Received: May 2, 2022. Revised: May 17 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Fixed Step Average and Subtraction Based Optimizer

Purba Daru Kusuma[1]* **Ashri Dinimaharawati[1]**

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

Abstract: This work proposes a new metaheuristic algorithm: a fixed-step average and subtraction-based optimizer (FS-ASBO). This algorithm is the improved version of the average and subtraction-based optimizer (ASBO). There are several improvements related to the original ASBO. First, the proposed algorithm replaces the randomized step size in the guided movement with the fixed step size. Second, the proposed algorithm adds an exploration mechanism after the guided movement in every iteration when the new candidate fails to find a better solution. This proposed algorithm is then implemented into a simulation to evaluate its performance. Through simulation, the proposed algorithm is challenged to solve theoretical optimization problems and real-world optimization problems. The 23 well-known benchmark functions represent the theoretical optimization problem. Meanwhile, the housing optimization problem represents the real-world one. In the simulation, the proposed algorithm is compared with particle swarm optimization (PSO), marine predator algorithm (MPA), Komodo mlipir algorithm (KMA), static Komodo algorithm (SKA), and ASBO. The result shows that this proposed algorithm is competitive to solve theoretical problem and superior to solve real-world problem. The proposed algorithm outperforms all sparing algorithms in solving seven functions. In housing optimization problem, it creates 12%, 10%, 8%, 11%, and 10% better total gross profit than the ASBO, PSO, MPA, KMA, and SKA.

Keywords: Average and subtraction-based optimizer, Metaheuristic, Housing optimization problem, Multimodal, Swarm intelligence.

1. Introduction

Optimization is a popular process conducted in many sectors. Its popularity comes from two circumstances. First, people or organizations always try to meet their objectives or goal. On the other hand, they always face limited resources in hand. Then, they become the objective of any optimization work to find the most efficient way to achieve any goal using limited resources. Due to its characteristics, optimization has become a popular process in many studies in operations research, such as manufacturing [1], health care [2], transportation [3], education [4], and so on.

There are several standard terms used in optimization. The purpose or goal of the work is called an objective. In some studies, it is like a soft constraint. It can be maximization or minimization. Maximization is related to the return that is wanted to achieve, such as revenue [5], profit, number of customers, etc. Contrary, minimization is related to resources that are utilized, such as cost [3], energy consumption [6], travel distance [7], processing time [8], delay [9], and so on. Meanwhile, the optimization process should be conducted within certain areas called boundaries, problem spaces, or constraints. For example, in the vehicle routing problem, vehicles should operate within a specific range of operational time [10], and goods loaded cannot surpass a particular maximum capacity [3]. The other example in the healthcare system is that all patients should be served within the given rooms [2] or nurses [2]. Besides operations research, optimization is also used in other sectors, such as telecommunication [11], energy [12], image processing [13], and so on. The optimal solution is an arrangement of resources that most meets the objective.

One popular method used in the optimization is metaheuristic method. This method uses an approximate approach so that it cannot guarantee the real optimal solution or global solution [14]. It tries its best effort to find the near-optimal solution or acceptable solution. Although it does not guarantee the global solution, the metaheuristic method is still popular and widely used due to its flexibility in solving many complex optimization problems within the given computational resources [14]. It is contrast with the exact method that guarantees the global solution but needs excessive computational resources. In many complex optimization problems [14], this method is impossible to conduct. As an approximate method, metaheuristic works at some stochastic level where the solution is generated randomly. Then, this solution is improved during the iteration until the termination criteria are met, or the maximum iteration is reached. Two mechanisms are always conducted in many metaheuristic algorithms: exploitation and exploration. Exploitation means that the algorithm tries to find a better solution near the current solution. On the contrary, exploration means that the algorithm tries to find an alternative solution within the problem space.

To date, there are hundreds of metaheuristic algorithms. Many of them were inspired by the nature mechanism, especially animals. The mechanism of the animal-inspired many algorithms is during the mating, foraging, or a combination of them. The mating process is adopted due to its characteristic of creating better offspring by selecting the parents, such as in genetic algorithm (GA) [15], red deer algorithm (RDA) [16], and so on. Foraging is adopted due to its circumstances that in the real world, the animal always tries to find food, but they never know where the best food lies in their habitat. The example of algorithms inspired by the foraging process is the particle swarm optimization (PSO) [17], artificial bee colony (ABC) [18], cat and mouse-based optimizer (CMBO) [19], spotted hyena optimizer (SHO) [20], whale optimization algorithm (WOA) [21], grey wolf optimizer (GWO) [22], tunicate swarm algorithm (TSA) [23], marine predator algorithm (MPA) [24], and so on. Meanwhile, several algorithms combine both the mating process and the foraging process by benefiting from these two processes. An example of these algorithms is Komodo mlipir algorithm (KMA) [25], and so on.

One of the newest metaheuristic algorithms is the average and subtraction-based optimizer (ASBO). This algorithm was proposed by Dehghani in 2022 [26]. The core concept of this algorithm is to move toward the best solution and avoid the worst solution. This algorithm consists of three sequential movements in every iteration [26]. In the first phase, a movement toward the average between the best and worst solutions is conducted. In the second phase, movement with the direction equal to the subtraction of the best solution with the worst solution is conducted. In the third phase, movement in the opposite way toward the best solution is conducted. In every movement, a new solution is accepted only if it is better than the current solution.

As a brand-new algorithm, the improvement related to this algorithm is very potential. Studies conducted to improve or implement this algorithm have not existed yet. Moreover, in its first publication, ASBO is tested to solve the only theoretical mathematic problem [26]. Through simulation, ASBO outperformed GA, PSO, gravitational search algorithm (GSA), teaching learning-based optimization (TLBO), GWO, WOA, TSA, SHO, and MPA [26]. However, this algorithm has not been tested to solve real-world problems. Based on this circumstance, studies to improve ASBO by modifying its exploration-exploitation strategy or hybridizing it with other methods are possible.

Due to this problem, this work aims to propose a new metaheuristic algorithm developed to improve ASBO. The improvement is conducted by modifying the exploration and exploitation mechanisms in ASBO. This modification is adopted from the other metaheuristic algorithm.

Several contributions are conducted in this work. These contributions are as follows.

1) This work becomes the first work that improves the ASBO algorithm due to its newest algorithm.
2) This work replaces the randomized step size with the fixed step size during the movement.
3) This work adds an exploration mechanism after the guided movements are conducted every iteration.
4) This work enriches studies in ASBO with a real-world optimization problem which has not been conducted yet in the earlier study so that the practical performance of the algorithm is evaluated too rather than just theoretical performance.

The methodology used in this work is as follows. In the beginning, the ASBO algorithm is reviewed so that its mechanism, advantage, and disadvantage are analyzed. Then, based on this analysis, the improvement of ASBO is developed by hybridizing ASBO with other methods used by other algorithms. This improved version is then implemented into a simulation to evaluate its performance. The proposed algorithm is used to solve both a theoretical mathematical problem and the real-world optimization problem in this work. The 23 benchmark functions are used as the theoretical problem, while the housing optimization problem is chosen as the real-world optimization problem. Then, the analysis related to the simulation result is conducted. In the end, this paper is written as a presentation of this whole process.

The remainder of this paper is structured as follows. The original form of ASBO is reviewed in section two. Based on this review, the proposed algorithm's model is explained in section three, which consists of the conceptual model, the algorithm presented in pseudocode, and the mathematical model. The simulation that is conducted to evaluate the proposed algorithm's performance in solving theoretical mathematic problems and real-world optimization problems, and its result, is shown in section four. The more profound analysis related to the findings is discussed in section five based on the simulation result. In the end, the conclusion of this work and the future research potential regarding this work are summarized in section six.

2. Related works

ASBO is a population-based metaheuristic algorithm. This algorithm consists of several autonomous agents that find a better solution in every iteration. This algorithm is also a swarm-based intelligence where collective intelligence is shared among agents [27]. In this context, this collective intelligence is the best and worst solution. These two solutions are selected among agents in every iteration. These best and worst solutions are not the best and worst solutions so far during the iteration but the best and worst solutions in the current iteration. Then, every agent moves toward the best solution and away from the worst solution. Besides the best solution in every iteration, the global best solution is also introduced. Global best solution is the best solution so far along with the iteration. In every iteration, this global best solution is updated. After the iteration ends, this global best solution becomes the final solution.

In ASBO, three movements are conducted sequentially in every iteration [26]. The first movement is toward the average location between the best solution and the worst solution. The second movement is toward the vector from the worst solution to the best solution. The third movement is a movement away from the best solution. This third movement represents the exploration strategy. In every movement, the agent will move to the new location (solution) if the new location is better than its current location. Otherwise, the agent stays in its current location. The detailed explanation related to every movement is described below.

In the first movement, the target is between the best solution and the worst location [26]. It is obtained by finding the average location between the best and worst solutions. After this average based target is obtained, there are two possible movements. If the target is better than the agent's current location, the agent sets movement toward this target. Otherwise, the agent sets movement away from this target. The step size of this movement is set randomly.

The second movement obtains a vector between the best solution and the worst solution [26]. This vector is calculated by subtracting the best solution from the worst solution; Then, the agent sets movement based on this vector. Like the first movement, the movement step size is randomized.

In the third movement, a vector between the agent's current location and the best solution is obtained [26]. This vector is obtained by subtracting the agent's current location with the best solution. Then, the agent sets a movement based on this weighted vector with a randomized step size.

There are several notes due to the exploration-exploitation strategy conducted in ASBO. First, besides ASBO, Dehghani also used the concept of best and worst solutions in his other metaheuristic works, such as football game optimizer (FBGO) [28], dart game optimizer (DGO) [29], shell game optimizer (SGO) [30], and hidden object game optimizer (HOGO) [31]. These four algorithms are the metaheuristic algorithms inspired by game mechanics. The similarities among these algorithms are moving toward the best solution and moving away from the worst solution. However, in ASBO, the process tends to be deterministic. Moreover, ASBO is simpler than FBGO, DGO, SGO, and HOGO. The other difference is that in ASBO, each agent focuses only on the best and worst solutions. Meanwhile, other randomized selected agents are also considered in the four algorithms.

ASBO can also be seen as an algorithm that combines mating and foraging processes in the animal-inspired algorithm, although it is not declared explicitly. The average mechanics between the best and worst solutions is like the balance mating process or crossover. RDA and KMA also conduct the mating process. In KMA [25], there is only one mating process between a female and the highest quality big male. Meanwhile, the mating process is more massive in RDA. In RDA [16], every commander creates a group of harems. Then, a certain portion of harems mates with its commander while other harems mate with other commanders. Then, every stag mates with the nearest harem.

The concept of the guided movement in ASBO is like the foraging mechanism, such as in KMA and MPA. In KMA, a big-male moves toward other better big males and moves away from other worse big males [25]. Meanwhile, the small male moves toward the big males with a certain step size. In MPA, every prey moves with a certain step size [24]. This guided movement is generally used in PSO as the early version of swarm intelligence. In PSO, every agent moves toward the global best and the local best with a certain weight to find a better solution [17]. However, there is the main difference between ASBO and several other algorithms, such as PSO and MPA. Rather than conducting weighted accumulation like in PSO, ASBO selects the best one among the movement. Meanwhile, like in PSO, the exploitation and exploration are conducted during the guided movement.

In its original form, ASBO does not provide any parameter for adjustment. This circumstance has some consequences. In general, many metaheuristic algorithms are equipped with room for adjustment. This adjustment is needed to adapt to the problem that it faces. Appropriate adjustment gives a good optimization result. On the contrary, an improper adjustment will end with a bad result. Due to no parameter for the adjustment, the user of ASBO cannot improve the performance but, on the other hand, is not worried about the wrong adjustment.

Based on this review, there are several possibilities regarding the exploration and exploitation strategy in ASBO. First, modification can be conducted in the guided movement. Second, modification can also be conducted by enriching the exploration strategy outside the guided movement. Third, modification can be conducted to give space for adjustment.

3. Model

The conceptual model of the proposed algorithm is as follows. As an improved version of ASBO, the main concept of ASBO is still used. The proposed algorithm still conducts three guided movements that are conducted sequentially. Every movement creates a candidate. Candidate whose fitness is the best becomes the selected candidate. Then, this candidate is compared with the current solution. This candidate replaces the current solution only if it is better than the current solution. Otherwise, the agent stays in its current location. In every iteration, the best and worst solutions are selected as the input for the movements. The global best solution is updated in every iteration. Finally, the global best solution becomes the final solution.

There are modifications conducted in this proposed algorithm. First, this proposed algorithm replaces the randomized step size used in the original ASBO with the fixed step size used in PSO [17]. Second, the proposed algorithm deploys fully randomized exploration if there is no guided movement candidate in the iteration that is better than the current solution. In this process, exploration is conducted within the problem space. This concept is like the exploration process in the artificial bee colony (ABC). In ABC, a bee will find another solution somewhere else within the problem space after several times; this bee fails to find a better solution near this current solution [18]. Different from other algorithms, several candidates are generated randomly in this exploration strategy, and a candidate whose fitness is the best becomes the selected candidate. This selected candidate replaces the current solution if it is better than the current solution.

This conceptual model is then transformed into an algorithm and mathematical model. The algorithm is shown in algorithm 1. Meanwhile, several annotations used in the mathematical model are described as follows.

The explanation of algorithm 1 is as follows. The algorithm consists of two steps: initialization and iteration. In the initialization, all initial solutions are generated. Then the iteration runs until the maximum

b_l	lower bound
b_u	upper bound
c_{g1}	first guided movement candidate
c_{g2}	second guided movement candidate
c_{g3}	third guided movement candidate
c_{gbest}	best guided movement candidate
c_e	exploration candidate
c_{ebest}	best exploration candidate
f	fitness function
t	iteration
t_{max}	maximum iteration
U	uniform random
x	current solution
x_{best}	best solution
x_{worst}	worst solution
x_{gbest}	global best solution
x_{t1}	first movement target
x_{t2}	second movement target
x_{t3}	third movement target
x_{av}	average based location
w_1	first movement step size
w_2	second movement step size
w_3	third movement step size

algorithm 1: FS-ASBO main algorithm

```
output: x_gbest
begin
//initialization
 for i=1 to n(X) do
  initialize(x_i)
 end for
//iteration
 for t=1 to t_max do
  find(x_best)
  find(x_worst)
  update(x_gbest)
  set first movement candidate(c_g1)
  set second movement candidate(c_g2)
  set third movement candidate(c_g3)
  select (c_gbest)
  update (x_i, c_gbest)
  if not move(x_i) then
   for j=1 to n(C_e) do
    generate exploration candidate(c_e,j)
   end for
   select (c_ebest)
   update (x_i, c_ebest)
  end if
 end for
end
```

iteration. At the beginning of the iteration, the best and worst solutions are selected. Then the global best solution is updated. Then, all three guided movements are conducted sequentially. Then, the best candidate among these movements is selected. This selected candidate is then used to update the current solution. If the agent stays in the current location, exploration is conducted by generating several exploration candidates. Then, the best candidate is selected among these exploration candidates. Once again, this selected candidate is used to update the current solution.

During initialization, all initial solutions are generated randomly within the problem space. It used uniform distribution so that every possible solution has equal opportunity. This process is formalized by using Eq. (1).

$$x = U(b_l, b_u) \tag{1}$$

In the beginning of the iteration, the best solution and the worst solution are selected. Then, this best solution is used to update the global best solution. These processes are formalized by using Eqs. (2) to (4).

$$x_{best} = x \in X | \min\left(f(x)\right) \tag{2}$$

$$x_{worst} = x \in X | \max\left(f(x)\right) \tag{3}$$

$$x'_{gbest} = \begin{cases} x_{best}, f(x_{best}) < f\left(x'_{gbest}\right) \\ x'_{gbest}, else \end{cases} \tag{4}$$

The explanation of Eqs. (2) to (4) is as follows. Eq (2) states that the best solution is the solution whose fitness score is the lowest one (minimization). Eq. (3) states that the worst solution is the solution whose fitness score is the highest. Eq. (4) states that the best solution will replace the global best solution if better than the global best solution.

The first movement is the agent's movement related to the average between the best solution and the worst solution. Like the original ASBO, the candidate of this movement may move toward this average solution or move away from the average solution. It depends on the fitness score of this average solution. This process is formalized by using Eqs. (5) to (7). Eq. (5) shows that the average location is obtained by finding the average value between the best and worst solutions. Then, Eq. (6) states that the first movement target is toward the average solution if this average solution is better than the current solution. Otherwise, the first movement target is away from the average solution. Finally, Eq. (7) shows that the first movement candidate moves from the current solution toward the first moving target with the fixed weighted step size.

$$x_{av} = \frac{x_{best}+x_{worst}}{2} \tag{5}$$

$$x_{t1} = \begin{cases} x_{av} - 2x, f(x_{av}) < f(x) \\ x - x_{av}, else \end{cases} \tag{6}$$

$$c_{g1} = x + w_1 . x_{t1} \tag{7}$$

The second guided movement is based on the subtraction of the best solution with the worst solution. This movement is formalized using Eq. (8) and Eq. (9). Eq. (8) indicates that the second movement target is obtained by subtracting the best solution from the worst solution. Then, Eq. (9) states that the second movement candidate moves from the current solution toward the second movement target with a fixed weighted step size.

$$x_{t2} = x_{best} - x_{worst} \tag{8}$$

$$c_{g2} = x + w_2 . x_{t2} \tag{9}$$

The third guided movement is the movement-related of the current solution away from the best solution. This is conducted to explore another solution. This movement is formalized using Eqs.

(10) and (11). Eq. (10) shows that the third movement target is away from the best solution. Then, Eq. (11) states that the third movement candidate moves from the current solution toward the third movement target with a fixed step size.

$$x_{t3} = x - 2x_{best} \tag{10}$$

$$c_{g3} = x + w_3 . x_{t3} \tag{11}$$

The next process is selecting the best-guided movement candidate. The guided movement candidate whose fitness score is the best will be chosen. Then, this selected candidate is used to update the current solution. This process is formalized using Eqs. (12) and (13). Eq. (12) is used as the candidate selection. Meanwhile, Eq. (13) states that this candidate will replace the current solution if it is better than the current solution.

$$c_{gbest} = c_g | \min\left(f(c_{g1}), f(c_{g2}), f(c_{g3})\right) \tag{12}$$

$$x' = \begin{cases} c_{gbest}, f(c_{gbest}) < f(x) \\ x, else \end{cases} \tag{13}$$

The exploration is conducted if the agent stays at its current solution. This process is formalized by using Eqs. (14) to (16). Eq. (14) states that the exploration candidate is generated randomly within the problem space. Eq. (15) states that the best exploration candidate is a candidate among the exploration candidates whose fitness score is the best. Finally, Eq. (16) states that this selected candidate will replace the current solution if it is better than the current solution.

$$c_e = U(b_l, b_u) \tag{14}$$

$$c_{ebest} = c_e \in C_e | \min(f(c_e)) \tag{15}$$

$$x' = \begin{cases} c_{ebest}, f(c_{ebest}) < f(x) \\ x, else \end{cases} \tag{16}$$

Based on this model, the algorithm's complexity can be presented as $O(t_{max}.n(X).n(C_e))$. It means the maximum iteration, population size, and the number of exploration candidates has an equal position as multipliers related to the algorithm's complexity. The challenge is finding the most appropriate combination to reach the acceptable solution while keeping the computational consumption low.

4. Simulation and result

This section implements the proposed algorithm into simulations to evaluate its performance. There are three simulations conducted for this process. The first simulation is conducted to evaluate the algorithm's performance in solving theoretical mathematic problems. The second simulation is conducted to evaluate the algorithm's convergence. The third simulation is conducted to evaluate the algorithm's performance in solving real-world optimization problems.

In the first simulation, the algorithm will be used to solve 23 benchmark functions. These functions are commonly used in many latest optimization studies, such as in MPA [24], RDA [16], KMA [25], HOGO [31], and so on. These functions are divided into three groups. The first group represents the high dimension unimodal function. The second group represents the high dimension multimodal function. The third group represents the fixed dimension multimodal function. The unimodal function is a function that has only one optimal solution called the optimal global solution [32]. This function does not have any optimal local solution.

Contrary, the multimodal function is a function that has several or many optimal solutions [32]. One solution is the global optimal or true optimal solution. The other optimal solutions are the local optimal solutions [32]. The first group consists of 7 functions: Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic. The second group consists of 6 functions: Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2. The third group consists of 10 functions: Kowalik, Six Hump Camel, Branin, Goldstein-Price, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10. The specification of the 23 functions is shown in Table 1.

In this first simulation, the proposed algorithm is compared with five other metaheuristic algorithms: PSO, MPA, KMA, stochastic Komodo algorithm (SKA) [33], and ASBO. PSO represents the old-fashioned algorithm [17] but is proven and widely used. On the other hand, MPA, KMA, SKA, and ASBO represent the shortcoming metaheuristic algorithms. Specifically, PSO is chosen due to its advantage as a metaheuristic that promotes guided movement toward the global best solution [17]. MPA is chosen due to its distinct mechanism that uses iteration to control the exploration-exploitation [24]. KMA was chosen due to its strategy that combines the crossover and the guided movement [25]. SKA is chosen as the improved version of the KMA. In SKA, the big males, females, and the small males are selected randomly [33], rather than based on

Table 1. Benchmark functions

No	Function	Model	Dim	Space	Target
1	Sphere	$\sum_{i=1}^{d} x_i^2$	10	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$	10	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	10	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i\rvert, 1 \le i \le d\}$	10	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1}(100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2)$	10	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1}(x_i + 0.5)^2$	10	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\ [0,1]$	10	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	10	[-500, 500]	-4189.8
9	Ratsrigin	$10d + \sum_{i=1}^{d}\left(x_i^2 - 10cos(2\pi x_i)\right)$	10	[-5.12, 5.12]	0
10	Ackley	$-20 \cdot exp\left(-0.2 \cdot \sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos 2\pi x_i\right) + 20 + exp(1)$	10	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	10	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d-1}\left((y_i - 1)^2\left(1 + 10\, sin^2(\pi y_{i+1})\right)\right) + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10{,}100{,}4)$	10	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{d-1}\left((x_i - 1)^2\left(1 + sin^2(3\pi x_{i+1})\right)\right) + (x_d - 1)^2\left(1 + sin^2(2\pi x_d)\right)\right\} + \sum_{i=1}^{d} u(x_i, 5{,}100{,}4)$	10	[-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	$\left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right).\left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right)$	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 2. Simulation result

Fun.	PSO	MPA	KMA	SKA	ASBO	FS-ASBO	Better Than
1	1.033×10^{3}	1.053	5.627×10^{2}	2.603	2.687×10^{-27}	**2.987×10^{-56}**	PSO, MPA, KMA, SKA, ASBO
2	8.863×10^{-2}	**0**	7.149×10^{2}	1.103×10^{-3}	**0**	**0**	PSO, KMA, SKA
3	1.898×10^{3}	5.984	1.434×10^{3}	3.104×10^{1}	1.107×10^{-7}	**6.219×10^{-56}**	PSO, MPA, KMA, SKA, ASBO
4	1.363×10^{1}	8.067×10^{-1}	1.379×10^{1}	1.523	5.767×10^{-11}	**7.799×10^{-29}**	PSO, MPA, KMA, SKA, ASBO
5	1.960×10^{5}	1.664×10^{1}	5.326×10^{4}	1.183×10^{2}	**8.610**	8.998	PSO, MPA, KMA, SKA
6	7.039×10^{2}	4.294	3.808×10^{2}	1.142	**1.378×10^{-4}**	5.527×10^{-1}	PSO, MPA, KMA, SKA
7	4.417×10^{-2}	8.663×10^{-3}	2.288×10^{-1}	2.258×10^{-2}	**6.509×10^{-3}**	1.492×10^{-2}	PSO, KMA, SKA
8	-1.660×10^{3}	-1.876×10^{3}	**-3.377×10^{3}**	$-2{,}445 \times 10^{3}$	-2.317×10^{3}	-2.408×10^{3}	PSO, MPA, ASBO
9	5.328×10^{1}	1.301	3.703×10^{1}	1.273×10^{1}	**6.774×10^{-1}**	1.774	PSO, KMA, SKA
10	8.767	**6.157×10^{-1}**	8.587	4.835	1.013	1.805	PSO, KMA, SKA
11	8.959	4.683×10^{-1}	6.369	5.607×10^{-1}	7.309×10^{-2}	**6.491×10^{-2}**	PSO, MPA, KMA, SKA, ASBO
12	5.084×10^{2}	1.130	1.014×10^{1}	6.322	**3.806×10^{-3}**	2.216×10^{-2}	PSO, MPA, KMA, SKA
13	1.192×10^{5}	3.382	6.075×10^{4}	2.242×10^{-1}	**2.937**	4.295	PSO, MPA, KMA
14	5.736	5.280	1.531×10^{1}	9.777	1.190	**1.086**	PSO, MPA, KMA, SKA, ASBO
15	1.975×10^{-2}	3.690×10^{-3}	1.502×10^{-2}	**6.190×10^{-4}**	6.331×10^{-2}	3.504×10^{-3}	PSO, MPA, KMA, ASBO
16	-1.028	-1.024	-1.027	**-1.032**	-1.185×10^{-1}	-1.023	ASBO
17	5.942×10^{-1}	8.417×10^{-1}	6.018×10^{-1}	**3.981×10^{-1}**	6.438×10^{-1}	4.083×10^{-1}	PSO, MPA, KMA, ASBO
18	5.402	5.106	3.036	5.613	**3**	**3**	PSO, MPA, KMA, SKA
19	-1.581×10^{-1}	**-3.860**	-8.977×10^{-1}	-2.866×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}	SKA
20	-2.519	-1.970	-2.864	**-3.255**	-1.436	-3.066	PSO, MPA, KMA, ASBO
21	-3.879	-1.846	-7.396	-6.768	-8.744	**-1.015×10^{1}**	PSO, MPA, KMA, SKA, ASBO
22	-4.472	-1.721	-8.089	-8.068	-8.688	**-1.023×10^{1}**	PSO, MPA, KMA, SKA, ASBO
23	-4.163	-2.002	-7.299	-7.324	-8.270	**-8.531**	PSO, MPA, KMA, SKA, ASBO

the rank as in the original form of KMA [25]. ASBO is chosen to compare the performance of the proposed algorithm, which is the improved version of ASBO with its original form.

The setup for the adjusted parameters is as follows. The population size is 20. The maximum iteration is 100. These two parameters are applied to all algorithms. Specifically, in FS-ASBO, the number of candidates is 10 and all step sizes are 0.5. In PSO, all weights are set 0.5 to implement the balanced movement. In PSO, the big male is 20% of the population and there is only one female. The mlipir rate is set 0.2. In MPA, the fishing aggregate device is 0.2. In SKA, the big male weight is 0.5 while the small male weight is 0.75. The threshold between the big male and the female is 0.2. Meanwhile, the threshold between the female and the small male is 0.4. The simulation result is shown in Table 2. The best result is written in bold font.

Table 2 shows that the proposed algorithm has met the requirements for a metaheuristic algorithm. This algorithm can find the acceptable solution for all 23 functions. The proposed algorithm also finds the optimal global solution in three functions: Schwefel 2.22, Goldstein-Price, and Shekel 5.

Table 2 also shows that the proposed algorithm is competitive enough compared with other algorithms. It outperforms all sparing algorithms in solving seven functions. Two functions are the high dimension unimodal functions, one function is the high dimension multimodal function, and three functions are the fixed dimension multimodal

Table 3. Convergence result

Funt.	t_{max} = 70	t_{max} = 140	t_{max} = 210
1	3.689×10^{-38}	2.478×10^{-80}	1.781×10^{-122}
2	0	0	0
3	6.825×10^{-38}	4.087×10^{-80}	4.583×10^{-122}
4	8.563×10^{-20}	7.307×10^{-41}	6.386×10^{-62}
5	8.998	8.995	8.997
6	6.199×10^{-1}	3.593×10^{-1}	4.112×10^{-1}
7	1.403×10^{-2}	8.968×10^{-3}	1.384×10^{-2}
8	-2.267×10^{3}	-2.405×10^{3}	-2.421×10^{3}
9	2.182	1.273	1.273
10	1.738	1.700	1.535
11	8.838×10^{-2}	6.116×10^{-2}	4.552×10^{-2}
12	2.498×10^{-2}	2.142×10^{-2}	1.249×10^{-2}
13	5.109	4.111	3.568
14	4.881	3.939	3.854
15	4.081×10^{-3}	3.568×10^{-3}	2.776×10^{-3}
16	-1.023	-1.029	-1.030
17	4.083×10^{-1}	4.080×10^{-1}	4.072×10^{-1}
18	3	3	3
19	-4.954×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}
20	-3.059	-3.056	-3.120
21	-9.226	-10.153	-10.153
22	-8.953	-10.403	-10.403
23	-8.569	-10.044	-10.536

functions. Its gap is very wide in solving three unimodal functions: Sphere, Schwefel 1.2, and Schwefel 2.21. The proposed algorithm is better than the original ASBO in solving 13 functions while draw in two functions. It outperforms PSO, MPA, KMA, and SKA in solving 21, 17, 20, and 17 functions respectively.

In the second simulation, the convergence behaviour of the proposed algorithm is observed. In this simulation, there are three values of the maximum iteration: 70, 140, and 210. In this simulation, the proposed algorithm is still implemented to solve 23 benchmark functions as in the first simulation. The result is shown in Table 3.

Table 3 shows that the proposed algorithm generally reaches its convergence in the earlier iteration. The proposed algorithm reaches convergence in the low iteration in solving 16 functions. Most of the converged functions in the early iteration are multimodal functions. Meanwhile, the result is improved in high maximum iteration in solving three unimodal functions: Sphere, Schwefel 1.2, and Schwefel 2.21.

The proposed algorithm is implemented in the third simulation to solve the housing development optimization problem. In this problem, the developer should optimize the land utilization. Land utilization can be achieved by developing houses as many as possible [34]. Land becomes a limited resource in this problem, becoming a constraint [34].

In the housing development problem, developer develops several house types. Every house type needs specific land use, cost, and price. The gross profit is obtained by subtracting the price from the cost. In the optimization problem, the house type represents the dimension. Although some house types are more profitable than others, they need wider land. Moreover, some houses with the same land use create other gross profit. Although some house types are more profitable than others, the developer must build all house types. This circumstance should be done to meet various customer segments and needs [35]. Some customers may need a house with a wider house because they have more family members. Contrary, some customers have a limited budget, so they need more affordable house types.

The scenario of this simulation is as follows. A developer should utilize 300,000 square-meter land allocated for houses. Land used for public infrastructures, such as roads, playgrounds, sports clubs, and so on is excluded from this land. This developer will develop three house types. The number of houses that must be built for every house type ranges from 100 to 1,200 units. The detailed specification for every house type is shown in Table 4. The objective is to maximize the total gross profit. The total gross profit is obtained by accumulating the gross profit for all built houses.

The simulation setup for this problem is as follows. The population size is set at 20. There are two values for the maximum iteration: 20 and 40. Due to the number of house types, this dimension is 3. Like in the first simulation, in this third simulation, the proposed algorithm is compared with ASBO, PSO, MPA, KMA, and SKA. The result is shown in Table 5

Table 4. House type specification [35]

House Type	Land Use (m^2)	Gross Profit (million rupiah)
type 42	108	28.2
type 45	108	52.0
type 54	120	99.8

Table 5. Real world housing optimization problem

Method	Total Gross Profit	
	t_{max} = 20	t_{max} = 40
FS-ASBO	188,950	189,004
ASBO	171.099	169,357
PSO	176.179	174,684
MPA	171.086	173,291
KMA	169.542	169,506
SKA	171,486	176,488

Table 5 shows that the proposed algorithm outperforms all other algorithms: ASBO, PSO, MPA, KMA, and SKA. It occurs in both maximum iterations. In the first maximum iteration, it is 10%, 7%, 10%, 11%, and 10% better than ASBO, PSO, MPA, KMA, and SKA respectively. In the second maximum iteration, it is 12%, 8%, 9%, 11%, and 7% better than ASBO, PSO, MPA, KMA, and SKA respectively.

5. Discussion

In this section, several findings related to this work will be discussed. These findings are obtained from the simulation result. First, in general, the proposed algorithm successfully becomes a good metaheuristic algorithm. It overcomes the challenge of finding the near-optimal or acceptable solution within the given iteration. This circumstance occurs in both unimodal and multimodal functions. Moreover, the problem dimension and space do not affect its performance. Its performance is still good enough in solving problems with narrow problem space, such as Goldstein-Price, and large problem space, such as Schwefel.

Second, overall, the proposed algorithm is superior in solving unimodal functions, as shown in the first group in Table 2. Due to its extreme superiority in solving half of the unimodal functions, it is shown that the precision of the solution found by the proposed algorithm is very high compared with the other algorithms. As floating point-based functions, the high precision result comes from the high precision solution.

Third, the proposed algorithm is competitive in solving the multimodal functions. However, its competitiveness is not so high as solving the unimodal functions. Compared with the ASBO, the proposed algorithm is superior in solving functions in the first and third groups. On the other hand, the original ASBO is superior in solving functions in the second group. Fortunately, the performance gap between the proposed algorithm and the original ASBO, where the ASBO is better than the proposed algorithm, is not significant.

Fourth, the simulation result shows that the convergence aspect of the proposed algorithm is good. The proposed algorithm can reach an acceptable and stable solution in low maximum iteration. It means that the computational consumption of the proposed algorithm is less than other sparing algorithms.

Fifth, the proposed algorithm is competitive in solving theoretical mathematic problems and real-world optimization problems. The real-world optimization problem, especially in the operation research, is very different from the theoretical mathematic problems, as shown in the 23 benchmark functions. Many problems in operations research are simple. The solution space in these many problems is an integer, such as the number of vehicles, number of products, stocks portfolio, etc.

The objective can be formalized as simple as linear functions, simpler than the Sphere or Schwefel 2.22. Many of them are just accumulating the score of each parameter within the solution, as shown in the housing optimization problem. For example, in the production process, the objectives are minimizing total tardiness, make-span, total cost, etc. The other example, in the transportation optimization problem, the objectives are minimizing travel distance, transportation cost, energy consumption, etc. In several optimization studies, the objective is just minimizing the number of unserved requests.

Based on this difference and its simplicity, the superiority of the proposed algorithm is not so high as the other sparing algorithms. In the theoretical mathematic problem, it is easy to see that the proposed algorithm is much better than the sparing algorithms. The proposed algorithm's superiority is less than 15% in solving the real-world optimization problem.

Even though this proposed algorithm has been tested in solving a housing optimization problem, many real-world optimization problems have been tested. It becomes the limitation of this work. First, this algorithm has not been tested in solving combinatorial problems, such as scheduling and timetabling. On the other hand, these operations research problems are implemented in many areas, from the production process to the university operations in arranging the classes, courses, and lecturers. This proposed algorithm still needs modification to solve a combinatorial problem. Second, this algorithm also needs to be implemented to solve the mechanical optimization problem where the problem space can be low precision floating point number and the objective function is more complex.

6. Conclusion

This work has demonstrated that the proposed algorithm, as the improvement of the average and subtraction-based optimizer, performs well as a metaheuristic algorithm. Its performance is proven to solve theoretical mathematic problems and real-world optimization problems. The proposed algorithm outperforms all sparing algorithms in solving seven functions. Its superiority is extreme in solving three functions: Sphere, Schwefel 1.2, and Schwefel 2.21. Compared with the original ASBO, the proposed algorithm is better in solving 11 functions and draw in solving 2 functions. Moreover, the proposed algorithm successfully achieves a globally optimal solution in three functions. The algorithm also outperforms all sparing algorithms in creating total gross profit in solving the housing optimization problem. Its performance is up to 12%, 8%, 10%, 11%, and 10% better than ASBO, PSO, MPA, KMA, and SKA respectively.

There will be many research potentials related to this work in the future. This algorithm should be implemented to solve many other real-world optimization problems. Besides, improving this algorithm by hybridizing this basic FS-ASBO with other methods is also challenging.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization: Kusuma; methodology: Kusuma; software: Kusuma, validation: Kusuma; formal analysis: Kusuma and Dinimaharawati; investigation: Kusuma; data curation: Kusuma; writing-original paper draft: Kusuma; writing-review and editing: Dinimaharawati; project administration: Dinimaharawati; funding acquisition: Kusuma.

Acknowledgments

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

References

[1] Y. Sun and X. Qi, "A DE-LS Metaheuristic Algorithm for Hybrid Flow-shop Scheduling Problem Considering Multiple Requirements of Customers", *Scientific Programming*, ID: 8811391, pp. 1-14, 2020.

[2] Z. A. Abdalkareem, A. Amir, M. A. A. Betar, P. Ekhan, and A. I. Hammouri, "Healthcare Scheduling in Optimization Context: A Review", *Health and Technology*, Vol. 11, pp. 445-469, 2021.

[3] P. D. Kusuma and M. Kallista, "Multi-Depot Capacitated Vehicle Routing Problem by Using Stable Marriage and K-Means Clustering to Minimize Number of Unserved Customers and Total Travel Distance", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 6, pp. 605-615, 2021, doi: 10.22266/ijies2021.1231.54.

[4] M. Lindahl, A. Mason, T. Stidsen, and M. Sorensen, "A Strategic View of University Timetabling", *European Journal of Operations Research*, Vol. 266, No. 1, pp. 35-45. 2018.

[5] M. Li, Z. Qin, Y. Jiao, Y. Yang, Z. Gong, J. Wang, C. Wang, G. Wu, and J. Ye, "Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning", In: *Proc. of The Web Conference*, 2019.

[6] K. Geng, C. Ye, L. Cao, and L. Liu, "Multi-Objective Reentrant Hybrid Flowshop Scheduling with Machines Turning On and Off Control Strategy Using Improved Multi-verse Optimizer Algorithm", *Mathematical Problems in Engineering*, ID: 2573873, pp. 1-18, 2019.

[7] A. K. M. F. Ahmed and J. U. Sun, "Bilayer Local Search Enhanced Particle Swarm Optimization for Capacitated Vehicle Routing Problem", *Algorithms*, Vol. 11, article ID: 31, pp. 1-22, 2018.

[8] N. Farmand, H. Zarei, and M. R. Barzoki, "Two Meta-heuristic Algorithms for Optimizing a Multi-objective Supply Chain Scheduling Problem in an Identical Parallel Machines Environment", *International Journal of Industrial Engineering Computations*, Vol. 12, pp. 249-272, 2021.

[9] I. S. Lee, "A Scheduling Problem to Minimize Total Weighted Tardiness in the Two-stage Assembly Flowshop", *Mathematical Problems in Engineering*, ID: 9723439, pp. 1-10, 2020.

[10] K. Ouaddi, Y. Benadada, and F. Z. Mhada, "Ant Colony System for Dynamic Vehicle Routing Problem with Overtime", *International Journal of Advanced Computer Science and Application*, Vol. 9, No. 6, pp. 306-315, 2018.

[11] Y. H. Santana, R. M. Alonso, G. G. Nieto, L. Martens, W. Joseph, and D. Plets, "Indoor Genetic Algorithm-Based 5G Network Planning Using a Machine Learning Model for Path Loss Estimation", *Applied Sciences*, Vol. 12, ID: 3923, pp. 1-18, 2022.

[12] M. A. Sobhy, A. Y. Abdelaziz, H. M. Hasanien, and M. Ezzat, "Marine Predators Algorithm for Load Frequency Control of Modern Interconnected Power Systems Including Renewable Energy Sources and Energy Storage Units", *Ain Shams Engineering Journal*, Vol. 12, No. 4, pp. 3843-3857, 2021.

[13] M. A. Elaziz, A. A. Ewees, D. Yousri, H. S. N. Alwefali, Q. A. Awad, S. Lu, and M. A. A. A. Qanees, "An Improved Marine Predators Algorithm with Fuzzy Entropy for Multi-Level

Thresholding: Real World Example of COVID-19 CT Image Segmentation", *IEEE Access*, Vol. 8, pp. 125306-125330, 2020.

[14] H. R. Moshtaghi, A. T. Eshlagy, and M. R. Motadel, "A Comprehensive Review on Meta-Heuristic Algorithms and Their Classification with Novel Approach", *Journal of Applied Research on Industrial Engineering*, Vol. 8, No. 1, pp. 63-69, 2021.

[15] 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.

[16] A. M. F. Fard, M. H. Keshteli, and R. T. Moghaddam, "Red Deer Algorithm (RDA): A New Nature-Inspired Meta-Heuristic", *Soft Computing*, Vol. 19, 14638-14665, 2020.

[17] D. Freitas, L. G. Lopes, and F. M. Dias, "Particle Swarm Optimization: A Historical Review up to the Current Developments", *Entropy*, Vol. 22, pp. 1-36, 2020.

[18] Y. Celik, "An Enhanced Artificial Bee Colony Algorithm based on Fitness Weighted Search Strategy", *Journal for Control, Measurement, Electronics, Computing, and Communications*, Vol. 62, No. 3-4, pp. 300-310, 2021.

[19] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm", *Sensors*, Vol. 21, ID: 5214, pp. 1-30, 2021.

[20] G. Dhiman and V. Kumar, "Spotted Hyenna Optimizer: A Novel Bio-inspired based Metaheuristic Technique for Engineering Applications", *Advances in Engineering Software*, Vol. 114, No. 10, pp. 48-70, 2017.

[21] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm", *Advances in Engineering Software*, Vol. 95, No. 12, pp. 51-67, 2016.

[22] F. Rezaei, H. R. Safavi, M. A. Elaziz, S. H. A. E. Sappagh, M. A. A. Betar, and T. Abuhmed, "An Enhanced Grey Wolf Optimizer with a Velocity-Aided Global Search Mechanism", *Mathematics*, Vol. 10, ID: 351, pp. 1-32, 2022.

[23] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A New Bio-Inspired based Metaheuristic Paradigm for Global Optimization", *Engineering Applications of Artificial Intelligence*, Vol. 90, No. 2, ID: 103541, 2020.

[24] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic", *Expert System with Applications*, Vol. 152, ID: 113377, 2020.

[25] Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm", *Applied Soft Computing*, Vol. 114, ID: 108043, 2022.

[26] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "A New Optimization Algorithm based on Average and Subtraction of the Best and Worst Members of the Population for Solving Various Optimization Problems", *PeerJ Computer Science*, Vol. 8, ID: e910, pp. 1-29, 2022.

[27] 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, pp. 1-17, 2021.

[28] M. Dehghani, M. Mardaneh, J. S. Guerrero, O. P. Malik, and V. Kumar, "Football Game Based Optimization: An Application to Solve Energy Commitment Problem", *International Journal of Intelligent Engineering & Systems*, Vol. 13, No. 5, pp. 514-523, 2020, doi: 10.22266/ijies2020.1031.45.

[29] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts Game Optimizer: A New Optimization Technique Based on Darts Game", *International Journal of Intelligent Engineering & Systems*, Vol. 13, No. 5, pp. 286-294, 2020, doi: 10.22266/ijies2020.1031.26.

[30] M. Dehghani, Z, Montazeri, O. P. Malik, H. Givi, and J. M. Guerrero, "Shell Game Optimization: A Novel Game-Based Algorithm", *International Journal of Intelligent Engineering & Systems*, Vol. 13, No. 3, pp. 246-255, 2020, doi: 10.22266/ijies2020.0630.23.

[31] M. Dehghani, Z. Montazeri, S. Saremi, A. Dehghani, O. P. Malik, K. A. Haddad, and J. M. Guerrero, "HOGO: Hide Objects Game Optimization", *International Journal of Intelligent Engineering & Systems*, Vol. 13, No. 4, pp. 216-225, 2020, doi: 10.22266/ijies2020.0831.19.

[32] K. Hussain, M. N. M. Salleh, S. Cheng, and R. Naseem, "Common Benchmark Functions for Metaheuristic Evaluation: A Review", *International Journal on Informatic Visualization*, Vol. 1, No. 4-2, pp. 218-223, 2017.

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

Received: May 2, 2022. Revised: May 17 2022.

[34] A. Utiarahman and A. S. Rauf, “Case Study of Housing Development with Optimization of Housing Land Utilization by Using Linear Programs”, *International Journal of Innovative Science and Research Technology*, Vol. 3, No. 11, pp. 307-313, 2018.

[35] A. C. Murti, M. I. Ghozali, and W. H. Sugiharto, “Optimization Model for Subsidized Housing using Linear Programming Associated with Land Suitability”, *Journal of Physics: Conference Series*, Vol. 1430, ID: 012056, 2020.