Comparative Study of Krill Herd Algorithm and Flower Pollination Algorithm

Viktor Tuba, Romana Capor-Hrosik and Milan Tuba

***Abstract*—Numerous applications deal with the hard optimization problems. In recent years various solutions for this kind of problems were proposed. Swarm intelligence algorithms represent efficient metaheuristics for finding the optimal solution for hard optimization algorithms. In this paper two recent swarm intelligence algorithms, krill herd algorithm and flower pollination algorithm will be tested and compared. Both algorithms were tested on CEC 2013 benchmark functions. Flower pollination algorithm obtained better and more stable results for some test functions while krill herd algorithm made some larger mistakes for some test functions.**

***Index Terms*—Hard optimization problems, optimization algorithms, krill herd algorithm, flower pollination algorithm.**

I. Introduction

IN everyday life we often have a need to find best way to use certain resources (usually time or money) to achieve some goal. In computer science this kind of problems are called optimization problems. In goal of formalizing these problems we introduced cost function (sometimes known as fitness function) K where $A \subset R^N$ that can be represented by:

$$K : A \rightarrow R \tag{1}$$

The goal of optimization is to find value x* that satisfies condition:

$$K(x) \leq K(x*) \tag{2}$$

Solving these problems can be very difficult and there are entire class of optimization problems that do not have deterministic method of solving. These problems are called NP difficult. It is characteristic for there problems that it is easy to test the solution, but it is hard to find best solution. Only known way to solve them is to use stochastic algorithms.

Famous example of NP difficult problem is traveling salesman. This problem is often used in lectures about NP difficult problems. Salesman needs to take a tour of N cities in order that will cost him minimally. He must not visit any city twice. Problem is represented by graph where nodes are cities and edges between cities have weight that represent cost of traveling between those two cites. This problem can be solved by heuristic that has complexity of N! so when we have more than 20 cities, calculation time is unreasonably long.

V. Tuba, R. Capor-Hrosik and M. Tuba are with Graduate School of Computer Science, John Naisbitt University, Bulevar umetnosti 29, Belgrade, Serbia.

Email: viktortuba@gmail.com, romanacapor@yahoo.com, tuba@ieee.org

M. Tuba was supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

Stochastic algorithms use random factors in their execution and set of rules for the search of the solution inside the domain of possible solutions. In the contrast to deterministic algorithms, that can be repeated with same result, stochastic algorithms always get different solution, but if algorithm is well designed and run long enough, it will always produce solution that is ”good enough”, or in other words solution that is within tolerance margin from optimal value. Usual use for these algorithms is to run them many times and take average result as final solution.

Most stochastic algorithms are based on imitation of some natural phenomena that we observed it provides good result, but it is not always completely understood how. Based on what natural phenomena algorithm is emulating, they are classified into types. There is three big types of stochastic algorithms: evolutionary, artificial immune systems and swarm intelligence.

Evolution algorithms are using the evolutionary idea of the survival of the fittest. Population of solutions is created and next generation is made by combining best solutions from previous generation. These algorithms also use concept of mutation as random factor. After first generation is created, population goes trough numerous iterations of breeding, where each generation is closer to solution, since only the best solutions were allowed to move into next generation.

Artificial immune systems use negative selection, they search for bad solutions and eliminate them from population. As their name suggests, they are inspired by natural immune systems in living beings. Some of the algorithms in this class are negative selection procedure and the clonal selection procedure.

Swarm intelligence algorithms represent each possible solution as an organism in some sort of swarm that searches for best position. Movement of each organism is influenced by his own memory, global data from swarm and random factor, exploration. Some of the famous swarm algorithms are ant colony optimization, artificial bee colony algorithm, particle swarm optimization and others.

In this paper a comparison of krill herd algorithm and flower pollination algorithm was done. Both algorithms belong to swarm intelligence class of stochastic algorithms. We will test them against CEC 2013 benchmark functions. In all benchmark functions goal is to find solution with minimal value.

II. Swarm intelligence algorithms

One of big problems in optimization is preventing the algorithm to get stuck in sub domain of search space that is not optimal. This happens when the problem have many dimensions or when problem have many local extremes.

Unconstrained and constrained optimization problems have been solved by many different techniques and methods. As an alternative to the traditional methods of operations research, heuristics and metaheuristics methods have been devised.

There are two major types of metaheuristics: inspired by nature and not inspired by nature. In this paper we are dealing with metaheuristics that are inspired by nature called swarm intelligence. Swarm algorithms are using collaborative behavior of simple individuals and replicate natural system in solving the problem. All members of the swarm are synchronized and coordinated in effort to reach the objective without central command. Swarms of worms, ants, bees, birds and fish were the main source inspiration for these methods. Efficiency analysis of swarm intelligence is given in [21].

Particle swarm optimization (PSO) is one of the first swarm algorithms [10]. It mimics social behavior of fish or birds. Original, and upgraded versions of PSO have successfully been applied on many global optimization problems [13].

Ant colony optimization (ACO) emulates the foraging and social behavior of ants. ACO models ants property of disposing a substance called pheromone on their way from nest to the food source. This well-known metaheuristic have many implementations that can be found in the literature. ACO was successfully applied on minimum weight vertex cover problem [7], minimum connected dominating set problem [8], and many others.

Cuckoo search (CS) is another representative of swarm intelligence, firstly introduced by Yang and Deb [23]. CS simulate search process by utilizing Levy flights and proved to be robust optimization technique for global optimization [2], image processing [5], etc.

Artificial bee colony (ABC) was inspired by foraging behavior of honey bee swarm. In ABC approach exploitation and exploration processes are guided by three types of bees: employed, onlookers and scouts. This technique proved to be an effective and efficient approach in solving numerical optimization problems [9]. Many upgraded and enhanced versions of ABC were proposed [4]. Also, many implementations of parallelized ABC can be found in the literature [12].

Bat algorithm new nature-inspired approach developed by Yang in 2010 [20]. This algorithm based on the echolocation behaviour of micro bats with varying pulse rates of emission and loudness. It was used in various applications such as handwritten digit recognition [15], parameter tuning for support vector machine [14], multilevel image thresholding [1], etc. Besides basic swarm intelligence implementations, many hybrid approaches can be found in the literature. A hybrid of ABC and firefly algorithm (FA) was developed for solving portfolio optimization problem [17]. In [3], ABC's scout mechanism was incorporated into FA for enhancing the exploration process for problems with entropy constraint [16]. SOA algorithm for global optimization was successfully hybridized with ABC approach [19]. Also, hybrid between SOA and FA was applied on constrained benchmarks [18].

III. Krill Herd Algorithm

Krill herd algorithm (KH) was proposed in 2012 by Gandomi and Alavi in [6]. This algorithm was inspired by by the social behavior of Antarctic krills.

Since the herding of the krill individuals represents a multi-objective process, it is directed towards two main goals increasing krill density and reaching the food. The time-dependent position of an individual krill in 2D surface is governed by the three basic factors [6]:

$$\frac{dX_i}{d_t} = N_i + F_i + D_i, \tag{3}$$

where N_i is the motion induced by other krill individuals, F_i is foraging motion, and D_i is the physical diffusion of the i-th krill. Motion caused by other krill individuals is defined by the following equation:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old}, \tag{4}$$

where

$$\alpha_i = \alpha_i^{local} + \alpha_i^{target}, \tag{5}$$

where N^{max} is the maximum induce speed, w_n is the inertia weight of the motion in the range $[0,1]$, N_i^{old} is the last motion induced, α_i^{local} is the local effect provided by the neighbors α_i^{target} is the target direction provided by the best krill individual.

The sensing distance for each krill individual can be determined using different heuristic methods. Here, it is determined using the following formula for each iteration:

$$d_{s,i} = \frac{1}{5N}\sum_{j=1}^{N}||X_i - X_j|| \tag{6}$$

where $d_{s,i}$ is the sensing distance for the i-th krill in the population, and N is the number of the krill individuals. The factor 5 in the denominator is empirically calculated [6]. In 6, if the distance of two krill individuals is less than the defined sensing distance, they are neighbors.

The known target vector of each krill individual is the lowest fitness of an individual krill. The global optimum is defined as followed:

$$\alpha_i^{target} = C^{best} K_{i,best} X_{i,best} \tag{7}$$

where C^{best} is the effective coefficient of the krill individual with the best fitness of the i-th krill individual. The value of C^{best} can be defined as follows:

$$C^{best} = 2(rand + \frac{l}{l_{max}}) \tag{8}$$

where $rand$ is a random number between 0 and 1, l is the current iteration numbers, and l_{max} is the maximum number of iterations.

As mentioned above, the krill motion consists of foraging motion, motion influenced by other individuals, and the physical diffusion. The foraging motion formulation is based on two main effective parameters: the food location, and the previous experience about the food location. Foraging motion of the i-th krill individual is formulated as follows:

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{9}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{10}$$

where V_f is the foraging speed, w_f is the inertia weight of the foraging motion, and it is defined in range [0,1], β_i^{food} is the food attractiveness, and β_i^{best} is the effect of the best krill found in the population so far. According to empirical test, the best value for the V_f is 0.02 ms_{-1}.

The effect of the food depends on its location. The center of the food is discovered first and it is used for formulation of food attraction. This can only be estimated. In [6], the virtual center of food concentration is estimated according to the fitness distribution of the krill individuals, which is inspired from the "center of mass". This center of food in each iteration is defined as:

$$X^{food} = \frac{\sum_{i=1}^{N} \frac{1}{k_i} X_i}{\sum_{i=1}^{N} \frac{1}{K_i}} \tag{11}$$

The food attraction of the i-th krill individual is defined as:

$$\beta_i^{food} = C^{food} K_{i,food} X_{i,food} \tag{12}$$

where C^{food} is the food coefficient, and it is defined as follows:

$$C^{food} = 2(1 - \frac{l}{l_{max}}) \tag{13}$$

Physical diffusion of the krill individuals is a random process, and it is used for exploration of the search space. It is formulated using maximum diffusion speed and a random directional vector:

$$D_i = D^{max} \delta \tag{14}$$

where D^{max} is the maximum diffusion speed, and δ is a random directional vector. Empirically calculated maximum diffusion speed is in the range [0.002,0.010] ms^{-1} [6].

Above defined motions frequently change the position of a krill individual towards the best fitness. Motions contain two global and two local strategies, which make KH very powerful algorithm [6]. The position of a krill individual in the time interval [t, t + Δt] is given below:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \tag{15}$$

Pseudo-code of the KH algorithm is given below [6].

Algorithm 1 Flower pollination algorithm [22]

Initialization
Generate random solutions for M flowers
repeat
  Find the best solution $f(x^*)$ among initialized flowers
  Motion calculation
  foraging motion
  physical diffusion
  Updating update the krill individual position in the population
until stop criteria=FALSE
return the best solution among all population

IV. Flower pollination algorithm

Flower pollination algorithm (FPA) represents one of the latest swarm intelligence algorithm proposed in 2012 by Yang [22]. It was inspired by natural phenomenon of pollen grains transfer from the stamens to the ovule-bearing organs or to the ovules (seed precursors) by themselves.

A single flower or pollen gamete will constitute a solution of the optimization problem, with the whole flower population being used while their constancy will be used as solution fitness. Pollen will be moved in the course of two operations: global and local pollination. Initially, random positions for the flowers were generated. Global pollination employs pollinators to carry pollen to long distances towards individual characterized by higher fitness while the local pollination occurs within limited range of individual flower thanks to pollination mediators like wind or water.

Global pollination occurs with probability p that is determined by so called switch probability while the local pollination takes place if global pollination is omitted. The first one constitutes of pollinator's movement towards best solution $x^*(k)$ found by the algorithm, with s representing N-dimensional step vector following a L'evy distribution:

$$L(s) \approx \frac{\lambda \Gamma(\lambda) \sin(\pi \frac{\lambda}{2})}{\pi s^{1+\lambda}}, \quad (s \gg s_0 > 0) \tag{16}$$

where Γ represents standard gamma function and parameters $\lambda = 1.5$, $s_0 = 0.1$ [24]. Local pollination is applied to two randomly selected specimens of the population. It is performed via movement towards them with randomly selected step size ϵ [22]:

$$s \leftarrow Levy(s_0, \gamma)$$
$$x = x_m(k) + s(x^*(k) - x_m(k)) \tag{17}$$

where m is the member index, k is the current iteration and $x^*(k)$ is the best solution in k^{th} iteration.

Local pollination is defined by the following equations:

$$\epsilon = rand(0, 1) \tag{18}$$
$$r, q \leftarrow integer\_rand(1, M)$$
$$x = x_m(k) + \epsilon(x_q(k) - x_r(k)) \tag{19}$$

where M is population size. Flower pollination algorithm is summarized in Algorithm 2.

Algorithm 2 Flower pollination algorithm [22]

Initialization
Generate random solutions for M flowers
Find the best solution $f(x^*)$ among initialized flowers
repeat
  for all members of the population **do**
    if *rand(0,1)*$< p$ **then**
      Perform global pollination according to equation 17
    else
      Perform local pollination according to equations 19
    end if
    Check if the new solution is better
    Find the best and copy the population
  end for Evaluate accuracy of the retinal vessel segmentation based on ground truth image
until stop criteria=FALSE
return the best solution among all population

V. Experimental Results

To test our proposed method we used Matlab R2016a and experiments were done on the platform with Intel ® Core$^{\text{TM}}$ i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS.

We tested krill herd algorithm and flower pollination algorithm on ten standard benchmark functions considered in CEC 2013 competition [11]. In Table I are listed used function along with their optimal values. In this paper we tested KH and FPA on five unimodal and five basic modal functions.

TABLE I
Benchmark functions

No	Function	Optimal
	Unimodal functions	
1	Sphere function	-1400
2	Rotated high conditioned elliptic function	-1300
3	Rotated bent cigar function	-1200
4	Rotated discus function	-1100
5	Different powers function	-1000
	Basic modal functions	
6	Rotated Rosenbrocks function	-900
7	Rotated schaffers F7 function	-800
8	Rotated Ackleys function	-700
9	Rotated weierstrass function	-600
10	Rotated Griewanks function	-500

Parameters for both algorithms were set empirically considering recommendations proposed by authors in [6] and [22]. Population in FPA was set to 25 and the maximal number of iterations was 2000. Parameter p was set to 0.8. Parameters for KH were set as follows: population size was 25, maximal number of iteration was 2000 (same as for FPA), foraging speed V_f was 0.02 and diffusion speed (D^{max}) to 0.005. The algorithms were tested on 30 independent runs. For each function mean, standard deviation, the best and the worst value of the function were calculated. The obtained results were presented in Table II.

As it can be seen, both algorithms found the optimal function value for f_1 (sphere). Standard deviation is 0 for both functions which means that KH as well as FPA successfully determined optimal function value every time. KH algorithm found almost exact optimal value for and f_5 (different powers function) while FPA found the optimal value in each run. KH algorithm was completely unable to find optimal solution for functions f_2, f_3 and f_4. For this function obviously some specific parameter settings are needed and probably more iterations. FPA has also a problem with f_3 where mean was approximately -1116 while the optimal value is -1200. FPA was able to find almost optimal value in at least one run, but it also made some major errors in some other runs (the worst run obtained -771.93) but still a lot better then KH.

For functions f_8 and f_{10} obtained function values by both algorithms were similar and for f_{10} close to the optimal ones. FPA made the largest error for function f_9 where the mean errors made was larger then 5. In case of the KH algorithm this error was slightly lower. Standard deviation for FPA and KH (excluding the functions f_2, f_3 and f_4) was the largest for f^7. FPA had less standard deviation and better mean solution comparing to KH.

KH algorithm obtained better mean value for f_6 and it also had smaller standard deviation which means that KH is more stable then the FPA for this function. Even though KH obtained better overall mean function value, FPA was able to find the optimal value in at least one run.

If we exclude functions f_2, f_3 and f_4 form this analysis it can be concluded that FPA and KH are competitive and that both show similar quality which can be increased by adjusting the parameters or increasing the iteration numbers. KH was not able to deals with mentioned three functions which gives FPA a slight advantage.

VI. Conclusion

In this paper we performed comparative study of two novel swarm optimization algorithms, krill herd and flower pollination algorithm. Both algorithms were tested on ten CEC 2013 benchmark functions. Based on the experimental results it can be concluded that FPA perform slightly better than KH algorithm because it showed good results for all 10 functions while KH was not able to find even a close solution for three functions. KH algorithm obtained good results for the rest of the test functions which shows that KH can be used but some improvements are necessary in some cases. In further work, modification of KH and FPA can be proposed and tested against several other swarm optimization algorithms.

References

[1] A. Alihodzic and M. Tuba, "Bat algorithm (BA) for image thresholding," *Recent Researches in Telecommunications, Informatics, Electronics and Signal Processing*, pp. 17–19, 2013.

[2] N. Bacanin, "Implementation and performance of an object-oriented software system for cuckoo search algorithm," *International Journal of Mathematics and Computers in Simulation*, vol. 6, no. 1, pp. 185–193, December 2010.

[3] N. Bacanin and M. Tuba, "Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint," *The Scientific World Journal, special issue Computational Intelligence and Metaheuristic Algorithms with Applications,*, vol. 2014, no. Article ID 721521, p. 16, 2014.

TABLE II
COMPARISON OF KH AND FPA

Function	Alg.	Mean	St. dev.	Best	Worst
f_1	KH	-1400.0000000	0.0000000	-1400.0000000	-1400.0000000
	FPA	-1400.0000000	0.0000000	-1400.0000000	-1400.0000000
f_2	KH	387280.2317293	286818.0221234	103192.7439208	904093.8408671
	FPA	**-1299.9023110**	0.0889395	-1299.9830370	-1299.6865120
f_3	KH	621902.2290795	1069172.4829060	17414.5575920	2952745.7197942
	FPA	**-1116.351252**	144.4921849	-1199.9682280	-771.9309584
f_4	KH	3761.89868351	2552.4289585	643.3451174	7285.6516169
	FPA	**-1098.334102**	0.8955666	-1099.7512370	-1096.5191830
f_5	KH	-999.9999238	0.0000062	-1000.0000000	-999.9998226
	FPA	**-1000.0000000**	0.0000000	-1000.0000000	-1000.0000000
f_6	KH	**-899.6049742**	0.6253705	-899.9997206	-898.3369359
	FPA	-897.5948996	4.1407605	-900.0000000	-890.1875773
f_7	KH	-771.5861781	7.5942124	-785.7652709	-763.2931962
	FPA	**-789.8087548**	5.3349824	-796.6628086	-777.5942683
f_8	KH	**-679.7501401**	0.0302581	-679.7995897	-679.7100870
	FPA	-679.6151865	0.0734112	-679.7542732	-679.4893379
f_9	KH	**-596.7403810**	0.8755057	-598.1855848	-595.7863809
	FPA	-594.5554265	0.5394321	-595.8160265	-593.8232158
f_{10}	KH	-499.7990839	0.0920729	-499.9247597	-499.6533806
	FPA	**-499.9643280**	0.0208393	-499.9876748	-499.9172826

[4] I. Brajevic and M. Tuba, "An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems," *Journal of Intelligent Manufacturing*, vol. 24, no. 4, pp. 729–740, August 2013.

[5] ——, "Cuckoo search and firefly algorithm applied to multilevel image thresholding," in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, ser. Studies in Computational Intelligence, X.-S. Yang, Ed., vol. 516. Springer International Publishing, 2014, pp. 115–139.

[6] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun Nonlinear Sci Numer Simulat*, vol. 17, pp. 4831–4845, 2012.

[7] R. Jovanovic and M. Tuba, "An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem," *Applied Soft Computing*, vol. 11, no. 8, pp. 5360–5366, December 2011.

[8] ——, "Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem," *Computer Science and Information Systems (ComSIS)*, vol. 10, no. 1, pp. 133–149, January 2013.

[9] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3021–3031, 2011.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, 1995.

[11] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, 2013.

[12] M. Subotic and M. Tuba, "Parallelized multiple swarm artificial bee colony algorithm (MS-ABC) for global optimization," *Studies in Informatics and Control*, vol. 23, no. 1, pp. 117–126, 2014.

[13] I. Tsoulos and A. Stavrakoudisb, "Enhancing pso methods for global optimization," *Applied Mathematics and Computation*, vol. 216, no. 10, pp. 2988–3001, July 2010.

[14] E. Tuba, M. Tuba, and D. Simian, "Adjusted bat algorithm for tuning of support vector machine parameters," in *Congress on Evolutionary Computation (CEC),*. IEEE, 2016, pp. 2225–2232.

[15] ——, "Handwritten digit recognition by support vector machine optimized by bat algorithm," in *24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,(WSCG 2016)*, 2016, pp. 369–376.

[16] M. Tuba, "Asymptotic behavior of the maximum entropy routing in computer networks," *Entropy*, vol. 15, no. 1, pp. 361–371, January 2013.

[17] M. Tuba and N. Bacanin, "Artificial bee colony algorithm hybridized with firefly metaheuristic for cardinality constrained mean-variance portfolio problem," *Applied Mathematics & Information Sciences*, vol. 8, no. 6, pp. 2831–2844, November 2014.

[18] ——, "Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems," *Neurocomputing*, p. 11, 2014.

[19] M. Tuba, I. Brajevic, and R. Jovanovic, "Hybrid seeker optimization algorithm for global optimization," *Applied Mathematics and Information Sciences*, vol. 7, no. 3, pp. 867–875, May 2013.

[20] X.-S. Yang, "A new metaheurisitic bat-inspired algorithm," *Studies in Computational Intelligence*, vol. 284, pp. 65–74, 2010.

[21] ——, "Efficiency analysis of swarm intelligence and randomization techniques," *Journal of Computational and Theoretical Nanoscience*, vol. 9, no. 2, pp. 189–198, 2012.

[22] ——, "Flower pollination algorithm for global optimization," in *International Conference on Unconventional Computing and Natural Computation*. Springer, 2012, pp. 240–249.

[23] X.-S. Yang and S. Deb, "Cuckoo search via levy flights," 2009, pp. 210–214.

[24] X.-S. Yang, M. Karamanoglu, and X. He, "Multi-objective flower algorithm for optimization," *Procedia Computer Science*, vol. 18, pp. 861–868, 2013.