International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 2, April 2024, pp. 2145~2155
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i2.pp2145-2155

Swarm flip-crossover algorithm: a new swarm-based metaheuristic enriched with a crossover strategy

Purba Daru Kusuma, Faisal Candrasyah Hasibuan
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Aug 9, 2023
Revised Oct 31, 2023
Accepted Nov 29, 2023

Keywords:

Artificial intelligence
Crossover
Metaheuristic
Optimization
Swarm intelligence

ABSTRACT

A new swarm-based metaheuristic that is also enriched with the crossover technique called swarm flip-crossover algorithm (SFCA) is introduced in this work. SFCA uses swarm intelligence as its primary technique and the crossover as its secondary one. It consists of three searches in every iteration. The swarm member walks toward the best member as the first search. The central point of the swarm becomes the target in the second search. There are two walks in the second search. The first walk is getting closer to the target, while the second is avoiding the target. The better result between these two walks becomes the candidate for the replacement. In the third search, the swarm member performs balance arithmetic crossover with the central point of the space or jumps to the opposite location within the area (flipping). The assessment is taken by confronting SFCA with five new metaheuristics: slime mold algorithm (SMA), golden search optimization (GSO), osprey optimization algorithm (OOA), coati optimization algorithm (COA), and walrus optimization algorithm (WaOA) in handling the set of 23 functions. The result shows that SFCA performs consecutively better than SMA, GSO, OOA, COA, and WaOA in 20, 23, 17, 17, and 17 functions.

Corresponding Author:

Purba Daru Kusuma
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: purbodaru@telkomuniversity.ac.id

1. INTRODUCTION

In recent years, many optimization studies utilized metaheuristics as the primary tool for optimization. Metaheuristic algorithms have also been combined with other techniques to solve various problems, especially in engineering. For example, grey wolf optimization (GWO) was combined with deep hybrid learning to classify lung cancer [1]. GWO was also utilized to solve the task allocation problem for multiple unmanned aerial vehicles (UAV) [2]. Harris hawk optimization (HHO) has been used to improve the quality of the kernel type in the support vector machine (SVM) technique [3]. Seagull optimization (SO) has been utilized to search the global maximum power point in the photovoltaic system [4]. The Archimedes optimizer algorithm (AOA) was used in the optimal power flow system by minimizing the emission, power loss, and fuel cost while maximizing the voltage profile [5].

Many metaheuristics developed in recent years were constructed based on swarm intelligence. Many of these metaheuristics use animal behavior as metaphors, such as walrus optimization algorithm (WaOA) [6], coati optimization algorithm (COA) [7], osprey optimization algorithm (OOA) [8], slime mold algorithm (SMA) [9], komodo mlipir algorithm (KMA) [10], zebra optimization algorithm (ZOA) [11], pelican optimization algorithm (POA) [12], golden jackal optimization (GJO) [13], clouded leopard optimization (CLO) [14], Siberian tiger optimization (STO) [15], marine predator algorithm (MPA) [16], Tasmanian devil optimization (TDO) [17], northern goshawk optimization (NGO) [18], and so on. Some metaheuristics utilized their member as metaphors, such as three influential member-based optimizations (TIMBO) [19], multileader optimization (MLO) [20], mixed leader-based optimization (MLBO) [21], hybrid leader-based optimization (HLBO) [22], and so on. Some metaheuristics do not use any metaphor for their name, such as total interaction algorithm (TIA) [23], golden search optimization (GSO) [24], three-on-three optimizer (TOTO) [25], multiple interaction optimizer (MIO) [26], attack leave optimization (ALO) [27], average subtraction-based optimization (ASBO) [28], particle swarm optimization (PSO) [29], and so on.

On the other hand, the crossover mechanism has become less popular. The coronavirus optimization algorithm (COVIDOA) is an example of a new metaheuristic that uses crossover as its primary search [30]. In COVIDOA, the parents are selected based on the roulette wheel regarding the quality of the current solution. In COVIDOA, the crossover technique is enriched with a frameshifting mechanism. The other example of a new crossover-based metaheuristic is the flower pollination algorithm (FPA) [31]. In FPA, the parents are selected randomly.

Despite the rapid development of metaheuristics, several issues accompany this trend. First, the extensive use of metaphors in these studies poses a challenge. Many metaheuristics employ metaphors to showcase their novelty, yet this approach hinders the investigation of their actual originality. Second, locating swarm-based metaheuristics that effectively incorporate crossover strategies proves difficult. Most of them utilize neighborhood searches with reduced search spaces as supplementary exploration. Last, the assessment of individual search methods within multi-strategy embracing metaheuristics needs to be revised. This circumstance makes it challenging to accurately gauge the contributions of each strategy in shaping the metaheuristic framework.

Based on this problem, the main objective of this work is to construct a new metaphor-free metaheuristic based on the swarm intelligence approach but also accommodates the flip and crossover mechanism. This objective is then translated into the scientific contributions provided by this work as follows: i) A new metaphor-free metaheuristic called swarm flip-crossover algorithm (SFCA) is introduced based on swarm intelligence as a core approach but enriched with flip and crossover mechanisms, ii) The comparative test is taken to assess the performance of SFCA as a whole package by confronting SFCA with five new swarm-based metaheuristics, and iii) The individual search test is taken to assess the performance of each search performed in SFCA.

The structure of this paper is arranged as follows. The background, problem statement, research objective, and contribution of this work are declared in section one. The studies regarding the metaheuristic, especially swarm intelligence, including several newest swarm-based metaheuristics, are reviewed in section two. The methodology, including the model and assessment scenario presentation, can be seen in section three. The assessment result and the discussion regarding the findings, complexity, and limitation are performed in section four. In the end, the conclusion of this work and the proposal for future studies are presented in section five.

2. RELATED WORKS

Swarm intelligence is a branch of the metaheuristic, especially the population-based one, with the uniqueness of constructing a collective of autonomous agents. Each agent works independently based on its understanding of its condition and environment. Rather than using centralized mechanisms like genetic algorithms and other evolution-based metaheuristics, decentralized behavior, i.e., autonomy, becomes the main distinction between swarm intelligence and other population-based metaheuristics. This strategy makes the members of the swarm become active entities. Besides the decentralized and self-organized system, swarm intelligence exploits the interaction among the swarm members to boost its performance.

The interaction among members becomes the backbone of the search process in swarm-based metaheuristics. Some swarm-based metaheuristics perform single-stage interaction, while others perform multiple-stage interaction. The fundamental concept of this interaction is determining the member or members chosen for each member to interact with. The global best member or the best member within the swarm is often used as a reference, such as in KMA [10], COA [7], ZOA [11], and so on. In some metaheuristics, such as TIA [23] and ZOA [11], each swarm member interacts with other members within the swarm. The interaction of the mixture between the global best member and the local best member can be found in PSO [29] and GSO [24]. Meanwhile, a member may interact with two randomly selected members, like in MPA [16], and SMA [9]. A randomly generated member around the space can also be utilized as a partner for interaction, like in POA [12] and COA.

The random search is also the secondary search in the swarm-based metaheuristic. Many recent swarm-based metaheuristics are enriched with the neighborhood search with reduced space during iteration. MPA becomes the early adaptor of this strategy [16]. Then some others also utilize this strategy, like ZOA

[11], COA [7], OOA [8], POA [12], and so on. Full random search is also utilized in some metaheuristics, such as in KMA during parthenogenesis [10], in SMA when a specific circumstance occurs [9], or in ALO when stagnation occurs [27].

Even so, the crossover process is less popular as a secondary or complementary process in developing a swarm-based metaheuristic. Meanwhile, evolution-based metaheuristics, such as genetic algorithm (GA), where the crossover becomes its backbone, are still widely utilized, and hybridized, such as in robotic navigation [32], food distribution [33], logistics and distribution centers for agricultural products [34], and so on. KMA is an example of a few swarm-based metaheuristics enriched with crossover. In KMA, the arithmetic crossover occurs between the moderate-quality swarm members with the highest-quality.

Based on this review, the opportunity to develop a new metaheuristic is still open. First, many interactions can be explored to become the strategy of the swarm-based metaheuristic. Second, there is a challenge to examine and promote crossover as a complement for the interaction-based search in the swarm-based metaheuristic due to its rarity rather than the neighborhood search, which is commonly utilized.

3. METHOD

3.1. Model

The conceptual design of SFCA can be perceived from its name, as it is a metaphor-free metaheuristic. SFCA contains three key terms: swarm, flip, and crossover. As a swarm intelligence, SFCA is constructed of a collection of autonomous agents called a swarm. These independent agents are then called swarm members. They move autonomously without any central command but with a certain degree of interaction among the swarm members, so their movement is not entirely random. The flip can be seen as a mirroring mechanism so that the swarm member will bounce to a new location within the space where this new location is the opposite of the current site. The crossover can be interpreted as the swarm member mixing with a reference to generate a unique solution.

This conceptual design is subsequently translated into a concrete strategy, wherein each swarm member conducts three sequential walks, termed "searches", during each iteration. In the first walk, each swarm member walks toward the best swarm member. In the second walk, each swarm member walks relative to the center of the swarm in two opposite directions. In the first direction, it walks towards the center of the swarm. In the second direction, it avoids the center of the swarm. The more favorable outcome between these two walks is then selected to replace the current solution. The third walk phase involves each swarm member deciding between executing a flip or crossover, determined stochastically with equal probabilities. Here, the center of the search space serves as a reference: a 'flip' results in a jump to the opposite space region, while a 'crossover' involves performing a balanced arithmetic crossover with the center. Figure 1 provides a visual representation of these walks. SFCA employs a stringent acceptance rule for swarm members to avert undesirable shifts towards inferior solutions. Figure 1(a) visualizes the walking of the swarm member toward the best swarm member in the first search. Figure 1(b) visualizes the walking toward and away from the center of the swarm as performed in the second search. Figures 1(c) and 1(d) visualize the third search where Figure 1(c) represents the flipping-based search while Figure 1(d) represents the crossover-based search.

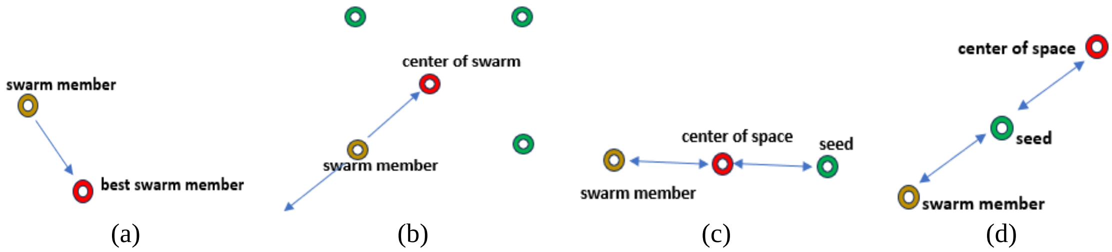

Figure 1. Visualization of the swarm, flip, and crossover: (a) walk toward the best swarm member, (b) walk relative to the center of the swarm, (c) flip, and (d) balance arithmetic crossover

The reasoning behind the strategy chosen for SFCA is as follows. The walk toward the best swam member is designed to improve the solution fast, especially in handling unimodal problems. In this case, walking toward and surpassing the best swarm member increases the probability of improvement. The walk relative to the swarm's center provides exploitation and exploration. The walk toward the center of the swarm intensifies the search within the region of the swarm, especially around the center. On the other hand, the walk away from the swarm's center provides an effort to expand the swarm size to explore a new location beyond the region of the swarm. The flip mechanism is designed to jump to another part of space. This walk can be seen as an exploration, especially when the current location is far from the center of space. On the other hand, the crossover can be perceived as exploitation because this walk pushes the swarm member to get closer to the center of space. The flip is designed to handle the optimization problem where the global optimal solution is far from the center of space. In contrast, the crossover is designed to handle the optimization problem where the global optimal solution is near the center of space.

The formalization of SFCA is presented by using pseudocode to figure out the whole process, while the mathematical formulation describes the more detailed presentation of the process. Meanwhile, the annotations used in this formalization are presented in Table 1. In Algorithm 1, the initialization phase is presented from lines 2 to 5. On the other hand, the iteration phase is presented from lines 6 to 15. The algorithm's output, i.e., the final solution, is the best swarm member.

Table 1. List of annotations

Notation	Description
c_c	Center of space
c_l	The lower boundary of space
c_u	The upper boundary of space
d	Dimension
f	Objective function
s	Swarm member
S	Swarm
s_b	The best swarm member
s_c	Center of the swarm
s_s	Swarm seed
t	Iteration
t_m	Maximum iteration
U_c	Continuous uniform random
U_d	Discrete uniform random

Algorithm 1. Swarm flip-crossover algorithm

```
1   begin
2     for all s in S
3       initialize si using (1)
4       update sb using (2)
5     end for
6     for t=1 to tm
7       generate sc using (3)
8       for all s in S
9         perform the first walk using (4) and (5)
10        perform the second walk using (6) to (8), (5)
11        perform the third walk using (8) and (5)
12        update sb using (2)
13      end for
14    end for
15  end
16  return sb
```

There are two processes performed during the initialization phase. The first one is generating the initial swarm member uniformly within the space, as stated in (1). Then, the updating process of the best swarm member is taken using (2) which represents the stringent acceptance procedure.

$$s_{i,j} = c_{l,j} + U_c(0,1)(c_{u,j} - c_{l,j}) \tag{1}$$

$$s_b' = \begin{cases} s_i, f(s_i) < f(s_b) \\ s_b, else \end{cases} \tag{2}$$

The first step in every iteration is finding the center point of the swarm. As a solution consists of multi-dimension parameters, the central issue is determined for every dimension. This center point is calculated using (3), representing the average of a certain extent for all swarm members. Equation (3) highlights that this center point needs to be weighted, which means that the quality of the swarm members needs to be considered.

$$s_{c,j} = \frac{\sum_S s_{i,j}}{n(S)} \tag{3}$$

The first walk consists of two processes. The first process is generating the seed based on the walk toward the best swam member. This process is stated in (4). The discrete uniform random in (4) means that the seed may be generated between the corresponding swarm member and the best swam member and can surpass the best swarm member. The second process is updating the corresponding member based on the quality of its seed, as stated in (5). This second process is also performed in the second and third walks.

$$s_{s,i,j} = s_{i,j} + U_c(0,1).(s_{b,j} - U_d(1,2).s_{i,j}) \tag{4}$$

$$s_i' = \begin{cases} s_{s,i}, f(s_{s,i}) < f(s_i) \\ s_i, else \end{cases} \tag{5}$$

The second walk is the guided search relative to the swarm's center. As mentioned previously, there are two sub-walks. The first sub-walk is walking toward the swarm's center, as stated in (6). The second sub-walk is walking away from the swarm's center, as stated in (7). Then the better among both sub-walks become the seed, as stated in (8).

$$s_{s1,i,j} = s_{i,j} + U_c(0,1).(s_{c,j} - U_d(1,2).s_{i,j}) \tag{6}$$

$$s_{s1,i,j} = s_{i,j} + U_c(0,1).(s_{i,j} - U_d(1,2).s_{c,j}) \tag{7}$$

$$s_{s,i} = \begin{cases} s_{s1,i}, f(s_{s1,i}) < f(s_{s2,i}) \\ s_{s2,i,else} \end{cases} \tag{8}$$

The third walk reflects the flip and crossover. The decision between flip and crossover is determined uniformly and equally, as stated in (9). If the flip is chosen, the seed is generated in the left region of the space if the swarm member is in the right area. Otherwise, the source is generated in the right part of the space. Equation (9) also states that the seed is generated right in the middle between the swarm member and the center of space if the crossover is chosen.

$$s_{s,i,j} = \begin{cases} c_{c,j} - |s_{i,j} - c_{c,j}|, U_c(0,1) < 0.5 \ \wedge s_{i,j} \geq c_{c,j} \\ c_{c,j} - |s_{i,j} - c_{c,j}|, U_c(0,1) < 0.5 \ \wedge s_{i,j} < c_{c,j} \\ \frac{c_{c,j}+s_{i,j}}{2}, U_c(0,1) \geq 0.5 \end{cases} \tag{9}$$

3.2. Assessment scenario

The assessment of swarm flip-crossover algorithm is performed by challenging it to tackle a wide range of optimization problems. In this work, two assessments are performed based on their objectives. The first assessment is a comparative assessment. SFCA, as the proposed metaheuristic, is confronted with other metaheuristics in this assessment. Its objective is to observe the comparative advantages, strengths, and weaknesses of SFCA. The second assessment is the individual search assessment. In this assessment, the performance of each search of SFCA is tested. Its objective is to evaluate the performance of each search constructed in SFCA.

In the comparative assessment, SFCA faces five new swarm-based metaheuristics: SMA, GSO, OOA, COA, and WaOA. All these metaheuristics are pure swarm-based metaheuristics. The reasoning for selecting these five metaheuristics as the confronters is as follows. First, all these metaheuristics are new. Second, these five metaheuristics represent a wide range of strategies. GSO represents the metaheuristic that performs guided search only with the mixture of the global best swarm member and the local best member becoming the reference. SMA represents metaheuristics that perform a guided search, full random search, and neighborhood search based on the circumstance faced by the swarm member. OOA, COA, and WaOA, are metaheuristics that perform both guided search and neighborhood search. Meanwhile, there is a difference in the references used during the guided search in these three metaheuristics. SMA is the only confronter with another adjusted parameter besides the swarm size and maximum iteration. In this assessment, the z parameter for SMA is set to 0.1. Meanwhile, for SFCA and all its confronters, the swarm size is set to 5 while the maximum iteration is set to 10. It means that this assessment represents the optimization process with low computational expense.

In the individual search assessment, each search is assessed individually. When a search is assessed, the other searches in SFCA are set to inactive. Unlike the comparative assessment, there is no confrontation with other metaheuristics in this assessment. Same as the comparative assessment, in this assessment, the swarm size is set to 5 while the maximum iteration is set to 10.

In both assessments, the use case is the set of 23 functions. This use case is commonly used in many studies proposing new metaheuristics. It covers various circumstances in optimization problems. It contains seven unimodal functions with only one optimal solution and 16 multimodal functions with multiple optimal solutions but only one global optimal one. All these unimodal functions have high problem dimensions. Meanwhile, the multimodal functions can be split into two groups. The first group consists of six functions with high problem dimensions, while the dimensions of the other ten functions are fixed. A detailed description of these functions is shown in Table 2.

Table 2. A detailed description of the set of 23 functions

No	Function	Dim	Space range	Optimal score
F1	Sphere	60	[-100, 100]	0
F2	Schwefel 2.22	60	[-100, 100]	0
F3	Schwefel 1.2	60	[-100, 100]	0
F4	Schwefel 2.21	60	[-100, 100]	0
F5	Rosenbrock	60	[-30, 30]	0
F6	Step	60	[-100, 100]	0
F7	Quartic	60	[-1.28, 1.28]	0
F8	Schwefel	60	[-500, 500]	$-418.9 \times dim$
F9	Ratsrigin	60	[-5.12, 5.12]	0
F10	Ackley	60	[-32, 32]	0
F11	Griewank	60	[-600, 600]	0
F12	Penalized	60	[-50, 50]	0
F13	Penalized 2	60	[-50, 50]	0
F14	Shekel Foxholes	2	[-65, 65]	1
F15	Kowalik	4	[-5, 5]	0.0003
F16	Six Hump Camel	2	[-5, 5]	-1.0316
F17	Branin	2	[-5, 5]	0.398
F18	Goldstein-Price	2	[-2, 2]	3
F19	Hartman 3	3	[1, 3]	-3.86
F20	Hartman 6	6	[0, 1]	-3.32
F21	Shekel 5	4	[0, 10]	-10.1532
F22	Shekel 7	4	[0, 10]	-10.4028
F23	Shekel 10	4	[0, 10]	-10.5363

4. RESULTS AND DISCUSSION
4.1. Assessment result

The result of the first assessment is presented in Tables 3 to 5. Table 3 presents the comparative result for the seven high dimension unimodal functions. Table 4 presents the comparative result for the six high-dimension multimodal functions. Table 5 presents the comparative result for the ten fixed-dimension multimodal functions. Three parameters are presented in Tables 3 to 5: the mean or average score, standard deviation, and mean rank.

Table 3 shows that SFCA is superior in overcoming the high-dimension unimodal functions. Among these six functions, SFCA performs as the best metaheuristic of the seven functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic). Meanwhile, four confronters also become the best performer in solving Schwefel 2.22. These confronters are SMA, OOA, COA, and WaOA. Moreover, SFCA also can find the global optimal solution in handling Sphere and Schwefel 2.22. Although most metaheuristics can find the global optimal solution of Schwefel 2.22, the exceptionally high consequence is faced by GSO as the only metaheuristic that fails to find the global optimal solution of this function. GSO is the worst performer in this first group of functions, as it has been in sixth place six times and fifth place once. Meanwhile, SMA is the second worst performer due to its position in fifth place five times and sixth place once. The performance gap between SFCA, GSO, and SMA in solving functions in the first group is wide.

Table 4 shows that SFCA is still superior in five functions but inferior in one function in solving the high-dimension multimodal functions. The five functions where SFCA becomes the best performer are Rastrigin, Ackley, Griewank, Penalized, and Penalized 2. Meanwhile, the function where SFCA becomes the second worst performer is Schwefel. The performance gap between SFCA and the second-best performer is wide in solving Rastrigin, Ackley, and Griewank, while this gap is narrow in solving Penalized and Penalized 2. Meanwhile, the performance gap between SFCA as the best performer and worst performer are wide in solving Rastrigin, Ackley, Griewank, Penalized, and Penalized 2.

Table 3. Comparison results on solving seven high dimension unimodal functions

F	Parameter	SMA	GSO	OOA	COA	WaOA	SFCA
1	mean	9.7176×10^{3}	6.1191×10^{4}	2.8849×10^{2}	1.1297×10^{3}	8.0190	0.0001
	std deviation	1.1115×10^{4}	1.0431×10^{4}	1.3270×10^{2}	4.3646×10^{2}	6.1864	0.0000
	mean rank	5	6	3	4	2	1
2	mean	0.0000	4.9391×10^{79}	0.0000	0.0000	0.0000	0.0000
	std deviation	0.0000	1.7586×10^{80}	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
3	mean	1.2468×10^{5}	1.8998×10^{5}	3.3150×10^{4}	4.6116×10^{4}	5.5093×10^{3}	1.4939×10^{3}
	std deviation	1.4286×10^{5}	9.5568×10^{4}	1.8788×10^{4}	1.7374×10^{4}	5.5602×10^{3}	2.9010×10^{3}
	mean rank	5	6	3	4	2	1
4	mean	3.9991×10^{1}	6.5303×10^{1}	1.4708×10^{1}	2.7755×10^{1}	4.0494	0.0254
	std deviation	2.6575×10^{1}	9.3254	3.7040	9.0652	1.6855	0.0160
	mean rank	5	6	3	4	2	1
5	mean	2.9292×10^{7}	1.1202×10^{8}	1.3455×10^{4}	1.4552×10^{5}	2.1437×10^{2}	5.8939×10^{1}
	std deviation	3.8990×10^{7}	4.8137×10^{7}	8.8386×10^{3}	9.3098×10^{4}	2.2047×10^{2}	0.0293
	mean rank	5	6	1	2	4	1
6	mean	1.0031×10^{4}	5.3961×10^{4}	2.6091×10^{2}	1.0383×10^{3}	2.3016×10^{1}	1.3198×10^{1}
	std deviation	1.0705×10^{4}	1.0267×10^{4}	1.2494×10^{2}	6.5590×10^{2}	1.1422×10^{1}	0.3443
	mean rank	5	6	3	4	2	1
7	mean	4.3465×10^{2}	1.3625×10^{2}	0.1862	0.5627	0.0638	0.0173
	std deviation	4.0570×10^{2}	5.2474×10^{1}	0.1030	0.2697	0.0371	0.0166
	mean rank	6	5	3	4	2	1

Table 4. Comparison results on solving six high-dimension multimodal functions

F	Parameter	SMA	GSO	OOA	COA	WaOA	SFCA
8	mean	-4.5652×10^{3}	-2.7963×10^{3}	-3.5439×10^{3}	-4.3278×10^{3}	-3.7470×10^{3}	-3.3106×10^{3}
	std deviation	1.3233×10^{3}	9.5385×10^{2}	5.1313×10^{2}	7.2595×10^{2}	5.7246×10^{2}	4.3474×10^{2}
	mean rank	1	6	4	2	3	5
9	mean	1.1687×10^{2}	6.2795×10^{2}	1.7180×10^{2}	1.8141×10^{2}	1.8793×10^{1}	0.0001
	std deviation	7.3826×10^{1}	6.4445×10^{1}	7.1581×10^{1}	4.6234×10^{1}	1.9264×10^{1}	0.0002
	mean rank	3	6	4	5	2	1
10	mean	8.0391	1.8867×10^{1}	4.2468	6.3539	0.8635	0.0012
	std deviation	3.8934	0.6352	0.5269	1.0980	0.4314	0.0004
	mean rank	5	6	3	4	2	1
11	mean	8.3380×10^{1}	5.7091×10^{2}	3.6909	1.1803×10^{1}	0.7367	0.0004
	std deviation	9.0354×10^{1}	1.2257×10^{2}	0.8468	3.4572	0.3899	0.0012
	mean rank	5	6	3	4	2	1
12	mean	4.3747×10^{7}	1.8014×10^{8}	3.0479	7.1408×10^{3}	1.2132	1.0212
	std deviation	6.1472×10^{7}	1.0941×10^{8}	0.6906	3.4572	0.1665	0.1060
	mean rank	5	6	3	4	2	1
13	mean	1.0905×10^{8}	4.8723×10^{8}	1.1491×10^{2}	6.1818×10^{4}	4.3027	3.1849
	std deviation	1.6513×10^{8}	2.0981×10^{8}	4.3842×10^{2}	9.5186×10^{4}	0.4336	0.0716
	mean rank	5	6	3	4	2	1

Table 5 indicates very tight competition in the third cluster of functions. In this cluster, the performance gap between the best and worst performers is generally narrow. The widest performance gap takes place in solving Goldstein Price. In this cluster, SFCA is in the first rank in six functions (Shekel Foxholes, Six Hump Camel, Hartman 3, Shekel 5, Shekel 7, and Shekel 10), second rank in one function (Kowalik), third rank in one function (Goldstein Price), and fourth rank in two functions (Branin and Goldstein-Price).

The result from Tables 3 to 5 is then transformed into the cluster-based superiority result. This summary is presented in Table 6. The data in Table 6 represents the number of functions in each cluster where swarm flip-crossover algorithm (SFCA) is better than its related confronter. Table 6 shows that, in general, SFCA is better than its confronters by outperforming SMA, GSO, OOA, COA, and WaOA in 20, 23, 17, 17, 17 functions consecutively. This distribution makes GSO the easiest confronter to outperform due to its inferiority to SFCA in all functions. The superiority of SFCA occurs in all three groups of functions.

The individual search result is presented in Table 7. Table 7 presents the best result for every function in bold font. The result shows that the walk relative to the center of the swarm becomes the primary contributor by producing distinct finest result in 13 functions. Most of these functions are fixed dimension multimodal functions. On the other hand, the walk toward the best swarm member becomes the best performer by achieving the best result in 8 functions. Most of them have high dimension functions. The contribution of the third walk is minimal.

Table 5. Comparison results on solving ten fixed-dimension multimodal functions

F	Parameter	SMA	GSO	OOA	COA	WaOA	SFCA
14	mean	1.1118×10^{1}	1.9176×10^{1}	1.1979×10^{1}	8.4474	1.0090×10^{1}	7.2195
	std deviation	4.5153	3.1879×10^{1}	5.5551	4.9013	3.8637	3.4174
	mean rank	4	6	5	2	3	1
15	mean	0.0805	0.0877	0.0126	0.0105	0.0025	0.0028
	std deviation	0.0675	0.1396	0.0171	0.0112	0.0021	0.0064
	mean rank	5	6	4	3	1	2
16	mean	-0.4764	0.4417	-1.0177	-1.0164	-1.0220	-1.0290
	std deviation	0.5079	4.9891	0.0136	0.0176	0.0101	0.0041
	mean rank	5	6	3	4	2	1
17	mean	1.5162	5.8448	0.4429	0.5030	0.4156	0.5663
	std deviation	2.3003	7.7717	0.0998	0.1934	0.0288	0.2735
	mean rank	5	6	2	3	1	4
18	mean	4.7935×10^{1}	6.9103×10^{1}	4.0553	7.9342	2.6964×10^{1}	1.3157×10^{1}
	std deviation	9.5125×10^{1}	1.3696×10^{2}	4.0079	9.5046	3.2071×10^{1}	1.4102×10^{1}
	mean rank	5	6	1	2	4	3
19	mean	-0.0495	-0.0043	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0070	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
20	mean	-1.8302	-1.8642	-2.8461	-2.7989	-2.8490	-2.6478
	std deviation	1.1560	0.6568	0.3028	0.3781	0.2101	0.3422
	mean rank	6	5	2	3	1	4
21	mean	-1.9561	-1.3389	-1.9334	-2.5711	-1.8204	-3.0673
	std deviation	2.3059	0.8458	1.5489	1.2333	0.7647	1.1105
	mean rank	3	6	4	2	5	1
22	mean	-1.6622	-1.5209	-1.4523	-2.3632	-1.9250	-3.3423
	std deviation	1.7844	0.6889	0.4693	0.9066	0.8939	1.6772
	mean rank	4	5	6	2	3	1
23	mean	-1.6778	-1.6927	-1.8564	-2.2863	-2.1163	-3.6338
	std deviation	0.9929	0.8594	0.4571	0.5919	0.7380	1.5299
	mean rank	6	5	4	2	3	1

Table 6. Cluster-based superiority result

Cluster	SMA	GSO	OOA	COA	WaOA
1	6	7	6	6	6
2	5	6	5	5	5
3	9	10	6	6	6
Total	20	23	17	17	17

Table 7. Result of the individual search assessment

Function	Average fitness score		
	1st Search	2nd Search	3rd Search
1	$\mathbf{1.7507\times10^{2}}$	3.3165×10^{2}	7.3780×10^{2}
2	**0.0000**	**0.0000**	**0.0000**
3	1.4562×10^{4}	$\mathbf{1.1847\times10^{4}}$	5.6749×10^{4}
4	1.3206×10^{1}	**7.8667**	3.3500×10^{1}
5	$\mathbf{4.8892\times10^{3}}$	1.4919×10^{4}	7.3647×10^{4}
6	$\mathbf{1.5484\times10^{2}}$	3.5648×10^{2}	6.3803×10^{2}
7	**0.1073**	0.1482	0.3469
8	-3.1101×10^{3}	$\mathbf{-3.2406\times10^{3}}$	-2.3645×10^{3}
9	$\mathbf{1.2831\times10^{2}}$	2.2271×10^{2}	1.8134×10^{2}
10	**3.3775**	4.3751	5.4326
11	**2.3422**	4.0022	6.4419
12	**2.2938**	3.0169×10^{1}	2.7584×10^{1}
13	1.1063×10^{1}	$\mathbf{1.0955\times10^{1}}$	1.4321×10^{4}
14	1.2043×10^{1}	$\mathbf{1.0381\times10^{1}}$	7.0548×10^{1}
15	0.0360	**0.0188**	0.0566
16	-0.9491	**-0.9769**	-0.7707
17	6.3957	**1.8074**	4.3450
18	4.9072×10^{1}	$\mathbf{2.2309\times10^{1}}$	4.2329×10^{1}
19	**-0.0495**	**-0.0495**	0.0000
20	-1.9025	**-2.1336**	-1.5071
21	-0.9193	**-1.8271**	-1.3047
22	-1.2594	**-1.7844**	-1.3865
23	-1.7217	**-1.8431**	-1.6642

4.2. Discussion

In general, the superiority of SFCA proves this work's contribution to improving the performance of existing metaheuristics. This superiority also shows that SFCA has good exploitation and exploration capabilities. The exploitation capability, commonly known as finding the better solution near the current solution, can be seen in the superiority of SFCA in solving the unimodal functions. Meanwhile, the exploration capability, commonly known as diversifying the possibility of search within space, can be seen in the superiority of SFCA in solving the multimodal functions with multiple optimal solutions but only one global optimal one. Moreover, by comparing the results in Table 6, it is shown that the exploitation capability of SFCA is as good as its exploration capability.

The result of the comparative assessment indicates that the stringent acceptance procedure is essential in the current development of metaheuristics. In general, the performance of SMA and GSO is worse than SFCA, OOA, COA, and WaOA. Moreover, the performance gap between these two metaheuristics is far worse than the four ones, especially in solving the high-dimension functions. Meanwhile, the competitiveness of OOA, COA, and WaOA relative to SFCA in the third cluster of functions indicates the contribution of the local search with narrowing space during the iteration. Among all metaheuristics used in this assessment, these three metaheuristics deploy the local search with narrowing space in a dedicated manner after the guided search.

The first and second assessments indicate the necessity of adopting multiple searches in multiple phases. SFCA, OOA, COA, and WaOA implement multiple searches in multiple phases. SMA implements multiple searches in a single phase using specific criteria to perform certain strategies. Meanwhile, GSO implements only one strategy: guided search without local search. Table 7 shows that although the walk toward the best swam member is superior to the other two walks, the performance of the whole package of SFCA, as shown in Table 3 to 5, is much better than the first walk when implemented individually.

By investigating the shape of the functions, such as in KMA, it is shown that the superiority of SFCA is found in all functions whether their global optimal solution is nearby or far from the central point of the space. This circumstance can be linked to all walks in SFCA. Through the first search, the swarm members tend to move toward the best swarm member. Through the second search, the movement of the swarm is pushed toward the center of the swarm. Meanwhile, regarding the uniformity during the initialization phase, the center of the swarm will be near the center of the space. The balance arithmetic crossover in the third phase also affects this circumstance. This walk makes the swarm member tend to get closer to the center of the space because the crossover takes place between the swarm member and the center of the space. This strategy poses the strength of SFCA in solving optimization problems if the global optimal solution is near the center of the space. Besides, SFCA is also competitive in solving problems where the global optimal solution is far from the center of the space, especially if it is compared with metaheuristics that deploy a single search without implementing the stringent acceptance approach.

The result of the individual search assessment exposes that both motion toward the best swarm member and the motion relative to the center of the swarm are complementary to each other. The motion toward the best swarm member provides superior exploration and exploitation capabilities as the first search contributes mainly to solving high dimension functions, whether they are unimodal or multimodal. On the other hand, the motion relative to the center of the swarm contributes mainly to balancing the exploitation and exploration capabilities as this search contributes dominantly to solving the fixed dimension multimodal functions.

The result of the individual search assessment also exposes the less contribution of the flipping and crossover methods than the directed search. The performance of the third search is behind the first and second searches. This result strengthens the circumstance that the directed search becomes more powerful than the crossover-based search as more swarm-based metaheuristics is introduced in the recent year rather than the crossover-based search. This circumstance can be used as motivation for further development of the crossover-based search to make it more competitive, especially when it competes with swarm intelligence.

The complexity of SFCA can be investigated based on the existence of the loop. In the initialization phase, only a nested loop consists of two loops. The swarm size controls the outer loop, while the dimension controls the inner loop. This circumstance makes the complexity of the initialization phase can be presented as $O(n(S).d)$. On the other hand, the nested loop in the iteration phase is different. There is a three-layer loop. The outer layer is controlled by maximum iteration. The swarm size controls the middle layer. Meanwhile, the inner loop consists of two sequential loops. The first loop is a loop for the whole swarm to find the center of the swarm with a loop controlled by the dimension in it. On the other hand, the second loop consists of a loop for the whole dimension but is performed three times as SFCA implements three walk strategies. Based on this explanation, the complexity in the iteration phase can be presented as $O(tm.n(S).(d(n(S)+3)))$.

5. CONCLUSION

A new metaheuristic called as swarm flip-crossover algorithm is presented in this paper (SFCA). The fundamental concept of SFCA is utilizing swarm intelligence as the core model with a balance arithmetic crossover strategy as complementary. This formalization of the model through pseudocode, and mathematical formulation has been presented as an interpretation of the fundamental concept. Both comparison and individual search assessments have been performed where SFCA is proven superior to its five confronters. SFCA performs consecutively better in 20, 23, 17, 17, 17 functions than SMA, GSO, OOA, COA, and WaOA. Meanwhile, based on the individual search assessment result, the walk toward and away from the center of the swarm performs superiorly. Then it is followed by the walk toward the best swarm member and the combined flip and crossover search. This result indicates that swarm intelligence is generally more potent than the evolution-based metaheuristic, where the crossover becomes its backbone.

There are several pathways for future studies. First, more assessment, especially with practical optimization use cases, is needed to evaluate the performance of SFCA more comprehensively. Second, the inferior performance of the evolution-based technique can be used as motivation for improvement because, until now, evolution-based metaheuristics are still widely utilized in various optimization studies. Third, inventing new references for the guided search is challenging, primarily to compete with the best swarm member.

ACKNOWLEDGEMENTS

The authors receive the financial support for publication of this paper by Telkom University, Indonesia.

REFERENCES

[1] R. Mothkur and V. B Nagendrappa, "An optimal model for classification of lung cancer using grey wolf optimizer and deep hybrid learning," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 1, pp. 406–413, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp406-413.

[2] Y. Wang, Q. Luo, and Y. Zhou, "Improved grey wolf optimizer for multiple unmanned aerial vehicles task allocation," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 1, pp. 577–585, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp577-585.

[3] H. T. Ibrahim, W. J. Mazher, and E. M. Jassim, "Modified Harris Hawks optimizer for feature selection and support vector machine kernels," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 2, pp. 942–953, Feb. 2023, doi: 10.11591/ijeecs.v29.i2.pp942-953.

[4] N. A. Windarko *et al.*, "Hybrid photovoltaic maximum power point tracking of Seagull optimizer and modified perturb and observe for complex partial shading," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 4571–4585, Oct. 2022, doi: 10.11591/ijece.v12i5.pp4571-4585.

[5] M. H. Ali, A. M. A. Soliman, and S. K. Elsayed, "Optimal power flow using Archimedes optimizer algorithm," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 13, no. 3, pp. 1390–1405, Sep. 2022, doi: 10.11591/ijpeds.v13.i3.pp1390-1405.

[6] P. Trojovský and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior," *Scientific Reports*, vol. 13, no. 1, May 2023, doi: 10.1038/s41598-023-35863-5.

[7] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, Jan. 2023, doi: 10.1016/j.knosys.2022.110011.

[8] M. Dehghani and P. Trojovský, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *Frontiers in Mechanical Engineering*, vol. 8, Jan. 2023, doi: 10.3389/fmech.2022.1126450.

[9] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.

[10] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo mlipir algorithm," *Applied Soft Computing*, vol. 114, Jan. 2022, doi: 10.1016/j.asoc.2021.108043.

[11] 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, doi: 10.1109/ACCESS.2022.3172789.

[12] P. Trojovský and M. Dehghani, "Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, Jan. 2022, doi: 10.3390/s22030855.

[13] N. Chopra and M. Mohsin Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Expert Systems with Applications*, vol. 198, Jul. 2022, doi: 10.1016/j.eswa.2022.116924.

[14] E. Trojovska and M. Dehghani, "Clouded Leopard optimization: A new nature-inspired optimization algorithm," *IEEE Access*, vol. 10, pp. 102876–102906, 2022, doi: 10.1109/ACCESS.2022.3208700.

[15] 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, doi: 10.1109/ACCESS.2022.3229964.

[16] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[17] 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, doi: 10.1109/ACCESS.2022.3151641.

[18] 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, doi: 10.1109/ACCESS.2021.3133286.

[19] F. Zeidabadi, M. Dehghani, and O. Malik, "TIMBO: Three influential members based optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 121–128, Oct. 2021, doi: 10.22266/ijies2021.1031.12.
[20] M. Dehghani *et al.*, "MLO: Multi leader optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 364–373, Dec. 2020, doi: 10.22266/ijies2020.1231.32.
[21] F. Zeidabadi, S. Doumari, M. Dehghani, and O. Malik, "MLBO: Mixed leader based optimizer for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 472–479, Aug. 2021, doi: 10.22266/ijies2021.0831.41.
[22] M. Dehghani and P. Trojovský, "Hybrid leader based optimization: a new stochastic optimization algorithm for solving optimization applications," *Scientific Reports*, vol. 12, no. 1, Apr. 2022, doi: 10.1038/s41598-022-09514-0.
[23] 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, Feb. 2023, doi: 10.22266/ijies2023.0228.20.
[24] 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, doi: 10.1109/ACCESS.2022.3162853.
[25] P. D. Kusuma and A. Dinimaharawati, "Three on three optimizer: a new metaheuristic with three guided searches and three random searches," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023, doi: 10.14569/IJACSA.2023.0140145.
[26] P. D. Kusuma and A. Novianty, "Multiple interaction optimizer: A novel metaheuristic and its application to solve order allocation problem," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 2, pp. 440–453, Feb. 2023, doi: 10.22266/ijies2023.0430.35.
[27] 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, Jun. 2023, doi: 10.22266/ijies2023.0630.19.
[28] M. Dehghani, Š. Hubálovský, and P. Trojovský, "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, Mar. 2022, doi: 10.7717/peerj-cs.910.
[29] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.
[30] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "COVIDOA: a novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle," *Neural Computing and Applications*, vol. 34, no. 24, pp. 22465–22492, Dec. 2022, doi: 10.1007/s00521-022-07639-x.
[31] M. I. A. Latiffi, M. R. Yaakub, and I. S. Ahmad, "Flower pollination algorithm for feature selection in tweets sentiment analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, doi: 10.14569/IJACSA.2022.0130551.
[32] F. M. Santa, F. H. M. Sarmiento, and H. M. Ariza, "Genetic algorithms applied to the searching of the optimal path in image-based robotic navigation environments," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, doi: 10.14569/IJACSA.2022.0130592.
[33] B. Wang, J. Wei, B. Lv, and Y. Song, "An improved genetic algorithm for the multi-temperature food distribution with multi-station," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130604.
[34] N. Wang, "Research on the optimization problem of agricultural product logistics based on genetic algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, 2022, doi: 10.14569/IJACSA.2022.0131295.

BIOGRAPHIES OF AUTHORS

Purba Daru Kusuma received his bachelor, and master's degrees in electrical engineering from Bandung Institute of Technology, Indonesia, in 2002 and 2009, respectively. He received his doctoral degree in computer science from Gadjah Mada University, Indonesia, in 2017. Currently, he is an assistant professor at Telkom University, Indonesia. His research area is artificial intelligence, machine learning, and operational research. He can be contacted by email: purbodaru@telkomuniversity.ac.id.

Faisal Candrasyah Hasibuan He is a computer engineering lecturer at Telkom University, Bandung, West Java, Indonesia. As an alma mater, he completed a bachelor's program where he works now. He was granted a master's degree in electrical engineering with a Computer Engineering specialization at Bandung Institute of Technology (ITB). The field of research interests are but are not limited to the embedded system, internet of things, and intelligent control systems. He can be contacted by email: faicanhasfcb@telkomuniversity.ac.id.