A Brief Review on Particle Swarm Optimization: Limitations & Future Directions

Shailendra S. Aote
Research Scholer, GHRCE &
Faculty, RGCER, Nagpur
shailendra_aote@rediffmail.com

Dr M M Raghuwanshi
Principal, RGCER

Dr. Latesh Malik
HOD, CSE, GHRCE

Abstract : Particle swarm optimization is a heuristic global optimization method put forward originally by Doctor Kennedy and Eberhart in 1995. Various efforts have been made for solving unimodal and multimodal problems as well as two dimensional to multidimensional problems. Efforts were put towards topology of communication, parameter adjustment, initial distribution of particles and efficient problem solving capabilities. Here we presented detail study of PSO and limitation in present work. Based on the limitation we proposed future direction.

I. INTRODUCTION

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates.

Particle Swarm Optimization (PSO) incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged. PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems, or the problems that can be transformed to function optimization problems. As an algorithm, the main strength of PSO is its fast convergence, which compares favorably with many global optimization algorithms like Genetic Algorithms (GA), Simulated Annealing (SA) and other global optimization algorithms. While population-based heuristics are more costly because of their dependency directly upon function values rather than derivative information, they are however susceptible to premature convergence, which is especially the case when there are many decision variables or dimensions to be optimized.

Particle swarm optimization is a heuristic global optimization method put forward originally by Doctor Kennedy and Eberhart in 1995. While searching for food, the birds are either scattered or go together before they locate the place where they can find the food. While the birds are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird is perceptible of the place where the food can be found, having the better food resource information. Because they are transmitting the information, especially the good information at any time while searching the food from one place to another, conduced by the good information, the birds will eventually flock to the place where food can be found. As far as particle swam optimization algorithm is concerned, solution swam is compared to the bird swarm, the birds' moving from one place to another is equal to the development of the solution swarm, good information is equal to the most optimist solution, and the food resource is equal to the most optimist solution during the whole course. The most optimist solution can be worked out in particle swarm optimization algorithm by the cooperation of each individual. The particle without quality and volume serves as each individual, and the simple behavioral pattern is regulated for each particle to show the complexity of the whole particle swarm.

In PSO, the potential solution called particles fly through the problem space by following the current optimum particles. Each particles keeps tracks of its coordinates in the problem space which are associated with the best solution (fitness) achieved so far. This value is called as pbest. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This value is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest (for lbest version). Acceleration is weighted by random term, with separate random numbers being generated for acceleration towards pbest and lbest locations. After finding the best values, the particle updates its velocity and positions with following equations.

$$V_i(k+1) = V_i(k) + c1 * rand() * (P_i(k) - X_i(k)) + c2 * rand() * (g(k) - X_i(k)) \quad (1)$$

$$X_i(k+1) = X_i(k) + V_i(k+1)$$

where

$V_i(k)$ is velocity of particle i at iteration k.

$X_i(k)$ is the position of particle i at iteration k.

$V_i(k+1)$ is velocity of particle i at iteration k+1.

$X_i(k+1)$ is the position of particle i at iteration k+1.

rand( ) is random number between (0,1)

c1 cognitive acceleration coefficient

c2 social acceleration coefficient.

Here our aim is to develop the algorithm which will solve various optimization problems efficiently.

II. REVIEW OF LITERATURE

This section briefs about the status of research work on the idea of Particle Swarm Optimization through survey.

James Kennedy and Russell Eberhart in 1995 [1] proposes particle swarm optimization concept in terms of its precursors, briefly reviewing the stages of its development from social simulation to optimizer. Discussed next are a few parameters that implement the concept. Implementation of one paradigm is discussed in more detail, followed by results obtained from applications and tests upon which the paradigm has been shown to perform successfully.

James Kennedy and Russell Eberhart in 1995 [2] also introduces a new form of the particle swarm optimizer, examines how changes in the paradigm affect the number of iterations required to meet an error criterion, and the frequency with which models cycle interminably around a nonglobal optimum. Three versions were tested: the “GBEST” model, in which every agent has information about the group’s best evaluation, and two variations of the “LBEST” version, one with a neighborhood of six, and one with a neighborhood of two. It appears that the original GBEST version performs best in terms of median number of iterations to convergence, while the LBEST version with a neighborhood of two is most resistant to local minima.

It is observed that the search process for PSO without the first part is a process where the search space statistically shrinks through the generations. On the other hand, by adding the first part**,** the particles have a tendency to expand the search space, that is, they have the ability to explore the new area. So the more likely have global search ability by adding the first part. Both the local search and global search will benefit solving some kinds of problems. There is a tradeoff between the global and local search for different problems, there should be different balances between the local search ability and global search ability. Considering of this, Yuhui Shi and Russell Eberhart in 1998 [3] brought the inertia weight w into the equation (1) as shown in equation (2). This ***w*** plays the role of balancing the global search and local search. It can be a positive constant or even a positive linear or nonlinear function of time.

$$V_i(k+1) = \boldsymbol{w} * V_i(k) + c1 * rand() * (P_i(k) - X_i(k)) + c2 * rand() * (g(k) - X_i(k)) \quad (2)$$

$$X_i(k+1) = X_i(k) + V_i(k+1)$$

It has been found that a large inertia weight facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration, i.e. fine-tuning the current search area.

In particle swarm, if the region converged to is a *local well* containing a *local minimum*, there may initially be hope for escape via a sort of momentum built into the algorithm via the inertial term; over time, however, particles’ momenta decrease until the swarm settles into a state of *stagnation*, from which the basic algorithm does not offer a mechanism of escape. While allowing particles to continue in this state may lead to *solution refinement* or exploitation following the initial phase of *exploration*, it has been observed empirically that after enough time, velocities may become so small that at their expected rate of decrease, even the nearest solution may be eliminated from the portion of the search space particles can practically be expected to reach in later iterations. In traditional PSO, when no better global best is found by any other particle for some time, all particles converge about the existing global best, potentially eliminating even the nearest local minimizer. Van den Bergh and A. P. Engelbrecht [4] appears to have solved this particular problem with his Guaranteed Convergence PSO (GCPSO) by using a different velocity update equation for the best particle since its personal best and global best both lie at the same point, which in traditional PSO inhibits the explorative abilities of the best particle, since it is so strongly pulled toward that one point, with only its weakening momentum and accelerations in the direction of that point keeping it exploring at all . GCPSO is therefore said to guarantee convergence to a local minimize.

There is still a problem, however, in that particles tend to converge to a local minimizer before encountering a true global minimizer. Addressing this problem, Van den Bergh developed multi-start PSO (MPSO)[5] which automatically triggers a restart when stagnation is detected. Restarting in MPSO refers to starting a new search with a different sequence of random numbers generated so that even initial positions are different than they were in previous searches. At restart, particles lose their memories of the previous search so that each search is independent of those previously conducted. After each independent search, the global best is compared to the best global best of previous searches. After a pre-specified number of restarts have completed, the best of all global bests is proposed as the most desirable decision vector found over all searches.

A hierarchical version of the particle swarm optimization (PSO) metaheuristic is introduced by Stefan Janson and Martin Middendorf in 2005 [6]. In the new method called H-PSO, the particles are arranged in a dynamic hierarchy that is used to define a neighborhood structure. Depending on the quality of their so-far best-found solution, the particles move up or down the hierarchy. This gives good particles that move up in the hierarchy a larger influence on the swarm. They introduce a variant of H-PSO, in which the shape of the hierarchy is dynamically adapted during the execution of the algorithm. Another variant is to assign different behavior to the individual particles with respect to their level in the hierarchy. A variant of H-PSO (AH-PSO) with a dynamically changing branching degree of the tree topology has been introduced which could improve the performance of H-PSO. Another extension of H-PSO is to use different values for the inertia weight of the particles according to their level in the hierarchy. It has been shown that this algorithm is able to reach a specified goal for every test function (except the Rastrigin function) faster than all other variants of PSO.

Chunming Yang and Dan Simon (2005) [7] develop a new approach towards better solution. In the New Particle Swarm Optimization Technique proposed here, each particle adjusts its position according to its own previous worst solution and its group's previous worst to find the optimal value. The strategy here is to avoid a particle's previous worst solution and its group's previous worst based on similar formulae of the regular PSO. Equation for velocity and position remains same but the term uses is worse position rather than the best one.

PSO has shown its fast search speed in many complicated optimization and search problems. However PSO could often easily fall into local optima. For better solution Hui Wang, Yong Liu, Sanyou Zeng , Hui Li and Changhe Li in 2007 [8] proposes opposition based PSO. OPSO presents to accelerate the convergence of PSO and avoid premature convergence on multi-modal functions. He proposed method employs opposition based learning for each particle and applies dynamic Cauchy mutation on the best particle. Some results show that particle in the PSO will oscillate between their previous best particle and global best particle found by all the particle before it converges. If the searching neighbors of the global best particle are added in each generation it would extend the search space of the best particle. It is helpful for the whole particles to move to the better positions. This can be accomplished by having Cauchy's mutation on the global best particle in every generations.

The fully informed particle swarm optimization algorithm (FIPS) developed by Marco A. Montes de Oca and Thomas Stutzle in 2008 [9] is very sensitive to changes in the population topology. The velocity update rule used in FIPS considers all the neighbors of a particle to update its velocity instead of just the best one as it is done in most variants. It has been argued that this rule induces a random behavior of the particle swarm when a fully connected topology is used. This argument could explain the often observed poor performance of the algorithm under that circumstance. But it is found to be more suitable on small search regions.

Many variants of the original particle swarm optimization (PSO) algorithm have been proposed. In many cases, the difference between two variants can be seen as an algorithmic component being present in one variant but not in the other. Marco A. Montes de Oca, Thomas Stützle, Mauro Birattari and Marco Dorigo in 2009 [10] proposes new PSO, where first they presented the results and insights obtained from a detailed empirical study of several PSO variants from a component difference point of view. In the second part, proposed a new PSO algorithm that combines a number of algorithmic components that showed distinct advantages in the experimental study concerning optimization speed and reliability and call this composite algorithm Frankenstein's PSO. Frankenstein's PSO is composed of three main algorithmic components, namely, 1) a time-varying population topology that reduces its connectivity over time, 2) the FIPS mechanism for updating a particle's velocity, and 3) a decreasing inertia weight. These components are taken from AHPSO, FIPS, and the time-decreasing inertia weight variant, respectively. The first component is included as a mechanism for improving the tradeoff between speed and quality associated with topologies of different connectivity degrees. The second component is used because the analysis showed that FIPS is the only algorithm that can outperform the others using topologies of different connectivity degree. Finally, the decreasing inertia weight component is included as a mean to balance the exploration-exploitation behavior of the algorithm.

Particle swarm optimization (PSO) is known to suffer from stagnation once particles have prematurely converged to any particular region of the search space. George I. Evers and Mounir Ben Ghalia in 2009 [11] proposed regrouping PSO (RegPSO) which avoids the stagnation problem by automatically triggering swarm regrouping when premature convergence is detected. This mechanism liberates particles from sub-optimal solutions and enables continued progress toward the true global minimum. Particles are regrouped within a range on each dimension proportional to the degree of uncertainty implied by the maximum deviation of any particle from the globally best position. Upon detection of premature convergence, the range in which particles are to be regrouped about the global best is calculated per dimension as the minimum of (i) the original range of the search space on dimension *j* and (ii) the product of the regrouping factor with the maximum distance along dimension *j* of any particle from global best.

III. LIMITATIONS IN EXISTING WORK

Original PSO approach (1995) is to optimize the solution using global best there is chance to trapped in local area. No suggestion is provided for such situation. As the algorithm considers the best value found by neighbors it is more efficient for small number of particles. As the number of particles increases, gbest version is more beneficial. A Modified Particle Swarm Optimizer (1998) works better but only small benchmark function it uses to test. There is difficulty to select probable value of inertia weight. The swarm and the queen: Towards deterministic and adaptive particle swarm optimization by Clerc M.(1999) did not clear whether optimal value is dependant on φ. This creates difficulty to select value of φ. All the three methods mentioned above did not give constant result. Sometimes Rehope method works better sometimes not. It occurs for every method used.

A new locally convergent Particle Swarm Optimizer by F. Van Den Berg and A. P.Engelbrecht (2002) tested for Unimodal functions only. How it performs for multimodal function is not defined here. A New Particle Swarm Optimization Technique by Chunming Yang and Dan Simon (2005) presents formulation of PSO and NPSO, each particle moves to a new position regardless of whether the new solution is better than the current one or not. Changes can be made so that it moves to a better solution unconditionally, but moves to a worse position according to some probability. Opposition-based Particle Swarm Algorithm with Cauchy Mutation has faster convergence on number of functions, still for some functions it falls into local optima which do not guarantee the further convergence. In Fully Informed Particle Swarm Optimization Algorithm, if the population is evenly distributed around a “funnel” in the landscape, the bias will produce good results, especially during the first iterations of the algorithm. When the region where the particles explore happens to be of lower quality than the particles’ previous best positions, the algorithm is in high risk of becoming trapped and being unable to improve any further. In this case, increasing the diversity of the population by making it larger, does not work because the larger the population, the stronger is the bias toward the centroid of the swarm. Enhancing the exploratory capabilities of the algorithm by using dynamic restarts provides some benefits but these are problem-dependent. Like any other algorithm, Frankenstein’s PSO(2009) has its own set of parameters that need to be set by the practitioner before trying to solve a problem. The final parameter settings will depend on the class of problems one is trying to solve and on the application scenario requirements. RegPSO(2009) appears to be a good general purpose optimizer based on the benchmarks tested, which is certainly encouraging; however, it is cautioned that the empirical nature of the experiment is not a theoretical proof that RegPSO will solve every problem well: certainly, its performance must suffer somewhere.

IV FUTURE SCOPE

Future work will try to understand where the algorithm suffers in order to understand any limitations and apply it to the proper contexts. One such difficulty seen already was with simple uni-modal functions, where regrouping is unnecessary since particles quickly and easily approximate the global minimizer to a high degree of accuracy, and where there is no better minimizer to be found. While the regrouping mechanism has been tested in conjunction with standard gbest PSO in order to demonstrate the usefulness of the mechanism itself, there does not seem to be anything to prevent the same regrouping mechanism from being applied with another search algorithm at its core. Performance may be improved in conjunction with an improved local minimizer such as GCPSO. The study of heterogeneity in PSO has not been done systematically and therefore there are still gaps in understanding of the effects of heterogeneity in PSO algorithms. Lot of scope is to have work on new type of particle swarm optimization.

Proper parameter & topology selection is also one of the research areas in PSO.

V CONCLUSION

Particle swarm optimization is a heuristic global optimization which is used in various real life applications. Here we have presented the concept of PSO and work carried out on PSO by different researchers. A detailed literature review is presented which is used to find out the limitations in various methods and which gives direction for future scope. Future scope identifies as work to be carried out towards topology of communication, parameter adjustment, initial distribution of particles and methods to deal with stagnation .

VI REFERENCES

[1] James Kennedy and Russell Eberhart, “Particle Swarm Optimization”,IEEE 1995.
[2] Russell Eberhart and James Kennedy, “A New Optimizer Using Particle Swarm Theory”, IEEE 1995.
[3] Yuhui Shi and Russell Eberhart, “A Modified Particle Swarm Optimizer”, IEEE 1998.
[4] F. Van Den Berg and A. P. Engelbrecht, “A New Locally Convergent Particle Swarm Optimizer”, IEEE 2002.
[5] Van den Bergh, a thesis on “An Analysis of Particle Swarm Optimizers”, Nov 2001.
[6] Stefan Janson and Martin Middendorf, “A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 35, NO. 6, DECEMBER 2005.
[7] Chunming Yang and Dan Simon, “A New Particle Swarm Optimization Technique”, 2005.
[8] Hui Wang, Yong Liu, Sanyou Zeng , Hui Li and Changhe Li, “Opposition based Particle swarm Algorithm with Cauchy Mutation”, IEEE 2007.
[9] Marco A. Montes de Oca and Thomas Stutzle, “Convergence Behavior of the Fully Informed Particle Swarm Optimization Algorithm”, 2008.
[10] Marco A. Montes de Oca, Thomas Stützle, Mauro Birattari and Marco Dorigo, “Frankenstein’s PSO: A Composite Particle Swarm Optimization Algorithm”, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 5, OCTOBER 2009.
[11] George I. Evers and Mounir Ben Ghalia, “Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks”, 2009.
[12] George I. Evers, a thesis on “AN AUTOMATIC REGROUPING MECHANISM TO DEAL WITH STAGNATION IN PARTICLE SWARM OPTIMIZATION”, 2009.
[13] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, Sankar Kumar Pal, “Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients”, 2007.
[14] M. A. Montes de Oca, J. Pena, T. Stutzle, C. Pinciroli, and M. Dorigo, “Heterogeneous Particle Swarm Optimizers”, Jan 2009.
[15] Magnus Erik, Hvass Pedersen, Andrew John Chipperfield, “Simplifying Particle Swarm Optimization”, 2009.
[16] Gary G. Yen and Wen Fung Leong, “Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS —PART A: SYSTEMS AND HUMANS, VOL. 39, NO. 4, JULY 2009.
[17] Qinghai Bai, “Analysis of Particle Swarm Optimization Algorithm”, CCSE, Vol 3, No. 1, February 2010.
[18] Magnus Erik, Hvass Pedersen, “Good Parameters for Particle Swarm Optimization”, Technical Report no. HL1001, 2010.
[19] Prithwish Chakraborty, Swagatam Das, Ajith Abraham, Václav Snasel and Gourab Ghosh Roy, “On Convergence of Multi-objective Particle Swarm Optimizers”, IEEE, 2010.