International Journal of Current Engineering and Technology
©2015 INPRESSCO®, All Rights Reserved

E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

Closeness based Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients

Sambhavi Tiwari†* and Archana Singh‡

†Department of Computer Science and Information Technology, SHIATS, Allahabad, India

Accepted 10 July 2015, Available online 14 July 2015, **Vol.5, No.4 (Aug 2015)**

Abstract

A Particle Swarm Optimization algorithm maintains a swarm of particles, where each particle has position vector and velocity vector which represents the potential solutions of the particles. These vectors are updated from the information of global best (Gbest) and personal best (Pbest) of the swarm. All particles move in the search space to obtain optimal solution. In this paper a new concept is introduced of calculating the velocity of the particles with the help of Euclidian Distance concept. This new concept helps in finding whether the particle is closer to Pbest or Gbest and updates the velocity equation accordingly. By this we aim to improve the performance in terms of the optimal solution within a reasonable number of generations.

Keywords: *Pbest; Gbest; optimization; uni-modal; multi-modal*

1. Introduction

Nature provides some of the efficient ways to solve problems. The nature inspired techniques generally work much better than other approach. Many intelligent algorithms are inspired by nature, including genetic algorithm, ant colony optimization, artificial immune systems, artificial fish swarm algorithm, and particle swarm optimization are based on the concept of population or swarms.

Inspired by the social cooperative and competitive behavior of bird flocking and fish schooling, Kennedy and Eberhart [R. Eberhart and J. Kennedy *et al*,1995] proposed a new optimization technique called particle swarm optimization (PSO).

The motivation behind this method was based on the simulation of animal social behaviors like fish schooling, bird flocking and many more.

PSO has drawn widespread attention in the last decades. Like other evolutionary algorithms particle swarm algorithm starts with the random initialization of a population of individuals in the search space. But in PSO there is no direct recombination of genetic material between individuals during the search. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual during the search. This algorithm works on the social behavior of particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual toward its own best location and toward the best particle of entire swarm at each generation (time step) [R. Eberhart and J. Kennedy *et al*,1995; M. Clerc *et al*,1999; M. Clerc and J.Kennedy *et al*, 2002].

In simple language, the particles are flown through a multidimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors, following two components are evaluated:

1) Position of the particle(X_{id})
2) Velocity of the particle(V_{id})

To calculate these two modules of PSO, a swarm of particles having position vector and velocity vector of the ith particle in the d-dimension search space can be represented as $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})$ and $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{id})$respectively. By using fitness function, can be unimodal or multimodal in nature suppose the best position of each particle i.e., best fitness value obtained by that particle at time t is Pbest $= (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{id})$, and the fittest particle found till now at time t is Gbest= $(p_{g1}, p_{g2}, p_{g3}, \ldots, p_{gd})$. Then, for calculating the new velocities and the positions of the particles for next fitness evaluation following equations is used:

$$V_{id} = V_{id}+c_1*rand_1\ (.)*(p_{id} - x_{id})+c_2*rand_2\ (.)*(p_{gd} - x_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

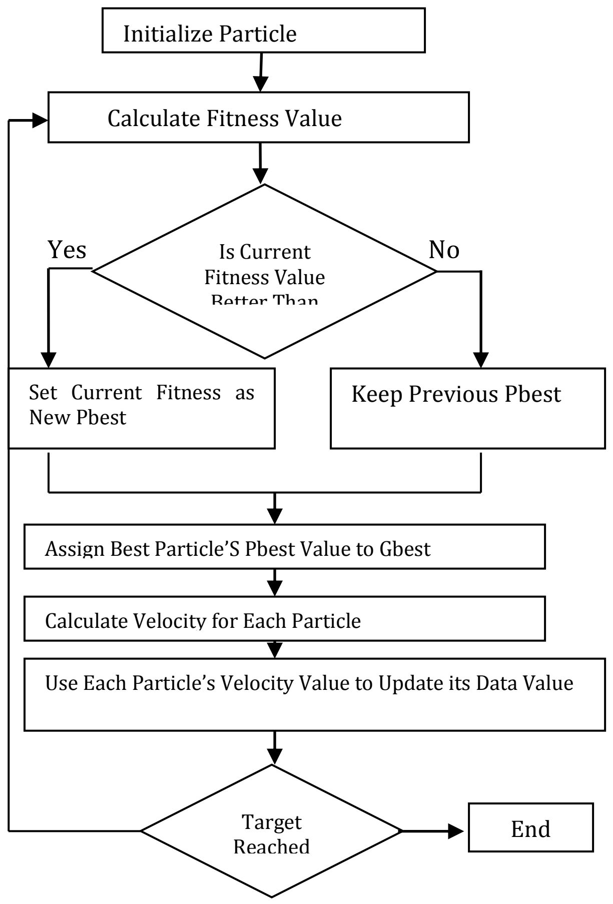
Where c_1 and c_2 are positive acceleration constants used to scale the contribution of cognitive and social components respectively and $rand_1$ and $rand_2$ are two

*Corresponding author **Sambhavi Tiwari** is a M.Tech Scholar and **Archana Singh** is working as Assistant Professor separately generated uniformly distributed random numbers in range [0,1].

The first part of (1) represents the previous velocity, which provides the necessary momentum for particles to roam across the search space. The second part, known as the cognitive component, represents the personal thinking of each particle. The cognitive component encourages the particles to move toward their own best positions found so far. The third part is known as the social component, which represents the collaborative effect of the particles, in finding the global optimal solution. The social component always pulls the particles toward the global best particle found so far. Initially, a population of particles is generated with random positions, and then random velocities are assigned to each particle. The fitness of each particle is then evaluated according to a user defined objective function. At each generation, the velocity of each particle is calculated according to (1) and the position for the next function evaluation is updated according to (2).Each time if a particle finds a better position than the previously found best position, its location is stored in memory.

Generally, a maximum velocity (V_{max}) for each modulus of the velocity vector of the particles (V_{id}) is defined in order to Control excessive roaming of particles outside the user defined search space. Whenever a V_{id} exceeds the defined limit, its velocity is set to $V_{max.}$

The flowchart for basic particle swarm optimization is shown below:


Initialize Particle
Calculate Fitness Value
Yes
Is Current Fitness Value Better Than
No
Set Current Fitness as New Pbest
Keep Previous Pbest
Assign Best Particle'S Pbest Value to Gbest
Calculate Velocity for Each Particle
Use Each Particle's Velocity Value to Update its Data Value
Target Reached
End

Each particle of swarm modifies its position according to:

1) Its current position.
2) Its current velocity.
3) Distance between its current position and Pbest.
4) And distance between its current position and Gbest.

In the next section, a discussion on related works on PSO is presented. Then our proposed work, result and conclusion are discussed in later sections.

2. Related Previous Work

Many improved versions have been reported in the literature due to its simplicity and effectiveness since it was first introduced. In this paper four variants of PSO are used i.e., basic PSO, Time varying inertia weight PSO, hierarchical PSO and MPSO which are discussed in research work [A. Ratnaweera, S. K. Halgamuge, and H. C. Watson *et al*, 2004] which stated that the lack of population diversity in PSO algorithm is a factor that makes them prematurely converge to local optima. Many approaches of diversity control have been introduced in order to avoid the whole swarm converging to a single optimum. In 2010, diversity control was implemented by preventing too many particles from getting crowded in one region of the search space. Another one which adds negative entropy into PSO in [X. Li *et al*,2004] to avoid premature convergence. Li (2004) developed a speciation based PSO, which dynamically adjust the number and size of swarms by constructing an ordered list of particles, ranked according to their fitness, with spatially close particles joining a particular species. Another an atomic swarm approach was adopted to track multiple swarms in dynamic environments by Blackwell and branke *et al*,2006. Recently a clustering PSO algorithm has been proposed in [S. Yang and C. Li *et al*,2010], where a hierarchical clustering method is used to produce multi swarms in effective regions in search space.

Hybrid evolutionary algorithms are becoming more and more popular due to their capabilities in handling problems that involve complexity, noisy environments, imprecision, uncertainty and vagueness. The first hybrid PSO algorithm was developed by by Angeline [P. Angeline *et al* ,1998],where a selection scheme was introduced. Recently, a PSO version with adaptive ω, $\eta 1$, and $\eta 2$, called adaptive PSO (APSO), has been proposed by Zhan *et al*. Liang *et al*. developed a comprehensive learning PSO (CLPSO) for multimodal problems. The other variant aims to improve PSO by introducing heuristic or non-heuristic mechanism, and there are various mechanisms including restarting mechanism, which has been widely used like Keiji Eatsumi *et al*. in 2009 proposed a restarting multi swarm PSO (RMSPSO) algorithm.

3. Proposed Work

In this paper we are using a new concept of closeness based evaluation on the swarm. As we know heuristic approaches have not necessarily been proven to produce the global minimum with every trial or to be applicable in all cases. Rather, they have been demonstrated to work well in general. Since, PSO is population based heuristic search and the speed of population based search heuristic can be measured in iterations, function evaluations or real time. Since, each particle evaluates its function value at each iteration the number of function evaluations conducted per iteration is equal to the number of search agents. Function evaluation seems to be most popular measure. Real time is not generally used since the time required to run simulation on one computer might not equal the time required on another computer, making real time comparison from paper to paper is practically impossible.

The optimization problem is then to find values of the variables that minimizes or maximizes the objective function while satisfying the constraints.

Generally in population based optimization method, it is desirable to encourage the individuals to wander through the entire search space without clustering around the local optima, during the early stages of the optimization. On the other side, during the latter stages, it is very important to enhance convergence toward the global optima, to find optimum solution efficiently.

Considering these concerns, in this paper we propose new concept of calculating the distance among the particles which helps in determining the closeness towards Gbest or Pbest, named as closeness based method. This concept is applied with HPSO-TVAC which yields a new algorithm termed as closeness based HPSO with TVAC, which uses TVAC as new parameter strategy for the PSO concept and on the basis of Euclidian distance between the particle and Pbest and particle and Gbest elements of the swarm are accelerating towards the optimal solution. This concept depicts the problem of minimization clearly.

Firstly, in TVAC like ratnaweera *et al* has proposed in his work, we reduce the cognitive component and increase the social component by changing the acceleration coefficient c1 and c2 with time. This is known as PSO-TVAC method.

Secondly, Kneddy and Eberhart *et al* proposed a version of PSO without the velocity of previous iteration. Later they concluded that since this version is very simple, it is ineffective in finding global optimia for complex problems. To overcome this problem ratnaweera *et al* proposed HPSO to provide the required momentum for particles to find global optimum solution in the absence of previous velocity term in (1).

Lastly, our new concept is introduced here to enhance the performance furthermore which yields good results. The closeness is calculated with the help of Euclidian distance among the particles. If the particle is more closer to Gbest then move that particle toward Gbest by reducing the cognitive factor c1 and increasing the social factor c2 by specific value in velocity update equation (1) else vice-versa.

Hence, a significant improvement of performance is observed with this new closeness based HPSO with TVAC method and also proves its acceptance for minimization problem of optimization.

Algorithm for Closeness based HPSO with TVAC:

```
1.  Generate random population of particle with
    random position and velocity in search space.

2.  Set the parameters of the algorithm as:

        $C_1$_min = 0.5
        $C_2$_max = 2.5
        $C_2$_min = 0.5
        $C_2$_max = 2.5
        W_min = 0.4
        W_max = 0.9
        $C_1 = 2.5$
        $C_2 = 0.5$

3.  Find initial function values of the swarm by using
    fitness function.

4.  Find the local best (Pbest) position of ith particle.
5.
6.  Find the global best(which is best among personal
    best (Pbest)) position of the swarm, i.e., Gbest.

7.  For i=1 to $I_{max}$
        Calculate the varying coefficient factors i.e.,
        $C_1$_var, $C_2$_var and W_var to upgrade the
        acceleration coefficients and inertia weight.

8.  Calculate the Euclidian distance for each particle
    between
        a. Particle's and Gbest's position ($D_{XG}$).
        b. Particle's and Pbest's position ($D_{XP}$).

9.  Update the value of inertia weight.
        W=W-W_var
10. Check if $D_{XG} < D_{XP}$
        then update $C_1 = C_1 - C_1\_var$
                    $C_2 = C_2 + C_2\_var$
        Else        $C_1 = C_1 + C_1\_var$
                    $C_2 = C_2 - C_2\_var$
11. Update the velocity ($V_{id}$) of each particle by the
    velocity vector equation.

        $\boxed{V_{id} = V_{id} + c_1 * rand_1(.) * (p_{id} - x_{id}) + c_2 * rand_2(.) * (p_{gd} - x_{id})}$

12. Update the position $X_{id}$ of each particle by position
    vector equation.  $\boxed{X_{id} = X_{id} + V_{id}}$
```

13. Update the velocity and position of Gbest particle.
14. Repeat step 3 to step 13 until termination criteria is met (maximum number of iteration).
15. Stop

4. Experimental Setting and Simulation Strategies

4.1 Benchmarks functions

To compare performance, both in terms of optimum solution after a predefined number of iterations, and the rate of convergence to optimum solution, of the new developments introduced in the proposal Closeness based(CB) HPSO with TVAC algorithm with basic PSO, PSO-TWIW, HPSO, and HYBRID(HPSO+TVAC+MPSO) algorithms, both uni-model and multi-model well known five benchmarks function are used. The global minimum of all the benchmarks function are at origin. Table I shows the mathematical representation of the benchmarks functions used in this dissertation.

Table I Benchmarks Functions

Name of the function	Mathematical representation
Sphere function	$f_1(x)=\sum_{i=1}^{n} x_i^2$
Rosenbrock function	$f_2(x)=\sum_{i=1}^{n}\left[100\left(x_{i+1-}x_i^2\right)^2+(x_i-1)^2\right]$
Rastrigin function	$f_3(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$
Griewank function	$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
Schaffer's f6 function	$f_6(x)=0.5-\frac{\left(\sin\sqrt{x^2+y^2}\right)^2-0.5}{\left(1.0+0.001\left(x^2+y^2\right)\right)^2}$

4.2 Experimental setting for Benchmarks functions

Each benchmarks function have some search range and initialization range in the search space. Table II shows the initialization range of the search for the benchmarks functions.

Table II Initialization for Benchmarks Functions

Function	Search Range	Initialization Range	V_{max}
Sphere	$(-100,100)^n$	$(-100,100)^n$	10
Rosenbrock	$(-100,100)^n$	$(-2.048,2.048)^n$	100
Rastrigin	$(-10,10)^n$	$(-5.12,5.12)^n$	10
Griewank	$(-600,600)^n$	$(-600,600)^n$	600
Schaffer's f6	$(-100,100)^2$	$(-100,100)^2$	10

4.3 Simulation strategy for proposed algorithm

Shi and Eberhart observe that the effect of population size have minimum significance on the performance of the PSO algorithm. However, in PSO research, it is quite common to limit the size of population to the range 20 to 60. Engelbrecht and Van Den Bergh suggested that there is a slight improvement of the optimal value with increasing the size of the population. Therefore in this paper, all benchmarks simulations were carried out with a population size of 10, 40 and 100. During the starting stage of PSO algorithm symmetric initialization method was used and during later stage asymmetric initialization method was used. In symmetric initialization, the population is initialized only in a portion of the n-dimensional search space. Since the above benchmarks function have the global minimum closed to the origin of the search space, hence asymmetric initialization method was used at later stage of the algorithm.

The benchmarks functions were tested on dimensions 2, 10, 20 and 30 with iterations 30,40,50 respectively. The values of mean median, maximum, minimum and standard deviation for 50 trials are carried out, and compared with the values of basic PSO, PSO-TVIW, HPSO, and HYBRID(HPSO+TVAC+MPSO) algorithms. And experiment results proved that CB HPSO with TVAC is better.

a. Mean: average of optimal fitness values taken through the simulation results.
b. Median: the median is middle of a distribution
c. Maximum: the maximum value of optimal fitness taken through the simulation results.
d. Minimum: the minimum value of optimal fitness taken through the simulation results.
e. Standard Deviation: Square root of variance.

Table III Comparison among different minimum values obtained for PSO variants where population size=5, Iter=100 and dimension=2

Function	PSO	PSO-TVIW	HPSO	HPSO+TVAC+MPSO
Rastrigin	3.203	5.459	3.888	3.173
Griewank	0.952	0.874	1.138	3.321
Rosenbrock	0.014	0.139	0.005	0.496
Schaffer's f6	0.056	0.182	0.056	0.056
Sphere	52.63	47.88	89.56	20.39

Function	HPSO+TVAC+MPSO	Closeness based
Rastrigin	3.173	3.165
Griewank	3.321	0.673
Rosenbrock	0.496	0.004
Schaffer's f6	0.056	0.307
Sphere	20.394	2.217

Conclusions

We have proposed an improved minimization based particle swarm optimization algorithm in this paper. This algorithm improves ability of finding minimum value for the benchmarks functions and to explore search space iteratively towards optimal solution with the help of Euclidian distance calculated each time among the particle's position with the Gbest and Pbest particle position. The proposed algorithm is tested on five well known benchmarks functions and fits best for two dimension particles. Experiments proved that the proposed algorithm outplays other PSO algorithm. The proposed algorithm has proper control on local optima and global optimum. This algorithm performs consistently and efficiently improves optimum solution in the search space.

References

R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in proc. 6th Int. Symp. Micro Machine Human Sci., 1995, pp.39-43.

Kennedy, J.; Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948.

M. Clerc, The swarm and the queen: Toward a deterministic and adaptive particle swarm optimization, in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 3, 1999, p. 1957.

M. Clerc and J.Kennedy (Feb. 2002), The particle swarm—Explosion, stability, and convergence in a multi-dimensional complex space, IEEE Trans. Evol. Comput., vol. 6, pp. 58–73,

A. Ratnaweera, S. K. Halgamuge, and H. C. Watson (Jun. 2004), Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 240–255.

(2010), Particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration, Engineering Applications of Artificial Intelligence, vol. 23, no. 8, pp. 1340–1349.

X. Xie, W. Zhang, and Z. Yang (2002), Dissipative particle swarm optimization, in Proc. Congr. Evol. Comput, vol. 2, pp. 1456–1461.

X. Li (2004), Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization, in Proc. Genetic Evol. Comput. Conf., pp. 105–116.

T. M. Blackwell and J. Branke (Aug. 2006), Multiswarms, exclusion, and anticonvergence in dynamic environments, IEEE Trans. Evol. Comput., vol. 10, no. 4, pp. 459–472.

S. Yang and C. Li (Dec. 2010), A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput., vol. 14, no. 6, pp. 959–974.

P. Angeline (1998), Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences, in Proc. 7th Conf. Evol. Program., pp. 601–610.