Received: November 1, 2023. Revised: December 1, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Migration-Crossover Algorithm: A Swarm-based Metaheuristic Enriched with Crossover Technique and Unbalanced Neighbourhood Search

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

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

Abstract: There has been a massive development of metaheuristic algorithms in the latest decade where swarm intelligence becomes the fundamental approach. Meanwhile, there is still no ideal metaheuristic that can solve all problems superiorly, as declared in the no-free-lunch (NFL) theory. This work introduces a novel swarm-based metaheuristic named as migration-crossover algorithm (MCA). In MCA, the swarm intelligence is enriched with the crossover technique and the neighbourhood search with unbalanced local search space. The global finest solution becomes the reference in the first step while the middle between two stochastically chosen solutions becomes the reference in the second step. The neighbourhood search is performed in the third step. The collection of 23 functions become the use case during the evaluation of MCA. In the first evaluation, MCA is compared with five new metaheuristics: total interaction algorithm (TIA), osprey optimization algorithm (OOA), migration algorithm (MA), coati optimization algorithm (COA), and walrus optimization algorithm (WaOA). The result reveals that MCA is finer than TIA, OOA, MA, COA, and WaOA in 20, 19, 17, 20, and 17 functions subsequently. The result of the second evaluation reveals that the global finest solution becomes the dominant contributor in the high dimension functions while the middle between two stochastically chosen solutions becomes the dominant contributor in the fixed dimension functions.

Keywords: Optimization, Metaheuristic, Swarm intelligence, Crossover, Neighbourhood search.

1. Introduction

There are many optimization studies utilized the metaheuristics to meet their objectives. In the biomedical system, dandelion optimization algorithm has been embedded to classify blood vessels for grading the diabetic retinopathy [1] while pelican optimization has been utilized to detect and classify tuberculosis based on the x-ray image of the chest [2]. In the agricultural sector, the red deer algorithm has been utilized to detect and classify plant diseases in the early season phase [3]. Bacterial colony optimization has been used to optimize the detection of DDoS attacks in the cloud system [4].

In recent years, many new metaheuristics have been developed based on the multiple solution-based metaheuristics. Another term for the metaheuristic consisting of multiple solutions is population-based metaheuristic. Moreover, within the population-based metaheuristics, many of them were developed based on the swarm intelligence approach rather than the evolutionary-based approach. Many of the swarm-based metaheuristics exploit the practices of animals during breeding or tracing for prey or food, like Komodo mlipir algorithm (KMA) [5], green anaconda optimization (GAO) [6], walrus optimization algorithm (WaOA) [7], reptile search algorithm (RSA) [8], coati optimization algorithm (COA) [9], zebra optimization algorithm (ZOA) [10], golden jackal optimization (GJO) [11], chameleon swarm algorithm (CSA) [12], cat and mouse based optimization (CMBO) [13], clouded leopard optimization (CLO) [14], northern goshawk optimization (NGO) [15], pelican optimization algorithm (POA) [16], snow leopard optimization (SLO) [17], red fox optimization (RFO) [18], Siberian tiger optimization (STO) [19], white shark optimization (WSO) [20], Tasmanian devil optimization (TDO) [21], osprey optimization algorithm (OOA) [22], and so on. Some metaheuristics use their main reference during the directed search for their name such as random selected leader-based optimization (RSLBO) [23], multi leader optimization (MLO) [24], mixed leader-based optimization (MLBO) [25], hybrid leader-based optimization (HLBO) [26], three influential member-based optimizations (TIMBO) [27], and so on. Some metaheuristics imitate the social behavior such as chef-based optimization algorithm (CBOA) [28], migration algorithm (MA) [29], and so on. Some metaheuristics promote their fundamental concept as their name, such as golden search optimization (GSO) [30], average subtraction-based optimization (ASBO) [31], four directed search algorithms (FSDA) [32], total interaction algorithm (TIA) [33], walk-spread algorithm (WSA) [34], attack-leave optimization (ALO) [35], and so on.

In general, there are some stagnations in the development of metaheuristics. First, many recent metaheuristics were built based on the swarm intelligence as a fundamental approach, so that some other approaches, such as evolutionary-based technique and neighborhood or local search become less popular. Second, many existing metaheuristics focus on the directed motion toward the global finest solution. Although this search is rational because it has been proven effective in many existing metaheuristics, the needs to introduce another approach is important to avoid the stagnation in the development of metaheuristics which may end with trivial refinement. Third, many swarm-based metaheuristics also deploy neighborhood search with the reduction of its local search space during iteration as additional search. The stagnation of the neighborhood search can be investigated as the iteration becomes the only parameter that controls local search space.

Related to this problem, this work introduces a novel metaheuristic that combines three branches of metaheuristic development called as migration-crossover algorithm (MCA). These branches include swarm intelligence, evolutionary-based technique, and neighborhood search. The swarm intelligence is presented by constructing MCA based on a set of autonomous agents called swarm and the deployment of directed search called migration. The evolutionary-based technique is presented using crossover technique, which is the fundamental strategy in the evolutionary-based technique. Meanwhile, a new type of neighborhood search is introduced by constructing the unbalanced local search space to refine the diversification capability of neighborhood search which was originally designed as intensification-based search.

The novelty and the scientific contribution of this proposed work are presented below.

- This work provides a novel metaheuristic that hybridizes three fundamental branches in metaheuristics: swarm intelligence, evolution-based system, and neighborhood search.
- A new kind of neighborhood search called unbalanced neighborhood search is introduced which is different from any existing neighborhood search.
- A comparative evaluation is taken to assess the contribution and refinement provided by MCA in the development of metaheuristic algorithm.
- The single search evaluation is taken to assess the contribution and importance of each search implemented in MCA.

This paper is formed as follows. Sect. 1 describes mainly the background of this work, problem formulation, objective, and scientific contribution of the provided work. The review of recent development of swarm-based metaheuristic is taken in sect. 2. The fundamental concept, formulation, and formalization of the provided algorithm are presented in sect. 3. The evaluation taken to assess the performance of MCA including its result is presented in sect. 4. Sect. 5 presents the discussion regarding the more comprehensive investigation regarding the evaluation result, the linkage to the theory, computing complexity of the algorithm, and the limitations of this work. In the end, sect. 6 consolidates the conclusion and the proposal for future development.

2. Related works

Swarm intelligence is a favorite approach used in many recent metaheuristics. It can also be seen as a recent milestone in the evolution of the development of metaheuristics. The history of metaheuristics development is started with the single solution-based metaheuristic where the system consists of single solution only that attempts to find a finer solution during the iteration within the search space. In this era, neighborhood searches are the most common. Through neighborhood search or local search, the solution attempts to find a finer solution by tracing opportunities close to the current solution [36]. Simulated annealing [36] and variable neighborhood search have become some famous metaheuristics that use this approach, and they are still utilized in many studies regarding the optimization until today.

The second milestone is the introduction of the population-based metaheuristic. This approach was developed to overcome the weakness of single solution-based metaheuristic. Population-based metaheuristic provides faster process as it consists of multiple solutions so that more solutions trace the whole search space. Evolutionary-based technique becomes the important concept in this era, where genetic algorithm (GA) become the crown of this concept. GA depends on the crossover technique to refine the quality of recent population and mutation process to diversify the population and create alternatives to avoid local optimal entrapment [37].

Then, swarm intelligence becomes the next milestone as it can be seen as the continuity of the population-based metaheuristic. Different from evolutionary-based techniques, swarm intelligence transforms the population into active and autonomous agents that trace the whole search space with a certain step size or speed. Swarm intelligence exploits the intensive interaction among agents to boost its speed and provide a finer final solution within less iteration. Particle swarm optimization (PSO) is the early form of swarm intelligence that utilizes the global finest and local finest as its reference [38]. Many swarm-based metaheuristics were introduced after the successful introduction of PSO.

The summary of recent swarm-based metaheuristics is presented in Table 1. This presentation includes their number of steps in every iteration, their references used in the directed search, the existence of crossover technique representing the evolutionary-based technique, and the existence of the neighborhood search. The last row consists of the provided metaheuristic to make clear positioning of the provided metaheuristic compared to these recent swarm-based metaheuristics.

Table 1. The summary of recent swarm-based metaheuristics including their steps, reference, existence of the crossover, and existence of neighborhood search

No	Metaheuristics	Steps	References during Directed Search	Crossover	Neighborhood Search
1	TIA [33]	1	all other agents	no	no
2	MA [29]	2	a stochastically chosen finer agent	no	reduced local space during iteration
3	COA [9]	2	global finest agents and a randomized agent within the space	no	reduced local space during iteration
4	OOA [22]	2	a stochastically chosen agent from a pool consisting of all finer agents and global finest agent	no	reduced local space during iteration
5	WaOA [7]	3	global finest agent and a stochastically chosen agent	no	reduced local space during iteration
6	GJO [11]	1	first finest agent and second-finest agent	no	no
7	POA [16]	2	a randomized agent within space	no	reduced local space during iteration
8	GSO [30]	1	a portion of global finest agent and a portion of local finest agent	no	no
9	NGO [15]	2	a stochastically chosen agent within swarm	no	reduced local space during iteration
10	ASBO [31]	3	global finest agent, the middle between finest agent and worst agent, the difference between finest agent and worst agent	no	no
11	GAO [6]	2	a stochastically chosen finer agent based on its normalized fitness and follow normal distribution	no	reduced local space during iteration
12	ALO [35]	3	global finest agent, the middle between global finest agent and a stochastically chosen agent, and the middle between two stochastically chosen agents	no	no
13	this work	3	global finest agent and two stochastically chosen agent	yes	reduced local space during iteration with unbalanced distance between the lower local boundary and higher local boundary.

The previous explanation and summary in Table 1 reveals that there is a chance to propose a new metaheuristic that combines three approaches: swarm intelligence, evolutionary-based technique, and neighborhood search in a balanced manner. As presented in Table 1, there is dominance of the swarm intelligence as a fundamental approach in the recent metaheuristics. Table 1 also reveals the variety in constructing the references used in the directed search. Meanwhile, some swarm-based metaheuristics are enriched with neighborhood search while others focus only on the directed search. Unfortunately, there is stagnation in the neighborhood search where there is common type of neighborhood search where the local space shrinks as the iteration continues. On the other hand, evolutionary-based technique becomes much less popular as complement for the swarm-based metaheuristic. It makes the clear and distinct position and contribution for this work to provide a new metaheuristic that combines the swarm intelligence, evolutionary-based technique, and neighborhood search. Moreover, proposing a new kind of neighborhood search also becomes an additional contribution to this work.

3. Model

The fundamental concept of MCA is the hybridization of swarm intelligence, crossover technique, and neighborhood search. There are two characteristics of swarm intelligence adopted in MCA. First, MCA is constructed by a certain number of autonomous agents. Each agent acts without any central command but based on three considerations: set of possible actions, perception of the environment, and interaction with other agents. In MCA, there are actions that are possibly taken by each agent. The first action is generating a new seed for refinement. The second action is replacing the current solution with this seed. The perception of the environment can be interpreted by measuring the quality of its current solution and its reference. The interaction with other agents is taken by selecting references from other agents. Second, MCA uses directed search which is finding a finer solution by moving to a new solution within the space based on the direction provided by the reference. The crossover mechanism is adopted by combining some values from the current solution and some values from its reference to generate a new seed. It means that the related agent and its reference become the parents of the seed. MCA also performs neighborhood searches but with some modifications where the lower local search space width may be different from the upper local search space width in the related dimension.

a	autonomous agent
A	set of agents (swarm)
a_b	global finest agent
a_{lo}	lower boundary
a_{llo}	lower local boundary
a_{hi}	higher boundary
a_{lhi}	higher local boundary
a_m	the middle between two stochastically chosen agent
a_r	a stochastically chosen agent
D	dimension
F	objective function
α	uniform floating point random number [0,1]
β	uniform integer random number [1,2]
s_c	crossover seed
s_d	directed search seed
s_f	final seed
t	iteration
t_m	maximum iteration
U	general uniform random
x	index for agent
y	index for dimension

algorithm 1: pseudocode of MCA

begin
 for x=1 **to** n
  initiate a_i using Eq. (2)
  update a_b using Eq. (3)
 end for
 for t=1 **to** t_m
  for x=1 **to** n
   run 1st search using Eq. (4) to Eq. (7)
   run 2nd search using Eq. (8) to Eq. (13)
   run 3rd search using Eq. (14) to Eq. (17)
   update a_b using Eq. (2)
  end for
 end for
 return a_b
end

MCA uses two references during the iteration process. The first reference is the global finest agent. This reference is chosen due to its popularity as it is also adopted in many swarm-based metaheuristic. The second reference is the middle between two stochastically chosen agents. These two agents are selected from the swarm. This reference is rare to use as many existing metaheuristics use a stochastically chosen agents within the swarm. These references are used in the directed search and the crossover.

This fundamental concept is then transformed into three sequential steps during the iteration. In the first step, each agent may perform a directed search toward the global finest agent and crossover with the global finest agent. The finer search between these two searches is selected for the seed in the first step. In the second step, each agent may perform a directed search relative to the middle between two stochastically chosen agents and crossover with the middle between two stochastically chosen agents. The finer search between these two searches is selected for the seed in the second step. Then, in the third step, each agent performs the neighborhood search where the local search space shrinks during the search space and the local search space width is affected by the distance between the related agent and the upper and lower boundaries.

A strict acceptance procedure is implemented in MCA. It means that every seed can substitute the current value of its parent only if this seed is finer than its parent. Moreover, an agent substitutes the value of the global finest agent only if it improves the global finest agent.

This overall strategy is then converted into a formal algorithm. It is formalized using algorithm 1 in pseudocode presentation. Moreover, the mathematical formulation to describe the algorithm in a more detailed manner is presented in Eq. (1) to Eq. (17). The notations utilized in this paper are presented below.

The presentation of swarm is presented in Eq. (1). It can be seen as a set of autonomous agents with a predefined swarm size.

$$A = \{a_1, a_2, a_3, \dots, a_n\} \tag{1}$$

There are two processes during the initialization. The first process is generating an agent that follows a uniform distribution within the search space as exhibited in Eq. (2). After that, the second process is updating the global finest agent based on the strict acceptance procedure as indicated in Eq. (3).

$$a_{x,y} = a_{lo,y} + \alpha\left(a_{hi,y} - a_{lo,y}\right) \tag{2}$$

$$a_b{}' = \begin{cases} a_x, f(a_x) < f(a_b) \\ a_b, else \end{cases} \tag{3}$$

The first search is formulated using Eq. (4) to Eq. (7). Eq. (4) reveals that the first directed search seed is obtained based on the migration toward the global finest agent. Eq. (5) reveals the stochastic crossover process between the related agent and the global finest agent based on the equal opportunity to produce the first crossover seed. Eq. (6) reveals the selection used to produce the first final seed. Eq. (7) reveals the updating process of the related agent based on the first final seed.

$$s_{d1,x,y} = a_{x,y} + \alpha\left(a_{b,y} - \beta a_{x,y}\right) \tag{4}$$

$$s_{c1,x,y} = \begin{cases} a_{b,y}, \alpha < 0.5 \\ a_{x,y}, else \end{cases} \tag{5}$$

$$s_{f1,x,y} = \begin{cases} s_{d1,x,y}, f\left(s_{d1,x,y}\right) < f(s_{c1,x,y}) \\ s_{c1,x,y}, else \end{cases} \tag{6}$$

$$a'_{x,y} = \begin{cases} s_{f1,x,y}, f\left(s_{f1,x,y}\right) < f(a_{x,y}) \\ a_{x,y}, else \end{cases} \tag{7}$$

The second search is specified using Eq. (8) to Eq. (13). Eq. (8) reveals the random selection process within the swarm based on the uniform distribution. Eq. (9) reveals that the reference is the middle between two stochastically chosen agents. Eq. (10) reveals that the second directed search seed is obtained based on the migration relative to the middle between two stochastically chosen agents and its direction is calculated based on the quality comparison between the related agent and this second reference. Eq. (11) reveals the stochastic crossover process between the related agent and the second reference based on the equal opportunity to produce the second crossover seed. Eq. (12) reveals the selection used to produce the second final seed. Eq. (13) reveals the updating process of the related agent based on the second final seed.

$$a_r = U(A) \tag{8}$$

$$a_{m,x,y} = \frac{a_{r1,y} + a_{r2,y}}{2} \tag{9}$$

$$s_{d2,x,y} = \begin{cases} a_{x,y} + \alpha\left(a_{m,x,y} - \beta a_{x,y}\right), f\left(a_{m,x,y}\right) < f(a_{x,y}) \\ a_{x,y} + \alpha\left(a_{x,y} - \beta a_{m,x,y}\right), else \end{cases} \tag{10}$$

$$s_{c2,x,y} = \begin{cases} a_{m,x,y}, \alpha < 0.5 \\ a_{x,y}, else \end{cases} \tag{11}$$

$$s_{f2,x,y} = \begin{cases} s_{d2,x,y}, f\left(s_{d2,x,y}\right) < f(s_{c2,x,y}) \\ s_{c2,x,y}, else \end{cases} \tag{12}$$

$$a'_{x,y} = \begin{cases} s_{f2,x,y}, f\left(s_{f2,x,y}\right) < f(a_{x,y}) \\ a_{x,y}, else \end{cases} \tag{13}$$

The third search is formalized using Eq. (14) to Eq. (17). Eq. (14) is used to calculate the lower local boundary. Eq. (15) is used to calculate the higher local boundary. Eq. (16) reveals that the third final seed is generated uniformly between the lower local boundary and the higher local boundary. Eq. (17) reveals the updating process of the related agent based on the second final seed.

$$a_{llo,x,y} = a_{lo,y} + \frac{t}{t_m}\left(a_{x,y} - a_{lo,y}\right) \quad (14)$$

$$a_{lhi,x,y} = a_{x,y} + \left(1 - \frac{t}{t_m}\right)\left(a_{hi,y} - a_{x,y}\right) \quad (15)$$

$$s_{f3,x,y} = a_{llo,x,y} + \alpha\left(a_{lhi,x,y} - a_{llo,x,y}\right) \quad (16)$$

$$a'_{x,y} = \begin{cases} s_{f3,x,y}, f\left(s_{f3,x,y}\right) < f(a_{x,y}) \\ a_{x,y}, else \end{cases} \quad (17)$$

4. Simulation and result

This section discusses the investigation conducted to assess the performance of MCA. There are two investigations in this work. The first investigation is the comparative evaluation whose objective is to assess the performance of MCA compared with the other metaheuristics. The second evaluation is the single search evaluation whose objective is to measure the performance of each search constructing the MCA. Comparative evaluation is important to measure the refinement of the MCA relative to the existing techniques. Moreover, the comparative evaluation is necessary to analyze the strengths and weaknesses of the MCA. On the other hand, the single search evaluation is needed to measure the contribution of each search in MCA since MCA is a multiple search metaheuristic.

The set consisting of 23 functions is used as the problem in both evaluations. This set of functions is chosen due to its coverage. It covers both unimodal and multimodal problems as it consists of seven high dimension unimodal (HDU) functions (F1 to F7), six high dimension multimodal (HDM) functions (F8 to F13), and ten fixed dimension multimodal (FDM) functions (F14 to F23). The main consideration of the unimodal functions is finding the global optimal as fast as possible as each function consists of single optimal solution. On the other hand, the main consideration of multimodal functions is avoiding the local optimal entrapment as each function consists of multiple optimal solutions but only one global optimal. This set of functions also covers various search spaces. Some functions have narrow search space while some other functions have large search space. The terrain of the functions also varied from smooth descent, wavy, and flat with very narrow slopes. Due to the character variety of the functions in this set, this set of functions has been used as theoretical problem during the assessment of many studies introducing new metaheuristics, such as KMA, TIA, GSO, ALO, and so on. A detailed description of this set of functions can be found in Table 2.

The MCA is compared with five new metaheuristics in the comparative evaluation. These metaheuristics are TIA, OOA, MA, COA, and WaOA. All these metaheuristics are new as they are introduced in 2023. TIA is the only comparator that adopts a single search strategy which is interaction with all other agents [33]. On the other hand, OOA [22], MA [29], COA [9], and WaOA [7] adopt multiple search strategies. These four comparators also deploy neighborhood search with reduction of the local space during the iteration. The other reason of choosing these five metaheuristics as comparators is the fact that they do not have any adjusted parameters except the swarm size and maximum iteration. This circumstance is chosen to make fair comparison because all these comparators will always be in their default settings.

In the comparative evaluation, there are only two independent parameters: the swarm size and the maximum iteration. In this work, the swarm size is set to 5 while the maximum iteration is set to 10. It means all the metaheuristics in this evaluation including MCA and its comparators are pushed to find the optimal solution quickly. The results are presented in Table 3 to Table 5 representing the HDU, HDM, and FDM functions respectively. There are three parameters presented in Table 3 to Table 5: mean, standard deviation, and the mean-based rank. Then, this result is summarized in Table 6 representing the superiority of MCA compared to its comparators in every group of functions based on the average fitness score (mean).

Table 3 reveals the superiority of MCA in overcoming the HDU functions. MCA becomes the finest performer in all functions in all seven HDU functions. MCA becomes the sole finest performer in six HDU functions (F1, F3, F4, F5, F6, and F7). Meanwhile, MCA is not the only metaheuristics that can find the global optimal in solving F2. Four comparators including OOA, MA, COA, and WaOA also perform as well as MCA in this function. The performance difference between MCA as the finest performer with the worst performer in solving HDU functions is wide except in F3. This result also demonstrates that MCA can perform well in functions with narrow search space such as F7 to the functions with large search space such as F1.

Table 2. Detail description of the set of 23 functions

No	Function	Model	Dim	Space	Target
1	Sphere	$\sum_{i=1}^{d} x_i^2$	70	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$	70	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	70	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i\rvert, 1 \le i \le d\}$	70	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1}(100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2)$	70	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1}(x_i + 0.5)^2$	70	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\ [0,1]$	70	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	70	[-500, 500]	$-2.9327\text{x}10^4$
9	Ratsrigin	$10d + \sum_{i=1}^{d}\left(x_i^2 - 10cos(2\pi x_i)\right)$	70	[-5.12, 5.12]	0
10	Ackley	$-20 \cdot exp\left(-0.2 \cdot \sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos 2\pi x_i\right) + 20 + exp(1)$	70	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	70	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d-1}\left((y_i - 1)^2(1 + 10\, sin^2(\pi y_{i+1}))\right) + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10,100,4)$	70	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{d-1}\left((x_i - 1)^2(1 + sin^2(3\pi x_{i+1}))\right) + (x_d - 1)^2(1 + sin^2(2\pi x_d))\right\} + \sum_{i=1}^{d} u(x_i, 5,100,4)$	70	[-50, 50]	0
14	Shekel Foxholes	$\left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
15	Kowalik	$\sum_{i=1}^{11}\left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	4	[-5, 5]	0.0003
16	Six Hump Camel	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
17	Branin	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[-5, 5]	0.398
18	Goldstein-Price	$(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)).(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	2	[-2, 2]	3
19	Hartman 3	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j - p_{ij})^2\right)\right)\right)$	3	[1, 3]	-3.86
20	Hartman 6	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j - p_{ij})^2\right)\right)\right)$	6	[0, 1]	-3.32
21	Shekel 5	$-\sum_{i=1}^{5}\left(\sum_{j=1}^{d}(x_j - c_{ji})^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.1532
22	Shekel 7	$-\sum_{i=1}^{7}\left(\sum_{j=1}^{d}(x_j - c_{ji})^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.4028
23	Shekel 10	$-\sum_{i=1}^{10}\left(\sum_{j=1}^{d}(x_j - c_{ji})^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.5363

Received: November 1, 2023. Revised: December 1, 2023.

Table 3. Assessment result in solving HDU functions

F	Parameter	TIA [33]	OOA [22]	MA [29]	COA [9]	WaOA [7]	MCA
1	mean	7.8639×10^1	3.7860×10^2	2.5961×10^1	1.3487×10^3	1.1957×10^1	2.1165
	std deviation	2.3505×10^1	1.2591×10^2	1.2859×10^1	4.0334×10^2	1.1914×10^1	1.3578
	mean rank	4	5	3	6	2	1
2	mean	1.4056×10^{79}	0.0000	0.0000	0.0000	0.0000	0.0000
	std deviation	6.4410×10^{79}	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	6	1	1	1	1	1
3	mean	1.2872×10^4	5.0540×10^4	4.4257×10^4	6.4490×10^4	1.4346×10^4	8.8824×10^3
	std deviation	8.7148×10^3	3.0795×10^4	2.8755×10^4	3.6292×10^4	1.5287×10^4	5.4592×10^3
	mean rank	2	5	4	6	3	1
4	mean	6.4029	1.7076×10^1	2.6616×10^1	3.4435×10^1	3.9455	2.1407
	std deviation	1.4432	6.3579	3.5162×10^1	1.2478×10^1	2.4903	1.0932
	mean rank	3	4	5	6	2	1
5	mean	1.6571×10^3	2.3965×10^4	5.8101×10^2	3.1608×10^5	3.1066×10^2	9.4781×10^1
	std deviation	6.7199×10^2	2.1067×10^4	5.2350×10^2	2.4683×10^5	3.1363×10^2	1.1948×10^1
	mean rank	4	5	3	6	2	1
6	mean	7.3895×10^1	3.2598×10^2	3.9857×10^1	1.2761×10^3	3.1030×10^1	1.7666×10^1
	std deviation	3.7860×10^1	9.7956×10^1	1.4545×10^1	5.2891×10^2	1.1322×10^1	1.5908
	mean rank	4	5	3	6	2	1
7	mean	0.0974	0.2071	0.0685	0.5529	0.0691	0.0600
	std deviation	0.0518	0.0912	0.0410	0.3054	0.0491	0.0293
	mean rank	4	5	2	6	3	1

Table 4. Assessment results in solving HDM functions

F	Parameter	TIA [33]	OOA [22]	MA [29]	COA [9]	WaOA [7]	MCA
8	mean	-2.5450×10^3	-3.7552×10^3	-4.1177×10^3	-4.5380×10^3	-3.9554×10^3	-7.0310×10^3
	std deviation	6.3078×10^2	6.7701×10^2	7.6943×10^2	8.6187×10^2	5.6751×10^2	8.5423×10^2
	mean rank	6	5	3	2	4	1
9	mean	1.2584×10^2	2.1672×10^2	3.8172×10^1	2.0248×10^2	1.6866×10^1	8.7556×10^1
	std deviation	4.6024×10^1	9.5589×10^1	2.1689×10^1	5.0751×10^1	2.2996×10^1	7.8962×10^1
	mean rank	4	6	1	5	2	3
10	mean	2.4081	4.2481	2.6111	6.5991	1.0024	0.3763
	std deviation	0.2794	0.3914	3.8286	1.0143	0.3197	0.1449
	mean rank	3	5	4	6	2	1
11	mean	1.6624	4.6119	1.1855	1.6505×10^1	0.8747	0.3802
	std deviation	0.1796	1.7679	0.2506	7.6192	0.3085	0.2533
	mean rank	4	5	3	6	2	1
12	mean	1.5426	4.5441	1.3821	2.0672×10^3	1.2140	1.0289
	std deviation	0.3277	1.6810	0.2194	5.8857×10^3	0.1539	0.1573
	mean rank	4	5	3	6	2	1
13	mean	6.0742	2.8223×10^2	4.9990	1.5059×10^5	4.0211	3.6486
	std deviation	1.0181	6.6558×10^2	0.8645	2.4455×10^5	0.2799	0.2952
	mean rank	4	5	3	6	2	1

Table 4 reveals the superiority of MCA in overcoming the HDM functions. MCA performs as the finest performer in five functions (F8, F10, F11, F12, and F13). Meanwhile, MCA is the on the third rank in solving F9 where MA and WaOA perform finer. The performance difference between the finest performer and the worst performer is wide in four functions (F10, F11, F12, and F13). Meanwhile, the performance difference between the finest and worst performers is narrow in F8 and F9. This result also indicates the superiority of MCA in solving problems with problem space from moderate to very large.

Table 5 reveals that MCA keeps maintaining its superiority in overcoming the FDM functions. MCA becomes the finest performer in six functions (F14, F19, F20, F21, F22, and F23). Meanwhile, MCA is on the second, third, fourth, and sixth ranks in solving F18, F17, F16, and F15 respectively. Table 5 also presents the fierce competition among these metaheuristics as the performance difference between the finest and worst performers in the FDM

Table 5. Assessment results in solving FDM functions

F	Parameter	TIA [33]	OOA [22]	MA [29]	COA [9]	WaOA [7]	MCA
14	mean	$1.1703x10^1$	$1.0635x10^1$	$1.1772x10^1$	9.5474	$1.0403x10^1$	8.5000
	std deviation	3.5754	3.4058	4.3150	4.2096	4.0650	4.0939
	mean rank	5	4	6	2	3	1
15	mean	0.0089	0.0150	0.0100	0.0116	0.0040	0.0184
	std deviation	0.0150	0.0276	0.0166	0.0110	0.0064	0.0225
	mean rank	2	5	3	4	1	6
16	mean	-1.0096	-1.0104	-1.0037	-0.9862	-1.0282	-1.0074
	std deviation	0.0309	0.0257	0.0377	0.1489	0.0054	0.0654
	mean rank	3	2	5	6	1	4
17	mean	1.4541	0.4486	0.4135	0.5322	0.4255	0.4361
	std deviation	1.5878	0.0677	0.0183	0.3386	0.0691	0.0719
	mean rank	6	4	1	5	2	3
18	mean	$2.5456x10^1$	8.2285	4.0948	$1.3085x10^1$	$2.4068x10^1$	6.9367
	std deviation	$2.9053x10^1$	$1.8180x10^1$	3.6014	$2.1243x10^1$	$2.2121x10^1$	9.2343
	mean rank	6	3	1	4	5	2
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.3429	-2.8659	-3.0598	-2.6866	-2.8492	-3.1935
	std deviation	0.3027	0.2352	0.1111	0.3585	0.2824	0.0956
	mean rank	6	3	2	5	4	1
21	mean	-1.6499	-1.7264	-2.9784	-2.1022	-1.4426	-4.2774
	std deviation	0.9179	0.9996	1.3144	0.3585	0.70006	2.1163
	mean rank	5	4	2	3	6	1
22	mean	-1.6241	-1.7943	-2.7443	-2.0167	-2.1611	-4.8805
	std deviation	0.9227	0.6339	0.9200	0.9322	1.4306	2.0506
	mean rank	6	5	2	4	3	1
23	mean	-1.8202	-1.9911	-3.1775	-2.4700	-2.1356	-4.0106
	std deviation	0.7988	0.7311	1.1947	1.1486	0.8634	2.2060
	mean rank	6	5	2	3	4	1

Table 6. Superiority of MCA based on the group of functions

Group	Number of Functions in Every Group Where MCA is Finer				
	TIA [33]	OOA [22]	MA [29]	COA [9]	WaOA [7]
1	7	6	6	6	6
2	6	6	5	6	5
3	7	7	6	8	6
Total	20	19	17	20	17

functions is narrow and this circumstance is applied to all ten functions in this group.

Table 6 resumes the superiority of MCA compared to all its comparators in all group of functions. Overall, MCA is finer than TIA, OOA, MA, COA, and WaOA in 20, 19, 17, 20, and 17 functions. As OOA, MA, COA, and WaOA also become the finest performer in F2 and all metaheuristics in this evaluation create same result in F19, it means that MCA is worse than TIA, OOA, MA, COA, and WaOA in only 2, 2, 4, 1, and 4 functions respectively. This evaluation reveals that MA and WaOA are comparators that the most difficult to beat. As MCA is worse only in a few functions, it reveals that the refinement of MCA relative to the existing metaheuristics is significant.

In the second evaluation, each search in MCA is assessed individually. As MCA consists of three sequential steps then there are three individual searches evaluated in this evaluation. The average fitness score becomes the only parameter evaluated in this second evaluation. The result is exhibited in Table 7. The finest result in each function is written in bold font.

Table 7 exhibits the significance of the first search. This first search creates the finest result in 14 functions. Meanwhile, the second search becomes the second-finest search as it creates the finest result in nine functions. Then the least significant contribution is provided by the third search as it produces the finest result in two functions. Meanwhile, all searches can find the optimal solution in F2. On the other hand, the first

Table 7. Result of individual search

F	Average Fitness Score		
	1st Search	2nd Search	3rd Search
1	**$1.9386x10^2$**	$4.7470x10^2$	$3.2003x10^4$
2	**0.0000**	**0.0000**	**0.0000**
3	**$3.6278x10^4$**	$4.8706x10^4$	$1.7334x10^5$
4	**$1.4194x10^1$**	$1.6887x10^1$	$5.2272x10^1$
5	**$1.1554x10^4$**	$2.7453x10^4$	$3.2941x10^7$
6	**$1.9377x10^2$**	$5.0243x10^2$	$3.0776x10^4$
7	**0.1777**	0.2768	$3.3780x10^1$
8	**$-5.4171x10^3$**	$-4.5454x10^3$	$-3.1654x10^3$
9	**$2.1127x10^2$**	$5.1916x10^2$	$7.2474x10^2$
10	**3.5097**	4.4463	$1.6658x10^1$
11	**2.9081**	5.3373	$2.8343x10^2$
12	**2.8619**	5.2167	$1.6658x10^7$
13	**$1.4587x10^1$**	$6.0812x10^2$	$7.3509x10^7$
14	$1.1507x10^1$	**9.9197**	$4.0983x10^1$
15	0.0364	**0.0128**	0.0476
16	-0.8560	**-1.0194**	-0.6972
17	6.0931	1.6640	**1.3381**
18	$2.2583x10^1$	$2.1386x10^1$	**$1.5542x10^1$**
19	**-0.0495**	**-0.0495**	-0.0005
20	-2.4141	**-2.6681**	-2.0291
21	-1.4660	**-2.2742**	-1.3143
22	-2.0362	**-3.0268**	-1.7059
23	-1.6770	**-2.5207**	-1.6665

and second searches perform equally in F19. By eliminating these two functions, the first search is dominant in the high dimension functions while the second search is dominant in the fixed dimension functions.

5. Discussion

The in-depth investigation can be started with the performance analysis based on the characteristic of the functions. The HDU functions are used to measure the intensification capability as they have only one global optimal [29]. On the other hand, the HDM functions are used to measure the diversification capability since they have multiple optimal solutions [29]. Meanwhile, the FDM functions have fewer optimal solutions, but their terrain is ambiguous. In some functions, the terrain is commonly flat with very narrow holes for the optimal solutions. The FDM functions are used to investigate the balance between the diversification and intensification capability [29].

Based on the previous explanation and the superiority of MCA in all groups of functions, MCA is proven as a complete metaheuristic. Its intensification and diversification capabilities are good. Moreover, it has balancing capability between the intensification and diversification as it is superior in the third group of functions. As the difference between the first step and the second step is on the reference as each step utilizes the directed search and crossover technique, it can be said that the global finest solution is the suitable reference to handle the high dimension functions. On the other hand, the middle between two stochastically selected solutions is suitable to handle fixed dimension multimodal functions.

The result of the comparative evaluation reveals that the hybridization of the swarm intelligence and crossover technique provides finer result. All the comparators deploy directed search which is the fundamental search in the swarm intelligence. Meanwhile, TIA is the only comparators that does not deploy neighbourhood search. Meanwhile none of the comparators deploy crossover technique.

This investigation is strengthened by the fact that WaOA and COA have a closer relationship with MCA rather than the other comparators. Both comparators deploy three searches. Two searches are directed search while the third search is neighbourhood search with reduction of the local space as iteration increases. The first difference between these two comparators is the number of steps during the iteration. There are three steps in WaoA while COA contains only two steps. Although both comparators deploy the directed search toward the finest solution and relative to a stochastically selected solution, the mechanism is different, and it becomes the second difference. In WaOA, the directed search toward the finest solution is conducted in a dedicated step but the probability of the finest solution is selected as reference is not absolute because the other finer solution also has chance to be selected as reference in this step. Meanwhile, the directed search relative to a stochastically selected solution is performed in a dedicated step. On the other hand, in COA, the directed search toward the finest solution and relative to a stochastically selected solution is performed in a dedicated step [29]. The first half of swarm performs the first directed search while the second half of swarm performs the second directed search. The fourth difference is that the pool used to select a solution stochastically. In WaoA, this random solution is selected among the population of the swarm [7]. On the other hand, in COA, the random solution is generated within the entire space [29].

The result of the single search evaluation exposes the importance of the reference used in the searching process. Both directed search and crossover-based search use reference. On the other hand, there is no reference used in the neighbourhood search. Although the neighbourhood search is proven superior in solving branin and goldstein-price, the neighbourhood search is superior in only two functions while the other searches are superior in many more functions.

The investigation regarding the computing complexity can be traced based on the number of loops used in MCA. In the initialization phase, its complexity can be presented as $O(n(A).n(D))$. This presentation means that the complexity during the initialization is equivalent to the swarm size or the number of decision variables. Meanwhile, in the iteration phase, its complexity can be presented as $O(3t_m.n(A).n(D))$. This presentation means that the complexity during the iteration phase is equivalent to the maximum iteration, swarm size, and the number of decision variables.

There are limitations in this work, especially the provided MCA despite its superior performance. First, the contribution of the neighbourhood search performed in the third step is not as significant as the combination of directed search and crossover technique performed in the first and second steps. Second, there are only five metaheuristics used as comparators in this work because it is impossible and not operable to investigate the performance of MCA with too many comparators. Third, there are a lot of techniques and stochastic distributions that already exist, but it is also impossible to accommodate too many techniques or stochastic distributions in a single metaheuristic. Fourth, there are also several other sets of functions, and a lot of practical problems can be used as use case to investigate the performance of MCA more comprehensively. These limitations can be used as baseline for further development by accommodating more but selected additional techniques and utilizing many practical problems as use cases to enrich the investigation of MCA.

6. Conclusion

This work presents a new metaheuristic called as migration-crossover algorithm (MCA). As the name suggests, MCA is swarm-based metaheuristic enriched with the crossover technique which is the fundamental search in the evolution-based metaheuristic and the neighborhood search which is the fundamental search in many single solution-based metaheuristic. This paper also presents the novelty of MCA in the equal opportunity between the directed search and the crossover technique in its first and second steps; and the unbalanced search space during the neighborhood search performed in the third step. Two evaluations have also been presented in this paper, including the comparative evaluation and single search evaluation. The result of the comparative evaluation exhibits the superiority of MCA over its comparators by being finer than TIA, OOA, MA, COA, and WaOA in 20, 19, 17, 20 and 17 functions respectively. On the other hand, the result of the single search evaluation exhibits the dominant contribution of the global finest solution in overcoming the high dimension functions and the middle of two stochastically selected solutions in overcoming the fixed dimension functions.

The hybridization between the swarm intelligence and evolutionary based technique can be used as inspiration for future development of metaheuristic. Many derivatives of swarm intelligence and evolutionary based techniques have not been explored yet. Moreover, there is a challenge to modify the neighborhood search to make it competitive again compared to the swarm intelligence or evolutionary based technique. Moreover, the practical use cases whether they are engineering, or non-engineering problems are important to use to evaluate the MCA more comprehensively.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

Acknowledgments

The funding for publication of this paper is provided by Telkom University.

References

[1] R. Ramesh and S. Sathiamoorthy, "Blood Vessel Segmentation and Classification for Diabetic Retinopathy Grading Using Dandelion Optimization Algorithm with Deep Learning Model", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 11-20, 2023, doi: 10.22266/ijies2023.1031.02.

[2] K. Manivannan and S. Sathiamoorthy, "Pelican Optimization with Majority Voting Ensemble Model for Tuberculosis Detection and

Classification on Chest X-Ray Images", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 1-10, 2023, doi: 10.22266/ijies2022.1231.14.

[3] D. Raja and M. Karthikeyan, "Red Deer Optimization with Deep Learning Enabled Agricultural Plant Disease Detection and Classification Model", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 21-30, 2023, doi: 10.22266/ijies2023.1031.03.

[4] M. Arunadev and V. Sathya, "DDoS Attack Detection using Back Propagation Neural Network Optimized by Bacterial Colony Optimization", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 301-312, 2023, doi: 10.22266/ijies2023.1031.26.

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

[6] M. Dehghani, P. Trojovsky, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 8, ID. 121, pp. 1-60, 2023.

[7] P. Trojovsky and M. Dehghani, "A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Walruses Behavior", *Scientific Reports*, Vol. 13, ID. 8775, pp. 1-32, 2023.

[8] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer", *Expert Systems with Applications*, Vol. 191, ID. 116158, pp. 1-33, 2022.

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

[10] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 49445-49473, 2022.

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

[12] M. S. Braik, "Chameleon Swarm Algorithm: A Bio-inspired Optimizer for Solving Engineering Design Problems", *Expert Systems with Applications*, Vol. 174, ID. 114685, pp. 1-25, 2021.

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

[14] E. Trojovska and M. Dehghani, "Clouded Leopard Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 102876-102906, 2022.

[15] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems", *IEEE Access*, Vol. 9, pp. 162059-162080, 2021.

[16] P. Trojovsky and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications", *Sensors*, Vol. 22, ID. 855, pp. 1-34, 2022.

[17] P. Coufal, S. Hubalovsky, M. Hubalovska, and Z. Balogh, "Snow Leopard Optimization Algorithm: A New Nature-Based Optimization Algorithm for Solving Optimization Problems", *Mathematics*, Vol. 9, ID. 2832, pp. 1-26, 2021.

[18] D. Polap and M. Wozniak, "Red Fox Optimization Algorithm", *Expert Systems with Applications*, Vol. 166, ID. 114107, pp. 1-21, 2021.

[19] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian Tiger Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *IEEE Access*, Vol. 10, pp. 132396-132431, 2022.

[20] M. Braik, A. Hammouri, J. Atwan, M. A. A. Betar, and M. A. Awadallah, "White Shark Optimizer: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization Problems", *Knowledge-Based Systems*, Vol. 243, ID. 108457, pp. 1-29, 2022.

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

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

[23] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "RSLBO: Random Selected Leader Based Optimizer", *International Journal of Intelligent*

Engineering and Systems, Vol. 14, No. 5, pp. 529-538, 2021, doi: 10.22266/ijies2021.1031.46.
[24] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. R. Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, “MLO: Multi Leader Optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 364–373, 2020, doi: 10.22266/ijies2020.1231.32.
[25] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, “MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 472-479, 2021, doi: 10.22266/ijies2021.0831.41.
[26] M. Dehghani and P. Trojovsky, “Hybrid Leader Based Optimization: A New Stochastic Optimization Algorithm for Solving Optimization Applications”, *Scientific Reports*, Vol. 12, ID: 5549, pp. 1-16, 2022.
[27] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, “TIMBO: Three Influential Members Based Optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 121-128, 2021, doi: 10.22266/ijies2021.1031.12.
[28] E. Trojovska and M. Dehghani, "A New Human-based Metaheuristic Optimization Method based on Mimicking Cooking Training", *Scientific Reports*, Vol. 12, ID. 14861, pp. 1-24, 2022.
[29] P. Trojovsky and M. Dehghani, “Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems”, *Computer Modeling in Engineering & Sciences*, Vol. 137, No. 2, pp. 1695-1730, 2023.
[30] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, “Golden Search Optimization Algorithm”, *IEEE Access*, Vol. 10, pp. 37515–37532, 2022.
[31] 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.
[32] P. Kusuma and A. Dinimaharawati, "Four Directed Search Algorithm: A New Optimization Method and Its Hyper Strategy Investigation", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 598-611, 2023, doi: 10.22266/ijies2023.1031.51.
[33] P. D. Kusuma and A. Novianty, “Total Interaction Algorithm: A Metaheuristic in Which Each Agent Interacts with All Other Agents”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 224-234, 2023, doi: 10.22266/ijies2023.0228.20.
[34] P. Kusuma and A. L. Prasasti, "Walk-Spread Algorithm: A Fast and Superior Stochastic Optimization", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 275-288, 2023, doi: 10.22266/ijies2023.1031.24.
[35] P. D. Kusuma and F. C. Hasibuan, “Attack-Leave Optimizer: A New Metaheuristic that Focuses on The Guided Search and Performs Random Search as Alternative”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 244–257, 2023, doi: 10.22266/ijies2023.0630.19.
[36] A. Kuznetsov, L. Wieclaw, N. Poluyanenko, L. Hamera, S. Kandiy, and Y. Lohachova, "Optimization of a Simulated Annealing Algorithm for S-Boxes Generating", *Sensors*, Vol. 22, ID. 6073, pp. 1-19, 2022.
[37] S. Katoch, S. S. Chauhan, and V. Kumar, "A Review on Genetic Algorithm: Past, Present, And Future", *Multimedia Tools and Applications*, Vol. 80, pp. 8091–8126, 2021.
[38] A. G. Gad, “Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review”, *Archives of Computational Methods in Engineering*, Vol. 29, pp. 2531–2561, 2022.