Study and Analysis of Particle Swarm Optimization: A Review

Hemlata S. Urade
Department of CSE
Rajiv Gandhi College of Engineering & Research Technology, Chandrapur (MS)

Prof. Rahila Patel
Department of CSE
Rajiv Gandhi College of Engineering & Research Technology, Chandrapur (MS)

ABSTRACT
Particle swarm optimization is a global optimization algorithm that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. This paper presents a review on PSO in single and multiobjective optimization. The paper contains the basic PSO algorithm and various techniques used in pre-existing algorithms. It also describes the simulation result which is carried out on benchmark functions of single objective optimization with the help of basic PSO. Study of literature shows future direction to enhance the performance of PSO.

Keywords
Optimization, Swarm intelligence, Particle Swarm optimization, multiobjective PSO, Dynamic PSO.

1. INTRODUCTION
Optimization is a mathematical discipline that concerns the finding of minima and maxima of functions, subject to so called constraints. Optimization originated in the 1940s, when George Dantzig used mathematical techniques for generating "programs" (training timetables and schedules) for military application. Since then, his "linear programming" techniques and their descendents were applied to a wide variety of problems, from the scheduling of production facilities, to yield management in airlines. Today, optimization comprises a wide variety of techniques from Operations Research, artificial intelligence and computer science. Thus it has been invented that optimization can be either single-objective or multiobjective.

An attempt to optimize a design or system where there is only one objective usually entails the use of gradient methods where the algorithms search for either the minimum or maximum of an objective function, depending on the goal. And if the problem having more than one objective then that will be the multiobjective optimization. The role of the optimization algorithm is then to identify the solutions which lie on the trade-off curve, known as the Pareto Frontier. Thus for the better Pareto-optimal set of multiobjective optimization problem many evolutionary algorithm has been invented, for example:

1. Evolutionary Genetic Algorithm (GA): These are most famous methods and used frequently in Evolution computation field. The search and application of GAs are diverse and deeply. The Genetic Algorithm originally described by the Holland. The emphasis of GA is usually on the recombination, with mutation treated as a 'background operator'.

2. Simulated Annealing Multiobjective Optimization (SA): Simulated annealing algorithm is a simple and robust technique, utilizing the principles of statistical mechanics simulating the way metal cool and annealing. SA can find global optimal solution among many local optimal solutions and more and more attention was pay attention to SA. Comparing with GA, SA takes up little memory and consumes less CPU time, because it finds the optimal solution using point-by-point iteration rather than a search over a population.

3. Ant Colony Optimization Algorithm (ACA): These ACA algorithm has been applied in combinatorial optimization successfully. Its biologic background decides it perfect for discrete combinatorial optimization. It's difficult for this technology to handle with continuous function optimization.

4. Particle Swarm Optimization (PSO): The Particle swarm optimization is a population based optimizing tool which worked out by the cooperation of each individual. The convergence ability in PSO is faster rather than other evolutionary algorithm (GA, SA, and ACA) and also it require the less parameter for calculating the optimizing value.

The rest of the paper is organized as follows: Section 2 represent the basic of PSO, Section 3 and Section 4 represent the overview on PSO in single objective optimization and multiobjective optimization, Section 5 describe the simulation work carried on basic PSO for the single objective benchmark functions, finally Section 6 outlines the future directions of our work and concludes the paper.

2. BASIC OF PSO
The Particle swarm optimization is a heuristic global optimization method put forward originally by Doctor Kennedy and Eberhart in 1995[1]. 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, conducted 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.

Particle swarm optimization is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Hypothesis are plotted in this space and seeded with an initial velocity, as well as communication channel between the particles. Particles then move through the solution space and are evaluated according to some fitness function after each timestamp. Over time particles are accelerated towards those particles within their grouping which have better fitness values.

PSO shares many similarities with evolutionary computation techniques such as Genetic algorithms (GA). The system is initialized with population of random solution and searches for optima by updating generation. In PSO, the potential solution called particles fly through the problem space by following the current optimum particles. Each particle 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[id]=v[id]+c1*r(id)*(pbest[id]-x[id])+c2*r*(id)(gbest[

id]-x[id])------ (a)

x[id] = x[id]+v[id]----------(b)

where,

v[id] is particle velocity

x[id] is the current particle

r(id) is random number between (0,1)

c1 and c2 are learning factors usually c1=c2=2.

The pseudo code of the procedure is as follows,

For each particle

Initialize particle

END

Do

For each particle Calculate fitness value.

If the fitness value is better than the best Fitness value (pbest) in history

Set current value as the new pbest.

END

Choose the particle with the best fitness value of all the particles as the gbest

For each particle

Calculate particle velocity according to equation (a) Update particle position according equation (b)

END

While maximum iterations or minimum error criteria is not attained.

Particles` velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

3. PSO IN SINGLE OBJECTIVE OPTIMIZATION

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. Ant colony optimization, Genetic algorithm, particle swarm optimization are various evolutionary algorithms proposed by researchers. Due to simplicity in PSO equation and fast convergence, PSO is found to be best among these. After analysing each parameter in PSO equation Yuhui Shi and Russell Eberhart [3] proposes new parameter “w” called inertia weight in the basic equation.

It can be imagined that the search process for PSO without the first part is a process where the search space statistically shrinks through the generations. It resembles a local search algorithm. Addition of this new parameter causes exploration and exploitation in search space. Firstly value of w was kept static. Later on it was kept linear from 0.9 to 0.4. Eberhart Russell and Shi Yuhui [4] in 2000 compare these results with constriction factor. A small change in equation has been made. When Clerc’s constriction method is used, φ is set to 4 and constant multiplier is thus set to 0.729. According to Clerc, addition of constriction factor may be necessary to insure convergence of particle swarm optimization algorithm.

Then equation (a) becomes

V[id]= K [v[id]+c1*r(id)*(pbest[id]-x[id])+c2*r*(gbest[id]-x[id])]

Where,

K= where $\varphi = c1+c2$, $\varphi > 4$

The PSO algorithm with the constriction factor can be considered as a special case of the algorithm with inertia weight.

F. van den Bergh, A. P.E ngelbrecht invented Guaranteed Convergence Particle Swarm Optimizer (GCPSO) [5]. The GCPSO has strong local convergence properties than original PSO. This algorithm performs much better with the small number of particle. The new phenomenon is defined called as stagnation by these GCPSO i.e., if a particle’s current position coincides with the global best position particle, then the particle will only move away from the point if its previous velocity and w are non-zero. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to premature convergence of the algorithm. In fact, this does not even guarantee that the algorithm has converged on a local

minimum it merely means that all the particles have converged on the best position discovered so far by the swarm.

Hierarchical PSO [7] is known as hierarchical version of PSO called as H-PSO. In this algorithm the particles are arranged in a dynamic hierarchy. In H-PSO, a particle is influenced by its own so far best position and by the best position of the particle that is directly above it in the hierarchy. In H-PSO, all particles are arranged in a tree that forms the hierarchy so that each node of the tree contains exactly one particle. If a particle at a child node as found a solution that is better than the best so far solution of particle at parent node, then these two particles are exchanged. In this algorithm the topology used as regular tree in which hierarchy is defined in terms of height and branching degree. This hierarchy gives the particles different influence on the rest of the swarm with respect to their fitness. Each particle is neighbored to itself and its parent in the hierarchy. Only the inner nodes on the deepest level might have a smaller number of children so that the maximum difference between the numbers of children of inner nodes on the deepest level is at most one. In order to give the best individuals in the swarm a high influence, particles move up and down the hierarchy.

New Particle Swarm Optimization(NPSO)[8] is the idea of NPSO came from our personal based experience that an individual not only learns from his or her own and other individuals' previous best, but also learns from his or her own and other individuals' mistakes. Each particle tries to leave its previous worst position and its group's previous worst position. NPSO has the better solution result than the originally PSO as more seeds are needed when the search dimension gets larger and other parameters might need to change too. The solutions might get out of local minima with more random numbers generated. Position change limits may be tuned too rather than original PSO. Eberhart proposed a discrete binary version of PSO for binary problems [9]. In their model a particle will decide on "yes" or " no", "true" or "false", "include" or "not to include" etc. also this binary values .In binary PSO, each particle represents its position in binary values which are 0 or 1. Each particle's value can then be changed (o r better say mutate) from one to zero or vice versa. In binary PSO the velocity of a particle defined as the probability that a particle might change its state to one. The novel binary PSO [9] solve the difficulties those are occurred in binary PSO. In this algorithm the velocity of a particle is its probability to change its state from its previous state to its complement value, rather than the probability of change to 1. In this new definition the velocity of particle and also its parameter has the same role as in continuous version of the PSO.

The original PSO is easily fall into local optima in many optimization problems. The problem of premature convergence is solved by the OPSO. It allows OPSO [10] to continue search for global optima by applying opposition based learning. The OPSO use the concept of Cauchy mutation operator. The OPSO based on opposition- based learning method. The OBL method has been given by Hamid R.Tizhoosh, it is explained as when evaluating a solution x to a given problem, we can guess the opposite solution of x to get better solution of x2`. By doing this the distance from the optima solution can reduce. The opposite solution x can be calculated as

$x` = a + b - x$

where x. R within [a, b]

Mendes [12] proposed another efficient approach which deals with stagnation is the fully informed particle swarm optimization algorithm (FIPSO) [12]. FIPSO use best of neighborhood velocity update strategy. In FIPSO each particle uses the information from all its neighbors to update its velocity. The structure of the population topology has, therefore, a critical impact on the behavior of the algorithm which in turn affects its performance as an optimizer. It has been argued that this happens because the simultaneous influence of all the particles in the swarm "confounds" the particle that is updating its velocity, provoking a random behavior of the particle swarm.

The original PSO has the homogeneity i,e the swarm group contain the same kind of particle. Very interesting and somewhere new technique called heterogeneous Particle swarm [13], contains the different kind of swarm. The four type of heterogeneity is invented, 1.Neighbourhood heterogeneity: This heterogeneity appears when particles have different neighborhood sizes. This kind of heterogeneity occurs when the population topology is not a regular graph, that is, when nodes may have different degrees. This type of heterogeneity allows some particles to be potentially more influential in the collective search process than others. 2. Model-of-influence heterogeneity: This type of heterogeneity occurs when particles in a swarm use different mechanisms for choosing their informers. 3.Update-ruleheterogeneity: If different particles use different rules for updating their position in the search space, we say that the swarm exhibits update rule heterogeneity Conceptually, PSO algorithms of this class exhibit one of the most extreme cases of heterogeneity because particles can explore the search space in completely different ways. 4.Parameter heterogeneity: The parameter heterogeneity is met with two condition (i) a group of particles must use the same update rule, and (ii) at least a pair of these particles must differ in their update rules' parameter settings.

4. PSO IN MULTIOBJECTIVE OPTIMIZATION

Particle Swarm Optimization Algorithm (PSO) takes great advantage on solving multiobjective problems, comparing with conventional methods. First, its high efficient search ability makes PSO get Pareto-optimal solutions; Moreover, PSO cannot obtain multiple non-dominated solutions according inherent parallel character; Furthermore, PSO can handle with all kinds of multiobjective functions and constrains; Finally, PSO combines with conventional methods with ease and we can get a better method for special problems. In fact, there have been several recent proposals to extend PSO to handle multiobjectives.

In 1999, J. Moore and R. Chapman [17] proposed the algorithm based on Pareto dominance. They emphasize the importance of performing both an individual and a group search (a cognitive component and a social component). However, the authors did not adopt any scheme to maintain diversity. X. Hu and R. Eberhart [18] in 2002, proposed an in which, only one objective is optimized at a time using a scheme similar to lexicographic ordering. Lexicographic ordering [19] tends to be useful only when few objective functions are used (two or three), and it may be sensitive to the ordering of the objectives. Fieldsend and Singh in 2002[20], proposed an approach in which they use an unconstrained elite archive (in which a special data structure called "dominated tree" is adopted) to store the non dominated individuals found along the search process. The archive interacts with the primary population in order to

define local guides. The approach is compared (Using four test functions and two metrics) against an algorithm similar to PAES and with a variation of original MOPSO. Their approach also uses "turbulence" Operator that is basically a mutation operator that acts on the velocity value used by PSO. It is important to note that the new version of MOPSO provided in the Fieldsend's paper does not have the problems of the original version in multifrontal problems.

PSO-based multiobjective optimization has following technique to get local best particle or global particle. a) Aggregating method: Parsopoules K E [21] first took PSO solving multiobjective problem using fixed weight, adaptive weight and vector evaluation method. Weighted-vector method often can't get appropriate weights for special optimization problem. Vector evaluation method always couldn't provide satisfied solutions for Multiobjective Optimization (MO) problems. b) Pareto-based method: Ray T and Liew K M [22] combined Pareto dominance with PSO to solve MO problems. They got a set of elitist solutions by Pareto dominance, from which a global optimal particle was chosen by roulette.

The approach uses a nearest neighbor density estimator to promote diversity. c) Sigma method: Mestaghim S [23] selected global best particle (or local best particle) according to the distance between current solutions and archive solutions. Sigma method began with the initialization of a number of solutions. If difference between fitness of the initial solutions was small, it would lead to selection press inadequate and PSO algorithm convergence slowly. d) Dynamic neighborhood method: Hu X and Eberhart R [24] defined one of objective as optimization objective, other objective as fixed objective. Each particle has different neighbors in each generation. This method is sensitive about the order of optimization objective and neighbor objective. e) Multi-population method: Konstantin's E.Parsopoulos et al [25] divided population into many subpopulations, and each subpopulation executes PSO independently according to one of the objective function. Every subpopulation exchange information with each other to obtain Pareto-optimal solution. Great number of particles increased computational time.

5. EXPERIMENTAL WORK

Here our aim is to solve multiobjective optimization problems with the help of dynamic PSO. Dynamic PSO can be defined as varying characteristics of PSO while experimentation is running. Characteristic include topology, swarm size, search space. If topology or swarm size can be change during process, then it is treated as dynamic particle swarm optimization.

We perform simulation to study PSO behavior in 2 and 10 dimensional spaces. Values of parameter involved in equation are considered as 0.7 for inertia, 1.49 for c1 and c2 and particle swarm size 25. Simulation has been carried out for 10000 iterations. Experiment has been conducted for standard benchmark functions. Following are results found during experimentation. It has been found that simple PSO performs well in 2 dimensions but performance degrades as dimension increases.

Table 1. Mean of Global best

Benchmark Functions	Formula	Mean of fitness values	
		2 Dim	10 dim
Ackley	$f(\bar{x}) = 20 + e - 20e^{-0.2\sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n}}} - e^{\frac{\sum_{j=1}^{n}\cos(2\pi x_j)}{n}}$ $-30 \le x_j \le 30$	**0.0027**	**0.0040**
Griewank	$f(\bar{x}) = 1 + \sum_{j=1}^{n} \frac{x_j^2}{4000} - \prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$ $-600 \le x_j \le 600$	**2.3985e-004**	**0.0222**
Sphere	$f(\bar{x}) = \sum_{j=1}^{n} x_j^2$ $-100 \le x_j \le 100$	**0.0826**	**3.4374**
Rastrigin	$f(\bar{x}) = 10n + \sum_{j=1}^{n} (x_j^2 - 10\cos(2\pi x_j))$ $-5.12 \le x_j \le 5.12$	**0.0022**	**0.0221**
Rosenbrock	$f(\bar{x}) = \sum_{j=1}^{n-1} \left(100(x_{j+1} - x_j^2)^2 + (1 - x_j)^2\right)$ $-30 \le x_j \le 30$	**1.4797e-004**	**0.1355**

6. CONCLUSION & FUTURE SCOPE

The review of PSO has studied with considering the preexisting algorithms. The basic steps of PSO explained which has been used in proposed approach. Result shows that simple PSO performs well in 2 dimensions as compared to 10 dimensions. Thus in future plan the multiobjective optimization problem will solve by Dynamic PSO.

Thus in future work, we will solve the multiobjective optimization problem by Dynamic PSO. These works divide in the following four modules.

1. To study the basic PSO and Dynamic PSO
2. Implementation of Dynamic PSO for to solve single objective optimization problem
3. Study of multiobjective optimization problem
4. Implementation of Dynamic PSO for to solve multiobjective optimization problem, which is our basic aim of project work.

Thus, Lot of work has been carried out to enhance the performance of PSO. Not a single algorithm, suited to solve all kinds of optimization problems. Still it can be achieved through,

1) Selection of proper population size
2) Proper adjustment of parameter values
3) Better velocity updates equation
4) Combining PSO with other techniques like ACO, GA.

7. REFERENCES

[1] James Kennedy and Russel Eberhart" Particle Swarm Intelligence", IEEE 1995.

[2] Russel 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] Xiang-Han Chen, Wei-Ping Lee, Chen yie Liao, Jag-Ting Dai,"Adaptive Constriction Factor for Location-related Particle Swarm", Proceedings of the 8th WSEAS International Conference on Evolutionary Computin,Vancouver, British Columbia, Canada, June 19-21, 2007

[5] F. Vanden Bergh, A. P.E. ngelbrecht " A New Locally Convergent Particle Swarm Optimizers" IEEE 2010

[6] Prithwish Chakraborthy, Swagatam Das, Ajith Abraham, Vaclav Snapseland Gourab Ghosh Roy " On convergence of Multi-objective particle swarm optimizer" IEEE 2010

[7] Stefan Janson and Martin Middendorf " A hierarchical particle swarm optimizer and its Adaptive variants

[8] Chunming Yang and Dan Simon, " A New Particle Swarm Optimization Technique" IEEE 2010

[9] Mjtavi Ahmadieh Kinanesar, A Novel Binary Particle Swarm Optimization" IEEE 2007

[10] Hui Wang, Youg Lie, Sanyou Zeng, Hui Li," Opposition based particle swarm algorithm with Cauchy Mutation" 2007

[11] Praveen Kumar Tripathi, Sanghmitra Bandyopadhyay, Sankar Kumar Pal Multi- Objective Particle Swarm Optimization with time variant inertia and acceleration coefficient " IEEE 2004

[12] Macro A. Montes de Oca and Thomas Stutzle," Fully Informed Particle Swarm Optimization," IEEE 2007

[13] Macro A. Montes de Oca, Jorge Pen a, Thomas Stutzle, Carlo Pinciroli and Macro Dorigo" Heterogeneous Paricle Swarm Optimizers" IEEE 2009

[14] Daniel Bratton, James Kennedy," Defining Standard for particle swarm optimization" IEEE 2007

[15] S. Janson and M. Middendorf," A hierarchical particle swarm optimizer and its adaptive variant" IEEE 2000

[16] S.-K.S. Fan and E. Zahara," A hybrid simplex search and particle swarm optimization for unconstrained optimization"

[17] J. Moore and R. Chapman, "Application of Particle Swarm to Multiobjective Optimization" : Dept. Comput. Sci. Software Eng., Auburn Univ.1999

[18] X.Hu and R Eberhart," Multiobjective optimization using dynamic neighbourhood particle swarm optimization," in Proc. Congr. Evolutionary computation (CEC'2002). Vol. 2.

[19] C.A. Coello Coello, D.A. Van Veldhuizen and G.B. Lamont, Evolutioary Algorithms for Solving Multi-Objective Problems. Norwell MA: Kluwer, 2002

[20] J.E. Fieldsend and S.Singh, " A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence," in proc. 2002 U.K. Workshop on Computational Intelligence, Birmingham, U.K., Sept. 2002

[21] Parsopoules K.E. Vrahatis MN, Particle Swarm Optimization Method in Multiobjective Problems[A]," Proceedings ACM Symposium on Applied computing[C] 2002

[22] Ray T, Liew K M," A Swarm Metaphor for Multiobjective Desing Optimization [J]", Engineering Optimization 2002

[23] Mostaghim S. Teich J," Strategies for Finding Local Guides in Multiobjective Particle Swarm Optimization (MOPSO) [A]," Proceedings of the IEEE Swarm Intelligence Symposium [C] 2003

[24] Hu X. Eberhart R," Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization[A]," Proceedings of the IEEE Congress on Evolutionary Computation [C]2002

[25] Konstantinos E. Parsopoulos, Dimitris K. Tasoulis and Michael N. Vrahatis. "Multiobjective optimization using parallel vector evaluated particle swarm optimization." In proceedings of the IASTED International Conference on Artificial Intelligence and Applications(AIA 2004).