Received: May 13, 2024. Revised: June 6, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A New Metaheuristic Called Stay-Jump Optimizer and Its Utilization on Economic Emission Dispatch Problem in Java-Bali Power Grid

Purba Daru Kusuma[1]*

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

Abstract: In the recent years, there are a lot of swarm-based stochastic optimization techniques i.e., metaheuristics were introduced. Most of these techniques were tested to solve the sets of theoretical functions. Some of them were enriched with practical tests where the common use cases are the mechanical engineering designs. On the other hand, the similar studies that utilized the optimization in power system are difficult to find. Moreover, the environmental issues become major considerations in engineering field. Based on this evidence, this paper constructs a new swarm-based optimization technique called stay-jump optimizer (SJO). The equal size swarm split is applied in the beginning of the process. Then, two directed searches toward the highest quality sub-swam members and two randomly selected higher quality sub swarm members are employed. The performance investigation is performed by employing SJO to find the optimal solution of 23 classic functions and the economic emission dispatch (EED) problem. The use-case for EED is the Java-Bali power grid system in Indonesia that consists of six steam power plants and two hydro-electric power plants. Five new optimization techniques including addax optimization algorithm (AOA), dollmaker optimization algorithm (DOA), giant armadillo optimization algorithm (GAO), zebra optimization algorithm (ZOA), and total interaction algorithm (TIA). The result shows that SJO is superior to its opponents as it is better than AOA, DOA, GAO, ZOA, and TIA in 17, 17, 16, 19, and 14 functions out of 23 functions respectively. SJO also becomes the best in the EED problem.

Keywords: Metaheuristic, Swam intelligence, Economic emission dispatch, Power grid.

1. Introduction

Environmental issues become one of major considerations in engineering field, especially in the optimization studies. Many optimization studies have included the environmental issue as one of their considerations besides the technical and economic aspects. Some issues that are related to the environmental aspects are renewable energy [1], carbon emission [2], waste [3], and so on.

One of the implementations of environmental issues in the power sector is the economic emission dispatch (EED) problem. EED is a branch of optimization in power systems whose nature is a multi-objective optimization problem where the objective is minimizing the operational/fuel cost and the emission reduction cost [4]. EED is a derivative of the economic load dispatch (ELD) problem where in its basic form, the objective is minimizing the operational cost [5]. As with many optimization studies, many studies in EED or ELD problems utilized metaheuristics as their optimization tools.

In recent days, there are a lot of new metaheuristics that were introduced in recent years. Many of them still utilized swarm intelligence as a baseline. Some of these metaheuristics were first introduced in 2024, such as addax optimization algorithm (AOA) [6], dollmaker optimization algorithm (DOA) [7], apiary organizational based optimization algorithm (AOOA) [8], focus and shake algorithm (FSA) [9], pufferfish optimization algorithm (POA) [10], Nizar optimization algorithm (NOA) [11], and so on. Some of others are first introduced in 2023, such as total interaction algorithm (TIA) [12], swarm magnetic optimizer (SMO) [13], four directed search algorithms (FDSA) [14], giant armadillo optimization (GAO) [15], green anaconda optimization (GAO) [16], golf optimization algorithm (GOA) [17], fully informed search algorithm (FISA) [18], walrus optimization algorithm (WaOA) [19], deep sleep optimizer (DSO) [20], and so on. Some others were first introduced in 2022, such as archery algorithm (AA) [21], zebra optimization algorithm (ZOA) [22], white shark optimization (WSO) [23], Siberian tiger optimization (STO) [24], prairie dog optimization (PDO) [25], and so on.

Unfortunately, the EED problem is not a popular problem that is employed in studies that propose new metaheuristics. In other words, studies that introduce a new metaheuristic and utilize the EED problem are hard to find. Most of these studies choose standard sets of functions like 23 classic functions [12] or IEEE standard functions [23] for the use cases. Some studies were also enriched with standard mechanical engineering design problems including the speed reducer, welded beam, tension/compression, and pressure vessel [7].

On the other hand, many studies that tried to solve the economic dispatch problem still employed old metaheuristics. These old metaheuristics were employed in their original form or have been improvised. Some of them are chaotic social group optimization [26], particle swarm optimization [27], simulated annealing [28], multi-verse optimizer (MVO) [29], and so on. Only a few of them employed new metaheuristics, such as technique of narrowing down area (ToNDA) [5], squirrel search optimizer (SSO) [30], and so on.

This work aims to introduce a new stochastic optimization technique called stay-jump optimizer (SJO). SJO utilizes the swarm intelligence approach so that it consists of a set of swarm members. Then, this swarm is split into two equal size sub swarms. SJO employs two directed searches for all swarm members. The first search is the motion toward the highest quality sub swarm members. The second search is the motion toward two randomly chosen higher quality sub swarm members. Term stay means the motion toward the reference from its own sub swarm while term jump means the motion toward the reference from another sub swarm.

There are two use cases chosen for the assessment. The first use case is the set of 23 classic functions representing the unconstrained problem. The second use case is the EED problem of the Java-Bali power system in Indonesia representing the constrained problem.

Based on this explanation, this paper contributes scientifically on several aspects as follows.

- This paper introduces a new metaphor-free stochastic optimization technique called stay-jump optimizer (SJO).
- SJO employs two equal size and non-sorted sub swarms.
- SJO introduces a novel searching technique where the swarm members interact with the entity within their sub swarm (stay) and entity from another sub swarm (jump).
- SJO is tested to find the optimal solution of both unconstrained and constrained problems.
- The EED problem is chosen as a contribution to solve the economic and environmental considerations.
- The Java-Bali power grid system was chosen as an alternative as this system is rare to find in many optimization studies.

The organization of the remainder of the paper is as follows. Section two consists of the review regarding the recent studies in the development of metaheuristics and the recent studies in economic dispatch problems. Section three provides the description of the model, including the model of the proposed technique and the model of the EED problem. Section four exhibits the assessment including the scenario and the result. Section five provides a comprehensive discussion regarding the result, findings, limitations, and computational complexity. Section six summarizes the concluding remarks and possibility for future studies.

2. Related works

In general, any studies in economic dispatch (ED) can be split into three parts. The first part is the model. In this model, the ED problem can be seen as ELD, EED, or unit commitment (UC) problem. Then, through this model, it can be seen whether the optimization is a single objective or multi-objective one. This model also provides insight whether other complementary parameters like ramp rate [28], valve point effect [5] or power loss [5] are considered or not. The second part is the use case. In general, the use case consists of the generating units that are set in the system including the number of generating units [28], the constants [28], power range [28], and so on. The use case also describes the power demand whether this system is a single demand, or multiple demands based on certain period. The third part is the optimization technique that is employed in this study. Almost all ED studies employ metaheuristic as optimization technique. This technique can be the existing one, expanded one, or new one. A summary of several recent ED studies is provided in Table 1.

Received: May 13, 2024. Revised: June 6, 2024.

Table 1. Summary of recent studies in economic dispatch problem

No	Reference	Type	Use Case	Technique
1	[26]	ELD	four systems (10 units, 20 units, 30 units, and 40 units)	Chaotic social group optimizer (improved version of the social group optimizer)
2	[27]	EED/ELD	two systems (13 units and 15 units)	Modified PSO (improved version of PSO)
3	[28]	EED	single system (8 units)	Simulated annealing (old metaheuristic)
4	[29]	UC	single system (10 units)	Parallel mirror technique-multi-verse optimizer (improved version of multi-verse optimizer)
5	[5]	ELD	four systems (2 units, 10 units, 13 units, and 40 units)	ToNDA (new metaheuristic)
6	[30]	ELD	four systems (7 units, 10 units, 20 units, and 28 units)	Squirrel search optimizer (new metaheuristic)
7	[31]	ELD	single system (10 units)	Teaching-learning based optimization (old metaheuristic)
8	[32]	ED	single system (IEEE 30-bus with 6 units)	Modified firefly algorithm (improved version of firefly algorithm)
9	[33]	UC	single system (IEEE 39 bus with 10 units)	Binary hybrid grey wolf optimizer (improved version of grey wolf optimizer)
10	[34]	EED	single system (6 units)	Quantum behaved artificial bee colony (improved version of artificial bee colony)
11	[35]	ELD	five systems (6, 10, 13, 40, and 140 units)	Golden jackal optimization (original form of the golden jackal optimization)

Table 2. Summary of recent studies in developing of metaheuristics

No	Reference	Metaheuristic	Use Case	Swarm Split
1	[6]	AOA	4 mechanical engineering designs	no
2	[12]	TIA	23 classic functions	no
3	[7]	DOA	23 classic functions and 4 mechanical engineering designs	no
4	[23]	WOA	CEC 2017 functions, CEC 2011 functions	no
5	[20]	DSO	23 classic functions, knapsack problem, travelling salesman problem, three mechanical engineering designs (I-beam, cantilever, and wind power and turbulence intensity)	no
6	[11]	NOA	60 unconstrained functions, 4 mechanical engineering designs	no
7	[8]	AOOA	23 classic functions	no
8	[22]	ZOA	23 classic functions, CEC 2015, CEC 2027, 4 mechanical engineering designs	partial
9	[16]	GAO	CEC 2011, CEC 2017, 4 mechanical engineering designs	no
10	this work	SJO	23 classic functions, Java-Bali power grid with 8 units	yes

On the other hand, many recent metaheuristics were developed based on swarm intelligence. Swarm intelligence is a derivative of the population-based metaheuristic but with specific nature where all members perform search actively and autonomously [36]. This approach is different from the evolution-based metaheuristic that employs a central command to determine which members become the parents to generate new members and which members should be eliminated from the population due to their poor quality [37]. In swarm-based metaheuristic, each member performs autonomously without any deliberate instruction from other or entities [36]. But each member performs a search based on certain references, such as the highest quality members, a randomly picked higher quality member, a randomly picked other members, and so on.

It is common for studies developing new metaheuristic to perform test to investigate the performance of their proposed technique. The standard unconstrained functions like 23 classic functions or CEC series become a favorite use case. These standard functions become a mandatory test for these studies. Meanwhile, some studies were enriched with tests employing mechanical engineering design problem representing the constrained problem. Most of these metaheuristics do not employ the swarm split while only a few of them employ this strategy. The summary of several new studies in metaheuristics including the use case and the implementation of the swarm split is provided in Table 2.

Based on this explanation, there is an opportunity to combine the development of new metaheuristic and its implementation to solve the economic dispatch problem. As mentioned previously, the ToNDA [5] and SSO [30] are new metaheuristics which their evaluation was the ELD only. There are not any standard unconstrained functions are employed to test both metaheuristics. On the other hand, all new existing metaheuristics which are exhibited in Table 2 do not take the economic dispatch problem as the constrained practical optimization problem. All of them employed standard unconstrained functions while mechanical engineering design problems are favorite option as the constrained and practical problems. Moreover, the swarm split is also not popular in many new swarm-based metaheuristics. This gap also becomes the opportunity to develop new metaheuristic which employs swarm split.

3. Model

3.1 The model of stay-jump optimizer

The stay-jump optimizer (SJO) is developed based on the concept that a swarm is split into two sub swarms. These sub swarms are equal in size if the swarm size is even. This swarm is split in the beginning of the optimization process and remains static until the optimization process ends. The swarm split is performed based on the index of the swarm member and not based on the quality of the members. The members whose index is even will be collected into the first sub swarm and the members whose index is odd will be collected into the second sub swarms.

In SJO, each swarm member performs two guided searches. The first search is the motion toward the highest quality sub swarm member. Meanwhile, the second search is the motion toward a randomly chosen member from the pool that consists of all higher quality sub swarm members plus the highest quality sub swarm members. This pool is called the superior pool.

Table 3. Annotations in SJO model

Notations	Description
b_l, b_u	lower and upper boundaries
f	objective function
i	index for swarm member
j	index for dimension
r_{u1}	floating point uniform random [0,1]
r_{u2}	integer uniform random [1,2]
r_{u3}	uniform random for population
s	swam member
S	swarm
S_s	sub swarm
S_f	the highest quality sub swarm member
sc	solution candidate
S_p	a pool consists of higher quality sub swarm member including the highest quality sub swarm member
s_{sel}	a randomly picked member
t	iteration
t_m	maximum iteration

The term stay-jump comes from the concept that a swarm member interacts not only with the reference within its own sub swarm but also another sub swarm. In the stay motion, the swarm member interacts with its own highest quality sub swarm member and a randomly selected member that is picked from its own superior pool. On the other hand, in the jump motion, the swarm member interacts with the highest quality sub swarm member and a randomly selected member from the superior pool from another sub swarm. Each motion generates a solution candidate. Two solution candidates that are generated from the stay and jump motions in the same stage are then compared to each other. The solution candidate whose quality is better then becomes the final solution candidate for the related stage. Then, if this solution candidate is better than the recent solution, this solution candidate replaces the existing solution.

The formalization of the SJO is provided in Eqs. (1) to (18). The annotations used in this paper are presented in Table 3. Meanwhile, the algorithm of SJO is provided using algorithm 1.

Eqs. (1) and (2) present the swarm split mechanism. Eq. (1) is used to collect the swarm members whose index is even. Eq. (2) is used to collect the swarm members whose index is odd.

$$S_{s1} = \{s_i \in S \wedge i \bmod 2 = 0\} \quad (1)$$

$$S_{s2} = \{s_i \in S \wedge i \bmod 2 = 1\} \quad (2)$$

The initialization phase is formalized using Eqs. (3) to (5). In this phase, all swarm members are generated based on the uniform random within the space as provided in Eq. (3). Then, Eqs. (4) and (5) are used to update the highest quality sub swarm members where Eq. (4) is for the first sub swarm and Eq. (5) is for the second sub swarm.

$$s_{i,j} = b_{l,j} + r_{u1}(b_{u,j} - b_{l,j}) \quad (3)$$

$$s'_{f1} = \begin{cases} s_i, f(s_i) < f(s_{f1}) \wedge s_i \in S_{s1} \\ s_{f1}, else \end{cases} \quad (4)$$

$$s'_{f2} = \begin{cases} s_i, f(s_i) < f(s_{f2}) \wedge s_i \in S_{s2} \\ s_{f2}, else \end{cases} \quad (5)$$

The searching processes in the iteration phase are formalized using Eqs. (6) to (17). But the updating process of the highest quality sub swarm members is still used in the iteration phase each time a motion ends. The searching process in every iteration ends with the updating process of the highest quality swarm member which is performed by choosing the better one between two highest quality sub swarm members as formalized using Eq. (18).

The first search is formalized using Eqs. (6) to (9). Eqs. (6) and (7) formalize the motion toward the highest quality sub swarm members. Then, Eq. (8) formalizes the selection for the final solution candidate of the first search. In the end, Eq. (9) formalizes the updating process of the swarm member based on the solution candidate generated in the first search.

$$sc_{11,i,j} = s_{i,j} + r_{u1}(s_{f1,j} - r_{u2}s_{i,j}) \quad (6)$$

$$sc_{12,i,j} = s_{i,j} + r_{u1}(s_{f2,j} - r_{u2}s_{i,j}) \quad (7)$$

$$sc_1 = \begin{cases} sc_{11}, f(sc_{11}) < f(sc_{12}) \\ sc_{12}, else \end{cases} \quad (8)$$

$$s'_i = \begin{cases} sc_1, f(sc_1) < f(s_i) \\ s_i, else \end{cases} \quad (9)$$

The second search is formalized using Eqs. (10) to (17). Eqs. (10) and (18) are used to formalize the construction of the higher quality pools. Eqs. (12) and (13) are utilized to pick a member from each pool. Eqs. (14) and (15) are utilized to perform the motion and generate the solution candidates. Eq. (16) is utilized to select the higher quality solution candidate. Eq. (17) is utilized to update the swarm member based on the final solution candidate in the second search.

$$S_{p1,i} = \{s_k \in S_{s1} \wedge f(s_k) < f(s_i)\} \cup s_{f1} \quad (10)$$

$$S_{p2,i} = \{s_k \in S_{s2} \wedge f(s_k) < f(s_i)\} \cup s_{f2} \quad (11)$$

$$s_{sel1,i} = r_{u3}(S_{p1,i}) \quad (12)$$

$$s_{sel2,i} = r_{u3}(S_{p2,i}) \quad (13)$$

$$sc_{21,i,j} = s_{i,j} + r_{u1}(s_{sel1,i,j} - r_{u2}s_{i,j}) \quad (14)$$

$$sc_{22,i,j} = s_{i,j} + r_{u1}(s_{sel2,i,j} - r_{u2}s_{i,j}) \quad (15)$$

$$sc_2 = \begin{cases} sc_{21}, f(sc_{21}) < f(sc_{22}) \\ sc_{22}, else \end{cases} \quad (16)$$

$$s'_i = \begin{cases} sc_2, f(sc_2) < f(s_i) \\ s_i, else \end{cases} \quad (17)$$

$$s_f = \begin{cases} s_{f1}, f(s_{f1}) < f(s_{f2}) \\ s_{f2}, else \end{cases} \quad (18)$$

The algorithm of SJO is provided by algorithm 1. Line 2 shows the swarm splitting process. The initialization phase is presented from lines 3 to 6. The iteration phase is presented from lines 7 to 15. Line 16 shows the highest quality swarm member as the final solution.

algorithm 1: stay-jump optimizer	
1	begin
2	split into S_{s1} and S_{s2} using Eq. (1) and Eq. (2)
3	for all $s \in S$
4	initialize s using Eq. (3)
5	update s_{f1} and s_{f2} using Eq. (4) and Eq. (5)
6	end
7	for t=1 to t_m
8	for all swarm members
9	first search using Eq. (6) to Eq. (9)
10	update s_{f1} and s_{f2} using Eq. (4) and Eq. (5)
11	second search using Eq. (10) to Eq. (17)
12	update s_{f1} and s_{f2} using Eq. (4) and Eq. (5)
13	end
14	update s_f using Eq. (18)
15	end
16	return s_f
17	end

3.2 Economic emission dispatch problem

The economic emission dispatch (EED) problem is an enhancement or derivative of the economic load dispatch (ELD) problem. Different from the classic ELD whose objective is singular, which is minimizing the fuel cost or operational cost, EED is a multi-objective problem. It has two objectives which are minimizing the fuel cost and emission reduction cost [28].

The system consists of a set of generating units or generators that are connected to the power grid. Each generating unit operates within its power range. Then, the provided power from each generating unit will be collected as a cumulative or total power. This total power should meet the power demand which is known as a priory or in advance.

The power demand varies within a period. There are periods where the demand is high as they are called peak hours or periods. On the other hand, there are periods where the demand is low. Some studies employ single power demand while others employ multiple power demand so that the ramp rate is introduced. The ramp rate limits the power range because the power of each generating unit cannot jump too high or fall too low from its current power.

As an optimization problem, EED consists of two parts: objective and constraints. In this paper, the objective is minimizing the fuel cost and emission reduction cost. Meanwhile, the constraints are as follows.

- The power should be within the power range (minimum power and maximum power).
- The total power should meet the power demand.
- The ramp rate is not considered as this paper employs single power demand.
- The power loss is not considered.
- The power demand is known in advance.

Table 4. Annotations of EED problem

Notation	Description
u	generating unit
U	set of generating units
p_{min}	minimum power
p_{max}	maximum power
p_{total}	total power
p_{demand}	power demand
l	index of generating units
c_{total}	total cost
c_{fu}	fuel cost
c_{em}	emission reduction cost
w_{fu}	fuel cost weight
w_{em}	emission reduction cost weight
α, β, γ	constants for cost functions

The model of the EED is provided from Eqs. (19) to (25). The annotations used in this model are provided in Table 4.

Below is the explanation of Eqs. (19) to (25). Eq. (19) describes the set of generating units. Eq. (20) conforms that the power should be within the power range. Eq. (21) states that the total power is the accumulation of power from all generating units. Eq. (22) states that this total power should meet the power demand. Eq. (23) states that the total cost is the weighted aggregate of the fuel cost and emission reduction cost. Eq. (24) presents the quadratic presentation of fuel cost while Eq. (25) presents the quadratic presentation of the emission reduction cost.

$$U = \{u_1, u_2, u_3, \dots, u_{n(U)}\} \tag{19}$$

$$p_{min,l} \le p_l \le p_{max,l} \tag{20}$$

$$p_{total} = \sum_{n(U)} p_l \tag{21}$$

$$p_{total} = p_{demand} \tag{22}$$

$$c_{total} = w_{fu} \sum_{n(U)} c_{fu,l} + w_{em} \sum_{n(U)} c_{em,l} \tag{23}$$

$$c_{fu,l} = \alpha_{fu,l} + \beta_{fu,l} u_l + \gamma_{fu,l} u_l^2 \tag{24}$$

$$c_{em,l} = \alpha_{em,l} + \beta_{em,l} u_l + \gamma_{em,l} u_l^2 \tag{25}$$

4. Simulation and result

The performance investigation for the proposed SJO is conducted by challenging it to solve both theoretical and practical problems. The set of 23 classic functions is chosen as the theoretical problem and the EED problem where the use case is Java-Bali power system [28] is chosen as the practical one. The first use case represents the unconstrained problem while the second use case is the constrained problem.

In both problems, there are five new swarm-based metaheuristics chosen as the opponents. These metaheuristics include AOA, DOA, GAO, ZOA, and TIA. Both AOA and DOA were first introduced in 2024. Both GAO and TIA were first introduced in 2023. ZOA was first introduced in 2022. In both cases, the swarm size is 10 and the maximum iteration is also 10.

The set of 23 functions covers wide range of functions. There are seven high dimension unimodal functions and sixteen multimodal functions. These multimodal functions can be split into six high dimension functions and ten fixed dimension functions. The detail description of these functions is

Received: May 13, 2024. Revised: June 6, 2024.

Table 5. A detailed description of the set of 23 functions

No	Dim	Space	Target
1	40	[-100, 100]	0
2	40	[-100, 100]	0
3	40	[-100, 100]	0
4	40	[-100, 100]	0
5	40	[-30, 30]	0
6	40	[-100, 100]	0
7	40	[-1.28, 1.28]	0
8	40	[-500, 500]	-1.2569×10^4
9	40	[-5.12, 5.12]	0
10	40	[-32, 32]	0
11	40	[-600, 600]	0
12	40	[-50, 50]	0
13	40	[-50, 50]	0
14	2	[-65, 65]	1
15	4	[-5, 5]	0.0003
16	2	[-5, 5]	-1.0316
17	2	[-5, 5]	0.398
18	2	[-2, 2]	3
19	3	[1, 3]	-3.86
20	6	[0, 1]	-3.32
21	4	[0, 10]	-10.1532
22	4	[0, 10]	-10.4028
23	4	[0, 10]	-10.5363

in Table 5 including the dimension, search space, and the target. f_1 to f_7 are the high dimension unimodal functions. f_8 to f_{13} are the high dimension multimodal functions. f_{14} to f_{23} are the fixed dimension multimodal functions.

The result of the first use case is provided in Tables 6 to 9. In Tables 6 to 8, the data includes the mean or average fitness score, the range of the fitness score, and the rank based on the mean. The decimal value lower than 10^{-4} is rounded to 0. Table 9 presents the summary of the superiority of SJO based on the function clusters.

The result in Table 6 indicates that SJO is very competitive in overcoming the high dimension multimodal functions. SJO is the first best in all seven functions and the distinct first best in six functions (f_1, f_3-f_7). In these six functions, the performance disparity between the best and worst players is wide. Meanwhile, all players perform equally in solving f_2. This result indicates that SJO has good exploitation capability as this first cluster of functions is designed to investigate the exploitation capability of any optimization technique [22]. Meanwhile, ZOA becomes the first best opponents and TIA becomes the third best opponents.

Table 7 indicates that SJO is competitive in overcoming the high dimension multimodal functions. SJO is the first best in three functions (f_9-f_{11}), second best in two functions (f_{12} and f_{13}), and fifth best in f_8. The performance disparity between the best and worst performers is wide in three functions (f_9-f_{11}), moderate in two functions (f_{12} and f_{13}), and narrow in f_8. This result also indicates that SJO has good exploration capability as the functions in this second cluster is designed to investigate the exploration capability of any optimization technique [22].

Table 6. Fitness score comparison in solving high-dimension unimodal functions

F	Parameter	AOA [6]	DOA [7]	GAO [15]	ZOA [22]	TIA [12]	SJO
1	mean	1.4580×10^2	2.8908×10^2	1.3936×10^2	3.8412	4.2423	0.0005
	range	3.2364×10^2	3.4514×10^2	2.3071×10^2	7.6293	4.8407	0.0014
	mean rank	5	6	4	2	3	1
2	mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	range	0.0008	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
3	mean	1.6805×10^4	8.5415×10^3	1.7536×10^4	1.6023×10^3	7.4243×10^2	1.7060×10^1
	range	4.2713×10^4	1.7582×10^4	4.1255×10^4	4.9968×10^3	4.2851×10^3	6.4742×10^1
	mean rank	5	4	6	3	2	1
4	mean	8.6519	1.1307×10^1	9.2217	1.5569	1.7209	0.0351
	range	7.8496	1.1598×10^1	8.8458	1.6753	1.0206	0.0323
	mean rank	4	6	5	2	3	1
5	mean	3.8532×10^3	1.4028×10^4	4.7597×10^3	8.5603×10^1	1.0705×10^2	3.8900×10^1
	range	1.0819×10^4	3.3984×10^4	1.2984×10^4	9.2650×10^1	7.7185×10^1	0.1419
	mean rank	4	6	5	2	3	1
6	mean	1.4564×10^2	2.4337×10^2	1.6731×10^2	1.1495×10^1	8.7414	7.2856
	range	2.5990×10^2	3.5898×10^2	2.4993×10^2	1.0263×10^1	4.3311	1.5455
	mean rank	4	6	5	3	2	1
7	mean	0.0614	0.1219	0.0999	0.0301	0.0307	0.0103
	range	0.1377	0.2019	0.1232	0.1039	0.0719	0.0285
	mean rank	4	6	5	2	3	1

Table 7. Fitness score comparison in solving high-dimension multimodal functions

F	Parameter	AOA [6]	DOA [7]	GAO [15]	ZOA [22]	TIA [12]	SJO
8	mean	-3.1986×10^3	-3.0602×10^3	-3.3776×10^3	-2.7228×10^3	-2.1793×10^3	-2.7516×10^3
	range	2.1430×10^3	2.3100×10^3	1.9624×10^3	1.8908×10^3	1.4314×10^3	2.1570×10^3
	mean rank	2	3	1	4	6	5
9	mean	1.7623×10^2	1.1724×10^2	2.4601×10^2	1.2168×10^1	2.6785×10^1	0.0079
	range	2.4460×10^2	1.3237×10^2	1.8740×10^2	4.0708×10^1	7.5430×10^1	0.0952
	mean rank	5	4	6	2	3	1
10	mean	4.1118	4.7896	3.7839	0.7393	0.8015	0.0045
	range	4.8198	2.6999	2.3291	0.9616	0.4911	0.0069
	mean rank	5	6	4	2	3	1
11	mean	2.5576	3.5832	2.3582	0.5267	0.6494	0.0072
	range	2.0330	3.1560	1.5741	0.7921	0.7851	0.1101
	mean rank	5	6	4	2	3	1
12	mean	2.3515	4.2141	2.7806	0.9641	0.5687	0.7663
	range	2.7317	1.3222×10^1	2.6451	0.6194	0.4663	0.5147
	mean rank	4	6	5	3	1	2
13	mean	9.0674	1.8674×10^1	1.1562×10^1	3.7761	2.9714	3.1263
	range	1.8220×10^1	3.7236×10^1	1.3471×10^1	1.3800	1.3895	0.4307
	mean rank	4	6	5	3	1	2

Table 8. Fitness score comparison in solving fixed dimension multimodal functions

F	Parameter	AOA [6]	DOA [7]	GAO [15]	ZOA [22]	TIA [12]	SJO
14	mean	8.0398	9.8137	6.7960	9.5049	9.4899	8.0239
	range	1.5990×10^1	1.5562×10^1	1.2595×10^1	9.6885	2.2364×10^1	9.6884
	mean rank	3	6	1	5	4	2
15	mean	0.0096	0.0091	0.0040	0.0053	0.0012	0.0030
	range	0.0314	0.0436	0.0075	0.0544	0.0027	0.0302
	mean rank	6	5	3	4	1	2
16	mean	-1.0232	-1.0182	-1.0218	-1.0110	-1.0307	-1.0289
	range	0.0490	0.0730	0.0558	0.2721	0.0091	0.0253
	mean rank	3	5	4	6	1	2
17	mean	0.4206	0.4208	0.4200	1.8887	0.4593	0.6627
	range	0.1285	0.0985	0.2558	7.4876	0.8650	2.1600
	mean rank	2	3	1	6	4	5
18	mean	3.9369	3.7278	4.4287	1.5888×10^1	3.9556	2.3876×10^1
	range	1.0600×10^1	1.1320×10^1	2.7058×10^1	6.8099×10^1	7.5034	8.1611×10^1
	mean rank	2	1	4	5	3	6
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	range	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.9445	-3.0285	-2.9654	-2.4926	-2.8123	-2.6540
	range	0.5358	0.6472	0.6020	1.8346	1.0489	1.2013
	mean rank	3	1	2	6	4	5
21	mean	-2.8372	-2.6092	-1.9835	-2.5803	-2.8776	-3.6759
	range	5.7116	3.4869	2.6923	4.4496	6.2029	7.6421
	mean rank	3	4	6	5	2	1
22	mean	-2.7731	-2.7430	-2.1624	-2.2842	-3.5489	-3.8019
	range	5.7805	3.5367	4.1648	2.9938	8.2968	6.2611
	mean rank	3	4	6	5	2	1
23	mean	-2.3960	-2.6782	-2.0878	-2.3917	-3.0522	-3.2722
	range	3.0851	2.9055	2.3436	3.4742	4.3516	6.0343
	mean rank	4	3	6	5	2	1

Received: May 13, 2024. Revised: June 6, 2024.

Table 9. Group-based superiority of SJO

Group	Number of Functions Where SSA is Better				
	AOA [6]	DOA [7]	GAO [15]	ZOA [22]	TIA [12]
1	6	6	6	6	6
2	5	5	5	5	4
3	6	6	5	8	4
Total	17	17	16	19	14

Table 8 indicates that SJO is also competitive in overcoming the fixed dimension multimodal functions. SJO becomes the first best in four functions (f_{19}, f_{21}-f_{23}), the second best in three functions (f_{14}-f_{16}), the fifth best in two functions (f_{17} and f_{20}), and the sixth best in f_{18}. This result also indicates the tough competition among players as the performance disparity between the best and worst performers is narrow. This result also shows that in general, all players have a balancing capability between the exploitation and exploration as these functions are designed for this consideration [22].

The result in Table 9 indicates the superiority of SJO compared to its opponents. SJO is better than AOA, DOA, GAO, ZOA, and TIA in 17, 17, 16, 19, and 14 functions out of 23 functions respectively. In general, the superiority of SJO occurs in the high dimension functions (the first and second clusters). Meanwhile, SJO is superior in the third cluster when only being compared to ZOA.

In the second investigation, SJO is challenged to tackle the EED problem with the use case is Java-Bali power grid system in Indonesia. This system is the biggest in Indonesia as it provides 500 kV power [28] as Java-Bali is the most populous and industrialized region in Indonesia. This system consists of eight generating units where two of them are hydroelectric power plants and six steam power plants [28].

Each power plant represents a generating unit. The two hydroelectric power plants are Cirata and Saguling power plants. Both Cirata and Saguling power plants are in West Java province. Meanwhile, the steam prower plants are Suralaya, Muaratawar, Tanjungjati, Gresik, Paiton, Grati. Suralaya power plant is in Banten province. Muaratawar and Tanjungjati power plants are in West Java province. Gresik, Paiton, and Grati power plants are in East Java province.

The specification of this system is provided in Tables 10 to 12 [28]. Table 10 provides the constants related to the fuel cost. Table 11 provides the constants related to the emission reduction cost. Table 12 provides the power range. Meanwhile, the result is presented in Table 13. This result is obtained based on the balance weight between the operation/fuel cost and the emission cost. As shown in Tables 10 and 11, the third and fourth generating units are the hydroelectric power plants while the others are the steam power plants.

The result shows that SJO performs as the best player in tackling this EED problem. On the other hand, DOA is the best opponent while GAO is the worst opponent. Besides, the range of the total cost between the best and worst players is narrow compared to the average total cost. This evidence shows the tough competition among players in solving this EED problem.

Table 10. Operational cost related constants [28]

Gen.	$\alpha_{fu,l}$	$\beta_{fu,l}$	$\gamma_{fu,l}$
1	57,543,208.0	3,332,794.0	-400.0
2	519,353,767.1	3,047,098.0	691.0
3	0.0	400.0	0.0
4	0.0	660.0	0.0
5	133,177,025.6	2,828,349.0	-80.0
6	180,205,527.9	2,104,640.0	218.0
7	140,621,312.5	2,545,832.0	203.0
8	112,522,922.1	5,877,235.0	-73.0

Table 11. Emission cost related constants [28]

Gen.	$\alpha_{em,l}$	$\beta_{em,l}$	$\gamma_{em,l}$
1	34,251,909.8	1,983,806.2	-236.7
2	72,202,664.7	423.6	96.2
3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	93,654,729.7	1,988,993.9	-56.9
6	123,428,443.8	1,441,534.9	149.5
7	140,621,312.5	2,545,832.5	62.1
8	24,146,549.8	1,261,209.3	-15.8

Table 12. Power range [28]

Gen.	p_{min}	p_{max}
1	1,610	4,200
2	934	2,308
3	404	1,008
4	208	700
5	848	2,400
6	1,080	4,714
7	360	900
8	305	1,610

Table 13. Average total cost

No	Metaheuristic	Total Cost (rupiah/hour)
1	AOA [6]	23,275,746,695
2	DOA [7]	23,243,749,572
3	GAO [15]	23.621,739,439
4	ZOA [22]	23,484,937,934
5	TIA [12]	23,281,509,625
6	SJO	23,242,945,283

5. Discussion

In general, the result shows that the objective in developing new swarm-based metaheuristics whose performance is acceptable is achieved. SJO can find the quasi-optimal solution for all problems whether the unconstrained problem as in 23 classic functions or the constrained one as EED problem. Its performance is superior in almost all high dimension functions and competitive in the fixed dimension functions and EED problem. In the context of SJO fails to become the best one, the performance disparity between SJO and the best player is narrow.

The assessment result of both assessments proves the no-free-lunch (NFL) theory. The wide performance disparity in solving most of high dimension functions and narrow performance disparity in solving all fixed dimension functions and EED problem shows that the performance of any optimization techniques highly depends on the problem they try to solve despites the nature of their techniques. This circumstance is also strengthened based on the dynamic in the mean rank. Although SJO performs very competitive in many functions, its performance is the worst in f_{18} and second worst in three functions (f_8, f_{17}, and f_{20}).

The complexity of SJO can be investigated by analysing the loop within its process. Based on this argument, the complexity during the initialization is different from the complexity during the iteration. There are a nested loop that consists of two loops during the initialization so that the complexity during this phase can be presented as $O(n(S).d)$. It means that the complexity is linear to the swarm size and the dimension of the problem. On the other hand, there are a nested loop that consists of four loops during the iteration phase so that the complexity during this phase can be presented as $O(t_m.d.n(S)^2)$. This presentation shows that the complexity during this phase is linear to the maximum iteration or dimension of the problems but quadratic to the swarm size. There are two loops for whole swarm during the iteration. The first is iteration so that all swarm members perform the searching process. The second is the iteration to construct the higher quality member pool.

Despites the achievement of this work as the acceptable performance of SJO in finding the quasi-optimal solutions, there are limitations regarding this work. In general, the real-world optimization problems are constrained problems. The EED problem is one of them. Meanwhile, there are many more optimization problems, especially in the engineering field that span from electrical, mechanical, industrial, and so on. Moreover, many of these problems are the combinatorial problems like flow shop scheduling [38], vehicle routing problem [39], course timetabling [40], and so on rather than numerical ones like EED or 23 classic functions. SJO should be tested to solve these various problems so that a more comprehensive investigation regarding its nature and performance can be conducted. Meanwhile, it is impossible to conduct all these tests in a single paper like this current work. Based on it, implementing SJO in various optimization problems is highly recommended.

6. Conclusion

A new swarm-based metaheuristics that is constructed based on equal size swarm split called stay-jump optimizer (SJO) has been presented in this paper. This presentation includes the concept, formalization, and the assessment. The result shows that the performance of SJO is acceptable as it can find the quasi-optimal solutions of both 23 classic functions and EED problem. The result also shows that SJO is superior to its opponents in solving high dimension functions and competitive in solving fixed dimension functions and EED problems. SJO is better than AOA, DOA, GAO, ZOA, and TIA in 17, 17, 16, 19, and 14 functions out of 23 functions respectively. Moreover, SJO also becomes the best technique in solving the EED problem. The result also strengthens the NFL theory as the wide performance disparity occurs in almost all high dimension functions and narrow performance disparity occurs in the fixed dimension functions and EED problem.

In the future, SJO can be tested to solve various real-world problems, especially with environmental issue consideration. Various tests can give better and comprehensive understanding regarding the performance and the nature of SJO including its strengths and weaknesses. Testing SJO to solve the combinatorial problems is also challenging as some modification is needed.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

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

Acknowledgments

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

References

[1] L. Wei, and G. An, "Optimization of Building Integrated Energy Scheduling Using an Improved Genetic Whale Algorithm", *Scientific Reports*, Vol. 14, No. 1, p. 2386, 2024.

[2] W. Yan, X. Wang, Y. Liu, X. M. Zhang, Z. G. Jiang, and L. Huang, "A Stochastic Programming Approach for EOL Electric Vehicle Batteries Recovery Network Design under Uncertain Conditions", *Scientific Reports*, Vol. 14, No. 1, p. 876, 2024.

[3] A. Manea, G. Dolci, and M. Grosso, "Life Cycle Assessment and Cost Analysis of An Innovative Automatic System for Sorting Municipal Solid Waste: A Case Study at Milan Malpensa Airport", *Waste Management*, Vol. 183, pp. 63-73, 2024.

[4] N. Kumar, S. Dahniya, and K. P. S. Parmar, "Multi-objective Economic Emission Dispatch Optimization Strategy Considering Battery Energy Storage System in Islanded Microgrid", *Journal of Operation and Automation in Power Engineering*, Vol. 12, No. 4, pp. 296-311, 2024.

[5] 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.

[6] T. Hamadneh, K. Kaabneh, O. Alssayed, K. Eguchi, S. Gochhait, I. Leonova, and M. Dehghani, "Addax Optimization Algorithm: A Novel Nature-Inspired Optimizer for Solving Engineering Applications", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 732-743, 2024, doi: 10.22266/ijies2024.0630.57.

[7] S. A. Omari, K. Kaabneh, I. A. Falahah, K. Eguchi, S. Gochhait, I. Leonova, Z. Montazeri, M. Dehghani, "Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 816-828, 2024, doi: 10.22266/ijies2024.0630.63.

[8] M. A. Al-Sharqi, A. T. S. Al-Obaidi, and S. O. Al-Mamory, "Apiary Organizational-based Optimization Algorithm: A New Nature-Inspired Metaheuristic Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 783-801, 2024, doi: 10.22266/ijies2024.0630.61.

[9] P. D. Kusuma, "Focus and Shake Algorithm: A New Stochastic Optimization Employing Strict and Randomized Dimension Mappings", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 551-562, 2024, doi: 10.22266/ijies2024.0630.43.

[10] O. Al-Baik, S. Alomari, O. Alssayed, S. Gochhait, I. Leonova, U. Dutta, O. P. Malik, Z. Montazeri, and M. Dehghani, "Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 9, p. 65, 2024.

[11] S. E. Khouni, and T. Menacer, "Nizar Optimization Algorithm: A Novel Metaheuristic Algorithm for Global Optimization and Engineering Applications", *The Journal of Supercomputing*, Vol. 80, pp. 3229-3281, 2024.

[12] 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.

[13] P. D. Kusuma, and F. C. Hasibuan, "Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 264-275, 2023, doi: 10.22266/ijies2023.0831.22.

[14] P. D. Kusuma, and A. Dinimaharawati, "Four Directed Search Algorithm: A New Optimization Method and Its Hyper Strategy Investigation", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 598-611, 2023, doi: 10.22266/ijies2023.1031.51.

[15] O. Alsayyed, T. Hamadneh, H. Al-Tarawneh, M. Alqudah, S. Gochhait, I. Leonova, O. P. Malik, and M. Dehghani, "Giant Armadillo Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 8, p. 619, 2023.

[16] M. Dehghani, P. Trojovsky, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 8, p. 121, 2023.

[17] Z. Montazeri, T. Niknam, J. Aghaei, O. P. Malik, M. Dehghani, and G. Dhiman, "Golf

Optimization Algorithm: A New Game-based Metaheuristic Algorithm and Its Application to Energy Commitment Problem Considering Resilience", *Biomimetics*, Vol. 8, p. 386, 2023.
[18] 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, p. e1431, 2023.
[19] P. Trojovsky, and M. Dehghani, "A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Walruses Behavior", *Scientific Reports*, Vol. 13, p. 8775, 2023.
[20] 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.
[21] 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.
[22] 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.
[23] 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, 2022.
[24] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian Tiger Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *IEEE Access*, Vol. 10, pp. 132396-132431, 2022.
[25] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie Dog Optimization Algorithm", *Neural Computing and Applications*, Vol. 34, pp. 20017-20065, 2022.
[26] 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.
[27] 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, p. 2921, 2023.
[28] 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.
[29] 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.
[30] 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.
[31] 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, p. 6933, 2023.
[32] B. M. Hussein, "Evolutionary Algorithm Solution for Economic Dispatch Problems", *International Journal of Electrical and Computer Engineering*, Vol. 12, No. 3, pp. 2963-2970, 2022.
[33] A. I. Abbas, and A. Answer, "The Optimal Solution for Unit Commitment Problem Using Binary Hybrid Grey Wolf Optimizer", *International Journal of Electrical and Computer Engineering*, Vol. 12, No. 1, pp. 122-130, 2022.
[34] H. S. Maharana, and S. K. Dash, "Quantum Behaved Artificial Bee Colony based Conventional Controller for Optimum Dispatch", *International Journal of Electrical and Computer Engineering*, Vol. 13, No. 2, pp. 1260-1271, 2023.
[35] 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, 2024.

[36] Y. Qawqzeh, M. T. Alharbi, A. Jaradat, and K. N. A. Sattar, “A Review of Swarm Intelligence Algorithms Deployment for Scheduling and Optimization in Cloud Computing Environments”, *PeerJ Computer Science*, Vol. 7, p. e696, 2021.

[37] S. Katoch, S. S. Chauhan, and V. Kumar, “A Review on Genetic Algorithm: Past, Present, and Future”, *Multimedia Tools and Applications*, Vol. 80, pp. 8091-8126, 2021.

[38] S. Singla, H. Kaur, D. Gupta, U. M. Modibbo, and J. Kaur, “No Idle Flow Shop Scheduling Models for Optimization of Machine Rental Costs with Processing and Separated Setup Times”, *Frontiers in Applied Mathematics and Statistics*, Vol. 10, p. 1355237, 2024.

[39] A. A. Londono, W. G. Gonzalez, O. D. M. Giraldo, and J. W. Escobar, “A Hybrid Heuristic Approach for the Multi-Objective Multi Depot Vehicle Routing Problem”, *International Journal of Industrial Engineering Computations*, Vol. 15, No. 1, pp. 337-354, 2024.

[40] P. D. Kusuma, and A. S. Albana, “University Course Timetabling Model in Joint Courses Program to Minimize the Number of Unserved Requests”, *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 10, pp. 121-127, 2021.