# GUARANTEED COVERAGE PARTICLE SWARM OPTIMIZATION USING NEIGHBORHOOD TOPOLOGIES

Prof. Mrs. R.A.Pimpale
Priyadarshini Institute of Engineering and Technology, Nagpur

**Abstract:**

The key behind the research represent in this paper is to understand the behavior of the particle swarm algorithm. This study proposes guaranteed convergence Particle Swarm Optimizer (GCPSO) with various topologies. The proposed GCPSO has evaluated the topology such as GBest, LBest, and Von Neumann Topology. It would be the most appropriate for different benchmark function such as Quadratic, Rosenbrock , Rastrigin, Griewank, Ackley, Shaffer's f6 for guaranteed coverage using GCPSO, faster convergence and to find better local optima (local PSO) for large no of particle swarm for unimodel and multimodel functions. There are several parameters that need to be defined in order to successfully use on PSO and GCPSO to solve a given problem.

**Keywords:** GCPSO, GBest, LBest, and Von Neumann Topology etc.

### 1.0 Introduction

Optimization is the process of trying to find the best possible solution to an optimization problem within a reasonable time limit. Particle swarm optimization is a population based stochastic optimization method, which was first introduced by James Kennedy and Russell C. Eberhart. This technique was inspired by social behavior of bird flocking and fish schooling. This kind of social optimization can be simulated by a particle swarm.

The PSO algorithms is iterative and involves particles considering the behavior of swarms in the nature, such as birds, fish, etc. developed the PSO algorithm. The PSO has particles driven from natural swarms with communications based on evolutionary computations. PSO combines self-experiences with social experiences. In this Algorithm, a candidate solution is presented as a particle. It uses a collection of flying particles (changing solutions) in a search area (current and possible solutions) as well as the movement towards a promising area in order to get to a global optimum.

A problem is given and its' fitness function (way to evaluate a proposed solution). It is defined a social network by setting to each individual a neighborhood. Then a population of individuals is randomly initialized and is set to be the problem solutions.

### 2.0 PSO Setting

The equations for updating the position and the velocity of a particle are the following:

$$x^i_{K+1} = x^i_{K} + V^i_{K+} \quad (1)$$

$$v^i_{K+1} = \omega V^i_{K+1} + c_1 r_1{}^i_K ( pbest_k - x^i_K ) + c_2 r_2{}^i_K (gbest_k - x^i_K) \quad (2)$$

Where,

$k$ is a unit pseudo time increment (iteration number)

$\omega$ is a real number

c1,c2 are real numbers usually $c1 = c2 = 2$.

$r_1{}^i_K$ , $r_2{}^i_K$ two real random numbers between 0 and 1.

The purpose of this paper is to determine whether the GCPSO is benefited from the effect of neighborhoods. This paper present the results of the experiments conducted and compared with the standard PSO. Three neighborhood topologies are applied to GCPSO and standard PSO tested on a variety of benchmark function. This paper represented results of experiments conducted and compares these results with standard PSO.

In this study, we propose GCPSO with various neighborhood topologies. We apply GC-PSO to the various network topologies as lbest, gbest and vonneumman. GC-PSO with various topologies is applied to eight test functions which are unimodal and multimodal. We investigate their behaviors and evaluate what kind of topology would be the most appropriate for each function

**2.1 The behavior of the algorithm.**

To this goal, it was decided to create a large number of sociometries and to evaluate their effect on the performance of the algorithm using a well-known series of benchmarks used in numeric optimization. These graphs were tested on following functions:

**3.0 Selected Benchmark Functions**

**3.1 Sphere function:**

This function is very simple. Any algorithm capable of numeric optimization should solve it without any problem. Its simplicity helps to focus on the effects of dimensionality in optimization algorithms. It is nonlinear, unimodal function with its global minimum located at

x = <0. . . 0> with f(x) = 0. This function has no interaction between its variables and gradient information always points toward the global minimum.

$$f(x) = \sum_{j=i}^{n} x_j^{\ 2} \qquad (3)$$

Where

$| x_j | \leq 100.0$ and f(x*)= f1(0,0…….,0)=0.0

**3.2 Quadratic Function**

Study of optimization problem is simplest for optimization of quadratic form, it distinguish maxima and minima in economic optimization problem. A standard quadratic optimization problem (StQP) consists of finding the largest or smallest value of a (possibly indefinite) quadratic form.

$$f(x) = \sum_{i=1}^{n} \left( \sum_{j=i}^{i} x_j \right)^2 \qquad (4)$$

**3.3 Ackley Function:**

Ackley has an exponential term that covers its surface with numerous local minima. The complexity of this function is moderated. An algorithm that only uses the gradient steepest descent will be trapped in local optima, but any search strategy that analyses a wider region will be able to cross the valley among the optima and achieve better results. In order to obtain good results for this function, the search strategy must combine the exploratory and exploitative components efficiently.

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n}} \sum_{j=1}^{n} xj\ 2\right) - \exp\left(\frac{1}{n} \sum_{k=1}^{n} \cos 2\pi xk\right) + 20 + e \qquad (5)$$

Where

$| x_j | \leq 32.0$ and f(x*)= f (0,0…….,0)=0.0

**3.4 Rosenbrock Function:**

Rosenbrock function is a non-convex function used as a performance test problem for optimization algorithms introduced by Howard H. Rosenbrock in 1960.

The global optimum is inside a long, narrow, parabolic shaped flat valley. Since it is difficult to converge the global optimum, the variables are strongly dependent, and the gradients generally do not point towards the optimum, this problem is repeatedly used to test the performance of the optimization algorithms.

$$f_1(x) = \sum_{j=1}^{n-1} [100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2] \qquad (6)$$

Where

$|x_j| \leq$ 30.0,

And

$$f_1(x^*) = f_1(1, 1, ..., 1) = 0.0$$

**3.5 Rastrigin Function:**

The Rastrigin function is a non-convex function used as a performance test problem for optimization algorithms. It is a typical example of non-linear multimodal function. It was first proposed by Rastrigin as a 2-dimensional function and has been generalized by Mühlenbein. This function is a fairly difficult problem due to its large search space and its large number of local minima. The function is multimodal. The locations of the minima are regularly distributed. The difficult part about finding optimal solutions to this function is that an optimization algorithm easily can be trapped in a local optimum on its way towards the global optimum.

$$f_1(x) = \sum_{j=1}^{n} [x_j^2 - 10cos(2\pi x_j) + 10] \quad (7)$$

Where

$|x_j| \leq$ 5.12,

And

$$f_1(x^*) = f_6(0, 0, ..., 0) = 0.0$$

**3.6 Schwefel Function**

The surface of Schwefel function is composed of a great number of peaks and valleys. The function has a second best minimum far from the global minimum where many search algorithms are trapped. Moreover, the global minimum is near the bounds of the domain.

$$f_2(x) = -\sum_{j=1}^{n} (x_j \sin(\sqrt{|x_j|})) \quad (8)$$

**3.7 Griewank-**

This function is strongly multi-modal with significant interaction between its variables, caused by the product term. This function has the interesting property that the number of local minima increases with dimensionality. However, the influence of the product term also diminishes dramatically in these circumstances.

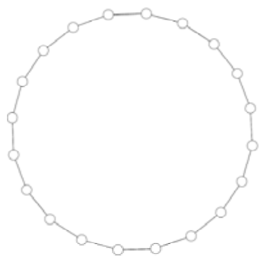

Fig. 4.1: lbest Topology

**4.0 Selected Topology Used**

The most important discovery was a significant difference between the different topologies. To make sure that these results were not only due to chance, we compared both measures and correlation was very high, indicating that there really was a relationship between topologies and performance.

**4.1 Neighborhood Topologies**

In the original PSO, two different kinds of neighborhoods were defined for PSO:

- In the lbest swarm topology include only K nearest neighbors this form ring like topology known as circle topology, only a specific number of particles (neighbor count) can affect the velocity of a given particle. The swarm will converge slower but can locate the global optimum with a greater chance.

The Von Neumann topology connects the particle in grid network structure where each particle is directly connected to four neighbors from above, below, to the left, to the right.

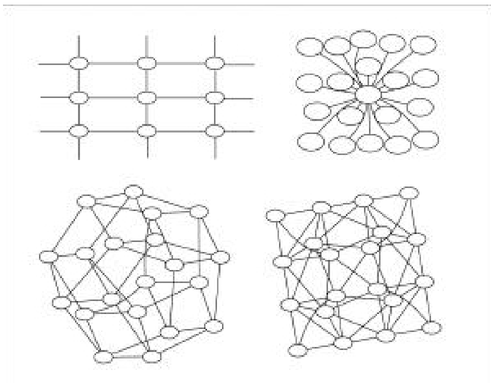

Fig. 4.2 Von Neumann topology

In the gbest swarm topology, all the particles are neighbors of each other; thus, the position of the best overall particle in the swarm is used in the social term of the velocity update equation. It is assumed that gbest swarms converge fast, as all the particles are attracted simultaneously to the best part of the search space. However, if the global optimum is not close to the best particle, it may be impossible to the swarm to explore other areas; this means that the swarm can be trapped in local optima.

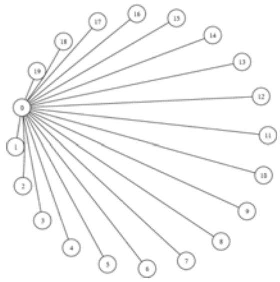

Fig 4.3: gbest topology

The swarms with the different neighborhoods were used to optimize several known functions and a statistical analysis was carried out over several dependents: best result at 200 iterations, number of iterations needed to meet the stopping criteria and whether the stopping criteria were met or not.

**4.2 Methodology**

Here methodology implied is practical, The PSO implementation was written in Eclipse. To evaluate the performance of optimization algorithms in a given problem, it is necessary to gather enough data to perform a statistical analysis with a certain degree of accuracy. It is important to have enough statistical power to be able to perform any kind of statistical inference.

According to the central limit theorem, the mean of a given population can only be approximated by the normal distribution when the sample size is 200000 (particles).

**5.0 Experiments and Discussion**

**5.1 PSO parameter**

| Parameter | Value |
|---|---|
| Particles | 200000 |
| c1,c2 | 1.49 |
| Interia weight(w) | 0.72 |
| fc | 5 |
| k | 2 |

Table 5.1: PSO parameter

Maximum fitness evaluation is 200000 (particles) and iteration is 200 for particle swarm optimization

| Function | PSO Topology | | |
|---|---|---|---|
| | gbest | lbest | Vn |
| ackley-pso | 4.623311993774259 | 7.549516567451 | 0.9313046018168 |
| griewank–pso | 0.0609815280001084 76 | 0.0 | 0.0 |
| quadric -pso | 5.9479459095650 | 2.0157562440 | 5.363409340669811 |
| rastrigin -pso | 60.69236105910704 | 70.64198213088629 | 72.63185481366625 |
| rosenbrock -pso | 0.0045108100377362 | 1.5164723107417424 | 0.06433960679577141 |
| schwefel -pso | 3671.7607784174506 | 5271.337754312777 | 4658.705304481281 |
| spherical -pso | 1.21936276871095 E-105 | 2.2463623964076037 E-93 | 1.847316112731218E-117 |

Table 5.2: PSO topology used for different function

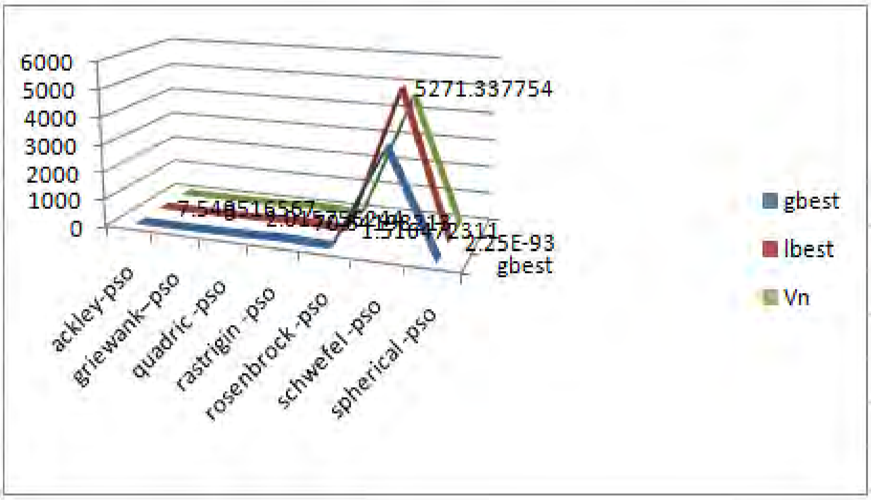


Fig. 5.1 PSO topology graph used for different function

Maximum fitness evaluation is 200000 (particles) and iteration is 200 for Guaranteed coverage particle swarm optimization

| Function | GCPSO Topology | | |
|---|---|---|---|
| | gbest | lbest | Vn |
| ackley-gcpso | 4.21140139 | 7.549516567 | 7.54951657 |
| griewank–gcpso | 0.07 | 0 | 0 |
| quadric –gcpso | 3.667565023 977721E | 2.095924921 | 4.0737437 |
| rastrigin –gcpso | 112.429893 | 71.63692099 | 76.611605 |
| rosenbrock -gcpso | 0.00282709 | 0.317493309 | 0.11205543 |
| schwefel -gcpso | 4797.07 | 5508.683595 | 4106.00722 |
| spherical -gcpso | 1.29E-161 | 5.59E-100 | 2.42E-120 |

Table 5.3: GCPSO Topology values for different functions

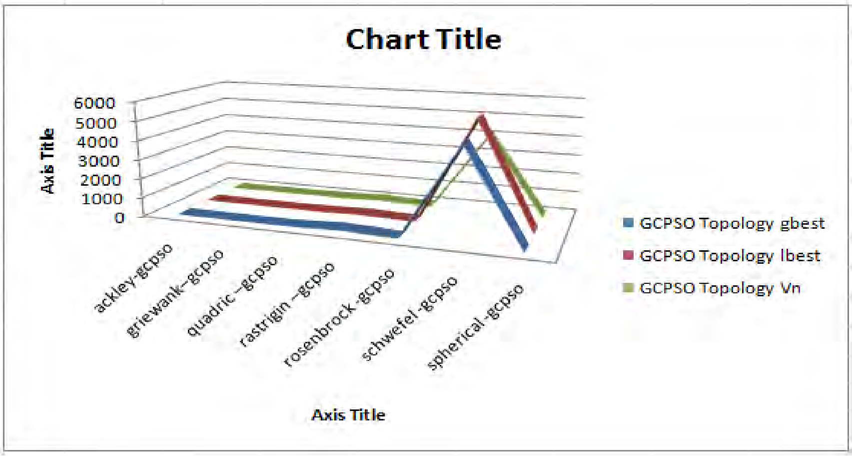


Fig 5.2 GCPSO Topology graph for different functions

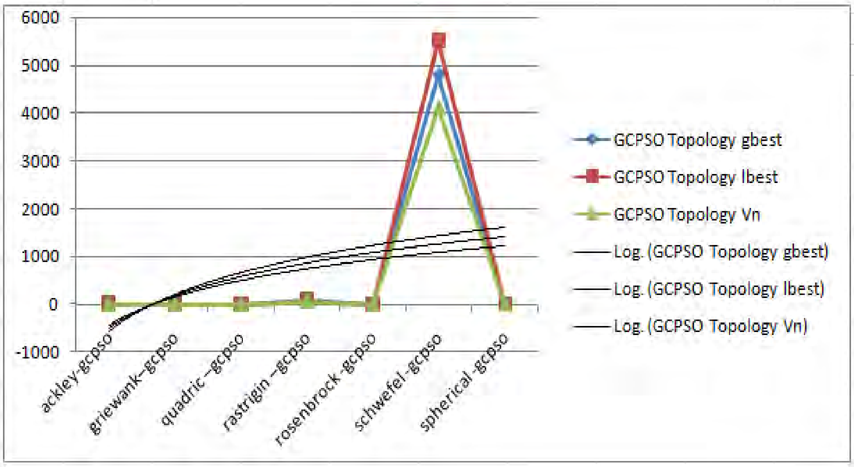


Fig 5.3 GCPSO Topology graph for different functions

The PSO implementation was written in Eclipse. For purposes of comparison, all the simulations use the same parameter settings for the PSO implementation except the inertia weight *w* and maximum velocity allowed. The population size (number of particles) is 200000. The dynamic range for each element of a particle is defined as (-100, 100), that is, the particle cannot move out of this range in each dimension and thus Xmax = 100. The maximum number of iterations allowed is 20000. If the PSO implementation cannot find an acceptable solution within 2000 iterations, it is ruled that it fails to find the global optimum in this run.

**5.2 Combining Results From the Functions**

The goal of this study is to find an algorithm that is able to maintain a good performance over all problems one may wish to solve. One of the common errors observed when demonstrating the capabilities of an algorithm is the use of a set of problems that is not representative of all the difficulties encountered in function optimization. This sometimes hides some of the vulnerabilities of the algorithm. For instance, if no functions are used with an abundance of local minima, it is impossible to measure the robustness of the algorithm to premature convergence.

When using many functions, results are usually presented independently on each of the functions used and there is no methodology to conclude which of the approaches has a good performance over all the functions. However, this considerably complicates the task of evaluating which approach is the best.

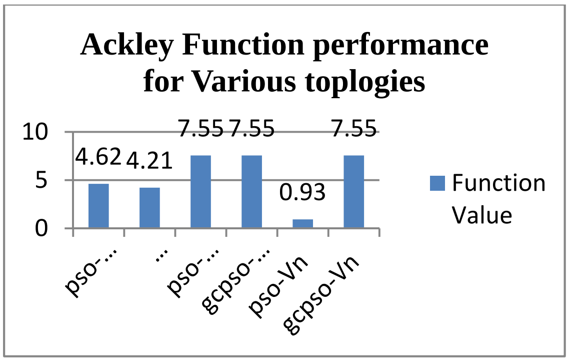


Fig 5.1: Ackley Function performance

In Ackley function it was observed that psolbest, gcpso-lbest and gcpso-vn has same function performance value and pso-vn had a best performance function.

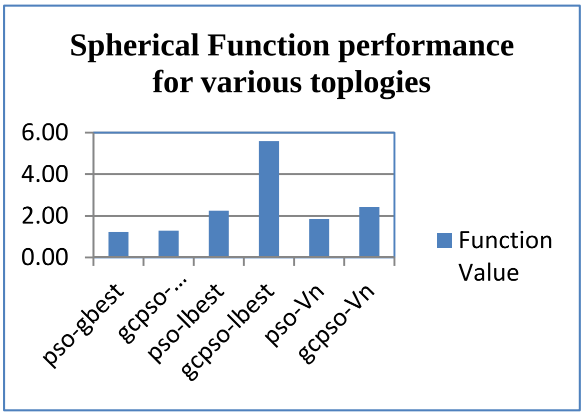


Fig. 5.2: Spherical Function performance

In spherical function Evaluation performance gbest toplogy for pso and gcpso is best topology. An algorithm capable of finding a good result on the Sphere function is definitely a good choice for global optima. So we need to implement it on some data for further processing.

It is not possible to combine raw results from different functions, as they are all scaled differently. For instance, almost any decent algorithm will find a function result less than 0.11 on griewank–gcpso function, but a result of 0.06 on Rosenbrock is considered good

There are several approaches available to combine the function out puts. One that was used in most of the papers (Kennedy and Mendes, 2002; Mendes et al., 2003b; Kennedy and Mendes, 2003; Mendes et al., 2003a; Mendes et al., 2004) was standardization. In this approach, the combination of function outputs is performed by standardizing the results of each function to a mean of 0.0 and standard deviation of 1.0. Results from all the trials for a single function are standardized to the same scale; as all of these problems involve minimization, a lower result is better, and after standardization that means that a negative result is better than average. After standardizing each function separately, we can combine them and find the average for a single condition. A standardized result Xs is computed according to formula, where μ is the average of the values and _ is the standard deviation.

$Xs = X - \mu$

This may seem to depart from what was said in the previous section. When combining the results from different measures, some of the precision is lost. The reason for doing this is to have some way of combining the results obtained in different functions so as to have a way of comparing algorithms among themselves.

If this measure is considered alone, then it should be regarded with great care. It should not be taken as an absolute truth, e.g., as a definite proof that one algorithm is better than another.

**6.0 Conclusions**

In this paper, we have analyzed the impact of the inertia weight and maximum velocity allowed on the performance of PSO. A number of experiments have been done with different inertia weights and different values of maximum velocity allowed. It is concluded that when Vmax is small (<= 2 for the f6 function), an inertia weight of approximately 1 is a good choice, while when Vmax is not small (>= 3), an inertia weight w =

0.8 is a good choice. When we lack knowledge regarding the selection of Vmax, it is also a good choice to set Vmax equal to Xmax and an inertia weight w = 0.8 is a good starting point. Furthermore if a time varying inertia weight is employed, even better performance can be expected.

Even though good experimental results have been obtained in this paper, different benchmark problem has been tested. The selection of the inertia parameter and maximum velocity allowed may be problem-dependent. To fully justify the benefits of selecting parameters as described in this paper, more problems need to be tested. By doing so, a clearer understanding of PSO performance will be obtained.

In unimodel optimization the stability of lbest topologies was poor for PSO. In multimodal optimization, neighborhood is desirable for larger area of search space being explored, it was noticed that VonNeumann topology should display better performance for PSO. Sphere function is definitely a good choice for guaranteed coverage of global minimum particle swarm optimization as well as global optima for particle swarm optimization

### 7.0 References


[1] Daniel N Wilke, "Analysis of the particle swarm optimization algorithm", University of Pretoria, 2005

[2] J. Kennedy and R. C. Eberhart. "Swarm Intelligence". Morgan Kaufmann. 2001

[3] A. P. Engelbrecht, "Fundamentals of Computational Swarm Intelligence", Wiley, 2005.

[4] Elisa Valentina Onet* and Ecaterina Vladu**," Nature inspired algorithms and artificial intelligence", Department of Computer Science, University of Oradea, Faculty of Electrical Engineering and Information Technology,

[5] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pages 1958-1962, 1999.

[6] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 2002.

[7] D. J. Watts. Small Worlds: The Dynamics of Networkds Between Order and Randomness. Princeton University Press, 1999.

[8] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. Nature, (393):440-442, 1998.

[9] X. Li. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. Lecture Notes on Computer Science, 3102:105-116, 2004.

[10] Shi, Y. and Eberhart, R. C. (1998c). Parameter selection in particle swarm optimization. In Evolutionary Programming VII, pages 591_ 600. Lecture Notes in Computer Science 1447.

[11] Shi, Y. and Eberhart, R. C. (2000). Experimental study pf particle swarm optimization. In The 4th World Multiconference on Systemics, Cybernetics and Informatics

[12] Particle Swarm Optimization Methods, Taxonomy and Applications, Davoud Sedighizadeh and Ellips Masehian, International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December 2009 1793-8201

[13] A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Dervis Karaboga · Bahriye Basturk, J Glob Optim (2007) 39:459–471 DOI 10.1007/s10898-007-9149-x

[14] Ackley D. H. Ackley. "A connectionist machine for genetic hillclimbing". Boston: Kluwer Academic Publishers, 1987.

[15] Rosenbrock, H.H. (1960). "An automatic method for finding the greatest or least value of a function". The Computer Journal **3**: 175–184. DOI:10.1093/comjnl/3.3.175. ISSN 0010-4620.

[16] A. Törn and A. Zilinskas. "Global Optimization". Lecture Notes in Computer Science, Nº 350, Springer-Verlag, Berlin, 1989.

[17] H. Mühlenbein, D. Schomisch and J. Born. "The Parallel Genetic Algorithm as Function Optimizer ". Parallel Computing, 17, pages 619–632, 1991.