Selective Crossover base on Fitness in Multi-Swarm Optimization

Chiabwoot Ratanavilisagul
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
E-mail: chaibwoot@hotmail.com

Boontee Kruatrachue
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: booontee@yahoo.com

Abstract— **In Multi-Swarm Optimization (MSO) a particle update its position in the direction follows the group best position and its previous best position. The crossover was used to prevent trapping in local optima and increasing convergence rate. The crossover updates position using position value in each dimension from the group best position or its previous best position. This creates "jumping" in position and hence avoids trapping and increase global search space. Both update methods can be interleaved in MSO to retain its local search ability while enhancing global search. Each position update causes a particle to move to its new position in spite of it lower fitness value. This occurs more often in update crossover due to longer distance to its previous position. This paper propose selective crossover where the new position is update only if the fitness is improved. A set of benchmark test functions is used to compare the convergence rate and the achieving optima point among each MSO implementation. The results of experiment show that selective crossover improve convergence speed and high optimizing followed by MSO with crossover and MSO.**

Keywords-swarm intelligence; Particle swarm optimization; Cauchy mutation

I. Introduction

Kennedy and Eberhart [1,2] introduced Particle Swarm Optimization (PSO) in 1995. It is motivated from the behavior of flying bird and their communication mechanism in solving optimization problems. PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted. Its main disadvantage is trapping in local optimum and premature convergence.

To avoid local optima and premature convergence, many researchers increased searching diversity of PSO.

Ratnaweera et.al. [3] used a mutation operator to solve the local optima problem and improve its performance.

Sun et.al. [4] proposed Quantum Particle Swarm Optimization (QPSO), where the quantum model was used to improve PSO performance. QPSO particles used a wave function instead of the standard position and velocity.

Jing et.al. [5] improved QPSO by using mutation mechanism. It parameters are the global best particle (gbest) and the mean value of all particles' previous best position (mbest). The parameters are mutated with Cauchy distribution. The result showed that QPSO with gbest and mbest mutation both performed better than the standard PSO.

Krohling et.al. [6][7] improved PSO from local minima using mutation with Cauchy probability distribution and Gaussian probability distribution.

On the other hand, many researchers [3][5][6][7] suggested that mutation decrease convergence rate. Some researchers proposed multiple swarms and exchanged information among them instead.

Al-Kazemi and Mohan [8] improved PSO by divided population into two groups. One group moves to the gbest while the other moves in the opposite direction.

Baskar and Suganthan [9] also divided population of PSO into two groups. They performing cooperation search by sharing the gbest information between them.

Peram et.al. [10] also divided population of PSO into two groups using different update equation. One group uses the standard PSO while the other uses the Fitness-to-Distance ratio PSO (FDR-PSO).

El-Abd and Kamel [11] also divided population of PSO into two groups. They exchanged the global best particle or the best particle. The experiment results showed that exchanging the best particles had better performance than exchanging the global best particle.

Changhe Li et.al. [12] divided population of PSO into multiple groups called fast multiple swarm (FMSO). It also used a selective Cauchy mutation to prevented trapping in local optimum. FMSO introduced novel position update technique using crossover operation. The crossover updates use position value in a dimension from the value of that dimension of either group best position or its previous best position.

The position generated from crossover updated often cause abrupt change in position and may results in worst position. This paper proposed selective update where the position is update only if the new position has better fitness.

The rest of this paper is organized as follows. Section 2 explains PSO. Section 3 explains the FPSO [12]. Section 4 explains selective crossover update technique. Section 5 describes the experiment setup and presents the experiment results. Section 6 concludes the paper with a brief summary.

II. Particle Swarm Optimization

In the standard PSO, Each particle in PSO shares the information with neighbor particle. Particles search space by following the previous global best positions and the previous best positions of itself.

The PSO algorithm starts with random particles position and velocity. Each generation, particle updates its position and velocity as follows:

$$X_i' = X_i + V_i' \tag{1}$$

$$V_i' = \omega V_i + \eta_1 rand()(P_i - X_i) + \eta_2 rand()(P_g - X_i) \tag{2}$$

where X_i' represents the current positions of i particle, X_i represents the previous positions of i particle, V_i represents the previous velocity of i particle, V_i' represents the current velocity of i particle, P_i represents the individual's best position, P_g represents the best position found in the whole swarm so far, $0 \le \omega < 1$ is an inertia weight, η_1 and η_2 are acceleration constants, rand() generates random number from interval [0,1]. A limit velocity calls $V_{\max}$. If calculate velocity of a particle exceeds this value, it will replace value of $V_{\max}$.

III. Fast Swarm Optimization

Each particle in PSO shares the information with neighbor particle. The information sharing mechanism increase PSO convergence rate and at the same time it may end in local optimum. In fact, the particles usually converge to local optima. To overcome this weakness, the Cauchy mutation operator is often used in PSO algorithm.

Normally, particles updated their position in each generation independent of the position fitness. In fact, their update usually leads them to local optimum. To solve this problem, FPSO algorithm proposed update particle position with Cauchy mutation and comparing with normal update based on position's fitness. The new position with better fitness is selected.

The basic steps of FPSO algorithm are as follows:

Step1: Generate initial particles by randomly generating the position and velocity for each particle.

Step2: Evaluate each particle's fitness.

Step3: For each particle, if the fitness is less than its previous best (P_i) fitness, update P_i.

Step4: For each particle, if the fitness is less than the best one (P_g) of all the particles, update P_g.

Step5: For each particle, do
   a) Generate a new particle p according to the formula (1) and (2).
   b) Generate a new particle p' according to the formula (3) and (4).

$$V_i' = V_i \exp(\delta) \tag{3}$$

$$X_i' = X_i + V_i'\delta_i \tag{4}$$

where X_i' is represent the current positions of i particle, X_i is represent the previous positions of i particle, V_i is represent the previous velocity of i particle, V_i' is represent the current velocity of i particle, δ and δ_i denote Cauchy random numbers.

   c) Compare p with p', choose the one with lesser fitness function.

Step6: If the stop criterion is satisfied, then stop, else go to Step 3.

IV. Proposed Work

This work study crossover operations on Multi-Swarm Optimization (MSO) based on FPSO (single swarm with Cauchy mutation).

We test various numbers of swarms that perform crossover operation. At first the best particle from a random swarm is selected. This particle crossover with all particles from a random swarm (RSC) or crossover over with all swarms (ASC).

The Algorithm for RSC is as followed:

Step1: Randomly select a best particle P_b from a random swarm.

Step2: Randomly select a swarm S.

Step3: For each particle P of swarm S.

For each dimension i of particle P_b,

If rand () < q_c crossover particle P with P_b as follows:

$$P_x[i] = (1-\alpha) * (P_x[i] + \alpha * P_b[i]) \tag{5}$$

$$P_v[i] = rand\ () * (P_b[i] - \alpha * P_x[i]) \tag{6}$$

where q_c is crossover rate, α is a random number from interval (0,1), P_x is position of particle P and P_v is velocity of particle P.

Step4: perform MSO update position for each particle in each swarm based on eq1-4.

Step5: if number of generation < $G_{\max}$

Go to step 1;

Else

Stop.

where $G_{\max}$ is the maximum generation.

For ASC the algorithm is the same except that all swarm is crossover with P_b.

The results of crossover operation causes a particle to move to its new position in spite of it lower fitness value. This occurs more often in update crossover due to longer distance to its previous positioning in comparing with normal update. So we proposed selective crossover operation where the new position is update only if the fitness is improved. The Step3 of the above algorithm is changed as follow:

Step3: For each particle P of swarm S.

For each dimension i of particle P_b,

If rand () < q_c crossover particle P with P_b as follows:

$$P_x[i] = (1-\alpha) * (P_x[i] + \alpha * P_b[i]) \tag{5}$$

$$P_v[i] = rand\ () * (P_b[i] - \alpha * P_x[i]) \tag{6}$$

If the particle P fitness is not improved,

Restore P to its previous position.

For RSC the selective version is called SRSC (SASC for ASC).

V. Experiments and Results

A. Introduction to the Benchmark Functions

The five benchmark functions with different properties are chosen to test crossover operations the equations are listed below:

1) Sphere function

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

Where $x \in [-5.12, 5.12]^n$

Sphere function is continues, convex and no has local minimum except the global one (single modal).

2) Rosenbrok's function

$$f(x) = \sum_{i=1}^{n} [100(x_i^2 - x_{i+1}) + (x_i - 1)^2]$$

Where $x \in [-2.048, 2.048]^n$

Rosenbrok's function has global optimum. It is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult. Rosenbrok's function is several local minima (multimodal).

3) Ackley's function

$$f(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})$$

$$-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$$

Where $x \in [-30, 30]^n$

Ackley's function has an exponential term that covers its surface with numerous local minima (several local minima) and regularly distributed. Ackley's function is multimodal.

4) Rastrigin's function

$$f(x) = 10n + \sum_{i=1}^{n} (x^2 - 10\cos(2\pi x_i))$$

Where $x \in [-5.12, 5.12]^n$

Rastrigin's function has the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the locations of the minima are regularly distributed. Rastrigin's function is multimodal.

5) Griewank's function

$$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1$$

Where $x\in[-300,300]^n$

Griewank's function is similar to Rastrigin function. It has many widespread local minima. However, the locations of the minima are regularly distributed. Griewank's function is multimodal.

B. *Parameters Setting for experiments*

Parameters are as follows for all experiments: acceleration constants of η_1 and η_2 are both set to be 1.496180 and inertia weight ω = 0.729844, as suggested by den Bergh [13], crossover rate q_c is 0.6, the number of population is 60, the number of swarms is 3 and each swarm's size is 20. Each function has 30 dimensions. For each function is run 50 times. The maximum generation is 1000.

C. *The experiment results and comparison experiment results*

The crossover experiments are performing on random and all swarm crossovers with and without fitness selection. They are also compared with MSO without crossover (NSC).

TABLE I. THE MAXIMUM(MAX) ,MINIMUM(MIN) AND AVERAGE(AVG) BEST FITNESS OVER 50 RUNS. THE CROSSOVER BY VARIOUS NUMBERS OF SWARMS.

Function		NSC	RSC	ASC
Sphere	Min	5.98E-32	3.28E-132	4.41E-276
	Avg	2.46E-26	4.33E-129	2.52E-272
	Max	4.93E-25	6.61E-128	5.08E-271
Rosenbrok	Min	1.22E-19	318.076	774.492
	Avg	3.23E-15	517.19	1086.14
	Max	3.83E-14	741.624	1415.37
Ackley	Min	2.12005	0	0
	Avg	5.69998	0	0
	Max	10.1482	0	0
Rastrigin	Min	41.7882	0	0
	Avg	94.0432	0	0
	Max	160.188	0	0
Griewank	Min	1.11E-16	0	0
	Avg	0.018836	0	0
	Max	0.151574	0	0

TABLE II. THE MAXIMUM(MAX) ,MINIMUM(MIN) AND AVERAGE(AVG) BEST FITNESS OVER 50 RUNS. THE SELECTIVE CROSSOVER BY THE FITNESS.

function		SRSC	SASC
Sphere	Min	2.14E-182	0
	Avg	1.96E-177	0
	Max	6.17E-176	0
Rosenbrok	Min	9.11E-19	7.24E-19
	Avg	1.56E-14	8.94E-15
	Max	6.75E-13	1.12E-13
Ackley	Min	0	0
	Avg	2.70E-15	6.39E-16
	Max	3.55E-15	3.55E-15
Rastrigin	Min	0	0
	Avg	0	0
	Max	0	0
Griewank	Min	0	0
	Avg	0	0
	Max	0	0

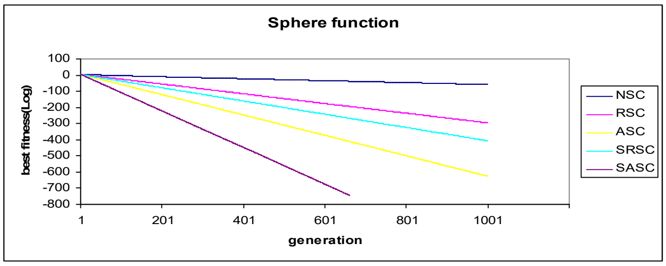

Figure 1. Comparison of the proess of mean fitness of the best particle between NSC, RSC, ASC, SRSC, SASC for 50 runs and for Sphere function, the verticle axis is the function value and the horizontal axis is the number of generation.

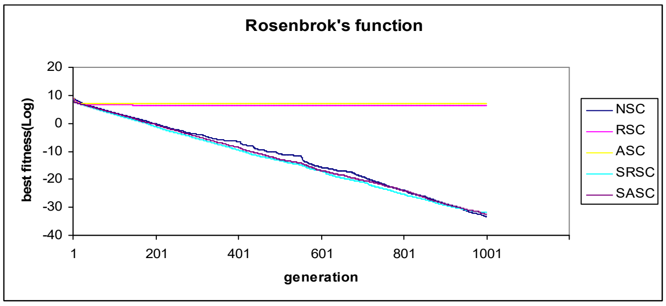

Figure 2. Comparison of the proess of mean fitness of the best particle between NSC, RSC, ASC, SRSC, SASC for 50 runs and for Rosenbrok's function, the verticle axis is the function value and the horizontal axis is the number of generation.

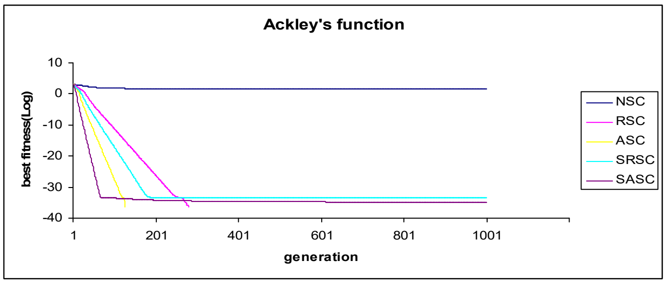

Figure 3. Comparison of the proess of mean fitness of the best particle between NSC, RSC, ASC, SRSC, SASC for 50 runs and for Ackley's function, the verticle axis is the function value and the horizontal axis is the number of generation.

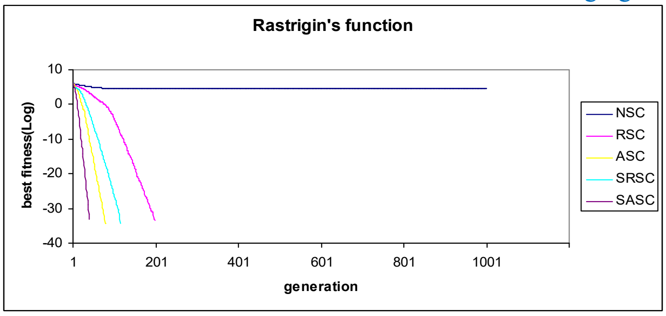

Figure 4. Comparison of the proess of mean fitness of the best particle between NSC, RSC, ASC, SRSC, SASC for 50 runs and for Rastrigin's function, the verticle axis is the function value and the horizontal axis is the number of generation.

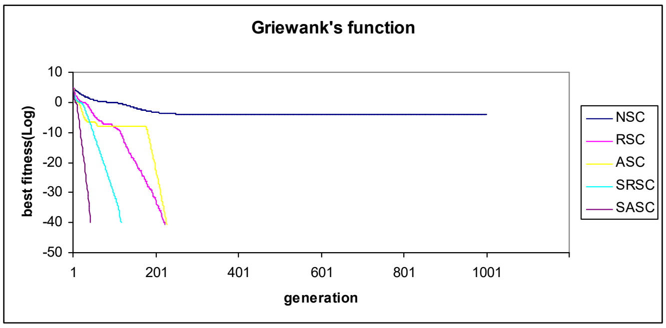

Figure 5. Comparison of the proess of mean fitness of the best particle between NSC, RSC, ASC, SRSC, SASC for 50 runs and for Griewank's function, the verticle axis is the function value and the horizontal axis is the number of generation.

The results are the about the same for all functions as seen from Fig. 1-5 Crossover operation for all groups is better except Rosenbrok's function. Selective version is better for all function independent of number of group performing crossover operation. MSO without crossover easily trap in local optima in all function except Rosenbrok's function.

From the results of Fig. 2 Rosenbrok's function, it is interesting to see that SASC, SRSC and NSC have very good convergence rate. RSC, ASC have very bad convergences rate and easily trapping in local.

VI. CONCLUSIONS

Result of experiments show that ASC has the fastest convergence rate because all swarms performs crossover operation in each generation. RSC has lower rate than ASC due to only single swarm in crossover operation.

The crossover operation increase probability of trapping in local optimum on multimodal with non regularly distributed locations minima (Rosenbrok's function). On the contrary, NSC without crossover performs better.

The crossover operation causes particles to “jump” in position. So, it easily goes out of local optima on multimodal with regularly distributed local minima.

In order to achieve good performance on both kind of functions (regularly and non regularly distributed locations minima), MSO with Selective Crossover is recommended.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, pp.1942-1948, 1995.

[2] R. C. Eberhart and J. Kennedy, A New Optimizer Using Particle Swarm Theory, Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp.39-43, 1995.

[3] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation, vol. 8,no. 3, pp. 240–255, 2004.

[5] Jing Liu, Wenbo Xu, Jun Sun, Quantum-behaved particle swarm optimization with mutation operator. Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence Pages: 237 - 240 , 2005.

[6] R. A. Krohling, Gaussian particle swarm with jumps, in Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK,pp. 1226-1231, 2005.

[7] R. A. Krohling, L. dos Santos Coelho, PSO-E: Particle Swarm with Exponential Distribution , in Proceedings of the IEEE Congress on Evolutionary Computation, pp1428- 1433, July 2006.

[8] B. Al-Kazemi and C. K. Mohan, “Multi-phase discrete particle swarm optimization” In Proc. of 4th Int. Workshop on Frontiers on Evolut.Alg., Research Triangle Park, NC, 2002.

[9] S. Baskar, and P. N. Suganthan, “A novel concurrent particle swarm optimization,” In Proc. of Cong. on Evolut. Comput., Portland, OR, pp. 792-796, 2004.

[10] T. Peram, K. Veeramachaneni, and C. K. Mohan, “Fitness-distance ratio based particle swarm optimization,” In Proc. of IEEE Swarm Intell. Symp., Indianapolis, IN, pp. 88-94, 2003.

[11] M. El-Abd, and M. Kamel, “Information exchange in multiple cooperating swarms,” In Proc. of Cong. on Evolut. Comput., Edinburgh, UK, pp. 138-142, 2005.

[12] Changhe Li, Qing Zhang, Yong Liu, and Lishan Kang, “A Fast Particle Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy,” Proceedings of the 2nd international conference Advances in computation and Intelligence 2007, Wuhan, China, pp.334-343.

[13] F. van den Bergh. An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.