Performance Comparison of Cat Swarm Optimization and Genetic Algorithm on Optimizing Functions

David
STMIK Pontianak, Pontianak, Indonesia
DavidLiauw@gmail.com

Tri Widayanti
STMIK Pontianak, Pontianak, Indonesia
triwidayanti69@gmail.com

Muhammad Qadafi Khairuzzahman
STMIK Pontianak, Pontianak, Indonesia
m.qadafi.k@gmail.com

***Abstract*— This study was conducted to find out the best performance resulted from Cat Swarm Optimization (CSO) and Genetic Algorithm (GA). CSO is one of the algorithms developed based on the behavior of a number of cats to solve optimization problems. It is noted that they include two problem solving modes such as seeking and tracing. The seeking mode happens passively and cautiously. Being cautious means bewaring of the prey or other cats. The latter mode, however, occurs when the cats are active in searching the prey by moving toward it. Since they spend most of the time to idle, the seeking mode is more frequently found. In order to maximize the improvement of tracing mode, a recent calculation method was added. This study implemented three experiment functions such as sphere, rastrigrin, and knapsack. Each experiment was conducted ten times to find out the quantity of iterations and time needed for each method. The results show that CSO is better than GA due to its performance in terms of iterations and time.**

Keywords*— *Cat Swarm Optimization, Genetic Algorithm, Optimizing Functions, Comparison, Optimizing Problems

I. Introduction

One of the problems existing in artificial intelligence is optimization. It refers to computation maximizing or minimizing the functions of aims by considering the problems [1]. There are two functions of optimization such as a linear function and a nonlinear function [1]. Various methods have been developed by researchers to discover solutions to optimizing problems [2]. An optimizing method can give the best, expected results. It requires fine strategy in decision making. Common methods used are Genetic Algorithm (GA) and Swarm Intelligence. The latter one is one of the techniques in artificial intelligence that is based on collective self-organizing behavior of decentralized system [3]. The application exists in Ant Colony Algorithm. Such the algorithm is used to solve optimizing problems inspired from the behavior of artificial ants [4]. Nevertheless, GA has been frequently researched in some of these recent years [5]. The inspiration of this algorithm comes from natural, one-way selection showing fine results resembling intentional optimization. It involves calculation based on metaheuristic framework [6]. This research focused on performance comparison of Cat Swarm Optimization (CSO) and GA becoming the evolution of solving optimizing problems. Other research applied CSO as a discrete solution [7]. It was implemented in the form of the binary to solve the benchmark and zero-one knapsack problems. The results were compared to the ones given by GA and several versions of discrete binary optimization. It was found that the method of CSO extremely improved results obtained.

In addition, there was research on Particle Swarm Optimization (PSO) and GA, heuristic seeking methods based on population [6]. The finding showed that PSO was as effective as GA when optimal global solutions were sought. However, PSO was more efficient in terms of computation. Specifically, functions were only evaluated through statistical analysis and formal hypothesis examination [6].

The problem formulated in this research was on performance comparison of CSO and GA based on iterations and time used. The aim was to find out the one with better, faster performance in solving optimizing problems.

There were three optimizing actions to take in this experiment such as minimizing the function of sphere with slight, local minima, minimizing the function of rastrigrin with a large amount of local minima, and knapsack problem.

II. Research Method

This research described literature of CSO and GA. Besides, calculation of an optimal score of aim functions was conducted by using an application developed with programming language, i.e. Python. CSO was developed in such a way so that aim functions could be adjusted, a number of generations were varied, calculation time was added, and random tracing techniques were elaborated.

Planning, analysis, implementation design, examination, and discussion were involved by using the following algorithms:

A. Cat Swarm Optimization (CSO)

CSO is concerned in new research through observation of the behavior of cats. Two sub-models, i.e. tracing and seeking modes exist. Previous research results indicated that CSO performed better than PSO did [8][9]. CSO is a metaheuristic method applied to solve optimization problems efficiently by observing a group of cats seeking the prey. These animals have similar behavior patterns, i.e. hunting and being curious about the prey. They, nonetheless, spend more time for passive activities like sleeping, being idle, and being silent. They further beware of the surrounding. When being silent, they keep observing the prey. After finding it, they directly take a quick action, i.e. hunting [8][9].

The method of CSO is, to restate, based on the behavior model of cats in solving optimizing problems. The cats represent solutions [14]. Meanwhile, the prey represents global optimum or the best possible result. The environment further means seeking areas. Each cat moves based on the condition of either of the two modes [7][8].

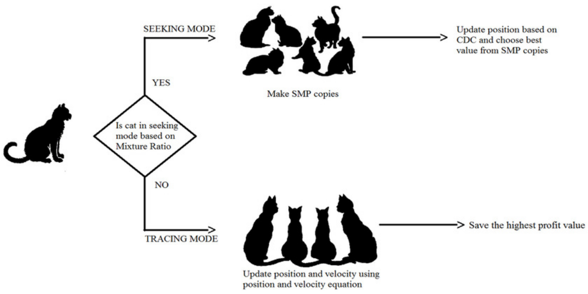

Fig. 1. Representation of CSO Algorithm [13]

Seeking mode refers to the passive situation when cats take a rest. They, nevertheless, keep bewaring of the surrounding. Seeking mode is divided into four factors such as Seeking Memory Pool (SMP), Seeking Range of the Selected Dimension (SRD), Counts of Dimension Change (CDC), and Self Position Considering (SPC). SMP is used to define the seeking memory size of each cat and indicate spots. The cats further select them based on the memorial pool. SRD, however, states the movement of selected dimensions. In seeking mode, if a dimension moves, the difference of new and old values should not exceed SRD. Another factor named CDC shows several changing dimensions. Finally, SPC is Boolean value used to decide a spot indicating the movement position of cats. SPC value does not influence SMP value. The following are steps of the seeking mode [8][9]:

1. Rise imitated j from the current position, where j is similar to SMP. If SPC value is true, j = (SMP-1). Next, maintain the current position as one of the candidates. Calculate the fitness value or aim function value as an early solution.
2. Each imitation is adjusted with CDC. Add or subtract SRD percent from the current value at random and change the previous value. In order to compute the number of dimensions to be modified, the formula is CDC * n.
3. Computing the modified value of each selected dimension, the following equation is used:

$$\text{Modification}_{i,d} = x_{i,d} + (-1)^{m} * \text{SRD} * x_{i,d}$$

4. Calculate the adjusted value (FS) and all candidate spots.
5. If all FSs are different, compute selected probability value of each candidate spot by using the following equation:

$$P_i = \frac{FS_i - FS_b}{FS_{max} - FS_{min}}$$

   On the contrary, control the selected probability value for all similar spots by 1.

6. Choose the movement spot of candidate spots and move the position of cats at random.

Tracing mode is, however, a model describing the situation when the cats follow target tracks. Once they are in tracing mode, they keep moving based on speed of each dimension. The three steps of tracing mode are as follows [10]:

1. Update the speed value of each dimension $V'_{k,d} = V_{k,d} + r * c\,(X_{best,d} - X_{k,d})$, where: d = 1,2,…,n and k = 1,2,…,m.
2. Check whether the velocity is maximal. If the new one exceeds the range, state a value that is similar to the limit.
3. Update the position of the $cats_k$ based on speed increase.

$$x_{k,d} = x_{k,d} + v_{k,d}$$

Next, Figure 2 on the flow chart of CSO is provided.

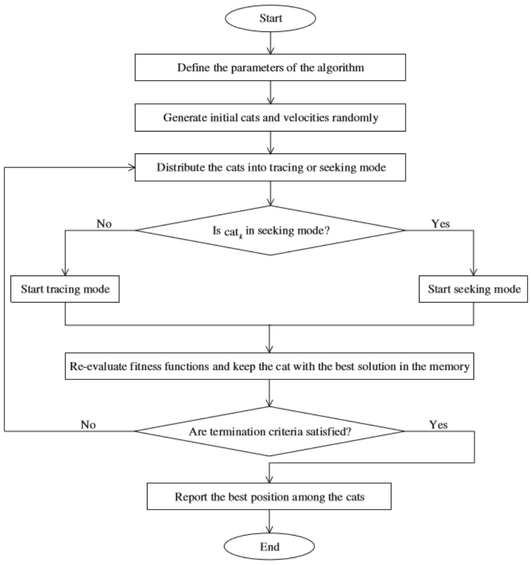

Fig. 2. Flow Chart of CSO [8][9].

B. Genetic Algorithm (GA)

GA was found by John Holland long time ago [10]. Such the algorithm functions to seek solutions of complex problems, particularly pertaining to criteria and constraints. GA applies the principle of science of evolution. Solutions represent individuals. More specifically, the solution value is represented in the form of the chromosome so that operation of improving the performance of individuals can be better [18].

In general, the steps of GA can be seen in Figure 3 [11].

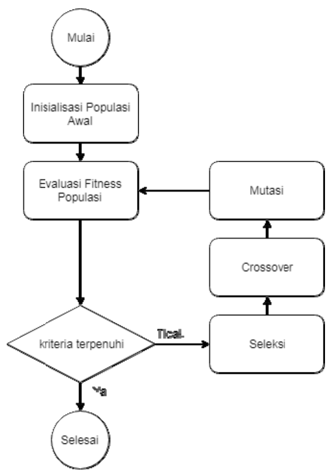

Fig. 3. Flow Chart of GA [9]

As shown in Figure 3, an early population of individuals is initiated by chance. After that, fulfilled fitness values of all individuals are calculated by using previously stated functions. If the criteria are fulfilled, GA is finished. They can be in forms of fulfilled fitness values meaning that repetition reaches the number of stated iterations or other certain conditions. If the criteria are not fulfilled, GA is conducted.

There are three stages of pure GA operation. The first one is selection. Here individuals of the population are selected through a method until the determined number is reached. Results of this operation produce pairs which are ready for the next stage. Common selection methods put into application are roulette wheel and elitism.

Following this, crossover is conducted. In pure GA, this stage involves pairs of individuals at random through the determined method. Single dot crossover, multiple dot crossover, and uniform crossover are frequently used methods. In general, there are two primary pairs used. Each process produces two recent individuals. Crossover process happens if a random, free number fulfills criteria of Pc (Probability of crossover) stated.

A final stage refers to mutation. Here a number of individuals are picked to determine a correct technique. Several familiar mutation techniques are single dot mutation and multiple dot mutation. Similar to crossover, mutation occurs if a random, free number fulfills criteria of Pm (Probability of mutation). Pm is usually smaller than Pc.

In this study, the method of tournament selection was used. Next, the best fitness value was stated for individuals of the population selected by chance.

III. PROPOSED NEW METHOD

Modification techniques used to improve convergence of CSO can be addition of a new parameter to equation of a renewed position and the use of new equation form of updated speed [17]. In this research, however, a new technique with a modified tracing method was proposed.

In tracing mode of CSO, the position of the cats was updated based on velocity increase [14][15]. In order to maximize the velocity, a recent calculation method was added to update the tracing mode after the result of each dimension was found $V'_{k,d}$. Random tracing method (r) was applied by adding or subtracting the velocity value selected with a small, random value and range r [-0.1, 0.1]. Therefore, the following formula was used:

$$v' = v + r\,(x_{max} - x_{min})$$

with variable domain $x = [x_{min}, x_{max}]$.

IV. EXPERIMENTAL RESULTS

Performance examination of the two algorithms was divided into three stages with different purposes. Proposed CSO performance was modified with a random tracing mode applying functions of standardized tests of optimization [16]. The following was description of each trial:

a. The first inputted data were on a general function, i.e. Sphere Function used to examine optimization of algorithm performance. This research definitely aimed to cognize performance of CSO and GA.
b. The second inputted data were on nonlinear equation system, i.e. Rastrigrin Function. The goal of this research was to cognize efficiency of CSO and GA in solving nonlinear equation problems.
c. The third stage was on Knapsack, an optimization problem. It concerned the ways people could bring a number of things without exceeding the capacity. CSO and GA were simulated based on optimization of sphere function, rastrigrin function, and knapsack problem. The computation was conducted through applications created with programming java language set on Intel Core i5 with 4GB memory. Results of CSO and GA were compared.

Examination included three problems. In order to retain the balance, several rules were equalized. They were as follows:

1. There were 10 individuals in the population.
2. For CSO, SMP value = 5.
3. For CSO, SRD value = 60%.
4. For CSO, c value = 1.
5. For GA, elitism selection was applied.
6. For GA, single dot crossover was applied.
7. For GA, single dot mutation was applied.
8. For CSO, nSeekerCat = 0.7 and nTrackerCat = 0.3. For GA, Pc = 0.7 and Pm = 0.1.
9. An early population was freely randomized.
10. Each trial was conducted 10 times.
11. Trials failed when there were more than 1000 generations.

In terms of the sphere function, the seeking region limit was from -5.12 to 5.12. The aim was to search a global, minimal value, where f(0,0) = 0. The following equation was used:

$$f(x) = \sum_{i=1}^{n} x_i^2 \quad ......(1)$$

TABLE I. EXAMINATION RESULTS OF SPHERE FUNCTION

No	CSO Iteration	CSO Time (s)	GA Generation	GA Time (s)
1	8	0.0387	31	0.203
2	12	0.0582	44	0.482
3	8	0.0295	29	0.231
4	7	0.0312	45	0.659
5	10	0.059	22	0.204
6	14	0.08	53	0.723
7	11	0.0448	49	0.695
8	25	0.102	33	0.422
9	8	0.028	40	0.786
10	10	0.0524	36	0.282
Average	11.3	0.05238	38.2	0.4687

Findings showed that CSO had faster performance in terms of number of iterations and work time. With around 11 iterations, the average work time was 0.0524. The comparison of CSO and GA was quite significant. It was noted that the number of iterations were threefold in terms of velocity and time.

Moreover, the seeking region limit of rastrigrin function was similarly from -5.12 to 5.12. The goal was to search a global, minimal value, where f(0) = 0. The following equation was used:

$$f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\Pi x_i) \right] \quad ......(2)$$

TABLE II. EXAMINATION RESULTS OF RASTRIGRIN FUNCTION

No	CSO		GA	
	Iteration	Time (s)	Generation	Time (s)
1	40	1.472	67	1.901
2	47	1.72	57	1.534
3	38	1.251	463	10.672
4	50	4.421	295	6.794
5	13	0.406	611	13.292
6	53	1.815	223	5.386
7	50	1.432	743	15.266
8	97	2.886	524	11.078
9	33	0.167	57	1.659
10	36	0.816	222	4.872
Average	45.7	1.6386	326.2	7.2454

Results indicated that CSO was much better than GA. This could be viewed from extremely significant iterations and work time.

Knapsack problem was on 20 things weighing 50kg. The following list and weight of things were provided:

TABLE III. LIST OF THINGS FOR KNAPSACK PROBLEM

No	Weight (Kg)	No	Weight (Kg)
1	3	11	16
2	5	12	18
3	4	13	9
4	2	14	13
5	1	15	8
6	11	16	19
7	10	17	2
8	18	18	6
9	4	19	10
10	11	20	10

For examination of knapsack problem, CSO applied dimensions based on existing things. The range of each dimension is from 0 to 1. If its value was more than 0.5, the represented thing was included. However, if it was equal to or less than 0.5, exclusion applied. The following examination results of knapsack problem were given:

TABLE IV. EXAMINATION RESULTS OF KNAPSACK PROBLEM

No	CSO		GA	
	Iteration	Time (s)	Generation	Time (s)
1	7	0.324	341	7.865
2	9	0.387	12	0.399
3	10	0.698	26	2.551
4	17	1.015	43	2.448
5	5	0.175	13	0.728
6	11	0.877	149	5.618
7	13	0.704	52	2.326
8	6	0.356	54	2.168
9	9	0.45	55	2.077
10	5	0.193	19	1.178
Average	9.2	0.5179	76.4	2.7358

It was realized that results also proved better performance of CSO.

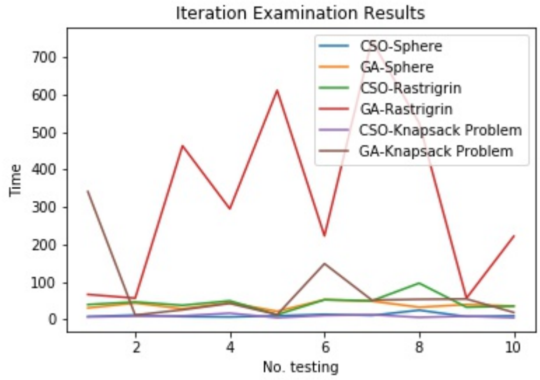

Fig. 4. Iteration Examination Results

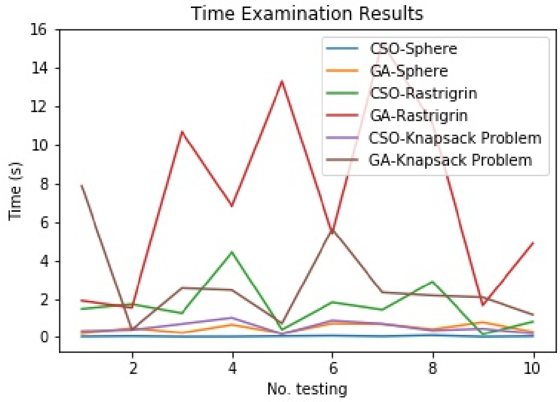

Fig. 5. Time Examination Results

Based on examination of the three cases (see Figures 4 and 5), it could be affirmed that CSO was more convergent in searching solutions of short iterations with quick processing time.

CONCLUSION

Metaheuristic approach of CSO is one of the best methods in solving optimizing problems. It is obviously proven that CSO is better than GA in terms of iterations and work time. Time required to obtain the most optimal result is through CSO. This happens due to absence of time required to code chromosomes. Meanwhile, GA has more randomized number of iterations. However, in order to obtain the most optimal result, modification of CSO is necessary. It is suggested that future research seeks an optimal value of optimizing problem and other aim functions (e.g. Schwefel, Rosenbrock, Griewank, and Ackley) through CSO and random tracing technique.

REFERENCES

[1] D. Solow. "Linear and nonlinear programming". *Wiley Encyclopedia of Computer Science and Engineering.*2007.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, *406*(6791), 39. 2000.

[3] C. Blum and D. Merkle. "Swarm intelligence. Swarm Intelligence in Optimization"; Blum, C., Merkle, D., Eds, 43-85. 2008.

[4] M. Dorigo and T. Stützle. "The ant colony optimization metaheuristic: Algorithms, applications, and advances". In *Handbook of metaheuristics* (pp. 250-285). Springer, Boston, MA. 2003.

[5] A. Abraham and L. Jain, “Evolutionary Multiobjective Optimization”. In *Evolutionary Multiobjective Optimization* (pp. 1-6). Springer, London. 2005.

[6] R. Hassan, B. Cohanim, O. De Weck, G. Venter. “A comparison of particle swarm optimization and the genetic algorithm”. In *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference* (p. 1897). 2005.

[7] Y. Sharafi, M.A. Khanesar, M. Teshnehlab, M. “Discrete binary cat swarm optimization algorithm”. In *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)* (pp. 1-6). IEEE. 2013.

[8] S.C. Chu, P.W. Tsai, J.S. Pan. “Cat swarm optimization”. In *Pacific Rim international conference on artificial intelligence* (pp. 854-858). Springer, Berlin, Heidelberg. 2006.

[9] S.C. Chu, P.W. Tsai. “Computational intelligence based on the behavior of cats”. *International Journal of Innovative Computing, Information and Control*, *3*(1), 163-173. 2007.

[10] M. Orouskhani, Y. Orouskhani, M. Mansouri, M. Teshnehlab. “A novel cat swarm optimization algorithm for unconstrained optimization problems”. *International Journal of Information Technology and Computer Science (IJITCS)*, *5*(11), 32. 2013.

[11] D. Whitley, “A genetic algorithm tutorial”, Statistics and computing, Vol.4, No. 2, pp. 65-85, 1994

[12] B.K. Panigrahi, Y. Shi, M. H. Lim (Eds.). “Handbook of swarm intelligence: concepts, principles and applications*”*. Vol. 8. Springer Science & Business Media. 2011.

[13] K. Selvakumar, K. Vijayakumar, C. S. Boopathi. “CSO Based Solution for Load Kickback Effect in Deregulated Power Systems”. *Applied Sciences*, *7*(11), 1127. 2017.

[14] P.W. Tsai, V. Istanda. “Review on cat swarm optimization algorithms”. In *2013 3rd International Conference on Consumer Electronics, Communications and Networks* (pp. 564-567). IEEE. November 2013.

[15] P.M. Pradhan, G. Panda. “Solving multiobjective problems using cat swarm optimization”. *Expert Systems with Applications*, *39*(3), 2956-2964. 2012.

[16] L. Guo, Z. Meng, Y. Sun, L. Wang. “Parameter identification and sensitivity analysis of solar cell models with cat swarm optimization algorithm”. *Energy Conversion and Management*, *108*, 520-528. 2016.

[17] M. Orouskhani, M. Mansouri, M. Teshnehlab, M. “Average-inertia weighted cat swarm optimization”. In *International conference in swarm intelligence* (pp. 321-328). Springer, Berlin, Heidelberg. June 2011.

[18] C.F. Juang. “A hybrid of genetic algorithm and particle swarm optimization for recurrent network design”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *34*(2), 997-1006. 2004.