Received: March 22, 2022. Revised: April 26, 2022. 

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Stochastic Komodo Algorithm

**Purba Daru Kusuma**[1*] **Meta Kallista**[1]

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

**Abstract:** A novel metaheuristic algorithm is proposed in this paper, namely stochastic komodo algorithm (SKA). This proposed algorithm is an improved version of Komodo mlipir algorithm (KMA), which is inspired by the behaviour of Komodo during foraging and mating. The improvement is conducted by simplifying the basic form of KMA. Like KMA, it consists of three types of Komodo: big male, female, and small male. Male Komodo focuses on intensification. On the other side, females conduct diversification or intensification based on the search space radius. It eliminates sorting mechanism at the beginning of the iteration. Rather than determined from the quality (fitness score), the distribution of the types of Komodo is conducted stochastically at the beginning of every iteration. This proposed algorithm is then tested by using ten functions. Five functions are unimodal, while the five others are multimodal. The proposed algorithm is also compared with several well-known algorithms: football game-based optimization, hide objects game optimization, cloud-theory-based simulated annealing, harmony search, and KMA. The result shows that this proposed algorithm is very competitive compared with these benchmark algorithms in both unimodal and multimodal functions. A female-dominant formation is proven to achieve optimal result.

**Keywords:** Komodo mlipir algorithm, Metaheuristic, Multi-agent, Multimodal, Swarm intelligence.

## 1. Introduction

Optimization is one of the popular subjects. It is implemented in many areas, such as manufacturing [1], logistics [2], transportation [3], education [4], communication [5], multimedia [6], and so on. In general, the objective of optimization is to find a solution, formation, or configuration that gives the optimal score, which can be either a minimum value (minimization) or a maximum value (maximation), depending on its objective. Therefore, there are many optimization methods, both mathematical and computational.

Metaheuristic algorithm is one popular method in optimization studies. This algorithm adopts an approximate approach. As an approximate method, a true optimal solution is not guaranteed to be found [7]. The metaheuristic algorithm focuses on finding near-optimal, sub-optimal, or acceptable solution. This is different from the exact method where true optimal solution is guaranteed to be found. But the exact method is not feasible to solve large space or high dimensional problems due to excessive computational consumption [7]. Fortunately, a metaheuristic algorithm can solve this problem. But, as an approximate method, it can be trapped in a local optimal solution while the true optimal solution is still somewhere else in the problem space.

Many shortcoming studies that propose new metaheuristic algorithm are trapped into three conditions. The first one is that many shortcoming algorithms use metaphors rather than declare distinct mechanism as the algorithm name [8]. These shortcoming algorithms used nature as the source of inspiration, especially animals, such as marine predator [9], monkey [10], whale [11], deer [12], penguin [13], dolphin [14], grasshopper [15], butterfly [16], and so on. The second one is that these algorithms become complicated in the process and in the calculation. The third one is that these studies focused on beating the previous algorithms by providing better result in achieving a near-optimal solution.

The examples are as follows. Fatholahi-Fard, Hajiaghaei-Keshteli, and Tavakkoli-Moghaddam [12] compared their proposed algorithm, red deer

algorithm (RDA) with genetic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO), imperialist competitive algorithm (ICA), and firefly algorithm (FA). Ding, Chang, Li, Feng, and Zhang [11] compared their algorithm, mixed-strategy-based whale optimization algorithm (MSWOA), with GA, PSO, gravitational search algorithm (GSA), ant lion optimization (ALO), whale optimization algorithm (WOA), and enhanced whale optimization algorithm (EWOA). Braik, Sheta, and Al-Hiary [10] compared their algorithm, capuchin search algorithm (CSA), with PSO, multi-verse optimizer (MVO), sine cosine algorithm (SCA), GSA, GA, and harmony search (HS). Dehghani, Dehghani, Mardaneh, Guerrero, Malik, and Kumar [18] compared their algorithm, football game-based optimization (FBGO), with GA, PSO, teaching-learning based optimization (TLBO), grasshopper optimization algorithm (GOA), emperor-penguins colony optimization (EPO), and shell game optimization (SGO), and hide objects game optimization (HOGO). Dehghani, Montazeri, Saremi, Dehghani, Malik, Al-Haddad, and Guerrero [19] compared their other algorithm, HOGO, with GA, PSO, GSA, TLBO, grey wolf optimization (GWO), GOA, spotted hyena optimizer (SHO), and EPO.

All these studies stated that their proposed algorithm outperformed the benchmark algorithms. Ironically, although these old-fashioned algorithms have been beaten many times by many shortcoming algorithms, they are still widely used in many optimization studies.

One latest metaheuristic algorithm is Komodo mlipir algorithm (KMA). This algorithm is inspired by the behavior of Komodo dragon, a monitor lizard that lives on Komodo island, Indonesia [20]. It is a combination between swarm intelligence and evolutionary algorithm. In it, the population of Komodo is divided into three groups: big male, female, and small male [20]. Each Komodo behaves based on the characteristics of the group. In this work, Suyanto, Ariyanto, and Ariyanto [20], who proposed this algorithm, compared their algorithm with several metaheuristic algorithms: genetic algorithm (GA), success-history based parameter adaptation differential evolution (SHADE), LSHADE with ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood (LSHADE-cnEpSin), equilibrium optimizer (EO), marine predator algorithm (MPA), and slime mold algorithm (SMA). Like other studies, this work claimed that KMA is better than these algorithms.

Despite its positive result in achieving near optimal and optimal solution, there are several notes for KMA. Both intensification and diversification are conducted by several mechanisms so that there is redundancy. Meanwhile, the effectiveness of every mechanism and parameter in this algorithm has not been explored yet. Calculation in KMA is also complicated. Based on these circumstances, there is a challenge to simplify this algorithm without losing its performance significantly.

Based on this problem, this work aims to propose a new metaheuristic algorithm based on KMA, namely stochastic Komodo algorithm (SKA). The contribution of this work is as follows.

1) This work modifies the KMA by simplifying the algorithm and reducing its redundancy.
2) This work investigates parameters in the SKA, which some of them are like KMA, and analyses their sensitivity.

The remainder of this paper is organized as follows. The detail model of KMA and the prospect of its improvement is explained in section two. The proposed algorithm is described in section three. The simulation and the result are shown in section four. More profound analysis and its findings are discussed in section five. The conclusion and future work are summarized in section six.

## 2. Related works

KMA is a metaheuristic algorithm that is inspired by the behavior of Komodo dragon [20]. Komodo dragon is a kind of monitor lizard that lives mostly on Komodo island, Indonesia. The big male Komodo eats prey such as deer, cow, goat, and so on. During eating, some other Komodo are attracted to join. But the big Komodo refuses to share. Some small Komodo eats prey that is left by the big Komodo. Female produces offspring by two ways. The first one is by mating with a male. The second one is called parthenogenesis or asexual reproduction. The KMA is developed based on this behavior.

In general, KMA is a population-based metaheuristic algorithm. In it, there are a several individuals in a population. KMA is also a swarm intelligence. As a swarm intelligence, there are certain number of agents (Komodo) that act autonomously to find a near-optimal or sub-optimal solution within the problem space [21]. Although each agent acts autonomously, there is collective intelligence that is shared among them [21]. But in KMA, this swarm intelligence is hybridized with evolutionary intelligence, where better solutions are produced by combining some selected previous solutions.

In KMA, the population of Komodo is divided into three types or groups: big male, small male, and female [20]. The main role of the big male is conducting intensification with certain diversification.

The small male focuses on intensification. Meanwhile, the female can engage in both intensification and diversification. The distribution of the Komodo depends on the quality of solution that is sorted in every iteration. High quality solutions become big males. A moderate quality solution becomes female. Low quality solutions become small male. The more detailed behavior of every type of Komodo will be described below.

There are two forces or circumstances conducted to the big male: attraction and distraction [20]. Attraction means a big male is attracted to other big males. Distraction means the big male avoids interaction with other big males. In general, a big male moves closer to other big males whose fitness score is better or by a certain probabilistic number. Else, this big male chooses to move farther away from the related other big males. This process is conducted by the accumulation of weighted vector between a big male and all other big males.

There are two actions that can be chosen by female Komodo in every iteration: mating with the best big male or asexual reproduction (parthenogenesis) [20]. The mating process is like cross-over in genetic algorithm. There are two new solutions produced by this mating process. The first solution is relatively near the female while the second solution is relatively near the best big male. The chosen offspring is a solution whose fitness score is better. In the parthenogenesis process, a female's next position is determined stochastically around its search space. The mating process is an intensification, while parthenogenesis can be either an intensification or diversification, depending on the search space size.

In general, the small male follows the big male [20]. This process is conducted by accumulating vectors between the small male and all the big males with a certain speed. But, in the multi-dimensional problem, not all dimensions are considered. The number of dimensions that are considered is determined by the mlipir rate, which is related to the number of dimensions that are modified.

Different from many algorithms, KMA implements a population adaptation scheme. In this scheme, the population size fluctuates depending on the circumstance. If stagnation occurs, the population size will increase. On the other hand, when there exist successive improvements, the population size decreases. The population size may increase until the maximum population is reached. On the other hand, the population size may decrease until a minimum population is reached.

There are several notes related to this original model of KMA. Several mechanisms are intensification while several others are diversification. It seems that there is duplication in both intensification and diversification. It is different from many well-known algorithms, such as genetic algorithm, simulated annealing, tabu search, harmony search, invasive weed optimization, and so on.

The division of an agent in KMA is like the division in red deer algorithm (RDA) or grey wolf optimizer (GWO). In RDA, the population is divided into commanders, hinds, and harems [12]. In GWO, there are four types of agents: alpha, beta, delta, and omega [22]. In all these algorithms, every type of agent works parallelly in every iteration. It is different from the artificial bee colony (ABC). In ABC, there are three phases of bees, employed bee, onlooker bee, and scout bee [23]. These three types are conducted serially [23]. Moreover, the dominance of the types of Komodo has not been explored. These notes then become the baseline to propose the improved version of KMA as stated in this work.

## 3. Model

The concept behind this proposed algorithm is as follows. Like KMA, a group of Komodo is divided into three types: big male, female, and small male. But the main difference between KMA and the proposed algorithm is that in the proposed algorithm, the role of every Komodo is determined stochastically in every iteration. On the other hand, in KMA, the division of the role is dynamic in every iteration based on the fitness score of each Komodo. The objective of this stochastic approach is to give every agent a chance to conduct various roles in every iteration without considering their fitness score. Moreover, this stochastic division eliminates the sorting process at the beginning of every iteration. That is why this proposed algorithm is named as stochastic Komodo algorithm (SKA).

The actions of every Komodo or agent are as follows. The big male always tries to get closer to other Komodo whose fitness score is better by accumulating the vector between this big male and all other Komodo whose fitness score is better. The female conducts parthenogenesis by creating a certain number of candidates (not only two) around it. Then the best candidate is selected from among them. If the fitness score of the best candidate is better than this female, this best candidate will replace the female. The small male tries to get closer to the highest quality Komodo with a certain speed. In this proposed algorithm, mlipir mechanism is eliminated. Every time a Komodo finishes its action, the highest quality Komodo is updated. The highest quality Komodo is Komodo whose fitness score is the best. This process

**Algorithm 1: stochastic Komodo algorithm**

**output**: $k_{best}$
//initiation
**for** i = 1 **to** $n(K)$
  $k_i$ = initialize($P$)
  $k_{best}$ = global-update ($k_i$, $k_{best}$)
**end for**
//iteration
**for** $t$ = 1 **to** $t_{max}$
 **for** $i$ = 1 **to** $n(K)$
  $r = U(0, 1)$
  If $r < g_1$ **then**
   $s(k_i)$ = big male
  **else**
   **if** $r < g_2$ **then**
    $s(k_i)$ = female
   **else**
    s($k_i$) = small male
  **if** $s(k_i)$ = big male **then**
   $k_i$ = big-male-move ($k_i$, $K$)
   $k_{best}$ = global-update ($k_i$, $k_{best}$)
  **if** $s(k_i)$ = female **then**
   **for** $j$ = 1 **to** $n(C)$
    $c_j$ = generate-candidate ($k_i$)
   **end for**
   $c_{best}$ = find-best-candidate ($C$)
   **if** $f(c_{best}) < f(k_i)$ then
    $k_i = c_{best}$
    $k_{best}$ = global-update ($k_i$, $k_{best}$)
  **if** $s(k_i)$ = small-male **then**
   $k_i$ = small-move ($k_i$, $k_{best}$)
 **end for**
**end for**

| | |
|---|---|
| $b_l$, $b_u$ | lower bound, upper bound |
| $c$ | candidate |
| $C$ | set of candidates |
| $g_1$ | threshold between big male and female |
| $g_2$ | threshold between female and small male |
| $k$ | komodo |
| $K$ | set of Komodo |
| $k_{best}$ | the best Komodo |
| $P$ | problem space |
| $r_s$ | search space ratio |
| $s$ | type of Komodo |
| $n_{bet}$ | number of better Komodo |
| $t$ | time / iteration |
| $t_{max}$ | maximum iteration |
| $U$ | uniform random |
| $w_1$, $w_2$ | big male weight, small male weight |

is conducted in every iteration until the maximum iteration is reached. In the end, the highest quality Komodo becomes the final solution.

This concept is then transformed into a mathematical model. Several annotations used in this model are as follows. Moreover, the algorithm of SKA is shown in Algorithm 1.

The explanation of Algorithm 1 is as follows. In the beginning, all Komodo are initialized randomly within the problem space. This process is formalized in Eq. (1). Then, best Komodo is updated. At the beginning of every iteration, a random number is generated. This random number is used to determine whether a Komodo becomes big male, female, or small male. Every big male or small male moves based on its characteristics. Meanwhile, every female generates candidates, select the best candidate, and evaluating whether this best candidate will replace this female. After finishing their job, the best Komodo is updated.

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

The big male updates its position based on all the other Komodo whose fitness is better than it. If there not exist Komodo whose fitness score is better, then this Komodo stays in its current location. This mechanism is formalized by using Eq. (2) and Eq. (3). Eq. (2) shows that if there exists Komodo whose fitness score is better than the big male, the candidate is determined by calculating the average position of all Komodo whose fitness score is better. Eq. (3) states that the new location of the big male is determined by the summation of its weighted current location and its weighted location of its candidate.

$$c_{b,i} = \begin{cases} 0, n_{bet,i} = 0 \\ \frac{\sum_{n_{bet,i}} k_j}{n_{bet,i}}, else \end{cases} \tag{2}$$

$$k_i' = w_1 k_i + (1 - w_1) c_{b,i} \tag{3}$$

The female generates several candidates near it. The location of every candidate is determined randomly and follows uniform distribution. This mechanism is formalized by using Eq. (4). Then, the location of this candidate is fitted so that it is still inside the problem space.

$$c_i = k_i + U(-0.5,0.5) r_s (b_u - b_l) \tag{4}$$

The small male updates its location based on its current location and the location of the best Komodo. This is formalized by using Eq. (5). Meanwhile, the updating process of the best Komodo is formalized by using Eq. (6).

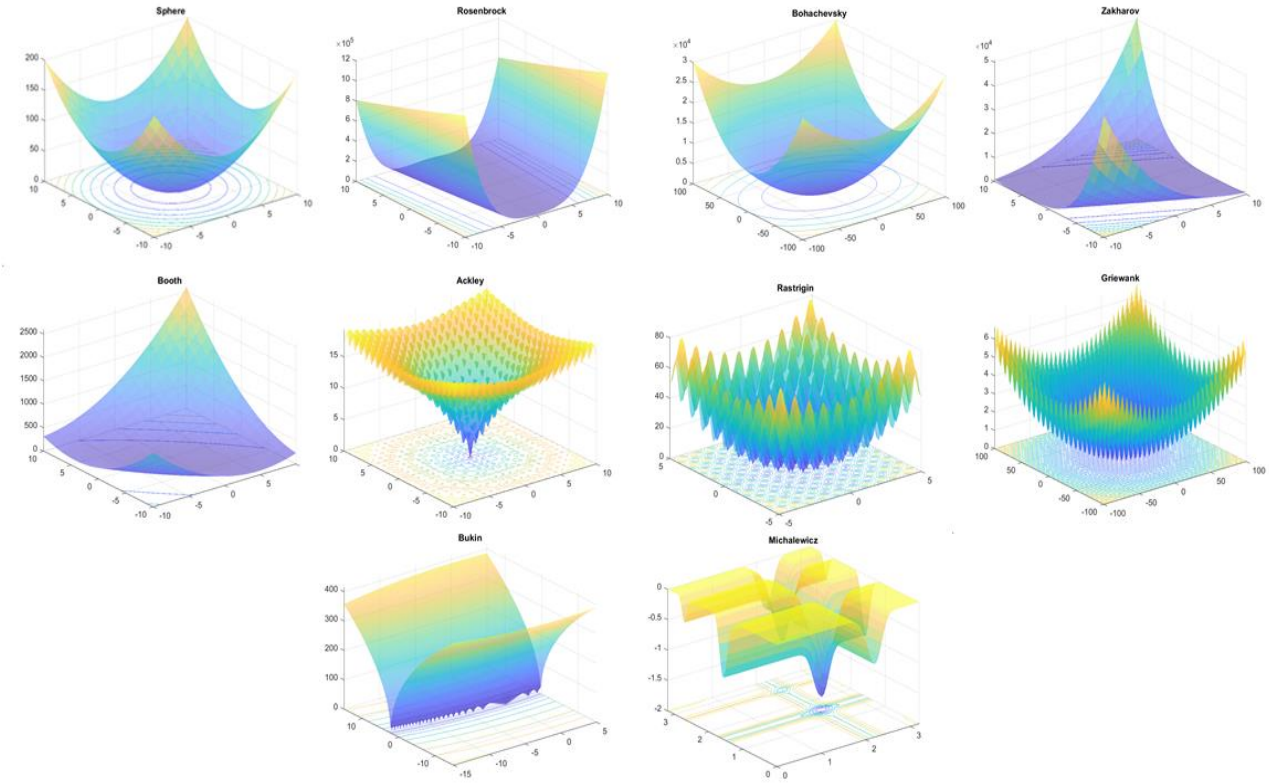


Figure. 1 Benchmark function

$$k_i' = w_2 k_i + (1 - w_2) k_{best} \quad (5)$$

$$k_{best}' = \begin{cases} k_i, f(k_i) < f(k_{best}) \\ k_{best}, else \end{cases} \quad (6)$$

The explanation of Eq. (5) and Eq. (6) is as follows. Eq. (5) states that the small male moves toward the highest quality big male with certain step size. Eq. (6) states that the current Komodo will replace the highest quality big male only it is better than the current highest quality big male.

## 4. Simulation and Result

Simulation is then conducted to evaluate this proposed algorithm. There are ten common benchmark functions in this work. Five functions are unimodal functions: Sphere, Rosenbrock, Bohachevsky, Zakharov, and Booth functions. The other five functions are multimodal functions: Ackley, Rastrigin, Griewank, Bukin, and Michalewicz functions. A two-dimensional description of ten benchmark functions can be seen in Fig. 1. Bohachevsky and Griewank functions represent problem with large problem space. The optimal fitness score for Michalewicz function is -4.6876 while the others are 0. The detailed specifications of these ten functions are shown in Table 1.

This proposed model is compared with three metaheuristic algorithms: cloud-theory based simulated annealing (CTA), HS, football game-based optimization (FBGO), hide object game optimization (HOGO), and KMA. The reason for choosing these algorithms is as follows. CTA and HS represent well-known and easy metaheuristic algorithms. CTA is chosen because it is a derivative of simulated annealing (SA) which offers better result by transforming the basic form of SA into population-based SA [24]. HS is chosen because it represents the non-population-based algorithm [25]. In it, diversification and intensification are conducted based on probabilistic calculation [25]. FBGO and HOGO represent the shortcoming game-based metaheuristic algorithm. HOGO [19] and FBGO [18] are also chosen because the adjusted parameters in these algorithms are minimal. Meanwhile, KMA is chosen because the proposed algorithm in this work is a derivative of the basic form of KMA. Based on that, it is important to compare the improvement proposed by this work with its original form.

The general setting of parameters in all algorithms is as follows. The maximum iteration is set at 100, except in HS, where the maximum iteration is 2000 because HS is not a population-based algorithm. The population size is set at 20. There are 30 runs for every function. The specific setting for every algorithm is described below.

Table 1. Benchmark functions

| No | Function | Model | Search Space | Dimension |
|---|---|---|---|---|
| 1 | Sphere | $\sum_{i=1}^{D} x_i^2$ | [-5.12, 5.12] | 5 |
| 2 | Rosenbrock | $\sum_{i=1}^{D-1}(100(x_{i+1}+x_i^2)^2+(x_i-1)^2)$ | [-5, 10] | 5 |
| 3 | Bohachevsky | $x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$ | [-100, 100] | 2 |
| 4 | Zakharov | $\sum_{i=1}^{D} x_i^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^4$ | [-5, 10] | 5 |
| 5 | Booth | $(x_1+2x_2+7)^2+(2x_1+x_2+5)^2$ | [-10, 10] | 2 |
| 6 | Ackley | $-20\cdot exp\left(-0.2\cdot\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i{}^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)+20 + exp(1)$ | [-32, 32] | 5 |
| 7 | Rastrigin | $10d+\sum_{i=1}^{D}\left(x_i^2-10cos(2\pi x_i)\right)$ | [-5.12, 5.12] | 5 |
| 8 | Griewank | $\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$ | [-600, 600] | 5 |
| 9 | Bukin | $100\sqrt{\lvert x_2-0.01x_1^2\rvert}+0.01\lvert x_1+10\rvert$ | $x_1\in[-15,-5]$ $x_2\in[-3,3]$ | 2 |
| 10 | Michalewicz | $-\sum_{i=1}^{D}\left((\sin x_i)\cdot\left(\sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2m}\right), m=10$ | $[0,\pi]$ | 5 |

The specific setting for the proposed algorithm is as follows. The formation of the Komodo is almost balance (0.35; 0.35; 0.3). The search space ratio is set 0.01. The big-male weight is set at 0.5. The small-male weight is set at 0.5. In CTA, the initial temperature is 10, terminating temperature is 0, and the Boltzmann constant is 0.0001. In HS, harmony memory considering rate is 0.8 and pitch adjusting rate is 0.5. In KMA, the big male proportion is 0.2, the number of females is 1, the radius of parthenogenesis is 0.5. In this simulation, the population size of the Komodo in KMA is constant so it is comparable with other algorithms. The result is shown in Table 2.

Table 2 shows that the proposed algorithm achieves the objective of metaheuristic algorithm. It can find the near optimal solution for all benchmark functions. As indicated in solving multimodal functions, it is proven that this proposed algorithm can avoid the local optimal trap.

Comparing among benchmark algorithms, the proposed algorithm is competitive enough. This algorithm outperforms all algorithms in solving three benchmark functions: Rosenbrock, Booth, and Griewank. This algorithm also outperforms four algorithms in solving six functions: Sphere, Bohachevksy, Zakharov, Ackley, Rastrigin, and Bukin. This algorithm outperforms three algorithms in solving Michalewicz function. The proposed algorithm outperforms KMA in solving nine functions. HOGO becomes the most difficult algorithm to beat. HOGO becomes the best algorithm in solving four functions. But the proposed algorithm is still better than HOGO in solving six functions.

The second simulation is conducted to observe the convergence of this algorithm. This simulation is conducted by running the algorithm with different maximum iterations. In this simulation, there are three values of maximum iteration: 25, 50, and 75.

Table 2. Simulation result

| Functions | Average Fitness Score | | | | | | Better than |
|---|---|---|---|---|---|---|---|
| | HS [25] | FBGO [18] | CTA [24] | HOGO [19] | KMA [20] | SKA | |
| Sphere | 0.0744 | 1.1020 | 0.0139 | **2.295 x $10^{-11}$** | 0.2096 | 0.0002 | HS, FBGO, CTA, KMA |
| Rosenbrock | 31.2933 | 553.3336 | 14.9298 | 4.2072 | 32.3806 | **3.7275** | HS, FBGO, CTA, HOGO, KMA |
| Bohachevsky | 0.1136 | 0.0162 | 0.0029 | **3.581 x $10^{-18}$** | 0.0988 | 1.932 **x $10^{-6}$** | HS, FBGO, CTA, KMA |
| Zakharov | 0.8468 | 21.1929 | 0.5871 | **9.736 x $10^{-6}$** | 11.3257 | 0.0012 | HS, FBGO, CTA, KMA |
| Booth | 0.0126 | 0.0498 | 2.235 x $10^{-5}$ | 0.0058 | 0.0126 | **1.88 x $10^{-6}$** | HS, FBGO, CTA, HOGO, KMA |
| Ackley | 4.2009 | 8.4469 | 17.1523 | **5.725 x $10^{-5}$** | 5.1953 | 0.3524 | HS, FBGO, CTA, KMA |
| Rastrigin | **3.4953** | 10.4677 | 14.7338 | 5.1754 | 9.5951 | 3.9895 | FBGO, CTA, HOGO, KMA |
| Griewank | 1.1714 | 3.1896 | 17.9473 | 0.2036 | 1.6560 | **0.2028** | HS, FBGO, CTA, HOGO, KMA |
| Bukin | 1.7777 | 0.1388 | 0.1255 | 3.0671 | **0.0136** | 0.0168 | HS, FBGO, CTA, HOGO |
| Michalewicz | **-4.4845** | -4.0125 | -3.4931 | -3.4559 | -3.2032 | -3.8639 | CTA, HOGO, KMA |

Table 3. Convergence result

| Function | Average Fitness Score | | |
|---|---|---|---|
| | $t_{max}$ = 25 | $t_{max}$ = 50 | $t_{max}$ = 75 |
| Sphere | 0.0004 | 0.0002 | 0.0003 |
| Rosenbrock | 252.3004 | 16.7223 | 4.7256 |
| Bohachevsky | 0.0006 | 0.0008 | 9.737 x $10^{-5}$ |
| Zakharov | 6.4849 | 0.5160 | 0.2454 |
| Booth | 1.692 x $10^{-6}$ | 6.520 x $10^{-7}$ | 3.780 x $10^{-6}$ |
| Ackley | 1.5253 | 0.3839 | 0.5962 |
| Rastrigin | 4.0836 | 4.9611 | 4.6044 |
| Griewank | 0.3752 | 0.2523 | 0.2005 |
| Bukin | 0.0305 | 0.0185 | 0.0136 |
| Michalewicz | -3.6197 | -3.4524 | -3.8218 |

Table 4. Formation test result

| Function | Average Fitness Score | | |
|---|---|---|---|
| | 0.6:0.2:0.2 | 0.2:0.6;0.2 | 0.2;0.2;0.6 |
| Sphere | 0.0004 | **6.391 x $10^{-5}$** | 0.0004 |
| Rosenbrock | 4.1617 | **3.7049** | 3.7706 |
| Bohachevsky | 0.0037 | **6.659 x $10^{-5}$** | 0.0025 |
| Zakharov | 0.8892 | **0.0006** | 0.0431 |
| Booth | 9.049 x $10^{-6}$ | **2.345 x $10^{-7}$** | 1.624 x $10^{-6}$ |
| Ackley | 1.9707 | **0.1778** | 2.2871 |
| Rastrigin | 3.3976 | **3.3260** | 4.8470 |
| Griewank | 0.3161 | **0.0822** | 0.3473 |
| Bukin | **0.0160** | 0.0206 | 0.0194 |
| Michalewicz | -3.7029 | **-3.9707** | -3.5994 |

The proposed algorithm, as shown in Table 3, can quickly achieve the convergence condition. In general, convergence occurs when the maximum iteration is less than 100. Convergence occurs when maximum iteration is 25 in solving Rastrigin and Michalewicz functions. Meanwhile, convergence occurs when the maximum iteration is 50 in solving Sphere and Ackley functions. Convergence occurs when maximum iteration is 75 in solving Bukin and Griewank functions.

The following simulations are conducted to evaluate the parameters sensitivity. The objective is to find which parameters significantly affect the performance and which parameters do not. These adjusted parameters do not affect the complexity or computational time. The first parameter is formation of the Komodo.

The third simulation is conducted to evaluate the relation of the Komodo formation and the algorithm performance. In the first simulation, the formation is equal. The formation is presented in (big male, female, small male). There are three scenarios: big male dominant, female dominant, and small male dominant. The result is shown in Table 4.

Table 5. Relation between search space ratio and fitness score

| Function | Average Fitness Score | | |
|---|---|---|---|
| | $r_s$ = 0.005 | $r_s$ = 0.05 | $r_s$ = 0.5 |
| Sphere | **2.321 x $10^{-5}$** | 0.0017 | 0.0963 |
| Rosenbrock | 39.3417 | **4.0048** | 28.2542 |
| Bohachevsky | **2.284 x $10^{-5}$** | 0.0017 | 0.1003 |
| Zakharov | 4.1326 | **0.0011** | 0.5438 |
| Booth | **1.448 x $10^{-7}$** | 1.949 x $10^{-6}$ | 6.671 x $10^{-5}$ |
| Ackley | 2.5683 | **1.0478** | 4.1469 |
| Rastrigin | 4.1801 | **3.2449** | 6.0456 |
| Griewank | 0.3233 | **0.2693** | 0.8895 |
| Bukin | **0.0213** | 0.0695 | 0.0823 |
| Michalewicz | -3.9649 | **-4.2858** | -3.9066 |

Table 4 shows that the relation between the formation of Komodo and the performance of the algorithm is various depends on the problem. Female-dominant formation tends to create better performance rather than male-dominant formation. Female-dominant formation achieve the best performance significantly in solving sphere, Bohachevksy, Zakharov, Booth, Ackley, and Griewank functions. On the other side, the formation does not affect the result in solving Rosenbrock, Rastrigin, Bukin, and Michalewicz functions.

The fourth simulation is conducted to evaluate the search space ratio during parthenogenesis and the algorithm performance. In this simulation, the formation of Komodo is set as female-dominant (0.2, 0.6, 0.2). There are three values of this ratio: 0.005, 0.05, and 0.5. The result is shown in Table 5.

Table 5 shows that the relation between the search space ratio and the algorithm performance is different depending on the problem to be solved. In general, moderate search space ratio is preferred. The result shows that moderate search space ratio performs as the best option in solving six functions. On the other side, low search space ratio is the best in solving four functions. The significance of the search space ratio is different between unimodal functions and multimodal functions. The search space ratio affects significantly in solving unimodal functions. On the other side, it is less significant in solving multimodal functions.

The fifth simulation is conducted to observe the relation between the big male weight and the algorithm performance. This simulation is conducted in small-male dominant formation. There are three values of the big male weight: 0.25, 0.5, and 0.75. The result is shown in Table 6. In this simulation, the search space ratio is set at 0.01.

Table 6. Relation between big-male weight and fitness score

| Function | Average Fitness Score | | |
|---|---|---|---|
| | $w_b$ = 0.25 | $w_b$ = 0.5 | $w_b$ = 0.75 |
| Sphere | 0.0003 | 0.0003 | 0.0003 |
| Rosenbrock | **3.8646** | 4.1395 | 8.4086 |
| Bohachevsky | 0.0054 | 0.0041 | **0.0011** |
| Zakharov | 0.3910 | 0.3286 | **0.2187** |
| Booth | 1.248 x $10^{-5}$ | 4.326 x $10^{-6}$ | **4.915 x $10^{-7}$** |
| Ackley | 2.1659 | 0.8505 | **0.8102** |
| Rastrigin | **3.6719** | 3.9134 | 5.6942 |
| Griewank | 0.2819 | 0.3056 | **0.2610** |
| Bukin | **0.0139** | 0.0163 | 0.0165 |
| Michalewicz | -3.4800 | -3.6754 | **-3.9131** |

Table 7. Relation between small-male weight and fitness score

| Function | Average Fitness Score | | |
|---|---|---|---|
| | $w_s$ = 0.25 | $w_s$ = 0.5 | $w_s$ = 0.75 |
| Sphere | 0.0004 | 0.0004 | **0.0003** |
| Rosenbrock | **3.7438** | 3.7506 | 4.5787 |
| Bohachevsky | 0.0036 | 0.0007 | **0.0002** |
| Zakharov | 0.0067 | **0.0028** | 0.2965 |
| Booth | 8.857 x $10^{-6}$ | 5.237 x $10^{-6}$ | **1.405 x $10^{-6}$** |
| Ackley | 2.5533 | 1.7182 | **0.6771** |
| Rastrigin | 7.8517 | 5.6671 | **4.4317** |
| Griewank | 0.4097 | 0.3103 | **0.2177** |
| Bukin | 0.0212 | 0.0189 | **0.0147** |
| Michalewicz | -3.6236 | -3.5800 | **-3.8558** |

Table 6 shows that the relation between the big-male weight and the performance of the proposed algorithm is various depends on the function to solve. Higher big-male weight improves the performance in six functions. Meanwhile, lower big-male weight improves the performance in three functions. In general, the big-male weight does not affect the performance significantly, except in Booth function.

The sixth simulation is conducted to observe the relation between the small male weight and the algorithm performance. This simulation is conducted in small-male dominant formation. In this simulation, there are three values of the small male weight: 0.25, 0.5, and 0.75. The result is shown in Table 7. In this simulation, the search space ratio is set at 0.01.

Table 7 shows that in general, the increase in the small-male weight improves the performance of the proposed algorithm. This circumstance occurs in eight functions: three unimodal functions and five multimodal functions. The exception is in solving Rosenbrock and Zakharov functions. Although the increase in this weight can improve the performance, its improvement is not significant, except in solving Ackley function.

## 5. Discussion

There are several findings due to the simulation result. First, this proposed algorithm can achieve the objective of metaheuristic algorithm. It can find the nearly optimal solution in all ten benchmark functions. Due to the multimodal functions, it is shown that this algorithm can avoid the local optimal trap.

The second finding is that this proposed algorithm is competitive enough compared with the benchmark algorithms (FBGO, HOGO, CTA, HS, and KMA). This competitiveness gives prospect that this proposed algorithm is promising to be used in solving the real-world optimization problems. Comparing with the original KMA, this proposed algorithm is also competitive enough.

There are several notes due to this competitiveness. This positive result is achieved by tuning parameters in the algorithm. Suyanto, Ariyanto, and Ariyanto [20] also tuned the KMA algorithm in their work so that it outperforms several well-known or short-coming algorithms. In general, every metaheuristic algorithm has several parameters that can be tuned to improve its performance based on the problem to solve. Based on it, although this proposed algorithm gives positive results in this work, there is not any guarantee that this algorithm is always better than the benchmark algorithms. By appropriate tuning, an algorithm can achieve its best result. On the other side, wrong tuning may produce an unpleasant performance. Second, the performance of any metaheuristic algorithm also depends on the problem to solve as it is stated in the no-free-lunch theory [26]. Although some algorithms are developed as general-purpose optimization algorithm, their performance would not be the best [26]. Although in general, this proposed algorithm is better than KMA, its performance is worse in solving Bukin function. Besides, better results can be simply achieved by expanding the maximum iteration or population size.

The third finding is that the convergence of this proposed algorithm is fast enough. As indicated in Table 3, a near optimal solution can be achieved before the $50^{th}$ iteration. After that, the fitness score tends to stagnant. Moreover, the acceptable optimal solution can be achieved in the $25^{th}$ iteration.

The fourth finding is that female affects significantly rather than male ones: big male or small male. Table 4 indicates this circumstance. In general, female-dominant formation achieves the best result compared with big-male dominant or small-male dominant formations. The reason is as follows. The role of the female is diversification. It implies that more females are required to explore the problem space. By increasing the proportion of females, exploration can be conducted more effectively. On the other side, females are also important in the intensification process by setting lower search space ratio.

The existence of the male is still important in the optimization process. Big-male and small-male are important in making convergence. Moreover, collective intelligence is conducted in male Komodo. The small-male focuses on the one best solution. On the other side, the big male interacts with several of the best solutions. This means that the small-male is designed as the intensification only while the big-male conducts intensification based on several alternatives.

The fifth finding is that the search space ratio is a sensitive parameter while big-male and small-male weights are less sensitive. The sensitivity of the search space ratio is more significant in solving unimodal problems than multimodal problems. In general, moderate search space ratio is preferred due to its characteristic in balancing the exploration and exploitation. On the other side, higher big-male and small-male weights are preferred because higher value of these weights can make the selected solution closer to the current best solution faster.

The complexity analysis of this proposed algorithm is as follows. Its complexity can be presented as $O(t_{max}.n(K).n(C))$. It is shown that the iteration depends on three parameters: maximum iteration, population size, and number of candidates. The assumption of this statement is that the number of candidates for female tends to be higher than better big male; and the female-dominant formation is chosen.

## 6. Conclusion

This work has demonstrated that the modification of the original KMA can provide better performance. This modified version is also simpler than the original one. Both diversification and intensification are still conducted properly without redundancy. By appropriate tuning, this proposed algorithm is competitive enough with the original KMA and other well-known algorithms. The simulation result shows that the female-dominant formation is preferred to male-dominant formation. The search space ratio becomes the most sensitive parameters with moderate value is preferred to make balance between intensification and diversification.

There are several future research potentials due to this work. This work is an early improved version of the KMA. It means that the other improvements are still possible and promising. Moreover, studies that

implement this proposed algorithm to be used to solve real world optimization problems are needed to give more comprehensive evaluation to this algorithm.

## Conflicts of interest

The authors declare no conflict of interest.

## Author contributions

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

## Acknowledgments

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

## References

[1] Z. Liang, P. Zhong, M. Liu, C. Zhang, and Z. Zhang, "A Computational Efficient Optimization of Flow Shop Scheduling Problems", *Scientific Reports*, Vol. 12, 2022.

[2] F. Pan, T. Fan, X. Qi, J. Chen, and C. Zhang, "Truck Scheduling for Cross-Docking of Fresh Produce with Repeated Loading", *Mathematical Problems in Engineering*, Vol. 2021, pp. 1-16, 2021.

[3] J. O. Mierzejewska, A. P. Maranda, and W. Maranda, "Selected Genetic Algorithms for Vehicle Routing Problem Solving", *Electronics*, ID: 3147, Vol. 10, pp. 1-34, 2021.

[4] F. H. Awad, A. A. Kubaisi, and M. Mahmood, "Large-scale Timetabling Problems with Adaptive Tabu Search", *Journal of Intelligent Systems*, Vol. 31, No. 1, pp. 168-176, 2022.

[5] A. E. Dinar, S. Ghouali, B. Merabet, M. Feham, M. S. Guellil, and E. K. Hussein, "5G Network Performance by Cell-Edge Servers Optimization Assignment (5GNP-CESOA)", *Procedia Computer Science*, Vol. 194, pp. 140-148, 2021.

[6] D. Teng, Y. Li, H. Yang, Z. Wei, and Y. Li, "Genetic Algorithm for Sparse Optimization of Mills Cross Array Used in Underwater Acoustic Imaging", *Journal of Marine Science and Engineering*, Vol. 10, pp. 1-11, 2022.

[7] 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.

[8] J. Swan, S. Adriaensen, A. E. I. Brownlee, K. Hammond, C. G. Johnson, A. Kheiri, F. Krawiec, J. J. Merelo, L. L. Minku, E. Ozcan, G. L. Pappa, P. G. Sanchez, K. Sorensen, S. Vob, M. Wagner, and D. R. White, "Metaheuristics in the Large", *European Journal of Operational Research*, Vol. 297, pp. 393-406, 2022.

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

[10] M. Braik, A. Sheta, and H. A. Hiary, "A Novel Meta-Heuristic Search Algorithm for Solving Optimization Problems: Capuchin Search Algorithm", *Neural Computing and Applications*, Vol. 33, pp. 2515-2547, 2021.

[11] T. Ding, L. Chang, C. Li, C. Feng, and N. Zhang, "A Mixed-Strategy-Based Whale Optimization Algorithm for Parameter Identification of Hydraulic Turbine Governing Systems with a Delayed Water Hammer Effect", *Energies*, Vol. 11, pp. 1-29, 2018.

[12] 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, pp. 14638-14665, 2020.

[13] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor Penguins Colony: A New Metaheuristic Algorithm for Optimization", *Evolutionary Intelligence*, Vol. 12, pp. 211-226, 2019.

[14] W. Qiao and Z. Yang, "Solving Large-Scale Function Optimization Problem by Using a New Metaheuristic Algorithm Based on Quantum Dolphin Swarm Algorithm", *IEEE Access*, Vol. 7, pp. 138972-138989, 2019.

[15] S. Arora, and P. Anand, "Chaotic Grasshopper Optimization Algorithm for Global Optimization", *Neural Computing and Applications*, Vol. 31, No. 8, pp. 4385-4405, 2019.

[16] S. Arora, and S. Singh, "Butterfly Optimization Algorithm: A Novel Approach for Global Optimization", *Soft Computing*, Vol. 23, No. 3, pp. 715-734, 2019.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[23] Y. Celik, "An Enhanced Artificial Bee Colony Algorithm Based on Fitness Weighted Search Strategy", *Automatika - Journal of Control, Measurement, Electronics, Computing and Communications*, Vol. 62, Nos. 3-4, pp. 300-310, 2021.

[24] E. Torabzadeh and M. Zandieh, "Cloud Theory-based Simulated Annealing Approach for Scheduling in the Two-stage Assembly Flowshop", *Advances in Engineering Software*, Vol. 41, Nos. 10-11, pp. 1238-1243, 2010.

[25] M. Dubey, V. Kumar, M. Kaur, and T. P. Dao, "A Systematic Review on Harmony Search Algorithm: Theory, Literature, and Applications", *Mathematical Problems in Engineering*, ID: 5594267, Vol. 2021, pp. 1-22, 2021.

[26] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, 1997