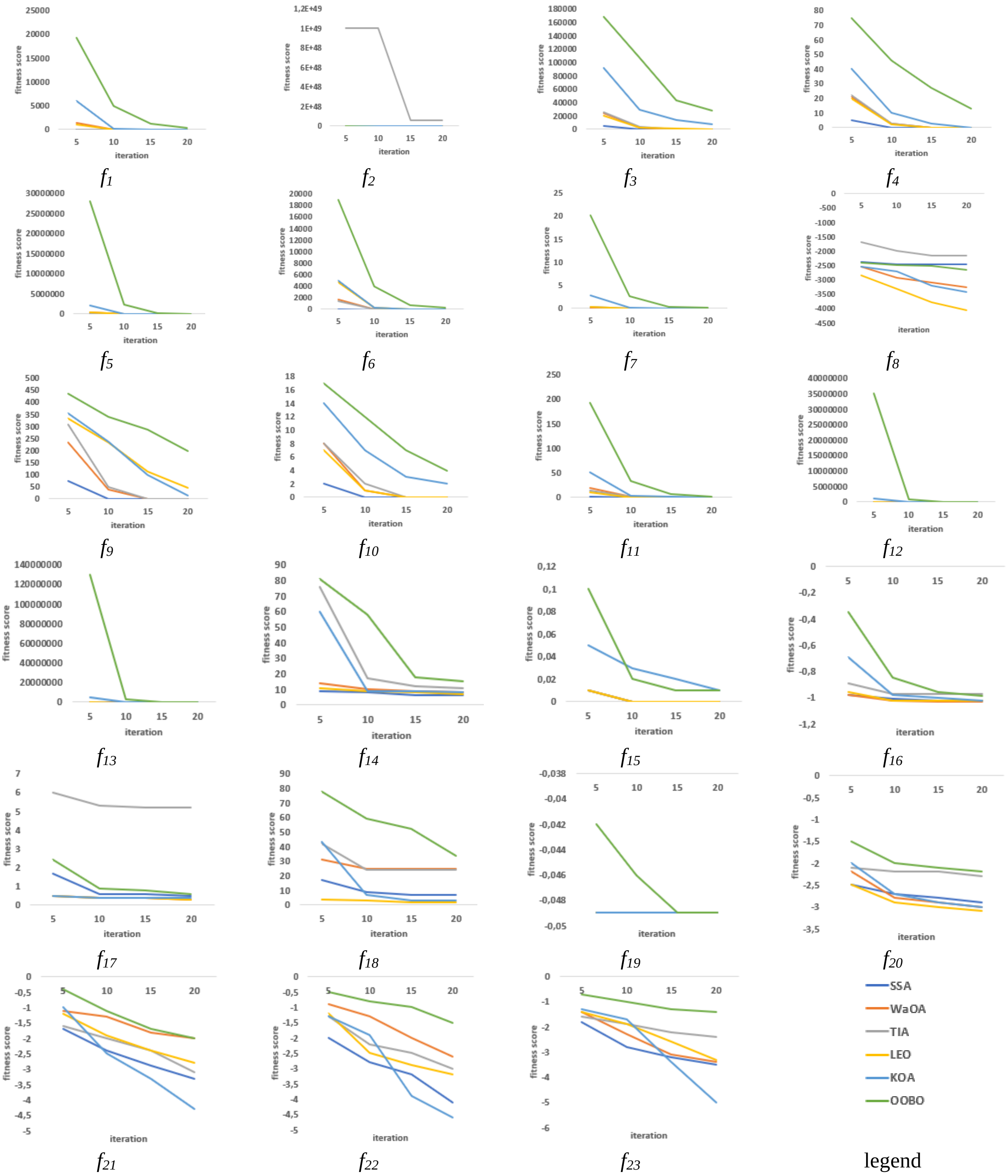

Figure. 1 Convergence curve

Notably, SSA secures the second-best position in solving f_5 and the third-best in handling f_6. Furthermore, SSA demonstrates the capability to identify the global optimal solution for two functions (f_1 and f_2). It is noteworthy that the performance variation among contenders in addressing high-dimensional unimodal (HDU) functions is substantial. However, it is reassuring that the performance gap between SSA and the best-performing algorithm, when SSA is not the top performer, remains comparatively narrow. This suggests that SSA maintains competitiveness even in scenarios where it may not outperform others.

Table 5 shows that SSA maintains its competitiveness in addressing high-dimensional multimodal functions. SSA excels as the best performer, successfully identifying the global optimal solution in two functions (f_9 and f_{10}). Additionally, SSA secures the second-best position in f_{13}, the third-best in f_{11}, and the fifth-best in both f_8

Received: January 29, 2024. Revised: March 11, 2024.

Table 3. A detailed description of the set of 23 functions

No	Function	Model	Dim	Space	Target
1	Sphere	$\sum_{i=1}^{d} x_i^2$	40	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$	40	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	40	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i\rvert, 1 \le i \le d\}$	40	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1}(100(x_{i+1}+x_i^2)^2+(x_i-1)^2)$	40	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1}(x_i+0.5)^2$	40	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\ [0,1]$	40	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	40	[-500, 500]	-1.2569x10^4
9	Ratsrigin	$10d+\sum_{i=1}^{d}\left(x_i^2-10cos(2\pi x_i)\right)$	40	[-5.12, 5.12]	0
10	Ackley	$-20\cdot exp\left(-0.2\cdot\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i{}^2}\right)-exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos 2\pi x_i\right)+20+exp(1)$	40	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2-\prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	40	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}\left((y_i-1)^2(1+10\,sin^2(\pi y_{i+1}))\right)+(y_d-1)^2\right\}+\sum_{i=1}^{d} u(x_i,10,100,4)$	40	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1)+\sum_{i=1}^{d-1}\left((x_i-1)^2(1+sin^2(3\pi x_{i+1}))\right)+(x_d-1)^2(1+sin^2(2\pi x_d))\right\}+\sum_{i=1}^{d} u(x_i,5,100,4)$	40	[-50, 50]	0
14	Shekel Foxholes	$\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
15	Kowalik	$\sum_{i=1}^{11}\left(a_i-\frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right)^2$	4	[-5, 5]	0.0003
16	Six Hump Camel	$4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]	-1.0316
17	Branin	$\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$	2	[-5, 5]	0.398
18	Goldstein-Price	$(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)).(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$	2	[-2, 2]	3
19	Hartman 3	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j-p_{ij})^2\right)\right)\right)$	3	[1, 3]	-3.86
20	Hartman 6	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j-p_{ij})^2\right)\right)\right)$	6	[0, 1]	-3.32
21	Shekel 5	$-\sum_{i=1}^{5}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.1532
22	Shekel 7	$-\sum_{i=1}^{7}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.4028
23	Shekel 10	$-\sum_{i=1}^{10}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.5363

Table 4. Fitness score comparison in solving high-dimension unimodal functions

F	Parameter	OOBO [24]	KOA [10]	LEO [22]	TIA [25]	WaOA [14]	SSA
1	mean	1.5906×10^2	0.4278	0.0005	0.0026	0.0027	0.0000
	std deviation	1.0811×10^2	0.2828	0.0006	0.0012	0.0020	0.0000
	mean rank	6	5	2	3	4	1
2	mean	0.0293	0.0000	0.0000	2.7875×10^{40}	0.0000	0.0000
	std deviation	0.1309	0.0000	0.0000	1.2466×10^{41}	0.0000	0.0000
	mean rank	5	1	1	6	1	1
3	mean	2.9615×10^4	7.0976×10^3	1.4671×10^2	1.2824×10^2	1.2287×10^2	1.1979
	std deviation	1.6145×10^4	6.6043×10^3	1.8780×10^2	2.7856×10^2	2.2463×10^2	2.5564
	mean rank	6	5	4	3	2	1
4	mean	2.9574×10^1	1.2553	0.0468	0.0747	0.1113	0.0002
	std deviation	3.0435×10^1	0.6516	0.0270	0.0231	0.0841	0.0001
	mean rank	6	5	2	3	4	1
5	mean	1.0552×10^4	4.6218×10^1	3.8957×10^1	3.8939×10^1	3.8981×10^1	3.8944×10^1
	std deviation	1.3173×10^4	7.6122	0.0238	0.0557	0.1022	0.0275
	mean rank	6	5	3	1	4	2
6	mean	2.1907×10^2	8.9857	9.2202	7.1436	8.1679	8.3415
	std deviation	1.7580×10^2	0.7629	0.8336	0.5768	0.4009	0.4114
	mean rank	6	4	5	1	2	3
7	mean	0.1041	0.0451	0.0186	0.0282	0.0209	0.0103
	std deviation	0.0491	0.0309	0.0092	0.0227	0.0109	0.0074
	mean rank	6	5	2	4	3	1

Table 5. Fitness score comparison in solving high-dimension multimodal functions

F	Parameter	OOBO [24]	KOA [10]	LEO [22]	TIA [25]	WaOA [14]	SSA
8	mean	-2.6930×10^3	-3.3212×10^3	-3.8118×10^3	-1.8465×10^3	-3.4249×10^3	-2.4801×10^3
	std deviation	3.8423×10^2	5.1787×10^2	5.6328×10^2	3.3977×10^2	6.1347×10^2	5.1069×10^2
	mean rank	4	3	1	6	2	5
9	mean	2.4260×10^2	1.7896×10^1	2.4827×10^1	0.0213	0.3565	0.0000
	std deviation	5.3199×10^1	3.3120×10^1	5.8181×10^1	0.0282	1.0715	0.0000
	mean rank	6	4	5	2	3	1
10	mean	4.0124	0.2649	0.0033	0.0102	0.9200	0.0000
	std deviation	1.3043	0.3028	0.0022	0.0022	4.1594	0.0000
	mean rank	6	4	2	3	5	1
11	mean	2.7589	0.1618	0.0019	0.0013	0.0204	0.0145
	std deviation	1.5013	0.1789	0.0046	0.0022	0.0458	0.4797
	mean rank	6	5	2	1	4	3
12	mean	3.6814	1.0647	1.0026	0.8061	0.8894	1.0716
	std deviation	1.2777	0.1592	0.1471	0.1230	0.1787	0.1378
	mean rank	6	4	3	1	2	5
13	mean	4.1189×10^1	3.5582	3.1121	3.1389	1.9832	3.0391
	std deviation	8.8187×10^1	0.1858	0.0683	0.1417	0.4465	0.0708
	mean rank	6	5	3	4	1	2

and f_{12}. Like the observations in the first set of functions, the performance gap between SSA and the best-performing algorithm in f_8, f_{11}, f_{12}, and f_{13} remains narrow. This indicates that SSA remains competitive even when it does not secure the top position in these specific functions.

Table 6 shows the fierce competition among contenders in solving the fixed-dimension multimodal functions. In general, the performance gap between the best performer and the worst performer is very narrow. SSA becomes the best performer in solving f_{19}. SSA becomes the second best in three functions (f_{14}, f_{21}, f_{22}), third best in four functions (f_{16}, f_{18}, f_{20}, f_{23}), fourth best in f_{15}, and fifth best in f_{17}.

Table 7 shows that SSA is very competitive or superior enough compared to its contenders. SSA is better than OOBO, KOA, LEO, TIA, and WaOA in 21, 13, 11, 16, and 14 functions. Among the group of functions, SSA is superior in solving high-

Table 6. Fitness score comparison in solving fixed dimension multimodal functions

F	Parameter	OOBO [24]	KOA [10]	LEO [22]	TIA [25]	WaOA [14]	SSA
14	mean	$1.6547x10^1$	9.9056	6.6511	$1.0213x10^1$	8.6932	6.6793
	std deviation	$2.0361x10^1$	3.7968	3.9255	3.5836	4.0645	3.0860
	mean rank	6	4	1	5	3	2
15	mean	0.0165	0.0058	0.0039	0.0027	0.0014	0.0042
	std deviation	0.0134	0.0077	0.0007	0.0052	0.0012	0.0134
	mean rank	6	5	3	2	1	4
16	mean	-1.0126	-1.0235	-1.0298	-1.0043	-1.0308	-1.0294
	std deviation	0.0296	0.0120	0.0047	0.0486	0.0022	0.0042
	mean rank	5	4	2	6	1	3
17	mean	0.6231	0.4034	0.3995	4.3802	0.3986	0.4450
	std deviation	0.6839	0.0071	0.0032	5.8361	0.0010	0.0818
	mean rank	6	3	2	4	1	5
18	mean	$1.2674x10^1$	4.3255	6.3861	$1.9642x10^1$	$3.0419x10^1$	6.7833
	std deviation	$1.2397x10^1$	5.5036	$1.6219x10^1$	$2.3282x10^1$	$3.0567x10^1$	9.2027
	mean rank	4	1	2	5	6	3
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.3071	-3.0536	-3.1582	-2.3200	-2.9928	-3.0116
	std deviation	0.4629	0.1264	0.1067	0.4982	0.1939	0.3038
	mean rank	6	2	1	5	4	3
21	mean	-1.1377	-4.2817	-2.9922	-2.2747	-2.1969	-4.1468
	std deviation	0.6270	1.5344	1.3533	1.4584	0.9962	1.7498
	mean rank	6	1	3	4	5	2
22	mean	-1.3504	-4.2240	-3.3315	-2.4322	-3.0662	-3.9907
	std deviation	0.4858	1.7837	1.2539	1.5683	1.1347	2.3736
	mean rank	6	1	3	5	4	2
23	mean	-1.5473	-3.9090	-3.2812	-2.5250	-3.0662	-3.2610
	std deviation	0.9755	1.3105	0.9552	1.1793	1.1347	0.9356
	mean rank	6	1	2	5	4	3

Table 7. Group-based superiority of SSA

Group	Number of Functions Where SSA is Better				
	OOBO [24]	KOA [10]	LEO [22]	TIA [25]	WaOA [14]
1	7	6	6	5	5
2	5	4	3	4	3
3	9	3	2	7	6
Total	21	13	11	16	14

dimensional unimodal functions. Meanwhile, SSA is less superior in solving fixed-dimension multimodal functions. Overall, SSA is superior to OOBO in all groups of functions. On the other hand, LEO is the most difficult contender to beat although SSA is slightly superior to LEO.

In the second assessment, SSA is challenged to solve the economic load dispatch problem where the system consists of 13 generators. These generators can be split into three groups {g_1 to g_3, g_4 to g_9, g_{10} to g_{13}}. The total demand is 1800 MW. The detailed description of the system is presented in Table 8 based on [1]. Table 8 presents the constants for the cost and the power range of each generator. The result is presented in Table 9.

Table 9 shows that the performance gap between the best performer and the worst performer in solving the economic load dispatch problem is very narrow.

Table 8. Constants of 13 Generators

Gen.	α	β	γ	p_{min} (MW)	p_{max} (MW)
1	550	8.1	0.00028	0	680
2	309	8.1	0.00056	0	360
3	307	8.1	0.00056	0	360
4	240	7.74	0.00324	60	180
5	240	7.74	0.00324	60	180
6	240	7.74	0.00324	60	180
7	240	7.74	0.00324	60	180
8	240	7.74	0.00324	60	180
9	240	7.74	0.00324	60	180
10	126	8.6	0.00284	40	120
11	126	8.6	0.00284	40	120
12	126	8.6	0.00284	55	120
13	126	8.6	0.00284	55	120

Table 9. Total fuel cost

No	Metaheuristic	Total Fuel Cost (USD/hour)
1	OOBO [24]	17,962
2	KOA [10]	17,941
3	LEO [22]	17,939
4	TIA [25]	17,951
5	WaOA [14]	17,940
6	SSA	17,942

The range is less than one percent of the average total fuel cost. LEO becomes the best performer while OOBO becomes the worst performer. SSA is in the fourth rank.

5. Discussion

In the broader context, the outcomes from both assessments underscore the intense competition among contemporary swarm-based metaheuristics, particularly emphasizing the rivalry between the proposed Stochastic Shaking Algorithm (SSA) and the five competing methods. SSA emerges as the top performer in eight functions, with five of them belonging to the high-dimensional unimodal category. Notably, SSA exclusively claims the top position in six functions. In contrast, the Language Education Optimization (LEO), identified as the most formidable metaheuristic to surpass, secures the sole best-performer status in only three functions. The performance disparity between the best and worst performers is notably wide in addressing high-dimensional unimodal functions. Conversely, this gap narrows considerably when tackling fixed-dimension multimodal functions. These findings substantiate the principles of the No Free Lunch (NFL) theory.

Fiercer competition occurs in solving the economic load dispatch problem where the range between the best and worst results is very narrow compared to the average result. By drawing back to the previous study [1], fierce competition also occurs. By comparing the result in this paper and in [1], all six metaheuristics in this current experiment perform better than the metaheuristics in the previous work. The worst result in this current work is provided by OOBO where the total fuel cost is 17,962 USD/hour. On the other hand, the best result in the previous work is 17,963 USD/hour [1].

The very narrow performance gap in the economic load dispatch problem can be analysed by investigating the cost function, especially the constants (α, β, γ) and the power range. In general, this cost function can be split into three parts: the constant part, the linear part, and the quadratic part. Based on the relation between the constants and the range, it is shown that the linear part has the most significant portion in generating cost while the quadratic part has the least significant portion because the value of γ is very small compared to the value of β. The multiplication of the quadratic power range and γ is relatively very small compared to the multiplication of β and the median value of the power range of the related generator. It means that the quadratic part in the cost function can be ignored. Then, the cost function is constructed mainly by the constant and linear parts. This circumstance makes this economic load dispatch problem easy to solve. In a more common perspective, the constant part can be seen as the fixed cost while the linear part can be seen as the variable cost.

Further investigation can be conducted by checking the range of values of α and β. In general, the range of β is small among 13 generators. The smallest value is 7.74 and the highest value is 8.6. This condition makes the gradient of the cost function almost similar. Moreover, the range of α in eight generators (g2 to g9) is also small compared to the value of α. In general, the first generator (g1) becomes the key determinant of the total cost as it has the highest α and highest maximum power.

The main limitation in this work can be drawn back to the various use cases, whether they are theoretical or practical. In this work, the set of 23 classic functions is chosen as the theoretical use case. On the other hand, there are other sets of theoretical use cases, for example the CEC series. The classic functions can be seen as the unconstrained problems while the CEC series are the constrained problems. Both classic functions and CEC series have limited solution space in each of the dimension. But each function in CEC series is also enriched with other constrains whether equality or inequality ones. In the real world, most of optimization problems are constrained problems. Although the constrained theoretical or standard functions like CEC series is not used in this work, this gap is filled with the economic load dispatch problem that represents the constrained practical problem. As previously explained, in ELD problem, there is equality constraint where the total power should meet the demand. In the practical area, there are several standard ELD problem like IEEE bus system, such as IEEE-30, IEEE-57, or IEEE-118 bus system.

Moreover, it is impossible to accommodate to many standard assessment use cases in a single paper. Many recent studies proposing new metaheuristics still use the set of 23 classic functions as the sole standard use case. This scenario can be found in many studies that propose Archery Algorithm (AA) [33], Coronavirus Herd Immunity Optimizer (CHIO) [34], KMA [29], Golden Jackal Optimization (GJO) [35], and so on. In general, the main objective of standard assessment is to evaluate the exploitation and exploration capabilities of the proposed metaheuristic. In this context, the set of 23 classic functions is enough and acceptable.

These findings can be used as a baseline and motivation for further studies. In the economic dispatch problems, other practical use cases can be used for more comprehensive analysis, for example, the Java-Bali electricity system in Indonesia. Besides the single objective of minimizing total fuel cost, other objective functions can be proposed, for example by introducing the environmental cost. Moreover, it will be challenging to implement SSA in other power systems with more various resources, especially the greener and renewable ones, such as wind, ocean wave, hydro, and so on, where the cost structure is different.

6. Conclusion

This research paper has introduced and thoroughly examined the stochastic shaking algorithm (SSA), covering its conceptualization, formulation, and assessment. The evaluation results indicate that SSA excels in addressing the set of 23 classic functions and exhibits competitiveness in tackling the economic load dispatch problem. In comparison to alternative metaheuristics—OOBO, KOA, LEO, TIA, and WaOA—SSA outperforms them in 21, 13, 11, 16, and 14 functions, respectively, out of the total 23 functions. In the theoretical assessment, LEO emerges as the most challenging contender to surpass, while OOBO is identified as the least challenging. Meanwhile, the second assessment witnesses the fierce competition, with a remarkably narrow performance gap between the best and worst performers.

Numerous avenues for future studies are identified. Firstly, a more extensive analysis could be undertaken by subjecting SSA to address additional theoretical problems, such as those presented in the IEEE CEC series. Secondly, SSA could be applied to solve various optimization problems within the energy sector, including power flow problems or battery storage problems. Lastly, further exploration can involve combining or modifying SSA to enhance its overall performance.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization: Kusuma; methodology: Kusuma, software: Kusuma, formal analysis: Kusuma and Prasasti; writing-original paper draft: Kusuma; writing-review and editing: Prasasti; supervision: Prasasti; project administration: Kusuma; funding acquisition: Kusuma.

Acknowledgments

This work is funded and supported by Telkom University, Indonesia.

References

[1] 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.

[2] K. M. D. Puspitasari, J. Raharjo, A. S. Sastrosubroto, and B. Rahmat, "Generator Scheduling Optimization Involving Emission to Determine Emission Reduction Costs", *IJE Transactions B: Applications*, Vol. 35, No. 8, pp. 1468-1478, 2022.

[3] N. Singh, T. Chakrabarti, P. Chakrabarti, M. Margala, A. Gupta, S. P. Praveen, S. B. Krishnan, and B. Unhelkar, "Novel Heuristic Optimization Technique to Solve Economic Load Dispatch and Economic Emission Load Dispatch Problems", *Electronics*, Vol. 12, No. 2921, 2023.

[4] 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. 6225, 2020.

[5] 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. 6933, 2023.

[6] S. Phommixay, M. L. Doumbia, and D. L. St-Pierre, "Review on The Cost Optimization of Microgrids via Particle Swarm Optimization", *International Journal of Energy and Environmental Engineering*, Vol. 11, pp. 73-89, 2020.

[7] V. K. Kamboj, C. L. Kumari, S. K. Bath, D. Prashar, M. Rashid, S. S. Alshamrani, and A. S. AlGhamdi, "A Cost-Effective Solution for Non-Convex Economic Load Dispatch Problems in

Power Systems Using Slime Mould Algorithm”, *Sustainability*, Vol. 14, No. 2586, 2022.

[8] 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, 2021, doi: 10.22266/ijies2021.1231.59.

[9] H. Sebaa, K. Khettab, and N. Chopra, “Solving Unit Commitment Problem for Microgrid Power Network Including Wind and Solar Sources Using Modified MVO Algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 11, pp. 1-677, 2022, doi: 10.22266/ijies2022.0630.01.

[10] 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. 470, pp. 1-54, 2023.

[11] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. M. Malik, and G. Dhiman, “Lyrebird Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems”, *Biomimetics*, Vol. 8, No. 507, pp. 1-62, 2023.

[12] P. D. Kusuma and M. Kallista, “Stochastic Komodo Algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[13] M. Dehghani, P. Trojovsky, and O. P. Malik, “Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems”, *Biomimetics*, Vol. 8, No. 121, pp. 1-60, 2023.

[14] P. Trojovsky and M. Dehghani, “A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Walruses Behavior”, *Scientific Reports*, Vol. 13, No. 8775, pp. 1-32, 2023.

[15] M. Dehghani, Z. Montazeri, E. Trojovska, and P. Trojovsky, “Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems”, *Knowledge-Based Systems*, Vol. 259, ID. 110011, pp. 1-43, 2023.

[16] 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, ID. 108457, pp. 1-29, 2022.

[17] 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, 2020, doi: 10.22266/ijies2020.1031.11.

[18] M. Dehghani, S. Hubalovsky, and P. Trojovsky, “Tasmanian Devil Optimization: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm”, *IEEE Access*, Vol. 10, pp. 19599-19620, 2022.

[19] M. Dehghani, S. Hubalovsky, and P. Trojovsky, “Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems”, *IEEE Access*, pp. 162059-162080, 2021.

[20] M. Dehghani and P. Trojovsky, “Osprey Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems”, *Frontiers in Mechanical Engineering*, Vol. 8, ID. 1126450, pp. 1-43, 2023.

[21] P. Trojovsky and M. Dehghani, “Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems”, *Computer Modeling in Engineering & Sciences*, Vol. 137, No. 2, pp. 1695-1730, 2023.

[22] P. Trojovsky, M. Dehghani, E. Trojovska, and E. Milkova, “Language Education Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems”, *Computer Modeling in Engineering and Sciences*, Vol. 136, No. 2, pp. 1527-1573, 2023.

[23] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, "Mother Optimization Algorithm: A New Human-Based Metaheuristic Approach for Solving Engineering Optimization", *Scientific Reports*, Vol. 13, No. 10312, pp. 1-26, 2023.

[24] M. Dehghani, E. Trojovska, P. Trojovsky, and O. P. Malik, “OOBO: A New Metaheuristic Algorithm for Solving Optimization Problems”, *Biomimetics*, Vol. 8, No. 468, pp. 1-48, 2023.

[25] 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, 2023, doi: 10.22266/ijies2023.0228.20.

[26] 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, 2023, doi: 10.22266/ijies2023.0630.19.

[27] P. Trojovsky and M. Dehghani, “Subtraction-Average-Based Optimizer: A New Swarm-Inspired Metaheuristic Algorithm for Solving Optimization Problems”, *Biomimetics*, Vol. 8, No. 149, pp. 1-42, 2023.

[28] M. Ghasemi, A. Rahimnejad, E. Akbari, R. V. Rao, P. Trojovský, E. Trojovská, and S. A. Gadsden, “A New Metaphor-less Simple Algorithm Based on Rao Algorithms: A Fully Informed Search Algorithm (FISA)”, *PeerJ Computer Science*, Vol. 9, No. e1431, pp. 1-24, 2023.

[29] Suyanto, A. A. Ariyanto, and A. F. Ariyanto, “Komodo Mlipir Algorithm”, *Applied Soft Computing*, Vol. 114, Art. No. 108043, 2022.

[30] M. S. Braik, "Chameleon Swarm Algorithm: A Bio-inspired Optimizer for Solving Engineering Design Problems", *Expert Systems with Applications*, Vol. 174, No. 114685, pp. 1-25, 2021.

[31] F. Rezaei, H. R. Safavi, M. A. Elaziz, and S. Mirjalili, “GMO: Geometric Mean Optimizer for Solving Engineering Problems”, *Soft Computing*, Vol. 27, pp. 10571-10606, 2023.

[32] M. Ghasemi, M. Zare, A. Zahedi, M. -A. Akbari, S. Mirjalili, and L. Abualigah, “Geyser Inspired Algorithm: A New Geological-inspired Meta-heuristic for Real-parameter and Constrained Engineering Optimization”, *Journal of Bionic Engineering*, 2023, online first.

[33] F. A. Zeidabadi, M. Dehghani, P. Trojovsky, S. Hubalovsky, V. Leiva, and G. Dhiman, “Archery Algorithm: A Novel Optimization Algorithm for Solving Optimization Problems”, *Computers, Materials & Continua*, Vol. 72, No. 1, pp. 399-416, 2022.

[34] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. A. Doush, “Coronavirus Herd Immunity Optimizer (CHIO)”, *Neural Computing and Applications*, Vol. 33, pp. 5011-5042, 2021.

[35] N. Chopra and M. M. Ansari, “Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications”, *Expert Systems with Applications*, Vol. 198, ID. 116924, pp. 1-15, 2022.