

Algorithm for modified particle swarm optimization economics essay

[Economics](#)



**ASSIGN
BUSTER**

Generally, in population-based optimization methods, it is desirable to encourage the individuals to wander through the entire search space, without clustering around local optima, during the early stages of the optimization. On the other hand, during the later stages of search, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently. The proposed Modified Particle Swarm Optimization (MPSO) have non linearly varying acceleration coefficients and inertia weight that will improve the quality of solution. The concept of time-varying acceleration coefficients (TVAC) and time-varying inertia weight factor (TVIW) is used to effectively control the global search and convergence to the global best solution. The major consideration of this modification is to avoid premature convergence in the early stages of the search and to enhance convergence to the global optimum solution during the later stages of the search. Considering these concerns, a method is proposed for time varying acceleration coefficients and inertia weight as a new parameter adaptation strategy for the PSO concept. The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search. Under this new development, the cognitive component is reduced and the social component is increased with time by changing the acceleration coefficients with time. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the

latter part of the optimization. In this modified PSO (MPSO), time varying acceleration coefficients and time varying inertia weight is used which varies in accordance with the iterations. The changes in the modified PSO (MPSO) with respect to original PSO are: The value of cognitive coefficient $C1$ is varied non-linearly as a Trigonometric function instead of the constant $C1$ as used in the standard PSO for proper global exploration of the search space. The value of social coefficient $C2$ is varied non linearly as a Trigonometric function instead of the constant $C2$ as used in the standard PSO for proper local exploitation and to increase the accuracy in calculating the optimum solution. The value of inertia weight ω_k is varied non linearly over the generations instead of the constant ω as used in the standard PSO.

4. 2 Algorithm for Modified Particle Swarm Optimization (MPSO):

If $P_i(k)$ & $P_i(k+1)$ are current position coordinate of each particle in k th and $(k+1)$ th iteration. $P_{best_i}(k)$ is the self best explored position found by the particle in k th iteration. $G_{best}(k)$ is global best position found among all the particles in k th iteration where k is the iteration counter and k_{max} is the maximum number of iterations. $R1$ and $R2$ are two random values with uniform distribution in the interval $[0, 0, 1, 0]$, $R1$ is called the cognitive random factor and $R2$ is called the social learning random factor. $C1$ is the cognitive acceleration coefficient (it controls the pull to the personal best position) and $C2$ is the social acceleration coefficient (it controls the pull to the global best position). $i = 1, 2, \dots, i$ is particle index. ω_k is inertia weight, whose value decreases during iterations from 0.9 to 0.4 in MPSO. The proposed algorithm for MPSO is: 1. Generate a random distribution of n

particle position. 2. Initialization of the parameters: k_{max} , C_{max} , C_{min} , ω_{max} , ω_{min} , β and n . 3. For $k = 1$ to k_{max} do. 4. Set $C1 = C_{min} + (C_{max} - C_{min}) * \sin(\pi (1 - k/k_{max})/2)$ and $C2 = C_{min} + (C_{max} - C_{min}) * \cos(\pi (1 - k/k_{max})/2)$. 5. Set $\omega_k = \omega_{min} + (\omega_{max} - \omega_{min}) * (1 - k/k_{max})^\beta$. Find particle with best position ($G_{best}(k)$), for $i = 1$ to n . 6. Find best position explored by i th particle up to now ($P_{best}(k)$). 7. Generate velocity vector for each particle:..... (4. 1) 8. Guide particle i to the new position:.....(4. 2) 9. Present $G_{best}(k_{max})$ as the optimum point. Here velocity vector formula in equation (4. 1) is modified by using time varying cognitive coefficient and social coefficient. The value of cognitive coefficient $C1$ varies from 2.5 (when $k = 0$) to 0.5 (when $k = k_{max}$) based on the following formula: $C1 = C_{min} + (C_{max} - C_{min}) * \sin(\pi (1 - k/k_{max})/2)$ (4. 3) The value of social coefficient $C2$ varies from 0.5 (when $k = 0$) to 2.5 (when $k = k_{max}$) as the relation: $C2 = C_{min} + (C_{max} - C_{min}) * \cos(\pi (1 - k/k_{max})/2)$ (4. 4) The value of inertia weight ω_k is based on the formula: $\omega_k = \omega_{min} + (\omega_{max} - \omega_{min}) * (1 - k/k_{max})^\beta$ (4. 5) Here, $C_{max} = 2.5$, $C_{min} = 0.5$, $k =$ iteration, $k_{max} =$ maximum iterations, $\beta = 0.4$, $\omega_{max} = 0.9$ and $\omega_{min} = 0.4$.

4.3 Description of the benchmark functions:

This is a very popular set of test functions which are commonly used to test the performance of evolutionary algorithms [13]. These functions are chosen due to their particularities, which include several factors, from one simple function with single minimum to one having a considerable number of local minima with similar fitness values. Each function is shown in figure for the full search space.

4.3.1. Dejong

This is a simple case with only one minimum. Any optimization technique should solve it without problem. Its simplicity helps to focus on the effects of dimensionality in optimization algorithms. Its global minimum locates at $x = [0, \dots, 0]^n$ with function $f(x) = 0$. Objective Function: $f(x) = \sum_{i=1}^n x_i^2$
 (4.6) Search Space: $\{x_i : -100 < x_i < 100\}$, Dimensionality: 10, 20 and 30.
 Global Minimum: at $x_i = 0$

•

Figure. 4. 1 DeJong function

4.3.2. Rastrigin

The Generalized Rastrigin Function is a typical example of a non-linear multimodal function, which is characterized by deep local minima arranged as sinusoidal bumps. The global minimum is at $x = [0, \dots, 0]^n$ with function $f(x) = 0$. This function is a difficult problem due to its large search space and its large number of local minima. An optimization algorithm can easily be trapped into a local minimum. Figure. 4. 2 Rastrigin function
 Dimensionality: 10, 20 and 30, Global Minimum: at $x_i = 0$. Objective Function: $f(x) = \sum_{i=1}^n (10.66 - x_i) \cos(x_i)$
 (4.7) Search Space: $\{x_i | : -10 < x_i < 10\}$

4.3.3. Griewank

This function is strongly multi-modal with significant interaction between its variables caused by the product term. The global optimum is at $x = [100, \dots, 100]^n$ with function $f(x) = 0$, which is almost indistinguishable from closely packed local minimum that surrounds it. This function has the interesting property that the number of local minima increases with

dimensionality. On one hand, this tends to increase the difficulty of the problem but on the other hand, because the distances between two adjacent local minimum are quite small, it is very easy to escape from them, in case of stochastic algorithms. Objective Function: $f(x) = 1 + \dots\dots\dots(4.8)$
 Search Space: $\{x_i \mid -600 < x_i < 600\}$ Dimensionality: 10, 20 and 30.
 Global Minimum: at $x_i = 0$. Figure. 4. 3 Griewank function

4. 4 Parameters Selection

Initial particles of the swarm are distributed throughout the search space using a uniform random number generator. To evaluate the performance of MPSO parameters used are: 1. Initialization: During the early stages of the development of the PSO method [1], symmetric initialization was widely used, where the initial population was uniformly distributed in the entire search space. In MPSO, asymmetric initialization method is used as proposed by Ratnaweera and Chang [13, 30], in which the population is initialized only in a portion of the search space. Since most of the benchmarks used, have the global minimum at or close to the origin of the search space, so asymmetric initialization method is used to observe the performance of the new developments. The most common dynamic ranges [8] used in the literature for the benchmarks considered were used and the same dynamic range is used in all dimensions, Table 4. 1 shows the range of population initialization and the dynamic range of the search. Here n is the dimension.

Table 4. 1 Range of search and range of initialization for the test functions

Function

Range of search

Range of initialization

Dejong F1(-100, 100) n (50, 100) n Rastrigin F2(-10, 10) n (2. 56, 5.

12) n Griewank F3(-600, 600) n (300, 600) n 2. Inertia Weight ω K: This

coefficient is used to control the trajectories of particles where k is the

iteration number, k_{max} is the maximum number of iterations, $\omega_{max} = 0.9$,

$\beta = 0.4$ and $\omega_{min} = 0.4$. It decreases from 0.9 to 0.4 over the iterations by

the relation given in equation (4.5). 3. Acceleration Coefficients ($C1$ and $C2$)

: These are mostly used in the community of particle swarms, the value $C1$

varies from 2.5 to 0.5 as in equation (4.3) and the value of $C2$ varies from

0.5 to 2.5 as in equation (4.4). $C_{max} = 2.5$ and $C_{min} = 0.5$. 4. Maximum

Velocity: V_{max} is set as the upper limit of the search space. Maximum

velocity for different functions is as follows: Table 4.2 Maximum velocity for

different functions

Function

V_{max}

F1100F210F36005. Population Size: There is no universal rule to determine

the number of particles in a swarm. In this benchmark test, swarm size is

taken to be forty. 6. Maximum number of iterations: it is a termination

criterion. This is the maximum number of iterations for which the program is

run. It is set to be different corresponding to the various test functions for

different dimensions in the experiment.

4.5 Test procedure for MPSO:

MPSO is a non-linearly decreasing inertia weight method. In this proposed method population-based search is applied for high diversity during the early part of the search to allow use of the full range of the search space and, during the later part of the search it finds the global optima efficiently. Here velocity vector formula is modified by using time varying cognitive coefficient, social coefficient and inertia weight. The behavior of C_1 , C_2 and ω with increasing iterations can be understood from the following Figure 4.

4. Figure 4.4 Behaviour of C_1 , C_2 and ω with increasing iterations As it can be seen from the Figure 4.4 during early part of the search, value of cognitive coefficient C_1 is maximum and it is decreasing from 2.5 (when $k=0$) to 0.5 (when $k = k_{max}$) by trigonometric function. Value of social coefficient C_2 is minimum during early stage of search and its value is increasing from 0.5 (when $k = 0$) to 2.5 (when $k = k_{max}$) by trigonometric function. Also here inertia weight ω_k is decreasing non-linearly with time and its value decreases from 0.9 to 0.4 over the iterations. To achieve high diversity during early part of the search, value of cognitive coefficient C_1 is kept maximum in the initial stage while for more accuracy in the end, value of social coefficient C_2 is kept maximum in the end over the generations.

5.1 System specifications: All the experiments were performed on system with Intel(R), core [TM] 2 Duo CPU and 1.97GHz, 2 GB of RAM. Software used is Matlab 7.6.0.324 (R2008a) version. Operating system is Microsoft Windows XP, version 5.1 (Build 2600: service pack 2).

4. 5. 2 Statistic values for measurement

Following measurements are used to evaluate the performance of MPSO: 1. Mean Value f_m : It is the arithmetic average of all the results from the tests on the same problem, which is denoted as: where n is the total amount of numerical tests or number of trials and f_i is the result for the i th test. This value describes the average performance of an algorithm. Algorithm runs for 50 trials for each function. 2. Best Result f_b : It is the best optimal solution obtained from all of the tests, which is used to assess the search ability of a global optimization algorithm i. e. whether this algorithm can find a global optimum with the best known result known or not. 3. Worst result f_w : It is the worst optimal solution obtained from all of the tests, which is used to evaluate the stability of a optimization algorithm. 4. Time T : is the evaluation time in seconds for running 50 trials of each experiment. 5. Standard deviation (std): standard deviation of the optimal mean values for 50 trials for each experiment is calculated.

4. 6 RESULTS AND DISCUSSION:

MPSO algorithm's performances on the three test benchmark functions for different dimensions are discussed as below:

4. 6. 1 Algorithm's performances on De Jong Function:

Performance of proposed algorithm for De Jong function is measured for the dimensions of 10, 20 and 30 corresponding to maximum iterations of 1000, 2000 and 3000 respectively with stopping criterion of 0. 01 as considered by Ratnaweera and Chang [13, 30]. Table 4. 3 Algorithm's performance on De Jong Function

Dejong function F1**Dimension****Maximum****Iterations****Kmax****Mean Optimal****value****Standard****Deviation****(std)****Time****(seconds)****Best value****fb****Worst value****fw**

1010000. 010. 0005. 7740. 0050. 0102020000. 010. 00013. 0330. 0070.

0103030000. 010. 00027. 0120. 0060. 079

Figure 4. 5 shows the comparison graph for MPSO with other PSO variants for Mean Optimal value of the Dejong function plotted in Matlab. As it can be seen in the Figure 4. 5 PSO-TVIW, PSO-TVAC, MPSO and PSO-RANDIW converges to same mean optimal value 0. 01 for dimension equal to thirty and maximum iterations i. e. kmax= 3000. Figure 4. 5 Comparison Graph of the Dejong function

4. 6. 2Algorithm's
<https://assignbuster.com/algorithm-for-modified-particle-swarm-optimization-economics-essay/>

performances of Rastrigin Function: To obtain the performance of proposed algorithm for Rastrigin function, it is measured for the dimensions of 10, 20 and 30 corresponding to maximum iterations of 3000, 4000 and 5000 respectively with stopping criterion of 0.01 [13, 30]. Table 4. 4 Algorithm's performance on Rastrigin function

Rastrigin functionF3

Dimension

Maximum

Iterations

Kmax

Mean Optimal

value

Standard

Deviation

(std)

Time (seconds)

Best value

fb

Worst value

fw

1030001. 242071. 13439. 5560. 0036. 96420400010. 13603. 20452. 4863. 00518. 17730500027. 43686. 18484. 34911. 94145. 885

Figure 4. 6 shows the comparison graph for MPSO with other PSO variants for Mean Optimal
<https://assignbuster.com/algorithm-for-modified-particle-swarm-optimization-economics-essay/>

value for the Rastrigin function plotted in Matlab. It can be seen in the Figure 4. 6 MPSO has the lowest mean optimal value of 27. 43684 for dimension equal to thirty. PSO-TVIW, PSO-TVAC, MPSO and PSO-RANDIW converges to mean optimal values as shown in figure for dimension equal to thirty and maximum iterations i. e. kmax= 5000. Figure 4. 6 Comparison Graph of the Rastrigin function

4. 6. 3 Algorithm's performances on Griewank Function: To obtain the performance of proposed algorithm for Griewank function is measured for the dimensions of 10, 20 and 30 corresponding to maximum iterations of 3000, 4000 and 5000 respectively with stopping criterion of 0. 01 [13, 30]. Table 4. 5 Algorithm's performance on Griewank function

Griewank function F4**Dimension****Maximum****Iterations****Kmax****Mean Optimal****value****Standard****Deviation****(std)****Time****(seconds)****Best value****fb****Worst value****fw**

1030000. 047680. 023058. 3400. 0090. 1072040000. 022130. 012478.

7290. 0020. 0463050000. 014970. 012497. 7700. 0010. 059

Figure 4. 7 shows the comparison graph for MPSO with other PSO variants for Mean

Optimal value for the Griewank function plotted in Matlab. Figure 4. 7

Comparison Graph of the Griewank functionAs it can be seen in the Figure 4.

7 MPSO has the lowest mean optimal value of 0. 014970 for the dimension of

thirty. PSO-TVIW, PSO-TVAC, MPSO and PSO-RANDIW converges to mean

optimal values as shown in figure for dimension equal to thirty and maximum iterations i. e. kmax= 5000.

•

4. 7 Comparison of mean optimal values for various time varying PSO

MPSO algorithm is implemented on three test bench functions and comparison is drawn between PSO-RANDIW, PSO-TVIW and PSO-TVAC [30] and the proposed MPSO for various dimensions and iterations. These test benchmarks functions are used because they are previously used by Ratnaweera and Chang [13, 30]. Comparison of Mean value and standard deviation of the optimal values for 50 trials for various PSO algorithms with MPSO is drawn in Table 4. 6. Table 4. 6 Mean optimal value and standard deviation of the optimal valuefor 50 trials for various PSO algorithms

Function**Dimension****Max. Iterations****Mean optimal value****(Standard Deviation)****PSO-TVIW****PSO-RANDIW****PSO-TVAC****MPSO**

Dejong F11010000. 01(0. 000)0. 01(0. 000)0. 01(0. 000)0. 01(0. 000)2020000. 01(0. 000)0. 01(0. 000)0. 01(0. 000)0. 01(0. 000)3030000. 01(0. 000)0. 01(0. 000)0. 01(0. 000)0. 01(0. 000)Rastrigin F21030002. 069(1. 152)4. 63(2. 366)2. 268(1. 333)1. 24207(1. 134)20400011. 74(3. 673)26. 293(8. 176)15. 323(5. 585)10. 13609(3. 204)30500029. 35(6. 578)69. 7266(20. 700)36. 236(8. 133)27. 43684(6. 184)Griewank F31030000. 0675(0. 029)0. 0661(0. 030)0. 05454(0. 025)0. 047685(0. 023)2040000. 0288(0. 023)0. 0272(0. 025)0. 0293(0. 027)0. 022135(0. 012)3050000. 0167(0. 013)0. 0175(0. 018)0. 0191(0. 015)0. 014970(0. 012)

4. 8 Discussion

The proposed Particle Swarm optimization (MPSO) algorithm has been analysed and tested using three test benchmark functions i. e. Dejong function, Rastrigin function and Griewank function. The evaluation of this proposed MPSO is based on these well-known and widely used benchmark test functions which contains characteristics that have difficult global

<https://assignbuster.com/algorithm-for-modified-particle-swarm-optimization-economics-essay/>

optimization problem for evolutionary algorithms. Comparison is drawn among the PSO-RANDIW, PSO-TVIW and PSO-TVAC and the proposed MPSO. On this basis, it can be concluded that MPSO outperforms the existing linear time varying PSO for all the three functions. MPSO achieved the best solution for the optimal values among the other three PSO variants.