A Survey On Particle Swarm Optimization Algorithms

Bhargav Joshi
Department of Computer Science and Software Engineering
Auburn University
Auburn, USA
bhargav.csse@gmail.com

Armin Khayyer
Department of Computer Science and Software Engineering
Auburn University
Auburn, USA
azk0100@auburn.edu

Ye Wang
Department of Computer Science and Software Engineering
Auburn University
Auburn, USA
yewang@auburn.edu

***Abstract*—This paper surveys on particle swarm optimization (PSO) approach to genetic algorithms. When four variants of PSO algorithms were subjected to solve Schaffer's F6 function, they were able to reach to the desired fitness, but they showed difference in the performance. This paper contains the performance evaluation and statistical analysis of class divisions through class equivalence derived using the f-test and t-test. PSO algorithms that use star topology shows better performance than PSO algorithms that use ring topology while solving the Schaffer's F6 function.**

Keywords—Genetic Algorithm, Class Equivalence, crossover

I. Introduction

Swarm optimization is an optimization technique used in evolutionary computations. Typically, several swarm optimization methods are genetic algorithms (GA) such as the particle swarm optimization (PSO) [1], the ant colony optimization (ACO), the differential evolution (DE), and the bacterial foraging optimization (BFO), etc. These swarm optimization methods utilize the cooperation and the competition among the swarm to find the global optimum. This paper focuses on the particle swarm optimization method for optimizing the Schaffer's F6 function.

The particle swarm optimization (PSO) method is a simple but powerful search technique inspired by the concept of social interaction of solving problems [2]. In particle swarm optimization, an individual is defined as a particle and the number of more than one particle defined formulate a swarm. Each particle is mainly composed of its current position, its velocity towards the local optima and the best solution found throughout iterations. As sociobiology expert E. 0. Wilson [3] has written, in reference to fish schooling, "In theory at least, individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpredictably distributed in patches" (p.209). This statement suggests that social sharing of information among cospeciates offers an evolutionary advantage: this hypothesis was fundamental to the development of particle swarm optimization [1][3]. Therefore, in PSO, a swarm of particles communicate with each other and try to reach the global optima of the function to find the best solution.

II. Methodology

A. Schaffer F6 function

The experiment was performed to solve Schaffer F6 function. Schaffer's F6 function has many local optima around the global optimum in the solution space [−100, 100]. Once getting into the local optimum, the particles hardly get out of local optima in the solution space [4]. The formula of Schaffer's F6 function is mathematically described as shown in figure 1.

$$\mathrm{F6}(x, y) = -(0.5 + \frac{(sin\sqrt{x^2 + y^2})^2 - 0.5}{(1 + 0.001(x^2 + y^2))^2})$$

Figure 1: Schaffer's F6 function

B. Particle Swarm Optimization (PSO) Algorithm

For computational systems, an algorithm has been developed to replicate the swarm behavior observed in the nature. Each individual in the swarm is defined as particles and each particle is initialized with (1) the x-vector: current position of the particle in the search space, (2) the p-vector: the location of the best solution found by the particle, (3) the v-vector: a gradient that suggests in which direction the particle will travel if not disturbed, (4) x-fitness: the fitness of x-vector, and (5) p-fitness: the fitness of p-vector [2].

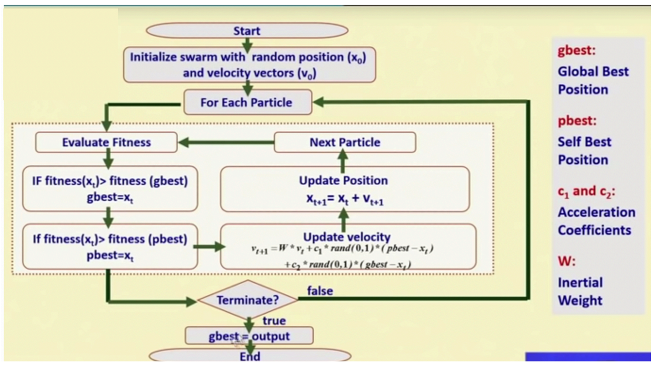
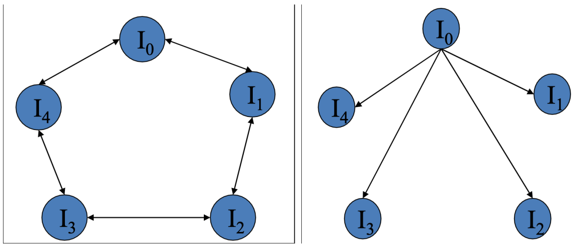


Figure 2 PSO algorithm flowchart [5]

Figure 2 shows a flowchart to implement particle swarm optimization. As per the flowchart, a swarm of particles are initialized with random position x-vector and velocity v-vector. Through iterative method, the algorithm evaluates the fitness of each particle. The local best and the global best fitness are updated as per the conditions shown in figure 2. Based on the local best fitness, the algorithm updates the velocity of the particle and updates its new position using the new velocity. The iteration terminates when satisfactory global best fitness is achieved.

C. Topology

Topology in PSO describes how particles are connected with each other. Topology structure can have a profound impact on the performance of PSO. Including too many particles in each neighborhood can have a negative impact on exploration and may lead to premature convergence [6]. The common topologies used in the literature are (1) ring topology and (2) star topology [2][1].



(a) Ring Topology **(b) Star Topology**

Figure 3 Basic Topologies [2]

Figure 3(a) represents the ring topology. In his topology, small local neighborhoods are defined for each particle. The social component reflects information exchanged within the neighborhood of the particle, reflecting the local knowledge of the environment. With reference to the velocity equation, the social contribution to particle velocity is proportional to the distance between a particle and the best position found by the neighborhood of particles. The velocity is calculated as,

$$V_{ij}(t+1)=V_{ij}(t)+C1R1_j(t)[y_{ij}(t)-x_{ij}(t)]+C2R2_j(t)[\hat{y}_{ij}(t)-x_{ij}(t)]$$

Here $\hat{y}_{ij}$ is the best position, found by the neighborhood of particle i in dimension j.

Figure 3(b) represents the star topology. In this topology, at each iteration, the neighborhood for each particle is the entire Swarm, that is the social component of the particle velocity update reflects information obtained from all the particles in the swarm. the velocity of particle i is calculated as,

$$V_{ij}(t+1)=V_{ij}(t)+C1R1_j(t)[y_{ij}(t)-x_{ij}(t)]+C2R2_j(t)[\hat{y}_j(t)-x_{ij}(t)]$$

Here, in the social component $\hat{y}$ is the global best particle's position, and y_{ij} is the particle's best position so far.

It should be noted that reducing the number of neighbors from the entire swarm to a few particles enables the PSO to avoid getting stuck in local optimum points, since particles in each neighborhood revolve mainly around the local optima found by the particles in that specific neighbor therefore the particles have the chance to probe the entire solution space. It also should be noted that neighborhoods overlap. A particle takes part as a member of a number of neighborhoods. This interconnection of neighborhoods also facilitates the sharing of information among neighborhoods, and ensures that the swarm converges on a single point, namely the global best particle.

D. *Update method*

There are two methods on how the particles are updated [2]:

- Synchronous Update
- Asynchronous Update

The synchronous PSO algorithm is shown in the flowchart of Figure 4. As shown in the algorithm, the particles' pBest and gBest updates are conducted after the fitness of all the particles has been evaluated. Therefore, this version of PSO is known as synchronous PSO (S-PSO). Because the pBest and gBest are updated after all the particles are evaluated, S-PSO ensures that all the particles receive perfect and complete information about their neighborhood, leading to a better choice of gBest and thus allowing the particles to exploit this information so that a better solution can be found. However, this possibly leads the particles in S-PSO to converge faster, resulting in a premature convergence [7].

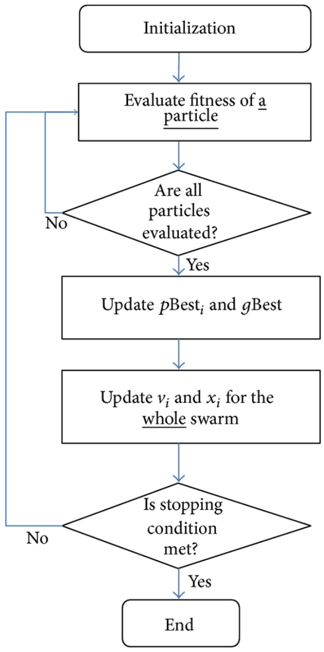


Figure 4 Synchronous Update [7]

Figure 5 shows a flowchart of asynchronous PSO. In Asynchronous PSO the particles are updated based on the current state of the swarm. A particle in A-PSO is updated as soon as its fitness is evaluated. The particle selects gBest using a combination of information from the current and the previous iteration [7].

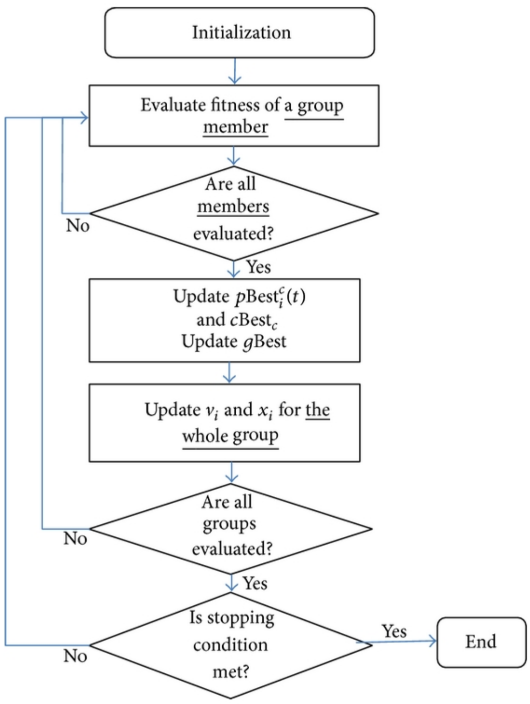


Figure 5 Asynchronous Update [7]

E. Variants of PSO algorithm

The section above describes a general approach towards the particle swarm optimization. Different variants of PSO algorithm have been developed based on the implemented topology and update methods. In this paper, four variants of PSO algorithm are analyzed.

- PSO, Ring Topology (Neighborhood Size = 3), Synchronous Update
- PSO, Ring Topology (Neighborhood Size = 3), Asynchronous Update
- PSO, Star Topology, Synchronous Update
- PSO, Star Topology, Asynchronous Update

III. Experiment

The experiment was conducted to solve Schaffer's F6 function. The variants of PSO algorithm mentioned in the section above were implemented in python3 environment. To obtain worthy results, each variant of the algorithm was run enough times to obtain 30 successful results that solved the function within 4000 function evaluations. Average and variance calculation were done on the results to find out the best performing variant. Student t-tests were done on the other variants with the best performing variant and each variant were categorized into their equivalence classes.

IV. Results

Number of function evaluations each variant took to solve the F6 function for 30 successful runs are posted in table 1.

Run #	**Function Evaluations**			
	Ring, Sync	**Ring, Async**	**Star, Sync**	**Star, Async**
1	302	2158	163	125
2	329	292	113	128
3	1039	1642	131	161
4	240	337	155	115
5	308	481	138	154
6	690	3778	159	128
7	745	104	113	130
8	1589	2902	123	143
9	1893	1853	165	198
10	310	738	132	123
11	292	1977	125	143
12	1522	1330	135	134
13	1571	660	118	183
14	906	218	145	146
15	400	612	129	168
16	321	2977	111	132
17	222	155	117	173
18	508	3158	145	144
19	424	371	145	117
20	322	2044	132	114
21	1514	1174	134	124
22	102	1972	159	169
23	1442	1835	148	155
24	209	373	146	141
25	2147	1183	158	156
26	832	731	155	154
27	3476	1681	144	138
28	1160	3426	139	162
29	306	2555	167	183
30	865	413	179	146

Table 1 Function Evaluations

Algorithm	**Average Function Evaluations**	**Variance**
PSO, Ring Topology, Synchronous Update	866	575860
PSO, Ring Topology, Asynchronous Update	1438	1177241
PSO, Star Topology, Synchronous Update	141	320
PSO, Star Topology, Asynchronous Update	146	469

Table 2 Basic statistics

Algorithm	**Equivalent Class**
PSO, Ring Topology, Synchronous Update	Class 2
PSO, Ring Topology, Asynchronous Update	Class 3
PSO, Star Topology, Synchronous Update	Class 1
PSO, Star Topology, Asynchronous Update	Class 1

Table 3 Class division through t-test

Table 2 shows basic statistical analysis of the results in table 1. Student t-test on the results in table 1 classified each variant into their equivalence classes. Table 3 shows the equivalence class for each variant with respect to the best performing variant. The t-test also showed that there was no statistical equivalence between class 2 and class 3.

V. Conclusions

The results showed that the variation of PSO with star topology and synchronous update was the best performing candidate. The t-test suggested that both variants with star topology showed class one equivalence. This happened because the Schaffer F6 function has many local optima therefore the capability of communicating with each particle in star topology helped achieve the global optima faster. Whereas in the variants of ring topology, it took a while to get to the global optima due to the fact that some of the neighborhoods might have been stuck in the local optima. It is also observable in the statistical analysis that synchronous update method performed better compared to asynchronous update method. In synchronous update method, the particles are updated once all particles' fitness is evaluated. This gives all particles more informed direction towards the global optima because the global best is updated at the end of the fitness evaluation of all particles. Whereas in asynchronous update a particle is updated as soon as the fitness is evaluated. Typically, asynchronous update method should give faster response but in case of Schaffer F6 function having many local optima, misguides the particle towards local optima and its direction could be set towards the local optima that could be away from the global optima.

VI. Breakdown of work

Each author of the paper contributed to their parts. Armin Khayyer developed the base code for implementing PSO, and coded topologies and synchronous update method. Ye Wang coded the asynchronous update method. Bhargav Joshi put together the final code with the functionality of GUI and added feature to collect the successful run of the program (less than 4000 function evaluations) for customizable number of runs. Bhargav Joshi also analyzed the collected data and derived the conclusions for the results from the study. Armin Khayyer provided support materials for the topology literature. Ye Wang found a research paper that explains synchronous and asynchronous update methods.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 2010, vol. 4, no. 3, pp. 1942–1948.

[2] G. Dozier, "Lecture notes on Advarsarial machine learning." Auburn, Alabama, USA, 2019.

[3] E. Wilson, *Sociobiology: The new synthesis*. Belknap Press, Cambridge, hlA, 1975.

[4] J. Liu, P. Li, Xiaoning Ma, and Jiaxun Xie, "Modeling, analysis and simulation on searching for global optimum region of particle swarm optimization," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 813–821.

[5] B. Mahanty, IIT Kharagpur, and NPTEL, "Lecture 38: Particle Swarm Optimization," NPTEL, India, 2018.

[6] D. Sanchez, "DS-PSO : Particle Swarm Optimization with Dynamic and Static Topologies DS-PSO : Particle Swarm Optimization with Dynamic and Static Topologies," 2017.

[7] N. A. Ab Aziz, M. Mubin, M. S. Mohamad, and K. Ab Aziz, "A Synchronous-Asynchronous Particle Swarm Optimisation Algorithm," *Sci. World J.*, vol. 2014, pp. 1–17, 2014.