



Effect of the Chaotic Crossover Operator on Breeding Swarms Algorithm

 Hüseyin Demirci¹,  Nilüfer Yurtay²

¹Corresponding Author; Sakarya University, Faculty of Computer and Information Sciences; huseyind@sakarya.edu.tr; +90 264 295 69 14

² Sakarya University, Faculty of Computer and Information Sciences; nyurtay@sakarya.edu.tr;

Received 18 September 2020; Revised 9 November 2020; Accepted 12 March 2021; Published online 18 March 2021

Abstract

In this paper we present effect of the chaotic crossover operator with different chaotic maps on the metaheuristic search algorithm Breeding Swarms algorithm which is the Particle Swarm Optimization's one of the genetic algorithm hybrid form. Some of the many optimization problems could have too many local extrema. Most of the time optimization algorithms could stuck on these extrema therefore these algorithms could have trouble with finding global extremum. To avoiding local extrema and conduct better search on search space, a chaotic number generator is used on Breeding Swarms algorithm's most of the random procedures. To test efficiency and randomness of the chaotic crossover operator, different chaotic maps are used on the Breeding Swarm algorithm. Test and performance evaluations are conducted on Multimodal and unimodal benchmark functions. This new approach showed us that modified Breeding Swarms algorithm yielded slightly better results than Particle Swarm Optimization and original Breeding Swarms algorithms on tested benchmark functions.

Keywords: optimization, chaos, particle swarm optimization, hybrid algorithm

1. Introduction

Optimization process is the selection of a value obtained from a dataset that its principles are defined before the process. Optimization problems can be solved by running many algorithms. No matter how on fast computers, these algorithms spend too much time to reach the exact solutions. Because of that, nature inspired meta-heuristic algorithms were developed. These algorithms generally try to find approximate solution rather than exact solution and run faster than the other algorithms. However, they have still many deficiencies. To solve these downsides, algorithms are combined with other algorithms that leads to the development of hybrid algorithms. Meta-heuristic algorithms are used in many optimization problem types such as single objective, multi objective and multi objective discrete problems. A helpful review about selecting proper algorithm for these problem types were made by Qi Liu et al. [1].

Particle Swarm Optimization (PSO) is a swarm based optimization algorithm which developed by Eberhart and Kennedy in 1995 [2]. This algorithm is very easy to implement and to code with any kind of programming language. PSO particles have its own memory. It can remember past experiences and share this experiences with other particles. With shared experience all particles towards to best solution and search other possibilities through that way. PSO and GA algorithms used in many real-world problems such as wireless sensor optimization and electrical load forecasting [3],[4].

This procedure may seem to be efficient but some bad particles could stuck on local extrema and waste computational time. To handle this problem, PSO has been hybridized with Evolutionary Algorithms such as Genetic Algorithm (GA) and Differential Evolution algorithm.

One of these hybridized algorithm is Breeding Swarms (BS) algorithm [5]. This algorithm combines GA with PSO and proposes new crossover operator Velocity Propelled Averaged Crossover (VPAC). PSO and GA uses aimed random search for optimization. While the algorithm's GA part conducts global search, PSO part makes the local search. Thus, the algorithm could escape local extrema. The lack of the algorithm is that GA part can stuck on local extrema.

In this paper we present Velocity Propelled Chaotic Crossover operator (VPCC) for improvement of this algorithm's GA part. GA and chaotic maps are used with together in many applications such as encrypting [6]. This crossover operator takes advantage of the unpredictability and randomness of the chaotic systems. With these new attributes BS algorithm escapes all of the local extrema and gets better results than the original algorithm.

2. Background and Related Works

2.1 Particle Swarm Optimization

Particle Swarm Optimization is a swarm based optimization algorithm [2]. This algorithm inspired from bird's behaviour of search for food. Main idea of the algorithm is that birds in the swarm inform the found food resource to each other and then the birds move toward to best food resource in the found resources. While birds moving to best resource, they search the search space for new food resources so that most of the search space will be scanned for alternate resources and this may lead to discovery of approximate best solution. Mathematical expression of particles movement given by equation (1)

$$x_i(t+1) = x_i(t) + v_i(t) \quad (1)$$

where x_i is the current value of i^{th} particle optimization variable, v_i is the velocity vector, t is the current iteration.

In original PSO algorithm velocity vector calculated as given in equation (2)

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \quad (2)$$

where $v_{ij}(t)$ is the velocity vector of the i^{th} particle at dimension $j=1\dots n$ in t^{th} iteration, $x_{ij}(t)$ is the position of the i^{th} particle at j^{th} dimension in t^{th} iteration, $y_{ij}(t)$ is the best solution's position of the i^{th} particle's at j^{th} dimension, $\hat{y}_j(t)$ is the best solution's position of all particles at j^{th} dimension, c_1 and c_2 are the personal and social learning coefficient respectively and the r_{1j} and r_{2j} are the random numbers generated in the range $[0.0,1.0]$. According to equation (2) particles use the knowledge of global best solution and personal best solution to move through the search space's corresponding dimension. The performance of the best solution is depends on c_1 and c_2 coefficients. If c_1 is bigger than c_2 , particles conduct local search. If c_2 is bigger than c_1 , particles conduct global search. Usually these parameters initially set equal to each other. Eberhart and Shi showed that c_1 and c_2 parameters can be set 1.494 [7].

General steps of PSO algorithm is shown below;

1. Generate the population. Each particle's velocity vector and variables are generated randomly
2. Calculate the fitness value. Each particle's fitness value calculated according to given fitness function
3. Determination of particle's best solution. Fitness value calculated in the previous step is compared with the best fitness value which is in the memory of the particle. If this value is better than the one in the memory, the value will be updated with better one
4. Determination of global best solution. Each particle's fitness value calculated in the second step compared with the global best fitness value. If this value is better than the one in the memory, the value will be updated with better one
5. Movement of the particle. Each particle's velocity vector will be calculated as stated in equation (2) and their positions will be calculated as stated in equation (1)
6. Until the stopping criterion to be achieved, procedures through 2 to 5 will be repeated

To determine the stopping criterion, two important issues must be considered. Firstly, the stopping criterion never has to cause the algorithm's early convergence. In this situation the algorithm makes only the local search, and this leads to stuck on the local extrema. Secondly, if the stopping criterion increases the costs of calculation, the algorithm cannot converge to the global minimum. Some of general stopping criteria are shown as below;

- Reaching the maximum iteration number,
- Reaching desired solution,
- The slowdown or lack of improvement

PSO algorithm has been changed through the time. Many researchers worked on the algorithm and developed some new features such as inertia weight [8], hybridization features and more.

First hybridization made by Angeline in 1998. This approach changes the bad particles velocity and position values by principle of tournament selection [9]. Another hybrid algorithm is NichePSO [10], this hybrid was developed using CGPSO [11]. This approach uses GA's techniques and divide population into sub populations and trains them with Kennedy's "cognition only model" [12]. Another hybrid algorithm proposes made by Løvebjerg in 2002. In this approach, GA, PSO and hill climbing algorithm were used. Particles select one of these algorithms for the optimization process. This model was called life cycle [13]. In 2003, Higashi and Iba combined Gaussian Mutation with PSO's velocity vector and position update formulations [14],[15].

2.2 Genetic Algorithm

Genetic algorithm was introduced first in 1975 by Holland [16]. Genetic algorithm is an evolutionary algorithm that inspired from biological mechanism. In this algorithm, each individual is called gene and the collection of these genes are called population. This algorithm contains two special steps. First one is crossover step that two genes are combined to generate a new one. Second one is mutation step that updates different parts of the generated new gene randomly. With these unique steps, randomly generated first population may convergence to good solution. Since GA's introduce, this algorithm is very successful for optimization problems. Like PSO, genetic algorithm was also changed through the time. Gene selection mechanism [17], crossover operators [18]-[20] and mutation operators [21] were introduced [22]-[25].

2.3 Breeding Swarms Algorithm

This algorithm proposed by M. Settles and T. Soule in 2005 [5]. In this algorithm, the population divided by predefined a constant value of breeding ratio. First part of population works exactly like PSO the other part of the population is discarded and replaced by GA. The population handling mechanism makes this algorithm is more robust to stuck on the local extrema. In this algorithm crossover operator called Velocity Propelled Averaged Crossover (VPAC) operator, Gaussian Mutation operator and Tournament Selection scheme have been used. Mathematical expression of VPAC given in equation (3)

$$\begin{aligned} c_1(x_i) &= \frac{p_1(x_i) + p_2(x_i)}{2.0} - \varphi_1 p_1(v_i) \\ c_2(x_i) &= \frac{p_1(x_i) + p_2(x_i)}{2.0} - \varphi_2 p_2(v_i) \end{aligned} \quad (3)$$

where, $c_1(x_i)$ and $c_2(x_i)$ are the positions of child 1 and child 2 at i^{th} dimension. $p_1(x_i)$ and $p_2(x_i)$ are the positions of the parent 1 and parent 2 at i^{th} dimension. φ is a random value generated in the range [0.0,1.0]. $p_1(v_i)$ and $p_2(v_i)$ are the velocity vectors of parent 1 and parent 2 at i^{th} dimension. Newly generated particles inherit their parent's velocity vectors and personal best values. According to [5], proposed work yielded slightly better performance and this algorithm is faster than GA and PSO.

2.4 Chaotic Systems

Chaos theory is a study field in mathematics. It is based on unpredictability of nature. In this theory, small changes in the initial conditions of dynamic system can make a very different outcome. It is called butterfly effect and this theory was summarized by Edward Lorenz.

In the recent studies, random number sequences are created by chaotic number generator and in some occasions it yielded better results. In [26] chaotic sequences were used in evolutionary algorithms for

performance improvement. Chaotic sequences in genetic algorithms were introduced in [27] and chaos embedded particle swarm algorithms were studied in [28].

3. Proposed Work

In related works hybrid versions of PSO gets better results than original PSO and the algorithm that hybridized. The efficiency of PSO depends on aimed random search mechanism. Aiming part of the mechanism can be achieved by personal best and global best values in equation (2) and random search part can be achieved by random values in equation (2).

PSO is hybridized with GA to improve the efficiency of PSO so that Breeding Swarm (BS) model were introduced [4], however they have some deficiencies. In genetics, to generate a new individual at least two parents are required. These two parents combine their DNA's and generate a new individual. In the combining step, half of the DNA from each parent combine with each other so that new individual inherits both parents' features. In real world, almost all the time one parent become dominant and other become recessive in this step Because of that new born individual generated from same parents were dissimilar. But in the VPAC operator's equation, this diversity cannot be modelled. Hence newly generated two child have little difference to each other. To simulate this diversity, we changed VPAC operator to VPCC operator. Mathematical expressions of VPCC operator is given in equation (4)

$$\begin{aligned}c_1(x_i) &= m_1 * p_1(x_i) + (1 - m_1) * p_2(x_i) - m_2 p_1(v_i) \\c_2(x_i) &= m_3 * p_1(x_i) + (1 - m_3) * p_2(x_i) - m_4 p_2(v_i)\end{aligned}\quad (4)$$

where m_1, m_2, m_3, m_4 are the random numbers generated from chaotic random number generator, the other parameters are same as VPAC parameters.

In chaotic number generator, we used logistic map because of its simplicity. Definition of the logistic map is given in equation (5)

$$\begin{aligned}map &= 4 * z * (1 - z) \\m_i &= (0.5 * r_1) + (0.5 * map)\end{aligned}\quad (5)$$

where m_i is the i^{th} random value in equation (4), z and r_1 are the random value generated by compiler's random number generator.

Scheme of the BS algorithm's creating new children is very simple. In equation (3), simply adds parent particles' values and divides to two after that subtracts this result from parent particles' velocity. In late stages of original BS algorithm, all particles' personal best values would be similar to each other also their velocity would be similar because they tend to go same point which is global best value.

Our approach in equation (4) uses four different random number created from equation (5) which is the equation of logistic map. This approach divides parent particles' values randomly and adds to each other, after that it subtracts this result from randomly multiplied parent particle's velocity. Hence the children created from four different chaotic random number, even the velocity of parents are very similar, created c_1 and c_2 would be very different from each other. By mean of this diversity, particles can free themselves from local optimum points. Because of the algorithm's PSO part does the global search and the GA part does the local search, proposed approach only affects algorithm local search capability. Our approach changes only crossover part of the BS algorithm. It benefits from randomness of chaotic systems and combines this randomness with crossover operation. Pseudo code of the original BS algorithm has given in Algorithm 1.

Since the only thing change was the crossover operation, original algorithm and modified algorithm have similar working time. There are only a few milliseconds difference between two algorithms.

The BS algorithm with new proposed crossover operator VPCC were tested on multimodal and unimodal benchmark functions. Social parameters c_1 and c_2 are set to 2.0, population size set to 125. We used Gaussian Mutation in both PSO and BS algorithms. Inertia weight is linearly decreased from 0.9 to 0.2, mutation rate was set to 1/dimension. All used parameters are shown in Table 1.

Algorithm 1 Pseudo Code of BS Algorithm

1	Initialize N population
2	Evaluate fitness value for each particle Sort the particles best to worst according to their fitness value
3	Discard worst N*breeding ratio particles Use PSO velocity updates on not discarded particles
4	Generate new particles by crossover and mutation from updated particles to replace discarded particles
5	Use PSO position update on not discarded particles
6	Use PSO position update on not discarded particles
7	Until stopping criteria go to line 2

Table 1 Parameters of PSO and BS

Parameter	PSO	BS
Population Size	125	125
Selection Scheme	-	Tournament
Tournament Size	-	3
Mutation Rate	-	1/Dimension
Mutation Variance	-	1.0 to 0.1
Social Parameters (c_1 & c_2)	2	2
Inertia Weight	0.9 to 0.2	0.9 to 0.2

4. Results

4.1 Test Problems

We used one unimodal and three multimodal benchmark functions for tests. These benchmark functions are popular and used in several studies [5],[9],[11],[14],[28].

First function is Ellipsoidal function which is unimodal. This function is described in equation (6).

$$f_1(x) = \sum_{i=1}^n ix_i^2 \quad (6)$$

Second function is Rosenbrock function which is multimodal. This function is described in equation (7).

$$f_2(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (7)$$

Third function is Griewank function which is multimodal. This function is described in equation (8).

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (8)$$

Fourth function is Ackley function which is multimodal. This function is described in equation (9).

$$f_4(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} \quad (9)$$

These functions initialization ranges are shown in Table 2.

Table 2 Initialization Range of Benchmark Functions

Function	Initialization Range
Ellipsoidal Function	(-100 to 100)
Rosenbrock Function	(-30 to 30)
Griewank Function	(-600 to 600)
Ackley Function	(-32,768 to 32,768)

All test functions were run with 10 and 20 dimension settings. For 10 dimension algorithms were run for 1000 iteration and for 20 dimensions algorithms were run for 2000 iteration. Each function was run for 15 times.

4.2 Results

PSO, BS and BS with VPCC algorithms results on benchmark functions are given in Table 3.

Table 3 Test Results of PSO, BS and BS with VPCC

Function	Dimension	PSO Mean Best (Std-Dev)	BS Mean Best (Std-Dev)	BS with VPCC Mean Best (Std-Dev)
Ellipsoidal Function	10	0,125176 (0,39708)	6,33E-08 (-2,26E-07)	2,42E-11 (8,98925E-11)
	20	6,115002 (9,104639)	0,744034 (0,789478)	4,2238E-05 (6,65888E-05)
Rosenbrock Function	10	4,778433 (2,487662)	2,110912 (2,352057)	0,80326 (1,111899)
	20	13,09686727 (13,8583444)	10,65285311 (2,23640593)	9,502027 (5,706486)
Griewank Function	10	0,004367103 (0,005704754)	0,063425016 (0,098349987)	0,02798 (0,057941)
	20	0,183087907 (0,229020138)	0,197518884 (0,191952133)	0,182558 (0,185198)
Ackley Function	10	1,325490902 (0,914628641)	0,822298525 (0,921644178)	2,3E-07 (7,17E-07)
	20	4,34005 (1,172947)	1,755973 (0,736546)	1,691745 (0,836093)

In every test cases the BS with VPCC algorithm reached the near global minimum point in a majority of trials. For the minority of the trials BS with VPCC algorithm has stuck on local minimums. In the other hand, most of the trials PSO and BS has stuck on local minimums.

Table 3 shows that Breeding Swarm algorithm with VPCC yielded better results than original PSO algorithm and BS algorithm. In 20 dimension problems BS with VPCC yielded slightly better performance. But in 10 dimension problems BS with VPCC yielded clearly better performance except in the Griewank function.

Figures 1, 2, 3, 4 shows the results of each algorithms performance on problems with 10 dimensions and 5, 6, 7, 8 shows the performance on the problems with 20 dimensions through the iterations.

In general, VPCC added version of BS algorithm outperformed the original BS algorithm in every test cases and outperformed the original PSO algorithm in most of the cases.

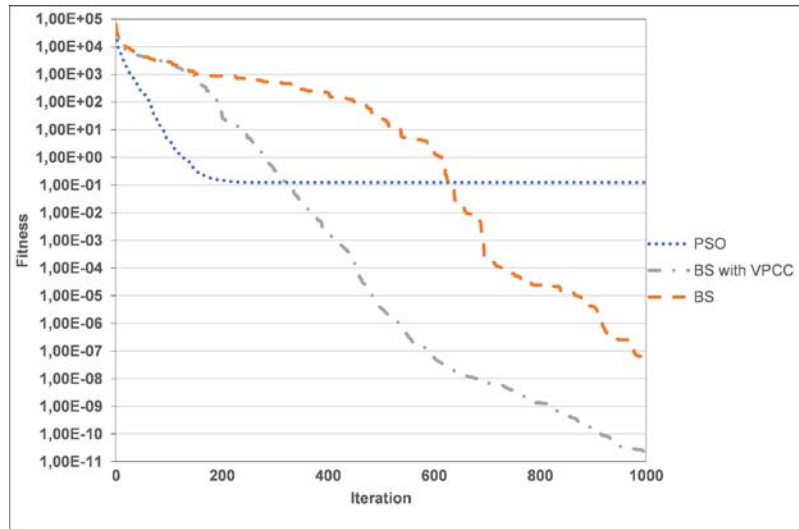


Figure 1 Ellipsoidal Function 10 Dimension

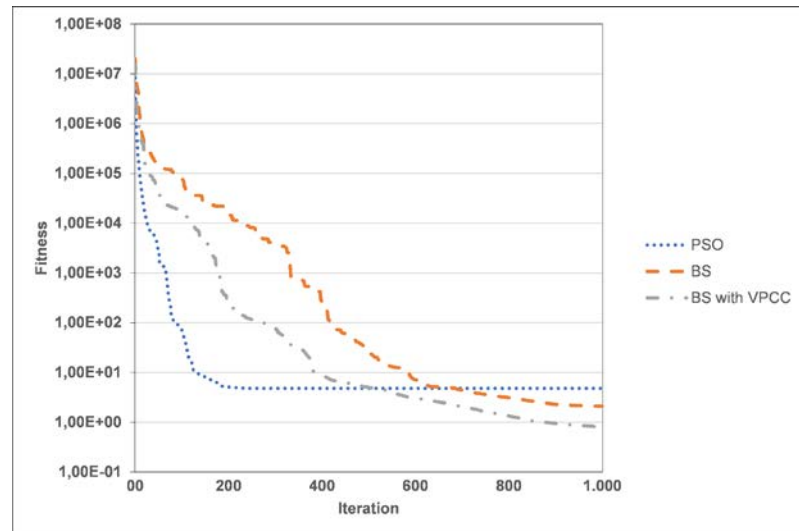


Figure 2 Rosenbrock Function 10 Dimension

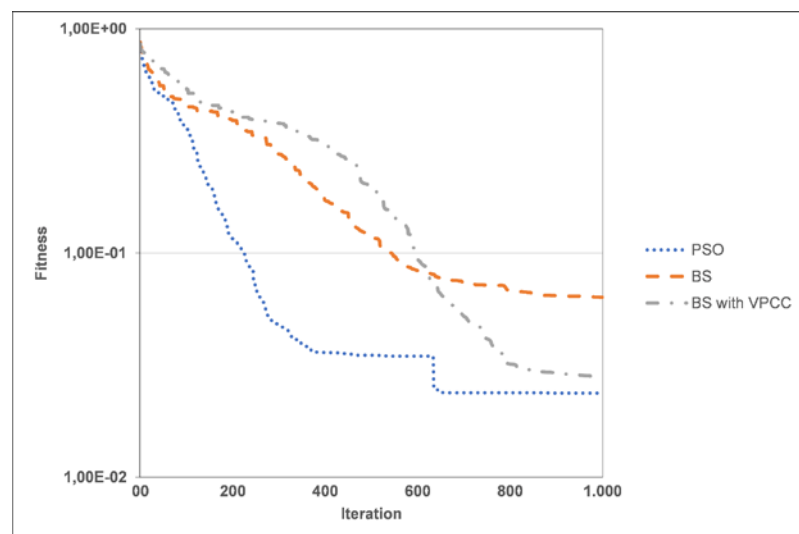


Figure 3 Griewank Function 10 Dimension

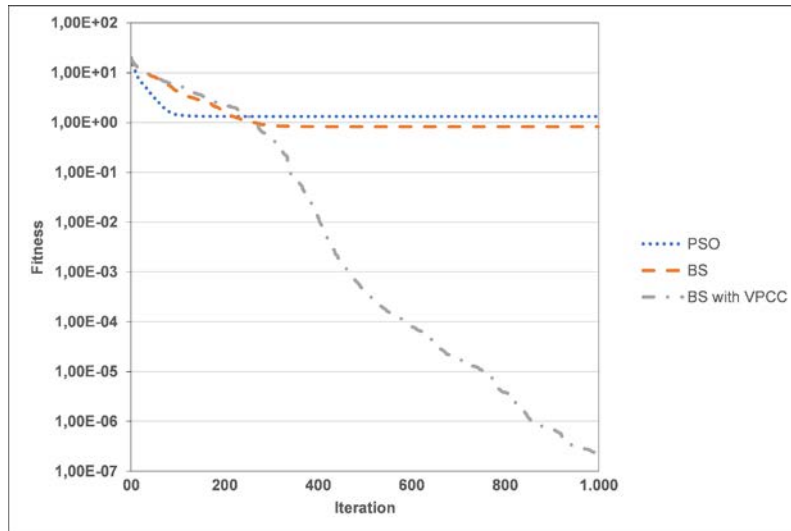


Figure 4 Ackley Function 10 Dimension

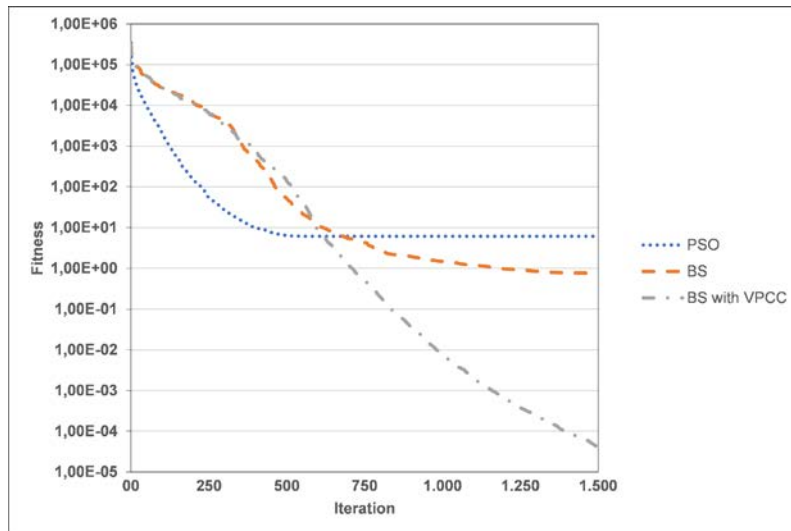


Figure 5 Ellipsoidal Function 20 Dimension

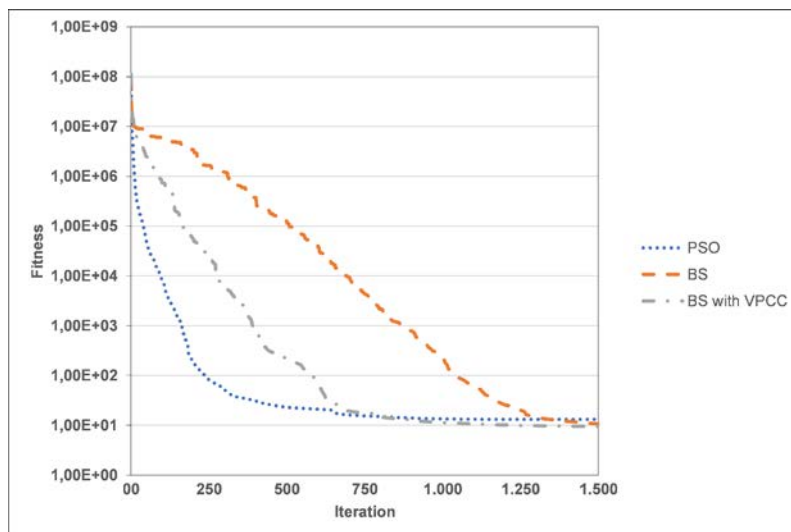


Figure 6 Rosenbrock Function 20 Dimension

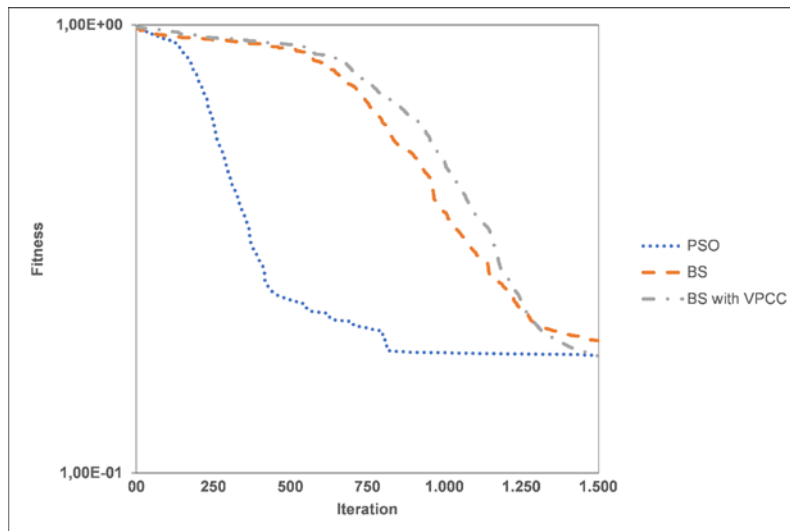


Figure 7 Griewank Function 20 Dimension

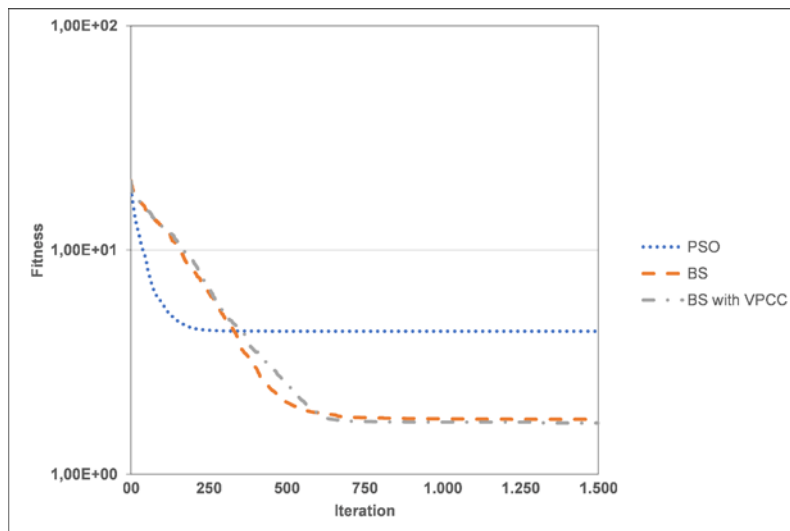


Figure 8 Ackley Function 20 Dimension

5. Conclusion

According to results, BS with VPCC algorithm is competitive with both PSO and BS algorithms. With the power of VPCC operator BS algorithm has overcome the early convergence problem and searched the entire search space. Because of the chaotic random number generator's effect on the crossover operator, the diversity of the population is improved. Therefore, the probability of finding the global best solution increases. In our research we used logistic map function for chaotic map for its simplicity. More complex chaotic maps might improve the randomness and gave better results but since the map complexity increases working time of the algorithm might be increase.

For future research, the VPCC operator which studied here or other operators or the other processes of PSO or GA can be combined with chaotic number generators. Different chaotic maps could be investigated because different maps might give different result with same algorithm.

References

- [1] Q. Liu, X. Li, H. Liu and Z. Guo, "Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art," *Appl Soft Comput.*, vol. 93, 2020.

- [2] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," *IEEE Int. Conf. Neural Networks*, pp. 1942-1948, 1995.
- [3] W. Yanmin, "Optimization of Wireless Sensor Network for Dairy Cow Breeding Based on Particle Swarm Optimization," *Int. Conf. Intell. Trans. Big Data & Smart City (ICITBS)*, pp. 524-527, 2020.
- [4] Y. Özger, M. Akpınar, Z. Musayev and M. Yaz, "Electrical Load Forecasting Using Genetic Algorithm Based Holt-Winters Exponential Smoothing Method," *Sakarya University Journal of Computer and Information Sciences.*, vol. 3, no. 2, pp.108-123, 2019.
- [5] M. Settles and T. Soule, "Breeding swarms: a GA/PSO hybrid," *ACM Conf. Genetic and Evol. Comput. (GECCO '05)*, pp. 161-168, 2005.
- [6] H. R. Vanamala and D. Nandur, "Genetic Algorithm and Chaotic Maps based Visually Meaningful Image Encryption," *TENCON 2019 - 2019 IEEE Region 10 Conf. (TENCON)*, pp. 892-896, 2019.
- [7] R.C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *IEEE Congr. Evol. Comput.*, pp. 84-88, 2000.
- [8] R. Eberhart and Y. Shi, "A Modified Particle Swarm Optimizer," *IEEE World Cong. Comput. Intel.*, pp. 69-73, 1998.
- [9] P.J. Angeline, "Using selection to improve particle swarm optimization," *IEEE World Cong. Comp. Intel.*, pp. 84-89, 1998.
- [10] R. Brits, A.P. Engelbrecht and F. van den Bergh, "A niching particle swarm optimizer," *Proc. Sim. Evol. Learn. SEAL.*, 2002.
- [11] F. van den Bergh and A.P. Engelbrecht, "A new locally convergent particle swarm optimizer," *IEEE Int. Conf. Syst., Man and Cyb.*, vol. 3, 2002.
- [12] J. Kennedy, "The particle swarm: social adaptation of knowledge," *Proc. of 1997 IEEE Int. Conf. Evol. Comput.*, pp. 303-308, 1997.
- [13] T. Krink and M. Løvebjerg, "The lifecycle model: combining particle swarm optimization, genetic algorithms and hillclimbers," *Conf. Parallel Probl. Solving Nat., 7th — PPSN VII*, pp. 621-630, 1997.
- [14] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," *IEEE Swarm Intel. Symp. (SIS)*, pp. 72-79, 2003.
- [15] A. Banks, J. Vincent and C. Anyakoha, "A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Nat. Comput.*, vol. 7, pp. 109-124, 2007.
- [16] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence :2nd ed.* University of Michigan Press, 1992.
- [17] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A (Amsterdam, Neth.)*, vol. 391, pp. 2193-2196, 2012.
- [18] Y. Kaya, M. Uyar and R. Tekin, "A Novel Crossover Operator for Genetic Algorithms: Ring Crossover," *arXiv preprint arXiv:1105.0355*, 2011.
- [19] M.J. Varnamkhasti, L.S. Lee, M.R.A. Bakar and W.J. Leong, "A Genetic Algorithm with Fuzzy Crossover Operator and Probability," *Adv. Oper. Res.*, vol. 2012, Article ID 956498, 2012.
- [20] D. Vrajitoru, "Crossover improvement for the genetic algorithm in information retrieval," *Inf. Process. Manage.*, vol. 34, pp. 405-415, 1998.
- [21] M. Srinivas, "Adaptive probabilities of crossover and mutation in genetic algorithms", *IEEE Int. Conf. Syst. Man. Cyb.*, vol. 24, pp. 656-667, 1994.
- [22] I. Abuiziah and N. Shakarneh, "A Review of Genetic Algorithm Optimization: Operations and Applications to Water Pipeline," *Int. J. of Math. Comput. Phys. Quan. Eng.*, vol. 7, pp.136-147, 2013.
- [23] R. Sivaraj and T. Ravichandran, "A Review of Selection Methods in Genetic Algorithm," *Int. J. Eng. Sci. Technol. (IJEST)*, vol. 3, pp. 3792-3797, 2011.
- [24] N. M. Razali and J. Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP," *Proc. World Cong. Eng.*, vol. 2, 2011.

- [25] O. Abdoun and J. Abouchabaka, "A Review of Selection Methods in Genetic Algorithm," *Int. J. Comp. App. (IJCA)*, vol. 31, pp. 49-57, 2011.
- [26] R. Caponetto, L. Fortuna, S.Fazzino and M.G. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 289-304, 2003.
- [27] L. J. Yang and T. L. Chen, "Application of Chaos in Genetic Algorithms," *Commun. Theor. Phys.*, vol. 38, pp. 168-172, 2002.
- [28] B. Alatas, E. Akin and A.B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons Fractals*, vol. 40, pp. 1715-1734, 2009.