International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Multi-Swarm Approach for Numeric Optimization

Ahmet Babalık*[1]

Accepted : 09/08/2018 Published: 28/09/2018

Abstract: In order to solve the numeric optimization problems, swarm-based meta-heuristic algorithms can be used as an alternative to solve optimization problems. Meta-heuristic algorithms do not guarantee finding the optimal solution but they produce acceptable solutions in a reasonable computation time. By depending on the nature of the problems and the structure of the meta-heuristic algorithms, different results are obtained by different algorithms, and none of the meta-heuristic algorithm could guarantee to find the optimal solution. Particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms are well known meta-heuristic algorithms often used for solving numeric optimization problems. In this study, a novel multi-swarm approach based on PSO and ABC algorithms is suggested. The proposed multi-swarm approach includes PSO and ABC algorithms together and replacing the swarm which achieves better solutions than the other algorithm in a pre-defined migration period. By this migration, swarm always include better solutions concerned to the algorithm which achieves better results. While running PSO and ABC algorithms competitively, this migration ensures to utilize better solutions of both the solutions of PSO or ABC algorithms, and the convergence characteristic of each algorithm provides different approximation to the solution space. Thus, it is expected to obtain successful solutions and increasing the success rate at each migration cycle. The suggested approach has been tested on 14 well-known benchmark functions, and the results of the study are compared with the results in literature. The experimental results and comparisons show that the proposed approach is better than the other algorithms.

Keywords: *Particle swarm optimization, artificial bee colony, multi-swarm, optimization, meta-heuristic*

1. Introduction

Numeric optimization is one of the most interested and investigated topic among the researchers in this field. Due to the recent trends and investigations in meta-heuristic or hyper-heuristic algorithms on optimization concept, there will be always a demand to adapt and investigate the area for obtaining better results even in accuracy and computational cost.

Considering the literature, there are several meta-heuristic algorithms exist which are used for numerical optimization problems. In the meta-heuristic algorithms concept, swarm intelligence is the design of algorithms that mimic the collective social insect and animal behavior [1]. These insects and animals include bees, termites, ants, wasps, fish, birds etc. Other systems such as the immune system can also be included since it contains a swarm of cells and molecules. Among the two fundamental concepts of the swarm intelligence, the first fundamental concept include self-organization that relies on positive feedback, negative feedback, fluctuations and multiple interactions. The second fundamental concept is the division of labor within the colony or swarm. These concepts are necessary for problem solving systems that self-organize and adapt to the given environment [2].

Swarm intelligence algorithms starts with a set of candidate solutions for searching solution space, and in each iteration cycle, the candidate solution set is updated according to the historical and other related information in order to obtain an acceptable solutions in solution space. Some of the most commonly used swarm intelligence algorithms are ant colony algorithm (ACO) [3], particle swarm optimization (PSO) algorithm [4], cuckoo search (CS) algorithm [5] and artificial bee colony (ABC) algorithm [6]. PSO algorithm is a well-known state-of-the-art method used for solving numeric optimization problems [4]. Including the different versions of PSO algorithm, it is the most commonly used meta-heuristic algorithm due to its easy implementation, few parameters and powerful exploration characteristics. On the other hand, ABC algorithm is one of the other popular and powerful meta-heuristic algorithm which is often used for comparison with the PSO algorithm [6]. Since every position of the solution is updated in every iteration cycle, the PSO algorithm is more effective than ABC algorithm in terms of convergence characteristics. Although PSO has a fast convergence rate, it can be easily trapped to local minima in multimodal problems [7, 8].

Multi-swarm algorithms are effective methods maintaining diversity of the swarm. According to the algorithmic structure, the information is exchanged between the sub-swarms or different swarms and the whole structure of the algorithm is generated on the cooperation aim. Many PSO or ABC based multi-swarm algorithms are presented in literature. Some of these studies include species based PSO [9], cooperative PSO [10], multi-swarm self-adaptive and cooperative PSO [11], multi-layer PSO [12], multi-swarm cooperative PSO [13] and chaotic multi-swarm PSO [14] based on PSO algorithm and migratory multi-swarm ABC [15], adaptive multi-population differential ABC [16], parallelized multiple swarm ABC [17] and multi-swarm ABC [18] based on ABC algorithm. These studies differ from each other by their cooperation mechanism.

Considering the updating mechanisms of the PSO and ABC algorithms, this study investigates a novel multi-swarm optimization algorithm for solving numeric optimization problems. Hereby the suggested approach tries to obtain the candidate solutions by using PSO and ABC algorithms separately in a defined number of iteration cycle, then the swarm which has better best solution is exchanged with the other swarm. This

[1] *Dept. of Computer Eng. Konya Technical University, Konya, TURKEY*
* *Corresponding Author: Email: ababalik@selcuk.edu.tr*

transfer process is repeated in a pre-defined number of times or until the stopping criterion is reached.

The rest of the paper is organized as follows; after presenting brief introduction to PSO and ABC algorithms in section 2.1 and 2.2 respectively, the suggested approach is introduced in section 2.3. The experimental results and discussion is given in section 3, and the study is concluded in section 4.

2. Materials and Methods

2.1. Particle Swarm Optimization

The PSO algorithm, inspired by the behavior of a school of flying birds, is a population-based stochastic algorithm which is based on social and psychological principles [4]. The PSO algorithm works by having a population namely "swarm" that contains candidate solutions namely "particles". The individuals in a particle swarm optimizer are "evolved" by cooperation and competition among the individuals themselves through generations. Each particle adjusts its flying according to its own flying experience and its companions' flying experience [19].

The particle swarm represents the interactions of a number of individuals that do not know what the goal is. The individuals know their immediate state and their best performance in the past. As individuals emulate the successes of their neighbors, the population begins to cluster in optimal regions of a search space. This leads to discover good solutions for difficult problems such as the nonlinearity, local optima, deceptive gradients and high dimension problems [4].

The PSO algorithm starts solving the problem with a randomly generated population in the search space. The population matrix consisting of n particles can be given as;

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nD} \end{bmatrix}_{nxD} \tag{1}$$

where n is number of particles and D is dimensionality of the problem. Each solution in the population is named as particle and can be given as;

$$X_i = [x_{i1}, x_{i2} \dots x_{iD}] \tag{2}$$

where i represents the i^{th} particle.

All particles have a fitness value and speed information. The fitness value is calculated according to the objective function. The best value of each particle obtained so far is called p^{best} and the best value of the population is called g^{best}. Initially, each particle is considered as p^{best}, and considering the minimization problem the p^{best} values of each particle are updated by using the following equation;

$$p_i^{best}(t+1) = \begin{cases} p_i^{best}(t) & f(X_i(t+1) \geq f(p_i^{best}(t)) \\ X_i(t+1) & f(X_i(t+1) < f(p_i^{best}(t)) \end{cases} \tag{3}$$

where $p_i^{best}(t+1)$ is the best value of the i^{th} particle obtained so far at time $t+1$ and f stands for the fitness function. After obtaining best value of each particle, the best value of the swarm g^{best} is also obtained using the following equation;

$$g^{best} = argmin\{f(p_i^{best})\} \quad i = 1,2,\dots,n \tag{4}$$

Each particle in the population is iteratively updated with respect to the previous value of the particle, and this update process is evaluated by using the following equation;

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{5}$$

where $V_i(t+1)$ represents the velocity vector of i^{th} particle and iteratively updated by using the following equation;

$$V_i(t+1) = \omega V_i(t) + c1R_1\left(p_i^{best} - X_i(\mathrm{t})\right) + c2R_2(g^{best} - X_i(\mathrm{t})) \tag{6}$$

where ω stands for the inertial weight, R_1 and R_2 are random numbers uniformly distributed within the range [0,1] and $c1$ and $c2$ are cognitive and social weighting factors of the algorithm respectively. Note that the velocity vector of each particle are generated randomly at the initialization phase of the algorithm. Inertial weight ω is a positive value providing global and local search stability suggested by Shi and Eberhart [19]. It is used to balance the local and global search characteristics of the PSO algorithm.

2.2. Artificial Bee Colony

The ABC algorithm is a swarm intelligence algorithm, inspired by the behavior of honey bee colonies, used for solving numeric optimization problems. The algorithm represents a collective intelligence of honey bee colonies that consist of three essential components: food sources, employed foragers and unemployed foragers, and has three phases that involve the employed bee, onlooker bee and scout bee.

The ABC algorithm starts with initialization of the food source positions which stands for the solutions similar to the particles in PSO algorithm. According to the search space of the optimization problem, a food source position can be formulized as follows;

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \tag{7}$$

where $i = \{1,2,\dots,n\}$ and n denotes the number of food sources, j= $\{1,2,\dots,D\}$ and D denotes the dimension of the problem and x_j^{min} and x_j^{max} stands for the lower and upper bounds of the j^{th} dimension of the problem. After this initialization step, employed bee, onlooker bee and scout bee phases are repeated sequentially until the a pre-defined iteration cycle number or stopping criteria reached. In the initialization of the algorithm, a counter for each food source is defined and these counters are used for occurrence of the scout bees.

2.2.1. Employed Bee Phase

Each employed bee is related with only one food source, and thus, the number of food sources is equal to the number of the employed bees. Each employed bee tries to find a new food source and access the quality of it. If the new food source which is also the candidate solution has better quality than the current one, the current food source is changed to the candidate one and the counter is reset; otherwise the counter of the food source is increased by 1. A candidate food source is obtained by using the following equation;

$$\upsilon_{\mathrm{ij}} = x_{\mathrm{ij}} + \phi_{\mathrm{ij}}(x_{\mathrm{ij}} - x_{\mathrm{kj}}) \tag{8}$$

where υ_{i} represents the candidate food source neighbour to x_{i}, υ_{ij} stands for the j^{th} dimension of the υ_{i}, j is the randomly chosen dimension for modification, k is index of the randomly chosen neighbour food source and ϕ_{ij} is a random number within the range [-1,1].

Fitness value of each solution is calculated by using the following equation;

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & f_i(t) \geq 0 \\ 1+|f_i| & \text{otherwise} \end{cases} \quad (9)$$

where fit_i is the fitness value of the i^{th} solution and f_i is the objective function.

2.2.2. Onlooker Bee Phase

Once all employed bees have completed their research in an iteration cycle, they transmit information about the amount of nectar (fitness value) of the food sources they have found to the onlooker bees via waggle dance.

Each onlooker bee selects a food source depending on the probability of their fitness values through roulette-wheel selection scheme and try to improve the solution. Hereby, food sources with better fitness values have a higher probability to be selected than others. Roulette wheel selection process is implemented by using the probability calculation given in (10).

$$P_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i} \quad (10)$$

where P_i and fit_i are the probability and fitness value of the i^{th} food source being selected respectively.

After calculating the probability value of each food source, a random number is generated within the range [0,1] concerning the roulette-wheel selection scheme. If P_i is greater than the generated number the onlooker bee produces a candidate food source position by using (8) such as employed bees, and after calculating the quality of the candidate solution, greedy selection is applied.

While updating the food sources, if the food source is changed with the candidate one, the counter which is also used in employed bee phase is set to 0, otherwise increased by 1.

2.2.3. Scout Bee Phase

At the end of each iteration cycle, the counter of each food source is checked whether they exceeded a pre-defined number called as "limit". The food sources which its counters has highest content is abandoned, a new food source is created using (7), this food source is assigned to an employed bee, and the counters of this food source are set to 0.

2.3. Suggested Multi-Swarm Approach

Evolutionary and swarm-based meta-heuristic algorithms produce successful results in solving optimization problems. In both evolutionary and swarm-based algorithms, candidate solutions are randomly generated initially, updated according to the strategies of the algorithms and evaluated the solution space. The use of multi-population or multi-swarm concepts is a method that increases the problem solving capabilities of nature-inspired algorithms. Multi population technique can easily be adapted to many nature-inspired algorithms. Basically in this technique, main population (or swarm) is divided into sub-populations (sub-swarms) and these sub-swarms evaluate the solution space. Also, the sub-populations communicate and interact with the each other. PSO and ABC are well-known swarm based algorithms in the literature. Therefore, we used these two algorithms in the proposed multi-swarm approach.

According to the suggested multi-swarm approach, the population is divided into two equal sub-populations and evaluated together within a pre-defined migration period. The first half of the population is evaluated by using the PSO algorithm, and the second half of the population is evaluated by using the ABC algorithm. The number of the iteration cycle of each period is calculated by dividing the total iteration cycle by the migration number as given in (11).

$$sc = \frac{MaxIT}{mp} \quad (11)$$

where sc stands for the number of the iteration cycles which is going to be run in a migration period, $MaxIT$ stands for the maximum iteration number and mp denotes the migration period.

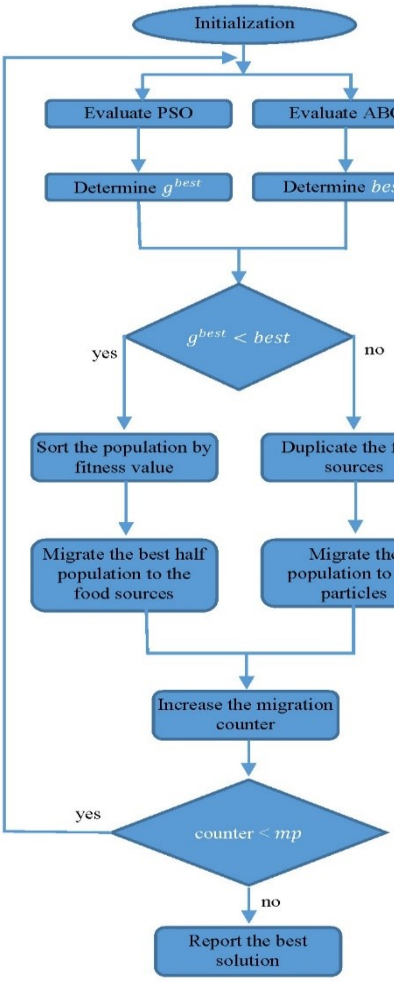
Initialization
Evaluate PSO
Evaluate ABC
Determine g^{best}
Determine $best$
$g^{best} < best$
yes
no
Sort the population by fitness value
Duplicate the food sources
Migrate the best half population to the food sources
Migrate the population to the particles
Increase the migration counter
counter < mp
yes
no
Report the best solution

Fig. 1. Flowchart of the suggested multi-swarm approach

For a minimization problem, after PSO and ABC algorithms evaluated within the sc cycle, the best solution obtained by PSO algorithm is compared with the best solution obtained by ABC algorithm. If the smaller solution value is obtained by the PSO algorithm, the population of the PSO algorithm is migrated to the population of the ABC algorithm. Hereby, the ABC algorithm is going to solve the problem by using the population of the PSO algorithm until the next sc cycle. Otherwise, the population of the ABC algorithm is migrated to the PSO algorithm.

As it is known, particles (solutions) of the PSO algorithm correspond to the food sources (solutions) of the ABC algorithm.

Even the particle size and the number of bees are equal, the number of food sources are half of the number of the particles. Because of the PSO and ABC algorithms characteristics, there is a ratio of 1/2 in population sizes preserving function evaluation times equal. Thus, the migration procedure includes migrating the first half of the solutions ordered by the fitness from PSO algorithm's particles to the ABC algorithm's food sources; migrating duplicate of the solutions from ABC algorithm's food sources to the PSO algorithm's particles.

According to the suggested approach, these evaluation and migration procedures are repeated migration period times. The flowchart of the suggested approach is given in Fig. 1.

Table 1. Benchmark Functions

F.ID	***Function***	***Search Range***	***C***	***Formulation***
1	Sphere	[−100, 100]	U	$f_1 = \sum_{i=1}^{n} x_i^2$
2	Schwefel2.22	[−10, 10]	U	$f_2 = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$
3	Rosenbrock	[−10, 10]	U	$f_3 = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
4	Noise	[−1.28, 1.28]	U	$f_4 = \sum_{i=1}^{n} ix_i^4 + \text{random}[0,1)$
5	Schwefel2.26	[−500, 500]	M	$f_5 = 418.98288727243369 * n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i\rvert})$
6	Rastrigin	[−5.12, 5.12]	M	$f_6 = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$
7	Ackley	[−32, 32]	M	$f_7 = -20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right\} - \exp\left\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right\} + 20 + e$
8	Griewank	[−600, 600]	M	$f_8 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
9	Penalized1	[−50, 50]	M	$f_9 = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{1}{4}(x_i + 1) u_{x_i,a,k,m} = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(x_i - a)^m & x_i < -a \end{cases}$
10	Penalized2	[−50, 50]	M	$f_{10} = \frac{1}{10}\{\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\,[1 + \sin^2(3\pi x_{i+1})]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_{i+1})]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$
11	Rotated Schwefel	[−500, 500]	R	$f_{11} = 418.9828 * \text{n} - \sum_{i=1}^{n} z_i\,,$ where $z_i = \begin{cases} y_i \sin(\sqrt{\lvert y_i\rvert}), \text{if} \lvert y_i\rvert \le 500 \\ 0, \text{otherwise} \end{cases}, y_i = y_i' + 420.96,$ where $y' = M * (x - 420.96), M$ is an orthogonal matrix
12	Rotated Rastrigin	[−5.12, 5.12]	R	$f_{12} = \sum_{i=1}^{n}[y_i^2 - 10\cos(2\pi y_i) + 10]$ where $y = M * x, M$ is an orthogonal matrix
13	Rotated Ackley	[−32, 32]	R	$f_{13} = -20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2}\right\} - \exp\left\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi y_i)\right\} + 20 + e$ where $y = M * x, M$ is an orthogonal matrix
14	Rotated Griewank	[−600, 600]	R	$f_{14} = \frac{1}{4000}\sum_{i=1}^{n} y_i^2 - \prod_{i=1}^{n}\cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$ where $y = M * x, M$ is an orthogonal matrix

3. Results and Discussion

In order to evaluate the suggested multi-swarm approach 14 well-known benchmark test functions are employed in this study. These functions and their characteristic properties are given in Table 1 [20]. According to their characteristic properties, these functions can be divided into 3 class including unimodal, multimodal and rotated functions, and there are 4 unimodal, 6 multimodal and 4 rotated functions exist in the benchmark test functions. Note that the objective of all of these functions are minimization, and the experiments are operated for 30 dimensional functions.

The suggested multi-swarm approach is evaluated by using the parameters given in Table 2.

Table 2. Parameter Settings

Parameter	PSO	ABC	PSO-ABC
PopSize	80	80	40+40
Dim	30	30	30
MaxIT	2,500	2,500	2,500
FEs	200,000	200,000	200,000
ω	0.7213	-	0.7213
c1, c2	1.1931	-	1.1931
mp	-	-	5

PopSize stands for the population size, *Dim* stands for the dimension of the problem, *MaxIT* stands for the maximum iteration number, *FEs* stands for function evaluations, ω stands for the inertial weight, *c1* and *c2* stands for the cognitive and social parameters of PSO algorithm respectively and *mp* stands for the migration period.

Table 3. Comparison of the results of the suggested approach with the results of the PSO and ABC algorithms

F.ID	***PSO***				***ABC***				***Suggested Approach***	
	Mean	***Std. Dev.***	***p***	***S***	***Mean***	***Std. Dev.***	***p***	***S***	***Mean***	***Std. Dev.***
1	**6.86E-48**	1.91E-47	1.42E-09	+	2.56E-16	9.45E-17	1.42E-09	+	1.88E-45	2.98E-45
2	**3.78E-26**	2.51E-26	8.10E-06	+	1.11E-15	1.82E-16	1.42E-09	+	7.07E-25	2.02E-24
3	2.02E+01	1.44E+00	4.78E-04	+	1.17E+01	1.32E+01	4.85E-01	-	**8.77E+00**	8.89E+00
4	**4.67E-03**	1.76E-03	5.55E-08	+	1.44E-02	4.15E-03	1.65E-05	+	9.44E-03	2.85E-03
5	2.85E+03	6.64E+02	1.37E-09	+	**7.81E-06**	3.25E-05	5.04E-06	+	1.90E+01	5.60E+01
6	2.99E+01	1.19E+01	9.68E-11	+	1.69E-11	4.39E-11	3.66E-10	+	**0.00E+00**	0.00E+00
7	**6.25E-15**	1.55E-15	4.93E-01	-	4.45E-13	2.12E-13	3.11E-10	+	6.54E-15	1.33E-15
8	2.27E-03	4.97E-03	2.08E-02	+	2.15E-11	3.98E-11	9.10E-11	+	**0.00E+00**	0.00E+00
9	**1.57E-32**	5.59E-48	NaN	-	2.26E-16	1.04E-16	9.73E-11	+	**1.57E-32**	5.59E-48
10	**1.35E-32**	5.59E-48	NaN	-	2.43E-16	9.78E-17	9.73E-11	+	**1.35E-32**	5.59E-48
11	4.86E+03	6.68E+02	1.89E-02	+	5.59E+03	5.39E+02	2.21E-07	+	**4.50E+03**	5.03E+02
12	3.41E+01	1.06E+01	5.61E-01	-	1.47E+02	3.46E+01	1.42E-09	+	**3.24E+01**	6.54E+00
13	**6.25E-15**	1.55E-15	4.54E-02	+	2.02E-06	2.86E-06	1.38E-10	+	6.96E-15	7.11E-16
14	2.07E-03	5.01E-03	2.27E-01	-	**1.36E-06**	2.83E-06	2.92E-06	+	1.38E-03	4.33E-03

F.ID stands for the function number, *Mean* stands for the average results of the 25 run, *Std. Dev.* stands for the standard deviation of the 25 run, *p* stands for the p value of Wilcoxon Test and *S* stands for the sign value of Wilcoxon Test.

As the suggested approach depends on both PSO and ABC algorithms, the results are compared with the results of the PSO and ABC algorithms. In order to compare results of the suggested approach with results of PSO and ABC algorithms, the function evaluations (FEs) are kept equal, all algorithms run 25 times, and the mean of the 25 runs and standard deviations are given in Table 3.

The better results are shown in bold in Table 3. In order to specify whether there is a significant difference exist among the algorithms, Wilcoxon test is utilized at 0.05 confidence interval, and the *p* values and *signs* are also given in Table 3. Hereby, a (+) sign indicates that the results are statistically different from each other ($p<0.05$) and a (-) sign indicates that the results are not different from each other ($p \geq 0.05$). The convergence graphs of each function is also given in Fig. 2, Fig. 3 and Fig. 4.

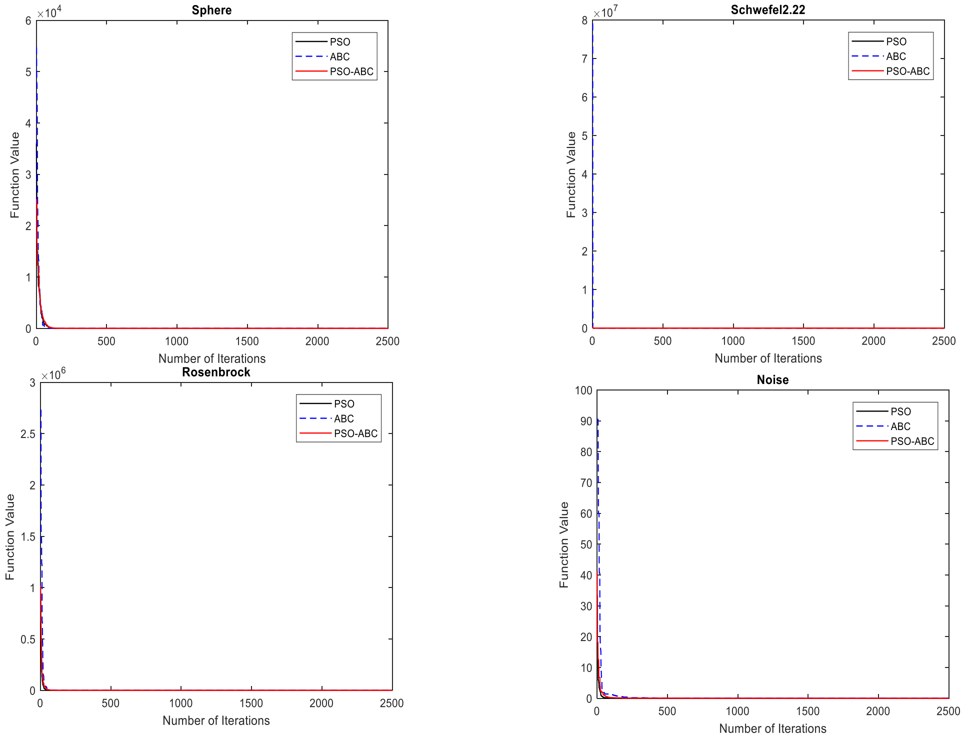

Fig.2 Convergence graphs of the unimodal functions

As seen from Table 3, the suggested multi-swarm approach achieves better results than the results of both PSO and ABC algorithms in benchmark functions f3, f6, f8, f11 and f12. In benchmark functions 9 and 10, the suggested approach achieves better results than the results of the ABC algorithm, and also these results are equal with the results of the PSO algorithm. According to the Wilcoxon test, it can be seen that the suggested approach is mostly differs from both PSO and ABC algorithms.

Due to the fact that PSO and the proposed approach present similar performance on the test set dealt with the study, a new comparison is considered and the results of the suggested multi-swarm approach is also compared with the results directly taken from [20], and the comparison is given in Table 4. In order to compare the results fairly, the FEs's are taken as equal in both this study and [20]. According to Table 4, the suggested multi-swarm approach achieves better results than the results of the other algorithms in benchmark functions f3, f6-f10 and f13. The average rank shows that the suggested approach is generally better than the PSO variants algorithm in terms of rank.

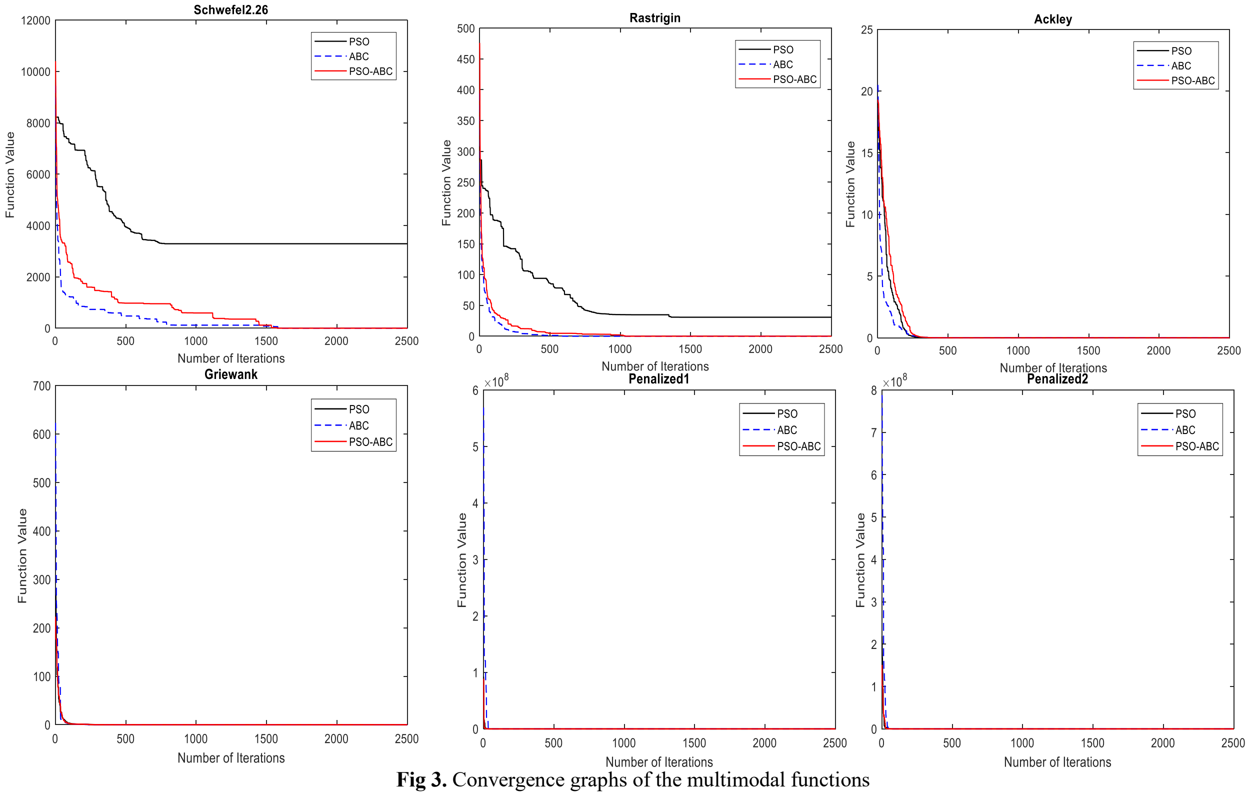

Fig 3. Convergence graphs of the multimodal functions

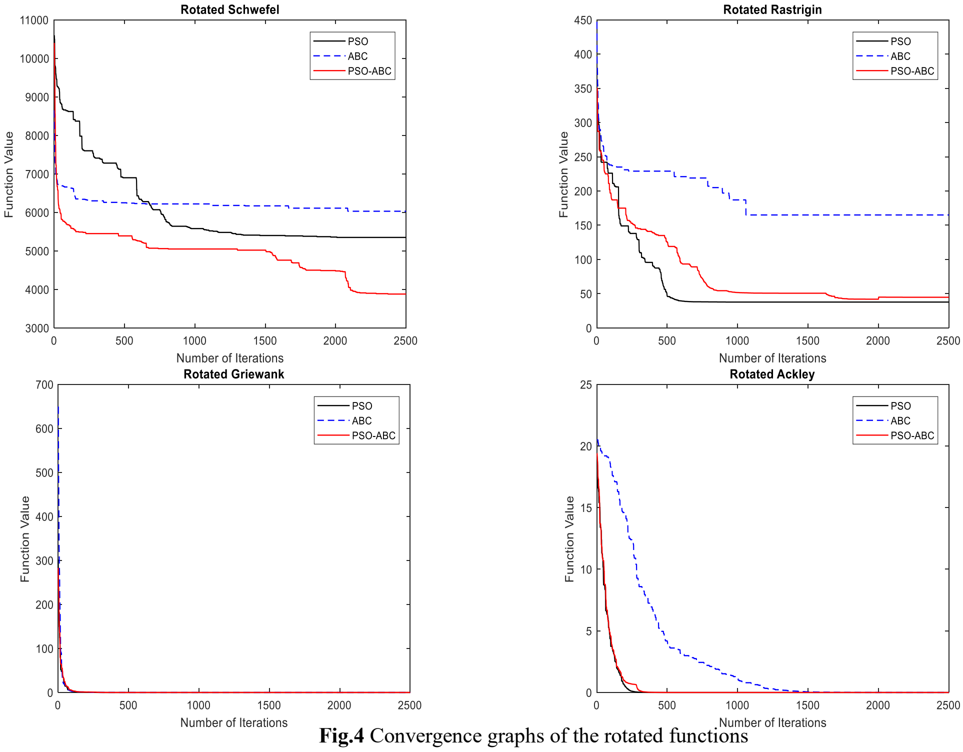

Fig.4 Convergence graphs of the rotated functions

Table 4. Comparison of the results of the suggested approach with the results in [20]

F.ID		*CLPSO*	*HPSO-TVAC*	*FIPSO*	*SPSO-40*	*LPSO*	*DMS-PSO*	*LFPSO*	*Suggested Approach*
1	Mean	1.58E-12	2.83E-33	2.42E-13	2.29E-96	3.34E-14	2.65E-31	4.69E-31	1.88E-45
	Std.Dev.	7.70E-13	3.19E-33	1.73E-13	9.48E-96	5.39E-14	6.25E-31	2.50E-30	2.98E-45
	Rank	8	3	7	1	6	4	5	2
2	Mean	2.51E-08	9.03E-20	2.76E-08	1.74E-53	1.70E-10	1.57E-18	2.64E-17	7.07E-25
	Std.Dev.	5.84E-09	9.58E-20	9.04E-09	1.58E-53	1.39E-10	3.79E-18	6.92E-17	2.02E-24
	Rank	7	3	8	1	6	4	5	2
3	Mean	1.14E+01	2.39E+01	2.51E+01	1.35E+01	2.81E+01	4.16E+01	2.38E+01	8.77E+00
	Std.Dev.	9.85E+00	2.65E+01	5.10E-01	1.46E+01	2.18E+01	3.03E+01	3.17E-01	8.89E+00
	Rank	2	5	6	3	7	8	4	1
4	Mean	5.85E-03	9.82E-02	4.24E-03	4.02E-03	2.28E-02	1.45E-02	2.41E-03	9.44E-03
	Std.Dev.	1.11E-03	3.26E-02	1.28E-03	1.66E-03	5.60E-03	5.05E-03	8.07E-04	2.85E-03
	Rank	4	8	3	2	7	6	1	5
5	Mean	3.82E-04	1.59E+03	9.93E+02	3.14E+03	3.16E+03	3.21E+03	1.37E+03	1.90E+01
	Std.Dev.	1.28E-07	3.26E+02	5.09E+02	7.81E+02	4.06E+02	6.51E+02	6.36E+02	5.60E+01
	Rank	1	5	3	6	7	8	4	2
6	Mean	9.09E-05	9.43E+00	6.51E+01	4.10E+01	3.51E+01	2.72E+01	4.54E+00	0.00E+00
	Std.Dev.	1.25E-04	3.48E+00	1.34E+01	1.11E+01	6.89E+00	6.02E+00	1.03E+01	0.00E+00
	Rank	2	4	8	7	6	5	3	1
7	Mean	3.66E-07	7.29E-14	2.33E-07	3.73E-02	8.20E-08	1.84E-14	1.68E-14	6.54E-15
	Std.Dev.	7.57E-08	3.00E-14	7.19E-08	1.90E-01	6.73E-08	4.35E-15	4.84E-15	1.33E-15
	Rank	7	4	6	8	5	3	2	1
8	Mean	9.02E-09	9.75E-03	9.01E-12	7.48E-03	1.53E-03	6.21E-03	8.14E-17	0.00E+00
	Std.Dev.	8.57E-09	8.33E-03	1.84E-11	1.25E-02	4.32E-03	8.14E-03	4.46E-16	0.00E+00
	Rank	4	8	3	7	5	6	2	1
9	Mean	6.45E-14	2.71E-29	1.96E-15	7.47E-02	8.10E-16	2.51E-30	4.67E-31	1.57E-32
	Std.Dev.	3.70E-14	1.88E-29	1.11E-15	3.11E+00	1.07E-15	1.02E-29	9.01E-31	5.59E-48
	Rank	7	4	6	8	5	3	2	1
10	Mean	1.25E-12	2.79E-28	2.70E-14	1.76E-03	3.26E-13	2.64E-03	1.51E-28	1.35E-32
	Std.Dev.	9.45E-13	2.18E-28	1.57E-14	4.11E-03	3.70E-13	4.79E-03	8.00E-28	5.59E-48
	Rank	6	3	4	7	5	8	2	1
11	Mean	4.39E+03	5.32E+03	4.41E+03	4.57E+03	4.50E+03	4.04E+03	5.51E+03	4.50E+03
	Std.Dev.	3.51E+02	7.00E+02	9.94E+02	6.28E+02	3.97E+02	5.68E+02	5.64E+02	5.03E+02
	Rank	2	7	3	6	4	1	8	5
12	Mean	8.71E+01	5.29E+01	1.50E+02	4.34E+01	5.34E+01	4.20E+01	1.79E+00	3.24E+01
	Std.Dev.	1.08E+01	1.25E+01	1.45E+01	1.74E+01	1.40E+01	9.74E+00	9.81E+00	6.54E+00
	Rank	7	5	8	4	6	3	1	2
13	Mean	5.91E-05	9.29E+00	3.16E-07	9.24E-02	1.55E+00	2.42E-14	1.65E-14	6.96E-15
	Std.Dev.	6.46E-05	2.07E+00	1.00E-07	3.20E-01	4.50E-01	1.52E-14	5.40E-15	7.11E-16
	Rank	5	8	4	6	7	3	2	1
14	Mean	7.96E-05	9.26E-03	1.28E-08	3.05E-03	1.68E-03	1.02E-02	1.48E-03	1.38E-03
	Std.Dev.	7.66E-05	8.80E-03	4.29E-08	5.70E-03	3.47E-03	1.24E-02	6.17E-03	4.33E-03
	Rank	2	7	1	6	5	8	4	3
	Avg.Rank	4.3	4.9	4.7	4.8	5.4	4.7	3.0	1.9

For each function first row indicates the mean result of the 25 run, the second row indicates the standard deviation of the 25 run, the third row indicates the rank of the algorithm and the row with title “Avg. Rank” indicates the mean rank of the algorithms.

4. Conclusion

The PSO and ABC algorithms are well-known swarm based numeric optimization algorithms. By depending on the nature of the problems and the structure of the meta-heuristic algorithms, different results are obtained by different algorithms, and none of the meta-heuristic algorithm could guarantee to find the optimal solution. A novel multi-swarm approach based on PSO and ABC algorithms is suggested in this study. The proposed approach has been tested on 14 well-known benchmark functions, and the results of the study are compared with the results in literature. It can be said that the suggested multi-swarm approach achieves successful results. Future works include utilizing different swarm based algorithms in the suggested approach. The effect of employing different number of sub-swarms are also be investigated in numeric optimization problems and real world engineering problems.

References

[1] E. Bonabeau, D. d. R. D. F. Marco, M. Dorigo, G. Théraulaz, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*: Oxford university press, 1999.

[2] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department2005.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 26, pp. 29-41, Feb 1996.

[4] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, ed: Springer, 2011, pp. 760-766.

[5] X. S. Yang and S. Deb, "Cuckoo Search via Levey Flights," *2009 World Congress on Nature & Biologically Inspired Computing (Nabic 2009),* pp. 210-214, 2009.

[6] D. Karaboğa, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Comput. Eng.Dep.2005.

[7] X. Xu, Y. G. Tang, J. P. Li, C. C. Hua, and X. P. Guan, "Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy," *Applied Soft Computing,* vol. 29, pp. 169-183, Apr 2015.

[8] V. R. Kulkarni and V. Desai, "ABC and PSO: A Comparative Analysis," *2016 Ieee International Conference on Computational Intelligence and Computing Research,* pp. 379-385, 2016.

[9] X. Li, "Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization," Berlin, Heidelberg, 2004, pp. 105-116.

[10] F. v. d. Bergh and A. P. Engelbrecht, "A Cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 8, pp. 225-239, 2004.

[11] J. Zhang and X. Ding, "A Multi-Swarm Self-Adaptive and Cooperative Particle Swarm Optimization," *Engineering Applications of Artificial Intelligence,* vol. 24, pp. 958-967, 2011/09/01/ 2011.

[12] L. Wang, B. Yang, and Y. Chen, "Improving particle swarm optimization using multi-layer searching strategy," *Information Sciences,* vol. 274, pp. 70-94, 2014/08/01/ 2014.

[13] B. Niu, Y. L. Zhu, X. X. He, and H. Wu, "MCPSO: A multi-swarm cooperative particle swarm optimizer," *Applied Mathematics and Computation,* vol. 185, pp. 1050-1062, Feb 15 2007.

[14] S. Mukhopadhyay and S. Banerjee, "Global optimization of an optical chaotic system by Chaotic Multi Swarm Particle Swarm Optimization," *Expert Systems with Applications,* vol. 39, pp. 917-924, Jan 2012.

[15] S. Biswas, S. Das, S. Debchoudhury, and S. Kundu, "Co-evolving bee colonies by forager migration: A multi-swarm based Artificial Bee Colony algorithm for global search space," *Applied Mathematics and Computation,* vol. 232, pp. 216-234, 2014/04/01/ 2014.

[16] J. J. Zhou, X. F. Yao, Y. Z. Lin, F. T. S. Chan, and Y. Li, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Information Sciences,* vol. 456, pp. 50-82, Aug 2018.

[17] M. Subotic and M. Tuba, "Parallelized Multiple Swarm Artificial Bee Colony Algorithm (MS-ABC) for Global Optimization," *Studies in Informatics and Control,* vol. 23, pp. 117-126, Mar 2014.

[18] D. Jia, S. Qu, and L. Li, "A Multi-swarm Artificial Bee Colony Algorithm for Dynamic Optimization Problems," in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, 2016, pp. 441-445.

[19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69-73.

[20] H. Hakli and H. Uguz, "A novel particle swarm optimization algorithm with Levy flight," *Applied Soft Computing,* vol. 23, pp. 333-345, Oct 2014.