

Hybridization of Particle Swarm Optimization - A Survey

Geetika¹, Jaswinder Singh²

¹Department of Computer Engineering, UCoE, Punjabi University, Patiala 147002, Punjab, India

²Assistant Professor, Department of Computer Engineering, UCoE, Punjabi University, Patiala 147002, Punjab, India

Abstract: Hybridization is the burning topic now-a-days. Therefore, extensive studies are taking place on this topic. It leads to more efficiency and robustness of the hybridized algorithms. Hybrid algorithms can be used to solve various set of problems like scheduling, engineering design problems, medical image processing, data clustering, geometric place optimization problems etc. In all, it can be said that hybridization is very promising technique. It offers potential advantages. Therefore, this paper describes the survey of various hybridized algorithms.

Keywords: Differential Evolution, Genetic Algorithm, Hybridization, Particle Swarm Optimization, Pursue Mobility Model

1. Introduction

With the advent of wireless networks, many fundamental changes can be observed in the field of networking and telecommunication. Optimization of node placement in all networks is a critical issue. Since Particle Swarm Optimization deals with this problem in a very effective way, so the placement of nodes in Wireless Sensor Networks can be dealt with Particle Swarm Optimization. Mobility is an important issue in infrastructure less networks. Previous studies show that hybridization is an effective way to solve the optimization problems. So it is expected that on hybridizing Particle Swarm Optimization with one of the Mobility model, the results would be more desirable.

2. Particle Swarm Optimization

Particle Swarm Optimization is abbreviated as „PSO“. It is a novel population-based stochastic search algorithm. This concept was given by Eberhart and Kennedy in 1995. It was influenced by the social behavior of bird flocking. It is a process of group communication to share individual knowledge. For example, when a group of birds or insects search food in the surroundings. It is not compulsory that all of them know where the food is. But if any member found a desired path to the food, the rest of the members will follow it quickly.

In PSO, each member of the population is called a particle and the population is called a swarm. First of all, population is randomly initialized. The particles can move in any direction. Each particle remembers the best previous position of itself and its neighbors. Particles dynamically adjust their own position and velocity derived from the best position. Finally, all particles fly towards the best.

2.1 Principle of PSO

PSO contains the set of moving particles known as „Swarm“. The following are the features of the particle:

- It has a position and a velocity.

- It remembers its best previous position found so far.
- It remembers its neighbours' best previous position.

Neighborhood can be defined as a physical neighbourhood. It takes distances into account. In practice, distances are computed at each step, which makes it quite costly. At each step, the behaviour of a given particle is a compromise between three possible choices:

- To follow its own way
- To move towards its previous best position
- To move towards the best neighbour

Y. Shi and R. Eberhart (1995) gave its mathematical equations as below:

$$v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id}) \quad (1)$$

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

Where v_{id} = velocity

x_{id} = present position

p_{id} = best previous position found so far

p_{gd} = globally best position

$rand()$, $Rand()$ = random functions in range [0,1]

w = inertia weight

c_1, c_2 = positive constants

2.2 PSO Algorithm Parameters

There are some parameters in PSO algorithm that may affect its performance. These are swarm size, number of iterations, V_{max} , acceleration constants and velocity elements.

- Swarm size: Swarm size or population size is the number of particles „ n “ in the swarm. Large number of particles increases the computational complexity per iteration. Hence, makes PSO more time consuming. Various researchers use an interval of $n \in [20, 60]$ for the swarm size.
- Iteration numbers: PSO algorithm also depends upon number of iterations. A small number of iterations may stop the search process prematurely. While a large number of iterations make PSO complex and more time consuming

- V_{max} : V_{max} is an important parameter. It determines fitness between the present position and the target position. If V_{max} is high, particles may overpass good solution. If V_{max} is low, particles may not explore sufficiently.
- Acceleration Constants: The acceleration constants c_1 and c_2 represent acceleration terms that pull each particle towards pbest and gbest positions. If the values are low, the particles will move away from the target. If the values are high, the particles will move towards the target.
- Velocity Components: There are three velocity components.
 - a) The term $w * v_{id}$ is called inertia component. This component represents a momentum which prevents change in the direction of the particles. It forces them to move towards the current direction.
 - b) The term $c_1(p_{id} - x_{id})$ is called cognitive component. It keeps a track of the best previous position of the particle. Therefore, it can backtrack to the previous best position whenever need arises.
 - c) The term $c_2(p_{gd} - x_{id})$ is called social component. It records the best position of the whole group. Hence each particle can move towards best position found by the neighborhood particles.

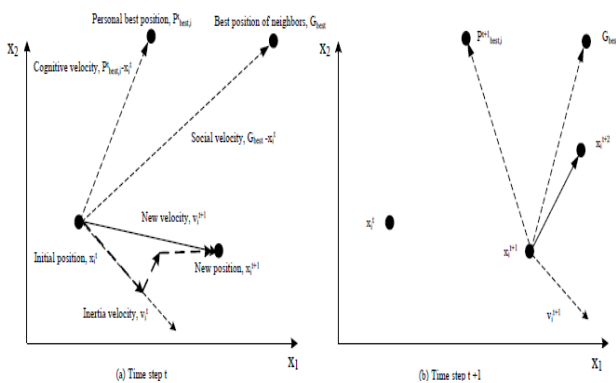


Figure 1: This figure illustrates how the three velocity components contribute to move the particle towards the global best position at time steps.

Source: Satyobroto Talukder (2011)

2.3 Advantages of PSO

- It is simple to implement.
- It quickly converges to a good solution.
- It makes use of only primary mathematical operators.
- It is fast, cheap and efficient.
- There are few parameters to regulate.

2.4 Limitations of PSO

The major drawback of PSO is „premature convergence“. It does not give quality solution as the number of iterations is increased.

3. Embedding of PSO with other Algorithms

This section contains survey on hybrid algorithms. In this, PSO is one of the prime algorithms. Researchers have enhanced the performance of PSO by incorporating various

algorithms like Genetic Algorithm, Hidden Markov Model, Differential Evolution, Simulated Annealing, Ant Colony Optimization etc. in it.

3.1 Hybridization of PSO with GA

Grimaldi et al. (2004) designed genetical swarm optimization (GSO). It is a new hybrid genetical swarm algorithm for electromagnetic optimization. It is a combination of GA and PSO. It is used for solving combinatorial optimization problems and electromagnetic optimization problems. In GSO, the population is divided into two parts. It is evolved with GA and PSO in each iteration. Then it is recombined to get the updated population. The whole process is repeated in the next iteration. A new parameter HC (or Hybridization Constant) is defined. It expresses the percentage of population that is evolved with GA in every iteration. When $HC = 0$, it implies that the procedure is pure PSO. It means the whole population is updated according to PSO operators. $HC = 1$ implies that the procedure is pure GA. If the value of HC lies between 0 and 1, it implies that the corresponding percentage of the population is developed by GA and the rest with PSO.

Juang (2004) proposed another hybridization strategy of PSO and GA called as HGAPSO. In this, the new term „elite is introduced. It is defined as the upper half of the best performing individuals in a population. The algorithm is enhanced by means of PSO instead of reproducing the population directly. It is used to design neural/fuzzy network.

Settles et al. (2005) introduced a GA and PSO hybrid called the breeding swarm (BS) algorithm. It combines the operators like selection, crossover and mutation of GA with the standard velocity and position update rules of PSO. Additional parameter called the breeding ratio is also included. It determines the size of the population which undergoes breeding in the current generation. In this algorithm, firstly „n“ best individuals are copied into a temporary population. Next, some individuals are selected to undergo the velocity and position update rules of PSO. Then selection of remaining individuals is done by some selection criteria for crossover and mutation criteria. Finally, the temporary population is mirrored into the working population and the fitness is evaluated. This process is repeated till we get optimum result. The BS algorithm was inspected on four unconstrained scalable optimization problems for different dimensions.

3.2 Hybridization of PSO with DE

Kannan et al. (2004) implemented DE to each and every particle for a finite number of iterations. Then every particle is replaced with the best individual obtained from the DE process.

Batauche et al. (2004) proposed an algorithm named DEPSO. Here, the DE operators are applied only to the best particle obtained by PSO. This is used for solving medical image processing problems. Das et al. (2008) proposed a scheme of adjusting velocities of the particles in PSO with a DE operator.

3.3 Hybridization of PSO with Ant Colony Optimization

Shelokar et al. (2007) proposed a hybrid of PSO with Ant Colony Optimization (ACO). It is called PSACO. It is a two stage algorithm. The PSO is used in the first stage and ACO is used in the second stage. Here, ACO works as a local search procedure to update the positions found by the particles. Hendtlass et al. (2001) hybridizes ACO with PSO. This algorithm records the best positions found so far. From these positions, best neighborhood is randomly selected.

3.4 Hybridization of PSO with Simulated Annealing

He et al. (2007) proposed another hybrid of PSO with Simulated Annealing (SA). It is used for solving constrained optimization problems. In this, in order to help the algorithm escape from local optima SA was applied to the best solution of the swarm.

3.5 Hybridization of PSO with Hidden Markov Model

A novel method of HMM learning has been developed by the researchers. L. Jiang et al. (2006) uses PSO for training HMM in speech recognition. The proposed approach is then compared with the Baum-Welch algorithm and genetic HMM training method. It was concluded that it is superior to both the Baum-Welch and GA-HMM training methods.

3.6 Hybridization of PSO with Pursue Mobility Model

The Pursue Mobility Model is influenced by the scenario where several nodes attempt to capture the single leading mobile node. This mobility model can be used in target tracking. The node being pursued is called as target node. The pursuer nodes are called as seeker nodes. The target node moves freely towards the target position. The seeker nodes try to copy the target node. For e.g. chasing of the thief by the police. Here, police track the thief.

The following are the steps of Pursue Mobility Model.

- Select and store the target position.
- Register the location of pursuing nodes.
- Update the current position of the pursuing node iff the distance of pursuing node is less than the distance of target node.

The new position is calculated as

```
new_pos = old_pos + acceleration (target-old_pos) +  
random_vector (3)
```

The value of random vector is calculated by an entity mobility model. Therefore, in the above e.g., the position of the police is evaluated by adding old position of the thief and acceleration value with random vector.

P. Manzoni et al. (2001) gave its pseudo code which is as follows.

```
public static Pos move_pursue(int node) {  
    Pos p=(Pos)loc.get(new Integer(node));  
    double d=p.distance(target);  
    // min movement  
    if(d<15.0)d=15.0;  
    // max movement  
    if(d>50.0)d=50.0;  
    int xi=(int)(Math.random()*d/3);
```

```
    int yi=(int)(Math.random()*d/3);  
    if (p.isLeft(target)) xi+=p.x;  
    else xi=p.x-xi;  
    if (p.isUp(target)) yi+=p.y;  
    else yi=p.y-yi;  
    return new Pos(xi,yi);  
}
```

4. Conclusion

PSO is a powerful optimization technique. It can be applied to wide range of optimization problems. But its performance can be enhanced many folds by hybridization. This technique can be used to solve complex problems such as engineering design problems, medical image processing etc. This paper will motivate and help the researchers who are working in this field to develop new hybrid algorithms. On a concluding note, it can be said that hybridization of algorithms is an interesting and promising field. It can give us more insights regarding the behavior and potential advantages and disadvantages of different algorithms.

References

- [1] C – F. Juang, “A hybrid of genetic algorithm and particle swarm optimization for recurrent network design”, IEEE Transactions On Systems, Man, And Cybernetics-Part B: Cybernetics 34, 2004, pp. 997–1006.
- [2] C. Onwubolu, C. Godfrey, B.V. Babu, “New Optimization Techniques in Engineering”, Berlin Heidelberg: Springer, 2004, ch. 1-8, pp. 1-220.
- [3] Chao-Wei Chou, Jiann-Horng Lin, Chorng-Horng Yang, Hsien-Leing Tsai, Ya-Hui Ou, "Constructing a Markov Chain on Particle Swarm Optimizer", Innovations in Bio-Inspired Computing and Applications (IBICA), 2012 Third International Conference on. IEEE, pp. 13-18, 2012.
- [4] Chao-Wei Chou, Jiann-Horng Lin, and Rong Jeng, "Markov Chain and Adaptive Parameter Selection on Particle Swarm Optimizer", International Journal on Soft Computing 4.2, 2013.
- [5] Chu, Suet Yee, "Particle swarm optimization", 2009.
- [6] E.A. Grimaldi, F. Grimacia, M. Mussetta, P. Pirinoli, R.E. Zich, “A new hybrid genetical – swarm algorithm for electromagnetic optimization”, In Proceedings of International Conference on Computational Electromagnetics and its Applications, Beijing, China, pp. 157–160, 2012.
- [7] Eberhart, Russell C., and James Kennedy, "A new optimizer using particle swarm theory", In Proceedings of the sixth international symposium on micro machine and human science, Vol. 1, 1995.
- [8] Eberhart, Russell C., and Yuhui Shi, "Particle swarm optimization: developments, applications and resources", In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Vol. 1, 2001.
- [9] H. Talbi, M. Batouche, “Hybrid particle swarm with differential evolution for multimodal image registration”, In Proceedings of the IEEE International Conference on Industrial Technology 3, pp. 1567–1573, 2004.

- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization", In Proceedings of IEEE International Conference on Neural Networks, volume IV, Perth, Australia, pp. 1942-1948, 1995.
- [11] Liping Xue, Junxun Yin, Zhen Ji, Lai Jiang, "A particle swarm optimization for hidden Markov model training", In Proceedings of the 8th International Conference on Signal Processing, Vol. 1. IEEE, 2006.
- [12] M. Settles, T. Soule, "Breeding swarms: A GA/PSO hybrid", In Proceedings of Genetic and Evolutionary Computation Conference, pp. 161-168, 2005.
- [13] Miguel Sánchez, Pietro Manzoni, "ANEJOS: A Java based simulator for ad hoc networks", Future Generation Computer Systems, Volume 17, Issue 5, March 2001, pp. 573-583.
- [14] P.S. Shelokar, P. Siarry, V.K. Jayaraman, B.D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization", Applied Mathematics and Computation 188, 2007, pp. 129-142.
- [15] Q. He, L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization", Applied Mathematics and Computation 186, 2007, pp. 1407-1422.
- [16] R. V. Kulkarni, Ganesh Kumar Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41.2, 2011, pp. 262-267.
- [17] S. Das, A. Abraham, A. Konar, "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives", Advances of Computational Intelligence in Industrial Systems, Studies in Computational Intelligence, Springer Verlag, Germany, 2008, pp. 1-38.
- [18] S. Kannan, S.M.R. Slochanal, P. Subbaraj, N.P. Padhy, "Applications of particle swarm optimization techniques and its variants to generation expansion planning", Electric Power Systems Research 70, 2004, pp. 203-210.
- [19] Shi, Yuhui, and Russell C. Eberhart, "Empirical study of particle swarm optimization", In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Vol. 3, 1999.
- [20] Shi, Yuhui, and Russell C. Eberhart, "Parameter selection in particle swarm optimization", Evolutionary Programming VII. Springer Berlin Heidelberg, 1998.
- [21] T. Hendtlass, M. Randall, "A survey of ant colony and particle swarm meta-heuristics and their application to discrete optimization problems", In Proceedings of the Inaugural Workshop on Artificial Life, pp. 15-25, 2001.