Parameter tuning for enhancing performance of a variant of particle swarm optimization algorithm

Ashok Kumar[1], Sheo Kumar[2], Rajesh Tiwari[2], Shalya Saxena[3], Anurag Singh[3]
[1]College of Computer Science and Information Technology, Teerthanker Mahaveer University, Moradabad, India
[2]Department of Computer Science and Engineering, CMR Engineering College, Hyderabad, India
[3]Department of Computer Science and Engineering, G L Bajaj Group of Institutions, Mathura, India

Article Info

Article history:

Received Mar 17, 2024
Revised Aug 6, 2024
Accepted Aug 11, 2024

Keywords:

Global optima
Local optima
Particle swarm optimization
Premature convergence
Stagnation

ABSTRACT

There is dependably an extraordinary requirement for new types of algorithms in the population-based improvement algorithm. These algorithms improve the execution of the current algorithm. Parameter change approach assumes an essential job in improving the execution of the PSO algorithm. A new algorithm called particle acceleration-based particle swarm optimization (PA-PSO) has been proposed. In this algorithm a particle acceleration parameter is tuned. This algorithm significantly improves the performance of the PSO–time varying acceleration coefficients (PSO-TVAC) algorithm. This algorithm reduces the time varying weight of inertia and the nonlinear acceleration coefficients in the equation of the PSO-TVAC velocity vector in each iteration. Particle movements in the n-dimensional search space are governed by the kinetics of the second motion equation. Experiments demonstrate that the proposed PA-PSO algorithm outperforms the existing PSO-TVAC algorithm on five well-known reference test functions. The algorithm possesses adequate control over the local as well as global optimums.

Corresponding Author:

Ashok Kumar
College of Computer Science and Information Technology, Teerthanker Mahaveer University
N.H.-9, Delhi Road, Moradabad, 244001, Uttar Pradesh, India
Email: ashchhonkar@gmail.com

1. INTRODUCTION

Optimization is the procedure to locate the best answer to meet a specific objective and, in the meantime, fulfill the prerequisite [1]. Instances of effective improvement ideas in nature are swarms that pursue fruitful individuals, colonies of ants looking to amplify their nourishment sources by ensuring plant lice colonies or flying creatures trying to limit the number of creatures surrounding the food by following and communication with each other and all simultaneously [2], [3]. An optimization issue is the issue of finding the most ideal arrangement among all arrangements [4].

There are many optimization techniques; some of them are deterministic such as algebraic techniques. Out of the stochastic optimization techniques, there are techniques like Branch-and-bound or Monte Carlo sampling, parallel techniques discussed in [5]. In addition to this there are heuristic and meta-heuristic optimization techniques such as genetic algorithms (GAs) [6], evolutionary strategies (ESs) [7], genetic programming (GP) [8], and ant colony optimization (ACO) [9].

In 1995, Drs. Russell Eberhart and James Kennedy introduced a different method called particle swarm optimization (PSO) [10] that is based on swarm intelligence. The functioning of this approach was impacted by biological assessment and natural species selection. This technique makes use of a population of distinct particles, each of which has a location, a speed, also a remembrance of the optimal physical condition discovered throughout the search [8]. Individual particle brings up to date its velocity depending on its momentum, remembrance, and participate information of the particles nearby it, by addition the particle's recently originate velocity to its present location. The particle will relocate to a fresh point in the exploration area [11], [12]. The PSO in its original form is well-defined in the speed update calculation defined by the (1):

$$v_{id}^{t+1} = v_{id}^{t} + c_1.r_1.(P_{id}\text{-}x_{id}^{t}) + c_2.r_2.(P_{gd}\text{-}x_{id}^{t}) \quad (1)$$

position upgrade equation is defined by the (2):

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

where particle position vector is denoted by x_{id} that contains the present position of each particle's solution in the search space while term v_{id} denote element velocity vector that contains the degree to which vector x_{id} will vary in both direction and magnitude in the next iteration.

SimilarlyThe value of P_{id} indicates the most optimal solution of the target equation found by an individual particle. In contrast, the symbol P_{gd} indicates the optimal global solution of the target equation found by the entire population of particles. The symbols c_1 and c_2 denote the corresponding learning factors that affect a particle's highest ranking and globally optimum location. The symbols r_1 and r_2 indicate random numbers.

The Scientist Shi and Eberhart in 1998 [13] established the inertia element ω, that improves substantially the PSO algorithm's search capabilities. The inertia element ω controls how prior velocities affect the current velocity. As a result, it modulates the exchange between a particle's global and local minima [14]–[15]. When ω is between 0.8 and 1.2, the PSO is most likely to discover the universal best after a reasonable number of repetitions. Secrest and Lamont [16] suggests starting with a big value of 1.4 for ω and linearly decreasing it to 0.5 for faster convergence [17]. The inertia factor (ω) might as (3):

$$\omega = \omega_{max} + (\omega_{max} - \omega_{min}) * \text{iteration}_{current}/\text{iteration}_{max} \quad (3)$$

the symbol ω represents the inertia factor, and the values ω_{max} and ω_{min} are assigned based on the problem's behavior iteration_{max} indicates the total number of iterations, whereas $\text{iteration}_{current}$ shows the present repetition digit. The velocity expression after incorporating the inertia component is as (4).

$$v_{id}^{t+1} = \omega.v_{id}^{t} + c_1.r_1.(P_{id} - x_{id}^{t}) + c_2.r_2.(P_{gd} - x_{id}^{t}) \quad (4)$$

The basis of the PSO method is the operation in (4) that changes v_{id}, which powers the particles to examine the most encouraging zones of the solution and then adds the velocity vector $v_{id}$to the vector x_{id} position to obtain another position. With unaltered v_{id} values, the particle basically takes uniform estimations straight through inquiries, and the past characteristics are the particle's v_{id}moment [18], [19]. If the scope is very small, it will have a huge computer memory and a lengthy calculation time [20]. According to previous research, 30-50 is an appropriate population size to achieve effective search space convergence and a fair computing time [21].

Many researchers continuously finding new PSO variants to optimizreal life problems. Kumar *et al.* [3] discuss PSO variants in terms of their diversity and conversion. Found that shifting the weight of inertia from ω_max at the beginning to ω_min at the maximum number of iterations improves the performance of the PSO algorithm significantly [22], [23]. Have seen that The particle swarm algorithm method's performance can be enhanced by adjusting the weight of inertia from ω_{max} at the start to ω_{min} at the most iterations possible [22], [23]. The calculation regarding inertia weight is as follows (5):

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min})\left(\frac{I_{max}-I_{current}}{I_{max}}\right) \quad (5)$$

where the numeric value of inertia ω_{min} is equal to 0.4 that represent smallest value and the value of ω_{max} is equal to 0.9 that is extreme value of inertia weight. symbol I_{max} and $I_{current}$ are maximum and current number of iteration correspondingly.

Presented an undetermined inertia weight value for tracking changing environments [24], [25]. The following equation describes how ω fluctuates randomly as (6):

$$\omega = 0.5 + \frac{r}{2} \tag{6}$$

where the symbol r denotes a uniformly distributed random number between 0 and 1.

At the beginning of the optimization process, the PSO-TVAC method aims to improve the exploration worldwide; at the end of the search, it encourages all particles to meet towards the worldwide optimum. The following equations determine how the weight of inertia, the coefficient of cognitive acceleration, and the coefficient of social acceleration evolve in this algorithm [26]-[28]:

$$c_1 = c_{1min} + (c_{1max} - c_{1min})\left(\frac{I_{max} - I_{current}}{I_{max}}\right) \tag{7}$$

$$c_2 = c_{2max} + (c_{2min}\text{-}c_{2max})\left(\frac{I_{max}\text{ - }I_{current}}{I_{max}}\right) \tag{8}$$

$$\omega = \omega_{min} + (\omega_{max}\text{- }\omega_{min})\left(\frac{I_{max}\text{ - }I_{current}}{I_{max}}\right) \tag{9}$$

Where the value of c_{1min} is equal to 0.5 that is minimum value of c_1 and the value of c_{1max} is equal to 2.5 that is maximum value of c_2. while the vale of c_{2min} is equal to 2.5 that is minimum value of c_2 and the value of c_{2max} is 2.5 that is maximum value of c_2 . The algorithm's performance is enhanced by changing c_1 from 2.5 to 0.5, c_2 from 0.5 to 2.5, and ω from 0.9 to 0.4 over the search space.

The work provides a more complete knowledge of PSO behavior and performance under a variety of environmental situations. The suggested technique makes the following key contributions:

- Determine the best answer in a minimum number of generations.
- Optimize irregular parameters to escape from local optima.
- Assess diverse solution through Proper tuning of random parameters.

2. METHOD

2.1. Mathematical analysis

Acceleration is a vector term that indicates the rate at which a substance changes its velocity and given by the equation $a = (v - u)/t$, where displacement in nth second is given by the equation, $s_n = d_n - d_{n-1}$ Where the symbol d_n and d_{n-1} being his displacement position of particle at n and n-1 th second respectively. Particle acceleration based PSO (PA-PSO) introduces the following modification over original PSO:

- Acceleration matrix of particles are calculated as follows:

A. Initialize initial velocity u_{id} matrix with current velocity matrix v_{id}.

$$u_{id}^t = v_{id}^t \tag{10}$$

B. Velocity matrix of particles are updated by using the velocity equation of basic PSO with value of c_1 , c_2 and ω is adjusting according to PSO-TVAC:

$$v_{id}^{t+1} = \omega.v_{id}^t + c_1.r_1.(P_{id} - x_{id}^t) + c_2.r_2.(P_{gd} - x_{id}^t) \tag{11}$$

C. Calculation of acceleration matrix:

$$a_{id}^{t+1} = v_{id}^t - u_{id}^t \tag{12}$$

- After calculating acceleration position of swarm is calculated by using the second equation of motion as follows:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} + a_{id}^{t+1}/(2 * (t + 1)) \tag{13}$$

2.2. Proposed PA-PSO algorithm

A new parameter acceleration to particle is found out after that position of particle is updated by using the second equation of motion.

```
Input: Population size (pop), dimension (dim), fitness function f(x) with its constraints
and maximum iteration (Imax).
```

```
Output: Function f(x)'s optimal fitness value and the particle's location at which it occurs.
The proposed algorithm PA-PSO includes the steps listed below:
1.  Create a position matrix x_id of order pop by dim and initialize it with random numbers
    within search space.
2.  Create a velocity matrix v_id of order pop by dim and initialize it with random numbers
    between 0 and 1.
3.  Calculate fitness vector f (x_id).
4.  Create vector personal best P_id and initialize it with fitness vector.
5.  Set the parameters' initial values to c_1min = c_2min = 0.5 and c_1max = c_2max = 2.5 and ω_min =
    0.4 and ω_max = 0.9.
6.  For iteration t = 1 to I_max
7.  Configure the algorithm's parameters as follows (7)-(9)
8.  For id = 1 to pop.
9.  Find Personal best P_id and Global best P_gd
    If f (x_id) < P_id
    P_id = x_id
10. Find Global best P_gd
    If f (x_id) < P_gd
    P_gd =x_id
11. Initialize initial velocity of particle as (10)
12. Utilize the velocity vector equation to update the particle's velocity as (11)
13. Calculate Acceleration of particle as (12)
14. Update the position of particle by the position vector as (13)
15. Repeat 7-12 steps (id becomes equal to pop).
16. Repeat 6-14 step (t becomes equal to I_max ).
17. Obtain global best position.
```

3. RESULT AND DISCUSSION

The proposed PA-PSO method is compared to the prior PSO-TVAC algorithm using five famous standard functions [18]. The origin contains the global minimum of all standard functions. Table 1 displays a mathematical illustration of the benchmark functions used in this study. Each benchmark function has a search and starting range in the search space. Table 1 displays the benchmark function search's initialization range. Where n is the number of dimensions in the search space. All tests are done using MATLAB 2013 running on MS Windows 7. The system has a 2.1 GHz core-i3 CPU and 2 GB of RAM.

Table 1. Initialization for benchmark functions

Function	Mathematical Representation	Search range	Initialization range	Vmax
Rastrigrin	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10Cos(2\pi x_i) + 10]$	$(-10,10)^n$	$(2.56, 5.12)^n$	10
Griewank	$f(x) = \sum_{I=1}^{n} \frac{x_i^2}{4000} - \prod_{i}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$(-600,600)^n$	$(300, 600)^n$	600
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100\ (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$(-100,100)^n$	$(15, 30)^n$	100
Sphere	$f(x) = \sum_{I=1}^{N} (x_i^2)$	$(-100,100)^n$	$(50,100)^n$	100
Schaffer's f_6	$f(x) = 0.5 - \frac{sin^2\sqrt{x^2 + y^2 - 0.5}}{[1.0 + 0.001(x^2 + y^2)]^2}$	$(-100,100)^n$	(15, 30)	100

Eberhart and Shi [20] point out that population size has minimal impact on the PSO method's performance. Though, in PSO studies, Thist is typical to restrict the population size to between 20 and 60. According to Bergh and Engelbrecht [29], increasing the population size results in a modest improvement in the ideal value. As a result, all benchmark functions are carried out with a population of 40. The PA-PSO method uses symmetric initialization at the beginning and asymmetric initialization at the end. In symmetric initialization, the initial population is spread evenly over the n-dimensional search space. In asymmetric initialization, the population is initialized in a subset of the n-dimensional search space. Because all benchmark functions have a global minimum near to the origin of the search space, an asymmetric initialization strategy is utilized later in the algorithm [30]. The Table 2 contains the median value corresponding to PSO-TVAC algorithm and PA-PSO method applied on five standard tasks. for each benchmark task, the value of mean, median, maximum, minimum and standard deviation for 50 trials are carried out but we are taking only median value to compared with the PA-PSO algorithm and PSO-TVAC algorithm. The experiments show that the proposed algorithmPA-PSO is better than the PSO-TVAC algorithm.

Table 2. Comparison of PSO-TVAC and PA-PSO over dimensions

Function name	Dimension	Iteration	PSO-TVAC	PA-PSO	Difference in median
Rastrigin	10	3000	Median:2.943032	Median: 1.989918	0.953114
	20	4000	Median:16.148309	Median: 12.934463	3.213846
	30	5000	Median:37.187278	Median: 28.853793	8.333485
Griewank	10	3000	Median: 0.057058	Median: 0.07622	-0.019162
	20	4000	Median: 0.031420	Median: 0.027027	0.004393
	30	5000	Median: 0.022178	Median: 0.011933	0.010245
Rosenbrock	10	3000	Median: 10.878330	Median: 5.503124	5.375206
	20	4000	Median: 20.144692	Median: 12.980339	7.164353
	30	5000	Median: 29.603539	Median: 19.316403	10.287136
Sphere	10	3000	Median: 0.013086	Median: 0.012597	0.000489
	20	4000	Median: 0.012546	Median: 0.012115	0.000431
	30	5000	Median: 0.012237	Median: 0.011950	0.000287
Schaffer's f_6	2	3000	Median: 0.004479	Median: 0.002941	0.001538
	2	4000	Median: 0.005316	Median: 0.002857	0.002459
	2	5000	Median: 0.005116	Median: 0.002931	0.002185

Figure 1 demonstrate that PA-PSO algorithm is giving better result as compare with PSO-TVAC algorithm on rastrigin function at 10, 20, and 30 dimensions on 3000, 4000, and 5000 iterations respectively. Similarly Figure 2 demonstrate that PA-PSO algorithm is giving better result as compare with PSO-TVAC algorithm on griewank function at 10, 20, and 30 dimensions on 3000, 4000, and 5000 iterations respectively. Figures 3 and 4 demonstrate that PA-PSO algorithm is giving better result as compare with PSO-TVAC algorithm on rosenbrock function and sphere function at 10, 20, and 30 dimensions on 3000, 4000, and 5000 iterations respectively.

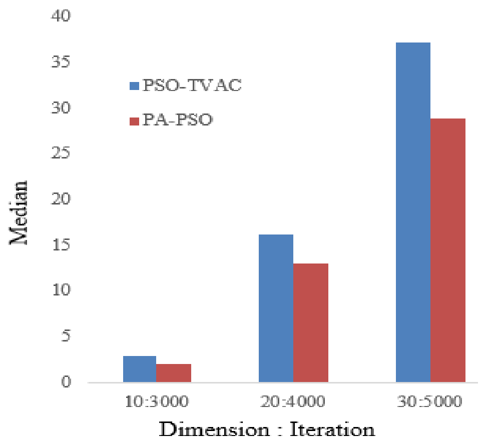

Figure 1. Comparison of PSO-TVAC and PA-PSO on rastrigrin function

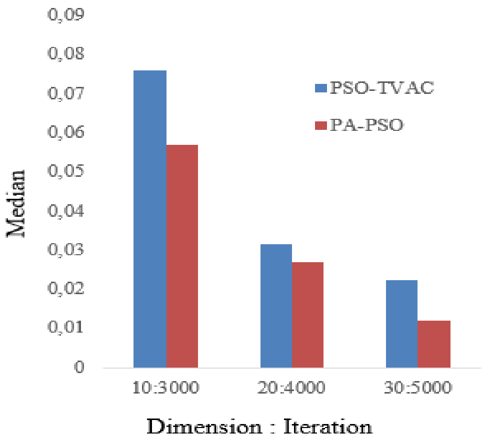

Figure 2. Comparison of PSO-TVAC and PA-PSO on griewank function

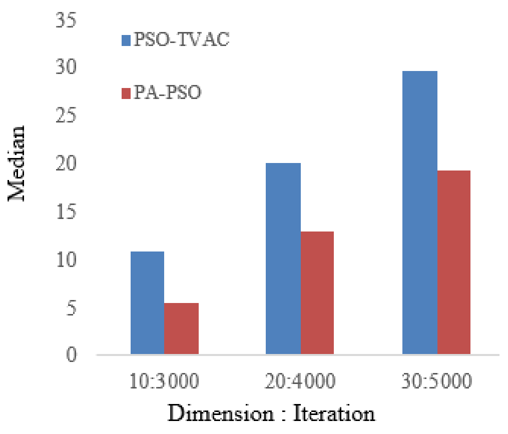

Figure 3. Comparison of PSO-TVAC and PA-PSO on rosenbrock function

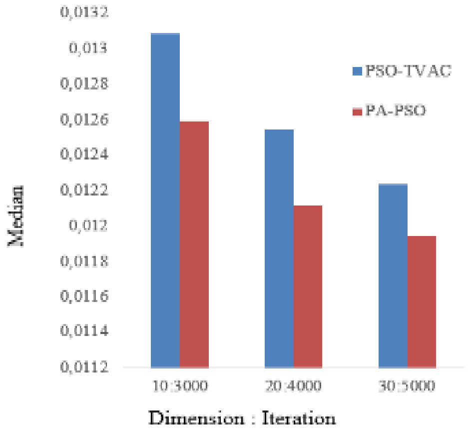

Figure 4. Comparison of PSO-TVAC and PA-PSO on sphere function

Figure 5 demonstrate that PA-PSO algorithm is giving better result as compare with PSO-TVAC algorithm on schaffer's f_6 function at 2 dimension on 3000, 4000, and 5000 iterations respectively. It is observed that in increasing dimension resultant solution gets worst but with increasing maximum iteration, the solution gets improved so benchmark functions Rastrigrin, Griewank, Rosenbrock and Sphere are verified on dimension 10, 20 and 30, with iterations 3000, 4000, and 5000 respectively for achieving better results. Only Schaffer's F6 function is tested on dimension 2, because it is a two-dimensional function. For each benchmark task a corresponding graph is plotted by taking the median values of the objective function both algorithm, PSO_TVAC and PA-PSO are plotted over 3000, 4000, and 5000 iterations respectively. The x-axis represents the dimension and iteration and y axis represents corresponding value in median. Figure 1 to Figure 5 shows the comparison of PA-PSO algorithm with PSO-TVAC algorithm for Rastrigrin, Griewank, Rosenbrock, Sphere, and Schaffer's f_6 functions respectively. The accepted solution is when the median of PA-PSO is lesser than median of PSO-TVAC. We can see that in most of the cases it is so. We can see that PA-PSO algorithm is doing well as compared with PSO_TVAC.

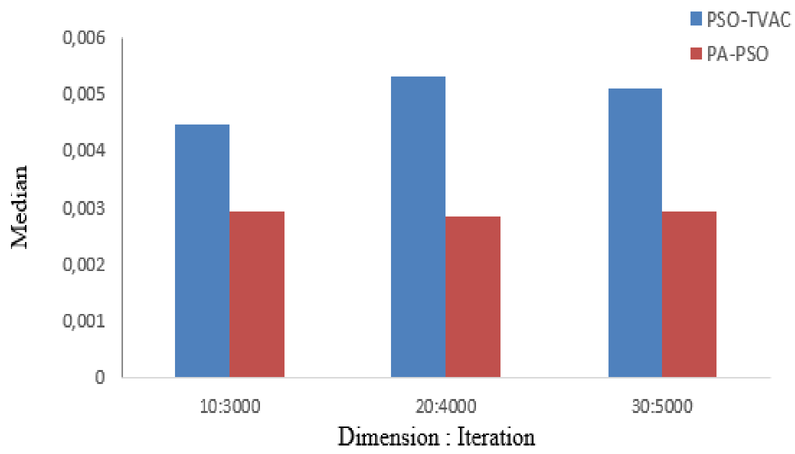

Figure 5. Comparison of PSO-TVAC and PA-PSO on schaffer's f_6 function

4. CONCLUSION

Proposed algorithm PA-PSO is tested on five well-known function and fortunately it gives good or improved results on all of them in compare of its counterpart PSO-TVAC. Median value of both algorithms over dimension: Iteration pair (l0: 3000,20: 4000,30:5000) are compared. The median difference depicts that for Rastrigrin and Griewank function, it shows the slight improvement over PSO-TVAC on all pair of dimension and iteration. For Rosenbrock function, it shows the great improvement with increasing dimension and iteration proposed algorithm's results gets better. In case of Sphere function, PA-PSO completely overpower PSO-TVAC on low dimension: iteration (10:3000) but results get approx. Equivalent on high dimension: iteration (20:4000, 30:5000). In case of Schaffer's f_6 function, PA-PSO gives the same result on each pair of dimensions: iteration. Proposed PSO reduced the median value from 0.0044 to 0.0029.

The examinations demonstrated that the proposed procedure PA-PSO remains more grounded than the other PSO-TVAC method. This calculation has sufficient power over the local optima and global and global optimal. It likewise shows consistent execution and improved ideal arrangements in the examine space. A great deal of future space has been seen amid the work. This algorithm can be connected to different applications as in order to limit the expense and vitality scattering in the remote sensor, decreasing the expense of building materials in the building site, and streamlining the utilization of assets in plants, and so forth. Thus it is evident that our proposed algorithm PA-PSO is giving the better results that PSO-TVAC.

REFERENCES

[1] G. Dantzig, *Linear programming and extensions*. RAND Corporation, 2018.
[2] I. Rechenberg, "Evolutions strategies in computational intelligence," *Frommann H boog, Stuttgart*, 1994.
[3] A. Kumar, B. Kumar, and B. D., "Particle swarm optimization: a study of variants and their applications," *International Journal of Computer Applications*, vol. 135, no. 5, pp. 24–30, Feb. 2016, doi: 10.5120/ijca2016908406.
[4] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report-tr06, Erciyes university, engineering faculty, computer engineering department*, vol. 200, pp. 1–10, 2005.

[5] J. R. Koza, "Genetically breeding populations of computer programs to solve problems in artificial intelligence," in *[1990] Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*, 1990, pp. 819–827, doi: 10.1109/tai.1990.130444.
[6] D. PARTRIDGE, "Handbook of genetic algorithms," *Connection Science*, vol. 3, no. 4, pp. 446–448, Jan. 1991, doi: 10.1080/09540099108946598.
[7] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Intelligent decision making through a simulation of evolution.," *Behavioral science*, vol. 11, no. 4, pp. 253–272, Jul. 1966, doi: 10.1002/bs.3830110403.
[8] J. H. Holland, "Adaptation in natural and artificial systems," *Ann Arbor: The University of Michigan press*, Aug. 1975, doi: 10.1145/1216504.1216510.
[9] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," *In Proceedings of the first European conference on artificial life*, vol. 142, pp. 134–142, 1991.
[10] A. Slowik, "Particle swarm optimization," in *The Industrial Electronics Handbook - Five Volume Set*, 2011, vol. 4, pp. 1942–1948, doi: 10.1007/978-3-319-46173-1_2.
[11] J. Kennedy, "Particle swarm: Social adaptation of knowledge," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1997, pp. 303–308, doi: 10.1109/icec.1997.592326.
[12] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43, doi: 10.1109/mhs.1995.494215.
[13] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1998, pp. 69–73, doi: 10.1109/icec.1998.699146.
[14] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 1999, vol. 3, pp. 1945–1950, doi: 10.1109/CEC.1999.785511.
[15] A. Stacey, M. Jancic, and I. Grundy, "Particle swarm optimization with mutation," in *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, 2003, vol. 2, pp. 1425–1430, doi: 10.1109/CEC.2003.1299838.
[16] B. R. Secrest and G. B. Lamont, "Visualizing particle swarm optimization - Gaussian particle swarm optimization," in *2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings*, 2003, pp. 198–204, doi: 10.1109/SIS.2003.1202268.
[17] A. Kumar, B. D. Patro, and B. K. Singh, "Parameter tuning for software effort estimation using particle swarm optimization algorithm," *International Journal of Applied Engineering Research*, vol. 14, no. 2, pp. 139–144, 2019.
[18] A. Kumar, B. K. Singh, and B. D. K. Patro, "Hybrid algorithm based on GA and PSO with improve convergence and diversity," *International Journal of Computer Science and Information Security*, vol. 15, no. 10, pp. 103–111, 2017.
[19] A. Kumar, B. R. A. J. E. S. H. K. SINGH, and B. D. K. Patro., "Diversity preserving auto improved-PSO for solving optimization problems," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 29, no. 6, pp. 531–550, 2017.
[20] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 2001, vol. 1, pp. 94–100, doi: 10.1109/cec.2001.934376.
[21] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 2001, vol. 1, pp. 81–86, doi: 10.1109/cec.2001.934374.
[22] J. Liu, W. Xu, and J. Sun, "Quantum-behaved particle swarm optimization with mutation operator," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2005, vol. 2005, pp. 237–240, doi: 10.1109/ICTAI.2005.104.
[23] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers and Operations Research*, vol. 33, no. 3, pp. 859–871, Mar. 2006, doi: 10.1016/j.cor.2004.08.012.
[24] A. J. Carlisle, "Applying the particle swarm optimizer to non-stationary environments," *Auburn University*, 2002.
[25] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, Jun. 2004, doi: 10.1109/TEVC.2004.826071.
[26] T. Xiang, X. Liao, and K. wo Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1637–1645, Jul. 2007, doi: 10.1016/j.amc.2007.02.103.
[27] A. I. Selvakumar and K. Thanushkodi, "A new particle swarm optimization solution to nonconvex economic dispatch problems," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 42–51, Feb. 2007, doi: 10.1109/TPWRS.2006.889132.
[28] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1998, pp. 84–89, doi: 10.1109/icec.1998.699327.
[29] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, Jun. 2004, doi: 10.1109/TEVC.2004.826069.
[30] H. Fan, "A modification to particle swarm optimization algorithm," *Engineering Computations (Swansea, Wales)*, vol. 19, no. 7–8, pp. 970–989, Dec. 2002, doi: 10.1108/02644400210450378.

BIOGRAPHIES OF AUTHORS

Prof. Dr. Ashok Kumar is Professor at college computing Science and Infromation Technology, Teerthanker Mahaveer University (TMU), Moradabad, Uttar Pradesh, India. He has completed his Ph.D degree in Computer Science and Engineering with specialization in Evolutionary Algorithms from Dr. A.P.J. Abdul Kalam Technical University, Lucknow, U.P., India, (Formerly UPTU, Lucknow). His Master in Engineering degree from Dr. B. R. Ambedkar University, Agra (Formerly-Agra University, Agra) in Computer Science and Engineering and also completed his Bachelor of Engineering in Computer Science and Engineering in 2000. He is an author of more than 26 Research papers and 04 Book Chapters. He also presented his papers in national and international conferences. Editorial board member of various national and international bodies. Many papers are indexed in Elsveir, Scopus, IEEE, WoS, and UGC. He has 24 years of rich experience in teaching and research. His research and skill areas are applying evolutionary algorithms to obtaion optimized results in real-life problems and machine learning. He can be contacted at email: ashchhonkar@gmail.com.

Prof. Dr. Sheo Kumar is currently working as a Professor in CMR Engineering College, Hyderabad from past six years. He was awarded his Ph.D. in Computer Science and engineering from Sri JJT University in the year 2017. His area of research is Data Mining, Artificial Intelligence and Machine Learning. He has guided more than 50 B.Tech, 14 M. Tech and 02 Ph.D Scholars. He is closely associated with students' welfare and learning system. He strongly believes in self learning and collaborative approach of learning. He can be contacted at email: sheo2008@gmail.com.

Prof. Dr. Rajesh Tiwari is working as a Professor in CMR Engineering College, Hyderabad, Telangana, India. He is having more than 26 years of Teaching and Industries experience. He has completed his Ph.D. in 2020 from Chhattishgarh Swamy Vivekanand Technical University, Bhilai, India, in the field of High-Performance Computing under Computer Science and Engineering Department. He is having 68 Research Publications, 1 Book, 10 Book chapters and 13 Patents. He has guided 27 M.Tech scholars and guiding 6 PhD scholars. He is member of various professional societies like IEEE, IE (India), ISTE and many more. Currently he is the chairperson of the Membership Development Committee of IEEE Hyderabad section of Region 10. He has chaired more than 7 National and International conferences. He has organized more than 10 International conferences. His research area is HPC, parallel computing, cloud computing, AI and ML and data science. He can be contacted at email: drrajeshtiwari20@gmail.com.

Shalya Saxena is working as an Assistant Professor in G L Bajaj, group of Institutions, Mathura, India. She is Pursuing her Ph.D in Computer Science and Engineering from GLA University, Mathura since 2023. She has completed his M.Tech. in 2023 from GLA University, Mathura, India and Completed her B.Tech in Computer Science and Engineering from GLA University, Mathura in 2018. She having more than 6 years of Teaching and Industries experience. Her area of research is discrete mathematics, cyber security. She can be contacted at email: shalya.saxena2014@gmail.com.

Anurag Singh is working as an Assistant Professor in G L Bajaj, group of Institutions, Mathura, India. he has completed his M.Tech. in 2018 from DEI University (Dayal Bagh University), Agra India and Completed his B.Tech in Computer Science & Engineering from from GLA institute of Technology and Management in 2014. He having more than 8 years of Teaching and Industries experience. His area of research is cyber security and web designing. He can be contacted at email: anuragsngh68@gmail.com.