Particle swarm optimization algorithm and its parameters: A review

Mudita Juneja, S.K.Nagar
Dept. of Electrical Engineering
Indian Institute of Technology (B.H.U.)
Varanasi, India

***Abstract*—In the year 1995, Dr R.C. Eberhart, who was an electrical engineer, alongwith Dr. James Kennedy, a social psycologist invented a random optimization technique whicha was later named as Particle Swarm Optimization. As the name itself asserts that this method draws inspiration from natural biotic life of swarms of flocks. It uses the same principle to find most optimal solution to problem in search space as birds do find their most suitable place in a flock or insects do in a swarm. The PSO algorithm is initialized with a horde of particles which are a collection of random feasible solutions. Every single particle in the swarm is initialised a random velocity and as soon as they are assigned a velocity these particles start moving in problem search space. Now from this space the algorithm draws the particle to most suited fitness which in turn pulls it to the location of best fitness achieved across the whole horde. The PSO update rule comprises of many distinguishing features which are adjusted and modified depending upon the area of application of algorithm. This paper gives a detailed description of the PSO algorithm and significance of the various parameters involved in its update rule. It also highlights the advantages and disadvantages of using PSO algorithm in any optimization problem.**

***Index Terms*— Computational intelligence, Evolutionary Algorithm, Genetic Algorithm, Exploration, Exploitation, Inertia weights, Acceleration constants.**

I. INTRODUCTION

Evolutionary techniques have seen a sudden surge in their popularity for their use in research domains. This popularity acknowledges their versatility as design tools and their impeccable ability to find optimal solution in complex multimodal search spaces. Genetic algorithms and Particle swarm optimization have emerged as effective and efficient algorithms for handling complex optimization problems. Genetic Algorithm is a learning methodology which is largely based on Darwin's principles. PSO working is as similar to what animals do in a herd or birds in a flock.

PSO technique was invented in the mid 1990s while attempting to replicate the well-orchestrated graceful motion of a bird flock, which was a part of a socio-cognitive study investigating the notion of collective intelligence in biological populations. It got recognition as a evolutionary technique soon after its invention [1]. The immensely efficient problem solving ability of this technique in various engineering and science applications has led to its widespread study and deep introspection. There are various research papers [2-19] available which give thorough knowledge of this method and its use. The PSO update rule includes some turning parameters that have greatly influenced the algorithm performance. It is very tedious task to balance the local and global search operations to find an optimal solution. With the help of PSO finding an optimum is much easier. The two search factors are inbuilt in update rule. Hence it finds most accurate and efficient solution to any problem in search space within the given span of time. It balances the tradeoff between the exploration and exploitation techniques much efficiently.

The following paper reviews the basic PSO technique, as an Evolutionary Algorithm, and emphasizes the importance of the parameters that are used in the update rule of this algorithm.

II. EVOLUTIONARY ALGORITHMS: PSO

A. Evolutionary Algorithms(EAs)

In recent years, Evolutionary Technique has evolved as an interesting research field. It is not only because of its efficiency in finding an optimum, but also due to its affinity with natural social systems. Since it is inspired by nature, these techniques are based on biological evolution. They use methods such as reproduction, mutation, recombination and natural selection to produce candidate solutions. Generally the problem space which consists of ordination undergoes operations like mutation, combination etc. The values that are to be retained are decided on the basis of a cost function which in turn regulates the suitability of resulting candidate solutions. These operations are then applied repeatedly. The improvisation in finding most optimal solution is due to the natural selection. The general scheme of EAs [20] is given in Fig. 1.

The basic idea and main features of EAs as in [20] is presented as a competition between a population of individuals The competition adds strength to the theory of "survival of fittest" under environmental influence leading to rise in fitness of population. The main features of EAs are as follows;

- EAs are population based, i.e., they process a whole collection of candidate solutions simultaneously.
- EAs mostly use recombination to mix information which is attained from large number of candidate solutions into a new solution.
- EAs are stochastic or random tools of optimization.

Genetic algorithm, one of the most important EA, refers to the model introduced by John Holland and his students in 1975. These algorithms encode potential solutions of the problem onto chromosomes and then apply recombination to these chromosomes while preserving their relevant information. In other words, GA utilizes recombination and selection operators

978-1-4673-9084-2/16/$31.00 ©2016 IEEE

to produce new sample points in sample space [21]. It can obtain near-optimal solution without the complete knowledge of task domain, by only manipulating chromosomes. This feature makes it versatile in solving any kind of optimization problems [22].

The concept of particle swarms was originated by a social psychologist, Kennedy and an electrical engineer, Eberhart to develop the idea of computational intelligence, utilizing the existing natural interactive systems. The first simulations [2] were influenced by social behavior and involved analogues of bird flocks searching for corn. This soon developed [3-4] into a powerful optimization method; Particle Swarm Optimization (PSO). PSO attempts to mimic the goal-seeking behavior of biological swarms.

In PSO algorithm, the collection of particles in search space aim to optimize a fitness function, in a way similar to movement of flocks of birds in natural environment in search of food. The particles are placed randomly in search space and they evaluate their quality or fitness at that position. Then, for a predefined number of iterations, each particle moves to a new location which gives better fitness than the previous position. This movement is based on the history of particles own best and current locations with those of the best positions attained by other particles in the swarm, with some random perturbations. Thus in subsequent iterations the swarm achieves the most optimum solution to the fitness function in the problem space, with a defined number of particles working together. [4] The fitness or objective function in PSO algorithm is a performance evaluation criterion that depends on the application area of the algorithm. The performance criterion is usually defined by a mathematical formulation to quantify the system performance achieved through a performance index.

Genetic algorithm and other population based search techniques are motivated by evolution as it occurs in nature. But PSO, as opposed to other techniques, is based on analogies with social behavior of animal and birds. There is no selection operation in PSO algorithm and thus, all the particles of the swarm are retained throughout the search process. The position and velocity of each of the particles is updated in every iteration, in accordance with particles own and group's best positions attained by far. Therefore, unlike other evolutionary algorithms, PSO, does not implement the policy of the survival of the fittest. [24]

B. *Basic Particle Swarm Optimization Algorithm*

The basic particle swarm optimization algorithm consists of a swarm of "n" particles, and the position of each particle represents a possible solution of the fitness function in D-dimensional search space. The particle changes its condition under the influence of three factors [5]:

- Its own inertia.
- Personal most optimal position.
- Swarm's most optimal position.

BEGIN

INITIALISE each candidate in the population randomly;
EVALUATE fitness of each candidate;
REPEAT UNTIL (*TERMINATION CONDITION* is satisfied) DO

1 SELECT parents;
2 RECOMBINE their pairs;
3 MUTATE the resulting offsprings;
4 EVALUATE new candidate solutions;
5 SELECT individuals for next generation based on fitness of evaluated candidate solutions;

OD

END

Fig. 1. General Scheme of Evolutionary Algorithms

In PSO algorithm, the speed and position of each of the particles in the swarm change according to the following equations:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 r_1^k\left(pbest_{id}^{k} - x_{id}^{k}\right) + c_2 r_2^k\left(gbest_{id}^{k} - x_{id}^{k}\right) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

Where,

v_{id}^{k} and x_{id}^{k} represents velocity and position of i^th particle (out of n particles) at d-dimension (out of D dimensions) in k^th iteration respectively.

$pbest_{id}^{k}$ and $gbest_{id}^{k}$ represents personal best position and global best position (i.e. group's best) of i^th particle (out of n particles) at d-dimension (out of D dimensions) in k^th iteration respectively.

w represents inertial weight attached to the particle's previously attained position.

c_1, c_2 represent acceleration constants.

r_1^k, r_2^k represent random numbers in the range of [0,1].

The velocity update in PSO consists of three parts [6]:

- Momentum: It represents the tendency of particle to move in the same direction as it was moving in the previous iteration. It incorporates the effect of previous velocity on current velocity of the particle.
- Cognitive part: It represents the pull to particle's velocity towards its own personal best (pbest). Referred to as "memory", "self-knowledge" or "remembrance".
- Social part: It represents the pull to particle's velocity towards swarm's best (gbest). Referred to as "cooperation", "social knowledge" or "shared information".

Figure 2 illustrates the flowchart of PSO algorithm [24].

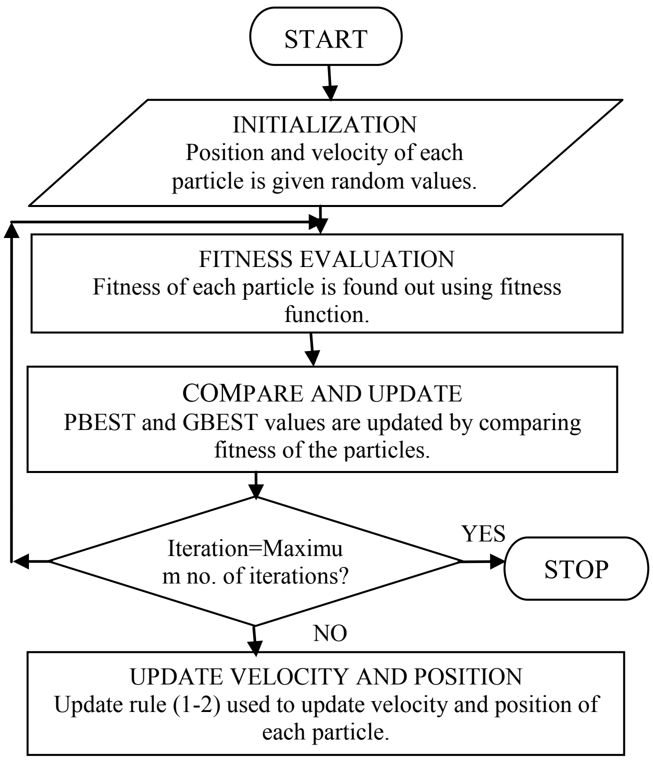

Fig. 2. Flowchart of PSO

C. *Inertia Weight*

In the search space particle's velocity on every dimension is fasten onto a maximum velocity, V_{max}. This maximum velocity determines the resolution or intent with which regions between the present position and the target position (best so far) can be sorted out [5]. Howsoever the use of hard bounds presents some problems. For instance if V_{max} is too high then particles might slip away from some good food locations ,whereas if V_{max} is too low then some good distant location will always be out of reach. Thus in order to reduce the importance of V_{max}, and better to say in order to knock out it altogether, and to sharpen the foraging ability of particles, a weight term was added to the PSO's update equations [5]. This weight term is called inertia weight (w) and it controls the effect which the last iteration speed has on the current speed. Larger value of w improves global search capability and smaller value of w improves the partial search capability of PSO algorithm. Generally, it is equal to 1, but eventually the search ability decreases and the particle get stuck at a non-optimum location. In experimental work, w is kept in between 0.9 to 0.4 and the values are decreased linearly so that the algorithm allows the particles to explore wider areas in beginning and nearby areas in later stages with reduced speeds. This setting gives a greater likelihood of reaching the target optimum position quickly [3]. Note that if we interpret $c_1 r_1^k(pbest_{id}^k - x_{id}^k) + c_2 r_2^k(gbest_{id}^k - x_{id}^k)$ as the external force f_i, acting on a particle, then the change in a particle's velocity (i.e., the particle's acceleration) can be written as (3);

$$\Delta v_i = f_i - (1 - \omega)v_i \tag{3}$$

Thus we can say that the constant $1 - \omega$ acts effectively as a friction coefficient. Hence ω is said to be the fluidity of medium in which a particle moves. Perhaps this is the reason why researchers have found that the best performance is obtained by setting ω initially to some relatively high value (e.g., 0.9).the high value ω of corresponds to a system where particles move in a low viscosity medium and perform extensive exploration. When ω is gradually reduced to a much lower value (e.g., 0.4), the system becomes more dissipative and exploitative and would be better at homing into local optima. It is even possible to start from values of $\omega > 1$, but that will make the swarm unsteady and doddery until and unless the value is reduced sufficiently to bring the swarm in a stable region [4].

D. *Acceleration constants c_1 and c_2*

These constants are related to the speed of flying of particles to the most optimist position of swarm and its own best position. They regulate the length and time taken by particle to reach most optimum position [4]. So that the particle land in a correct position, these constants must be properly selected. For too big a value of acceleration constants, the particle may fly past the correct position and for too small values, the particle will not be able to reach the target position. Generally each of these constants are set to 2 to make the times taken to move towards the particle's personal best and swarm's global best as equal and half the total time. These acceleration constants represent the weighing of acceleration terms towards pbest and gbest locations.

E. *Random numbers r_1 and r_2*

The pull on the particles towards pbest and gbest positions are regulated by adding random numbers in the update rules [25]. These are random fiction and determine the magnitude of random forces towards the two best positions. They add a random component to the PSO algorithm and help prevent the algorithm from getting stuck at a non-optimal local minimum or maximum solution.

F. *Size of population*

This is often set empirically on the basis of the dimensionality and perceived difficulty of a problem. Values in the range 20–50 are quite common. Swarm size varies from one application to the other and so is problem dependent.

III. Advantages and Disadvantages of PSO Algorithm

The PSO algorithm used in various optimization problems has certain advantages and disadvantages given below;

A. Advantages

- PSO algorithm does not involve selection operation or mutation calculation. The search can be carried out by repeatedly varying particle's speed.
- By learning from group's experiences, particles fly only to good areas (where there is a possibility of finding food).
- PSO algorithm is based on artificial intelligence and thus, can be applied into both scientific research and engineering applications [3].
- Simple calculations are involved in PSO algorithm and with development of newer evaluation techniques they are be done easily [3].

B. Disadvantages

- Standard PSO suffers from a substantial rise in search complexity with increase in dimension of search space.
- The method is vulnerable to partial optimism, which leads to a much less accurate regulation of its speed and the direction [3].
- Due to the lack of dimensionality [3] this method cannot be used for problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field

When an application task involves many parameters and the parameter dimensions are increasing in order to match the increase in task complexity, the solution space is expected to grow exponentially. Consequently the search becomes more and more hazardous and baffling. Time to time the PSO algorithms has been modified and altered to obtain better solutions than the standard PSO. But the search quality of these modified versions declines soon for complex tasks with high dimensional and multimodal objective functions. In addition to it the distribution and density of these optimal solutions often vary from function to function making it more difficult to design a general or universal strategy for all complex situations. This is mainly because the PSO has a high convergence speed and this often results in the loss of diversity during the optimization process. The undesirable and premature situation leads the particles to get trapped in local optimums. Hence it is unable to gain the best solution. To overcome the search difficulties described above, a new PSO approach with two special features was proposed; one with dimension partition and other with adaptive velocity control [26-27].

The most important reason for this premature convergence is the velocity update rule which only depends on information on same dimension. Thus when a specific dimension starts losing its diversity in position or gets trapped in local minima the search capability in that dimension decreases and so the overall search ability is hindered.

IV CONCLUSION

The paper reviews the basic PSO technique since its invention in 1995 by Kennedy and Eberhart. The basic idea of Evolutionary techniques is given and the importance of PSO over other techniques is described. The crucial parameters used in PSO update rule of the algorithm are discussed in detail giving the significance of each parameter.

Due to the flexibility and versatility of this algorithm, it has great potential for use in a variety of real life systems. It gives a promising optimization technique that can be used in problems with complex search spaces by adjusting some parameters which are problem-specific.

REFERENCES

[1] Devender Kumar Saini and Dr. Rajendra Prasad, "Order reduction of linear interval systems using particle swarm optimization", International Journal of Engineering Science and Technology, Vol. 2, pp.316-319, 2010.

[2] James Kennedy and Russell Eberhart, "Particle swarm optimization", Proc. IEEE Int. Conf. on Neural Networks, 1995, Vol. 4, pp. 1942-1948.

[3] Riccardo Poli, James Kennedy and Tim Blackwell, "Particle swarm optimization: An overview", Swarm Intelligence, 2007, pp. 33-57.

[4] Yuhui Shi and Eberhart, R.C., "Empirical study of particle swarm optimization", Evolutionary Computation, CEC 99, Proceedings of the 1999 Congress, Vol. 3, pp.1945-1950.

[5] Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science, CCSE, 2010, Vol. 3, pp. 180-184.

[6] Mudita Juneja and S.K.Nagar, "Comparative Study of Model Order Reduction using Combination of PSO with Conventional Reduction Techniques", Proc. IEEE Int. Conf. of Industrial Instrumentation and Control ICIC 2015; Pune, India; 2015: pp. 406-411.

[7] Russell C. Eberhart and Yuhui Shi, "Particle Swarm Optimization: Development, Applications and Resources", Proc. Of the 2001 Congr. On Evolutionary Computation, 2001, Vol. 1, pp. 81-86.

[8] James Kenedy, "Particle Swarm Optimization" in Encylopedia of Machine Learning, Washington DC, USA: Springer, 2010, pp. 760-766.

[9] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski, "Particle Swarm Optimization", AIAA Journal, Vol. 41, No. 8, pp. 1583-1589, 2003.

[10] Marco Dorigo et al., "Particle Swarm Optimization", Scholarpedia, Vol. 3, No. 11, pp. 1486, 2008.

[11] J. Robinson, S. Sinton and Y. Rahmat-Samii, "Particle swarm genetic algorithm and their hybrids: optimization of a profiled corrugated horn antenna", Proc. IEEE Int. Symp. Antennas Propagation, vol. 1, pp. 314-317.

[12] Russell C. Eberhart and Yuhui Shi, "Comparison between genetic algorithms and particle swarm optimization" in Evolutionary Programming VII, Springer Berlin Heidelberg, 1998, pp.611-616.

[13] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems", Proc. 2002 ACM Symp. Applied Computing (SAC'2002), pp. 603-607.

[14] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems", Evol. Comput., vol. 7, pp. 205-230, Fall 1999.

[15] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge Proc", IEEE International Conference on Evolutionary Computation, 1997, pp. 303-308.

[16] J.Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proc. IEEE International Conference on Neural Networks, 1995, vol. IV, pp. 1942-1948.

[17] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", Proc. 1997 Conf. on Systems Man and Cybernetics, 1997, pp. 4104-4109.
[18] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator", Proceedings of the 1999 Congress on Evolutionary Computation, 1999, pp. 1958-1962.
[19] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance", Proc. Congress on Evolutionary Computation 1999, 1999, pp. 1931-1938.
[20] A.E. Eiben, James E Smith, Introduction to Evolutionary Computing, Springer, 2015.
[21] D.A. Linkens and H.O. Nyongesa, "Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications", IEE Proceedings - Control Theory and Applications, Vol. 143, Issue 4, pp. 367-386, Jul 1996.
[22] D. Whitley, "A genetic algorithm tutorial", Statistics and Computing Springer, Vol. 4, Issue 2, pp. 65–85, 1994.
[23] Sudhir Y Kumar, PK Ghosh and S Mukherjee,"Model Order Reduction using Bio-inspired PSO and BFO Soft-Computing for Comparative Study", International Journal of Information Systems and Communications(IJISC), Vol. 1, pp. 43-53, Jun. 2011.
[24] Mudita Juneja and Sheela Tiwari, "Reduced order modeling of triple link inverted pendulum using particle swarm optimization algorithm", International Journal of Advancements in Electronics and Electrical Engineering, Vol. 3, Issue 3, pp. 77-82, 2014.
[25] S. Panda, J. S. Yadav, N. P. Patidar and C. Ardil , "Evolutionary Techniques for Model Order Reduction of Large Scale Linear Systems", Electrical, Electronics and Computer Science (SCEECS), IEEE Students' Conference , 2009, pp 1-6.
[26] Ruei-Yang Wang, Wei-Po Lee, Yu-Ting Hsiao, "A New Cooperative PSO Approach for the Optimization of Multimodal Functions", Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference , pp. 103-109.
[27] Toshiharu Hatanaka, Takeshi Korenaga, Nobuhiko Kondo, Katsuji Uosaki, "Search Performance Improvement for PSO in High Dimensional Space", in Particle Swarm Optimization, Japan: Intech, 2009.