

Sparse Based Particle Swarm Optimization Algorithm

Lalit Kumar

MANIT: Maulana Azad National Institute of Technology

Manish Pandey

MANIT: Maulana Azad National Institute of Technology

Mitul Kumar Ahirwal (✉ ahirwalmitul@gmail.com)

MANIT: Maulana Azad National Institute of Technology <https://orcid.org/0000-0001-5346-7008>

Research Article

Keywords: Optimization, Particle Swarm Optimization, Sparse Representation, Swarm Intelligence

Posted Date: May 11th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-253771/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Sparse Based Particle Swarm Optimization Algorithm

Lalit Kumar¹, Manish Pandey², and Mitul Kumar Ahirwal³

^{1,2,3}Maulana Azad National Institute of Technology, Bhopal, M.P., India

¹lalit212212@gmail.com, ²contactmanishpandey@yahoo.co.in, ³ahirwalmitul@gmail.com

Abstract: Particle Swarm Optimization (PSO) is the well-known metaheuristic algorithm for optimization, inspired from swarm of species. PSO can be used in various problems solving related to engineering and science inclusive of but not restricted to increase the heat transfer of systems, to diagnose the health problem using PSO based on microscopic imaging. One of the limitations with Standard-PSO and other swarm based algorithms is large computational time as position vectors are dense. In this study, a sparse initialization based PSO (Sparse-PSO) algorithm has been proposed. Comparison of proposed Sparse-PSO with Standard-PSO has been done through evaluation over several standard benchmark objective functions. Our proposed Sparse-PSO method takes less computation time and provides better solution for almost all benchmark objective functions as compared to Standard-PSO method.

Keywords: Optimization, Particle Swarm Optimization, Sparse Representation, Swarm Intelligence.

1 Introduction

Nature Inspired Algorithms (NIA) are the most prominent strategies developed from natural surroundings (Yang 2014) and mostly used in optimization tasks. Swarm based Intelligence algorithm is one of the category of NIA which is based on the cooperative behavior of swarm members. J. Kennedy and R. Eberhart (1995) proposed a Particle Swarm Optimization algorithm (PSO) which belongs to swarm based metaheuristic algorithm (Kennedy 2011; Marini and Walczak 2015). PSO is encouraged by the vibrant movement of insects, birds, fishes, etc. In the field of science and engineering, PSO has been utilized to tackle non-differentiable, non-linear, and multimodal optimization problems (Shi 2001; Poli 2008; Liu et al. 2016; Harrison et al. 2018) such as to achieve optimal risk investment portfolios problem (Zhu et al. 2011), to satisfy distribution of temperature and flow of heat (Farahmand et al. 2012), to integrated solar combined cycle power plants (Mabrouk et al. 2018) and many more. Limitations with respect to inadequate speed-up for reaching optimum point and unsatisfactory accuracy of PSO in some studies, motivate researchers for further development and enhancement in PSO. PSO is a simple to-actualize algorithm and furthermore has less customizable boundaries than practically equivalent other algorithms. In recent years many improvements have been done in PSO for many application areas of optimization such as Chaos based PSO (Gao et al. 2019) to improve convergence speed, diminishing population based PSO to meet the swarm on the most favorable point (Soudan and Saad 2008), FCPSO based on balancing the diversity of location to achieve convergence (Sahu et al. 2012), and many more. Nature inspired optimization techniques are also used in field of signal processing to improve the range of search space (Ahirwal

et al. 2014), to filter the noisy signals (Ahirwal et al. 2014; Ahirwal et al. 2015), to optimize adaptive noise canceller (Ahirwal et al. 2016).

Basically, the execution of PSO starts with a randomly distributed particles (population of solutions) inside the search space. As the iterations continue, the particles move as general group towards a most favorable point (Kennedy 2011). To appraise the optimality of each solution (particle) fitness function is evaluated in each cycle after that updation mechanism is applied to update the location of each particle so that they reach to optimal point and helps in convergence. This process of execution is repeated many times on particles to converge at global optima. It is challenging to process all the above steps within a reasonable amount of computation time if fitness objective function is more complex and it is difficult to achieve significant improvement also (Mahdavi et al. 2015; cheng 2016; Yang 2018; Sengupta et al. 2019). Several variants of Standard-PSO algorithms have also been developed to improve the computation time based on some application like task scheduling management (Pandey et al. 2010; Zhan and Huo 2012).

The main aim of this paper is to gain speed-up by reducing computation time and accomplish better efficiency with the help of proposed Sparse-PSO. The rest of the paper is organized as follows:

A basic description of the Standard-PSO algorithm has been provided in Section 2. The proposed Sparse-PSO algorithm has been explained in Section 3. Section 4 displays the experimental results for the proposed approach and compared it with results for Standard-PSO. Section 5 contains a summary and concluding remarks.

2 Standard-Particle Swarm Optimization

Particle Swarm Optimization is a very popular optimization algorithm as only few parameters are there in this algorithm. PSO consist of a group of particles know as swarm. PSO is an iterative method since many iterations of the procedure has to be done to achieve the optimal value. In PSO, firstly an initial population of particles is initialized by arbitrarily initializing the position and velocity vectors in the defined search space. Each particle has fitness value to check solution quality at each iteration. This fitness value is obtained through objective functions which may be of minimization or maximization type, depending upon the problem. In this algorithm, there are two significant things to note, personal best (P_b) solution accomplished up until now and the global best (G_b) particle which is the best solution among all individual best solutions achieved until now. The dynamic journey of the particle is controlled by its personal flying knowledge as well as the flying involvement of different particles (P_b , and G_b) in the swarm. Each particle's velocity and position are updated by using equations (1) and (2) respectively.

$$Vel_i^m(r+1) = W * Vel_i^m(r) + \Psi_1 h_1 * (P_{b_i}^m(r) - Pos_i^m(r)) + \Psi_2 h_2 * (G_b^m(r) - Pos_i^m(r)), \quad (1)$$

$$Pos_i^m(r + 1) = Pos_i^m(r) + Vel_i^m(r + 1), \quad (2)$$

where Vel describes the velocity direction, Pos describes the position of particle, r is the current iteration, m is the particular dimension at i^{th} particle, $W=1$ is the inertia weight, h_1 and h_2 are random numbers in the interval $(0,1)$, and Ψ_1 and Ψ_2 are positive acceleration constants.

In equation (1), first term in the velocity updation equation dominates the influence of earlier used velocities on the recent velocity or makes the particle move in the same direction with the same velocity (Shi and Eberhart 1998). Second term controls the position of particle by returning it to the previous position, if the previous position is better than the current position of particle according to the objective function. While third term controls the particle to follow the best particle in the swarm.

3 Proposed Sparse Particle Swarm Optimization (Sparse-PSO)

3.1 Proposed Initialization Process of Sparse-PSO

In this work, a particle is represented as a position vector of magnitude M , where M indicates the number of dimensions or variables. Our aim is to achieve the speed-up in the computation time of algorithm. In this effort, the population of particles called swarm is arbitrarily generated within defined boundaries of size N but in the form of sparse matrix representation to achieve our aim in reaching to the optimum point rapidly. After initialization of sparse matrix based population, each particle is evaluated according to the fitness function. However, this procedure is performed in the initial phase of algorithm.

3.2 Proposed Algorithm

In proposed method, the swarm of size N is initialized to sparse matrix as discussed in Section 3.1 (The details of the same has been described in the following sub-section). Next, test functions are used to evaluate the optimal solution for every particle of swarm set. At this point, quality of solution depends on the nature of test function that can be minimization or maximization problem. Following which, personal best and global best are selected and revised repeatedly in accordance with the quality of particles.

Sparse-PSO Algorithm:

Inputs Parameters :

Population of size N

Number of dimensions M

Number of iterations R

Acceleration constants $\Psi_1 = \Psi_2$

Density to define Sparsity of the matrix D

Output:

Global best solution

Respective position vector

Initialization:

- 1) Initially generate an arbitrarily sparse based swarm of particles of dimensions (N, M) in the search space by using below procedure, where N is the size of swarm and M is the magnitude of each particle in swarm
 - Calculate number of non-zero elements (C) such that $C = N * M * D$
 - Randomly generate C indices for each dimension of the swarm (a, b) , which pairwise define the location of non-zero values
 - For each (a, b) pair of indices generate a random value (V) in the specified range
 - Generate the sparse based swarm with given density D by assigning the value V to pairwise unique locations (a, b)
 - 2) **For** every particle
 - 3) Compute fitness of particle using objective function (Test Function)
 - 4) Set personal best (P_b)
 - 5) **End of for**
 - 6) Set global best (G_b)
 - 7) **While** number of iterations are not met
 - 8) **For** each particle
 - 9) Revise the velocity and location of particle by equations (1), and (2)
 - 10) Determine fitness of new revised particle
 - 11) Update personal best
 - 12) **End of for**
 - 13) Hence, update global best
 - 14) **End of while loop**
-

4 Experimental Results

4.1 Benchmark Functions

The efficacy of the proposed Sparse-PSO algorithm is verified using several experiments conducted on fourteen benchmark functions taken from (Momin and Yang 2013; Ardeh 2016) with different characteristics. The functions used to test our proposed method have been listed in Table 1, with their bound ranges and global minimum value where dimension, problem domain size, and optimal solution are denoted by M , $Lb \leq x_i \leq Ub$, and $f(X^*)$ respectively. Out of these functions, Brown function (f_1), Exponential function (f_2), Powell Sum function (f_3), Sphere function (f_4), Step function (f_5), and Zakharov function (f_6) are unimodal in nature whereas the Ackley function (f_7), Periodic function (f_8), Quartic function (f_9), Qing function (f_{10}), Salomon function (f_{11}), Styblinski-Tank function (f_{12}), the Xin-She Yang function (f_{13}), and Xin-She Yang N.2 function (f_{14}) are multimodal in nature.

4.2 Parameters Setup

The basic parameter settings are defined in Table 2. All experimental results of the algorithms namely Standard-PSO and Sparse-PSO are collected from 25 independent runs, each involving 2000 iterations.

4.3 Experiment Results

In this work, swarm size is taken as 150 and dimension as 30 for all the functions. Both the mentioned algorithms are implemented independently 25 times for 2000 iterations on each problem of benchmark function. The data achieved from 25 independent runs are given in Tables 3 and 4. Furthermore, we computed the wilcoxon rank sum test on all the functions for both the optimizers and results are provided in Table 6. Correlation of the computation time as mean time, best time, and worst time taken by both the algorithms during each run are showed in Table 3. By analyzing the outcomes of Table 3, it very well may be inferred that our proposed Sparse-PSO algorithm has a far superior execution time compared with the Standard-PSO algorithm. Also, Table 4 shows the accuracy achieved by Sparse-PSO and the Standard-PSO in the terms of mean (average of global values), best (minimum in global values), worst (maximum in optimum values), and standard deviation (deviation between global values).

Fig. 1, represents the mean of best computation time for all the functions used in this work and also concludes that proposed Sparse-PSO takes less execution time overall with respect to Standard-PSO. Convergence graphs for each function are designed in Fig. 2 where the horizontal axis shows the number of iterations and the vertical axis represents the costs of each benchmark function for all iterations. Fig. 2(a) shows

that for Brown function, the starting point of Sparse-PSO method is very close to the optimal point which is far better than Standard-PSO method. It can be seen from Fig. 2(b) that for Exponential function start point of proposed method is better than Standard-PSO. Next, both the methods converge to optimal point in almost same manner. Convergence graph for Powell Sum function denotes that starting point of Sparse-PSO method reaches near to the optimal point as compared to Standard-PSO method reported in Fig. 2(c). Fig. 2(d) represents the convergence graph for Sphere function which shows that start point as well as convergence curve of proposed method is better than Standard-PSO. Convergence graph for Step function are almost equal for both the algorithms is reported in Fig. 2(e). Fig. 2(f) shows the better convergence of Sparse-PSO method compared to Standard-PSO for Zakharov function. Fig. 2(g) demonstrates that in Sparse-PSO method the graph for Ackley function reaches global optimum point while in Standard-PSO approach the graph converges earlier before reaching the global optimum point. Sparse-PSO performs well for Periodic function presented in Fig. 2(h) as it converges firstly than Standard-PSO. Figures 2(i) and 2(j) determine that when the number of iterations is low the Sparse-PSO method enhances the results as compared to the Standard-PSO and later on both methods are showing similar results when number of iteration is high. Convergence graph for Salomon function reaches global optimum point for Sparse-PSO method firstly compared to Standard-PSO method as reported in Fig. 2(k). Convergence graph for Styblinski-Tank function represents that Sparse-PSO method is better than Standard-PSO because graph of Sparse-PSO reaches to the global optimal point while graph for Standard-PSO converges earlier as shown in Fig. 2(l). Both figures 2(m) and 2(n) demonstrate that starting point in convergence graph for Sparse-PSO method is better related to Standard-PSO method. It can be concluded from Fig. 2 that the proposed Sparse-PSO algorithm starts with better function value compared to Standard-PSO algorithm for all the functions and reaches to the global optimum. Table 4 shows that Sparse-PSO over performs Standard-PSO in terms of average speed-up of execution time for all benchmark functions.

4.4 Wilcoxon rank sum test

Wilcoxon F. (1945) proposed a non-parametric statistical test for significance analysis known as wilcoxon rank sum test (Wilcoxon 1992). The test assigns a rank to all the observations considered as a group and afterward sums the ranks of each group. Table 6 shows that score obtained by proposed Sparse-PSO method is significantly better on almost all functions compared to Standard-PSO method.

5 Conclusion and Future Directions

A new initialization technique on the basis of sparse representation has been proposed for PSO algorithm in this paper. The main objective of the proposed Sparse-PSO algorithm is to reduce the computation time required. Fourteen benchmark functions have been used to test the proposed method, and the outcomes of the same are compared with the standard approach. It is clear from results that the Sparse-PSO takes less computation time as

compared to the Standard-PSO. Results also shows that the Sparse-PSO method is more effective for multimodal functions such as Qing function, Salomon function, Styblinski-Tank function, Xin-She Yang function, and Xin-She Yang N.2 function. The proposed method achieves the global best value with less computation time compared to Standard-PSO. Furthermore, the wilcoxon test confirms that the Sparse-PSO method is significantly better than the Standard-PSO in terms to achieve optimum results for most of the functions. The future direction of this work would be to achieve more improvement by correcting and updating the used parameters and performing this method on more complex functions.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

Authorship contributions

All authors contributed to the study conception and design. Coding and analysis were performed by Mr. Lalit Kumar. The manuscript was written by Manish Pandey and Mitul Kumar Ahirwal. All authors read and approved this submitted manuscript.

References

- Ahirwal, M. K., Kumar, A. and Singh, G. K. (2014). Improved range selection method for evolutionary algorithm based adaptive filtering of EEG/ERP signals. In *Neurocomputing* (144, 282-294).
- Ahirwal, M. K., Kumar, A. and Singh, G. K. (2015). Sub-band adaptive filtering method for electroencephalography/event related potential signal using nature inspired optimization techniques. In *IET Science, Measurement & Technology*, 9(8), 987-997.
- Ahirwal, M. K., Kumar, A. and Singh, G. K. (2016). Study of ABC and PSO algorithms as optimized adaptive noise canceller for EEG/ERP. In *International Journal of Bio-Inspired Computation*, 8(3), 170-183.
- Ahirwal, M.K., Kumar, A. and Singh, G. K. (2014). Adaptive Filtering of EEG/ERP through Noise Cancellers using an Improved PSO Algorithm. In *Swarm and Evolutionary Computation*(Vol. 14, pp. 76-91). Elsevier.
- Ardeh, M. A. (2016). BenchmarkFens Toolbox- A collection of benchmark functions for optimization.
- Cheng, R. (2016). Nature inspired optimization of large problems.
- Farahmand, A., Payan, S. and Sarvari, S.H. (2012). Geometric optimization of radiative enclosures using PSO algorithm. In *International journal of thermal sciences* (60, pp.61-69).

- Gao, D., Li, X. and Chen, H. (2019). Application of improved particle swarm optimization in vehicle crashworthiness. In *Mathematical problems in Engineering*.
- Harrison, K. R., Engelbrecht, A. P. and Ombuki-Berman, B. M. (2018). Self-adaptive particle swarm optimization: a review and analysis of convergence. In *Swarm Intelligence*, 12(3), 187-226.
- Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*(pp. 760-766).Springer US.
- Liu, Z., Lu, J. and Zhu, P. (2016). Lightweight design of automotive composite bumper system using modified particle swarm optimizer. In *Composite Structures* (140, 630-643).
- Mabrouk, M.T., Kheiri, A. and Feidt, M. (2018). A systematic procedure to optimize Integrated Solar Combined Cycle power plants (ISCCs). In *Applied Thermal Engineering* (136, pp.97-107).
- Mahdavi, S., Shiri, M.E. and Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: A survey. In *Information Sciences*(295, pp.407-428).
- Marini, F. and Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. In *Chemometrics and Intelligent Laboratory Systems*(149:153-65).
- Momin, J. and Yang, X. S. (2013). A literature survey of benchmark functions for global optimization problems. In *Journal of Mathematical Modelling and Numerical Optimization*, 4(2), 150-194.
- Pandey, S., Wu, L., Guru, S.M. and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *24th IEEE international conference on advanced information networking and applications* (pp. 400-407).IEEE.
- Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimization. In *Journal of Artificial Evolution and Applications*.
- Sahu, A., Panigrahi, S. K. and Pattnaik, S. (2012). Fast convergence particle swarm optimization for functions optimization. In *Procedia Technology* (4, 319-324).
- Sengupta, S., Basak, S. and Peters, R.A. (2019). Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. In *Machine Learning and Knowledge Extraction*, 1(1), pp. 157-191.
- Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 congress on evolutionary computation*(vol. 1, pp. 81-86).IEEE.
- Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *International conference on evolutionary computation proceedings*, IEEE world congress on computational intelligence (Cat. No. 98TH8360, pp. 69-73). IEEE.
- Soudan, B. and Saad, M. (2008). An evolutionary dynamic population size PSO implementation. In *3rd International Conference on Information and Communication Technologies: From Theory to Applications*(pp. 1-5).IEEE.

- Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics*. New York, NY, (pp. 196-202). Springer.
- Yang, S. (2014). Nature-inspired optimization algorithms. Elsevier
- Yang, X.S. (2018). Optimization techniques and applications with examples. *John Wiley & Sons*.
- Zhan, S. and Huo, H. (2012). Improved PSO-based task scheduling algorithm in cloud computing. In *Journal of Information & Computational Science*, 9(13), pp.3821-3829.
- Zhu, H., Wang, Y., Wang, K. and Chen, Y. (2011). Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. In *Expert Systems with Applications*, 38(8), pp.10161-10169.

Figures

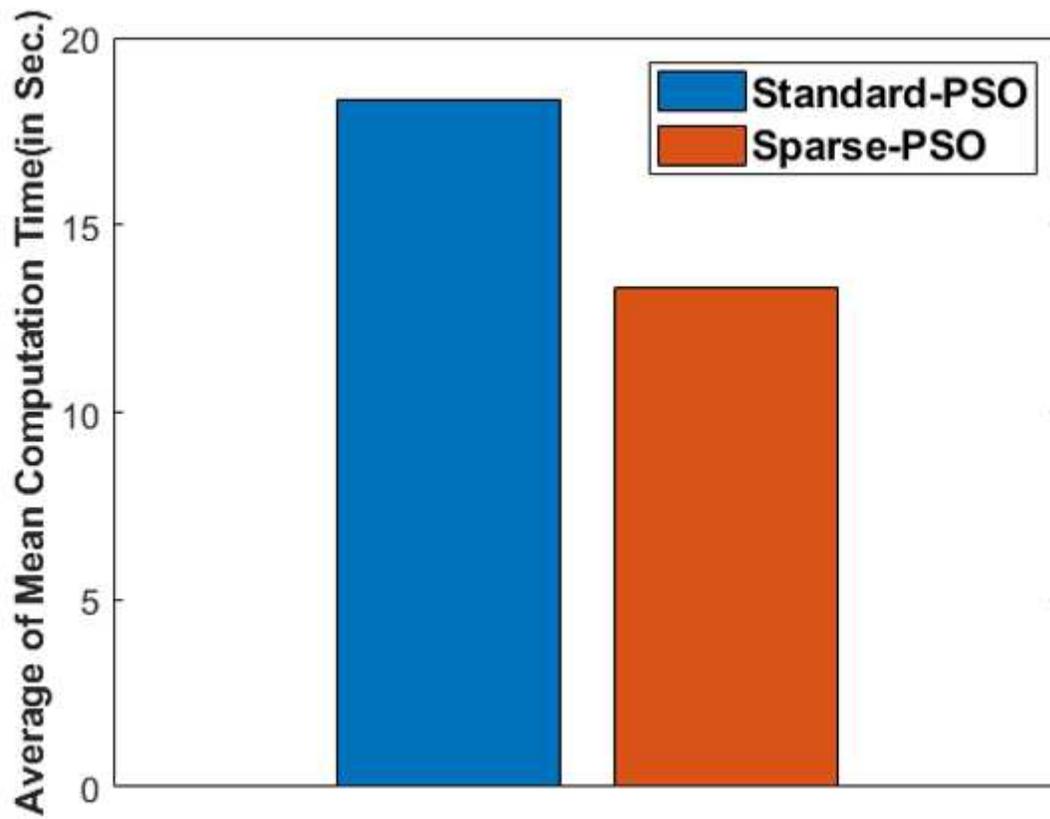


Figure 1

Overall average computation time for all functions (f1 to f14)

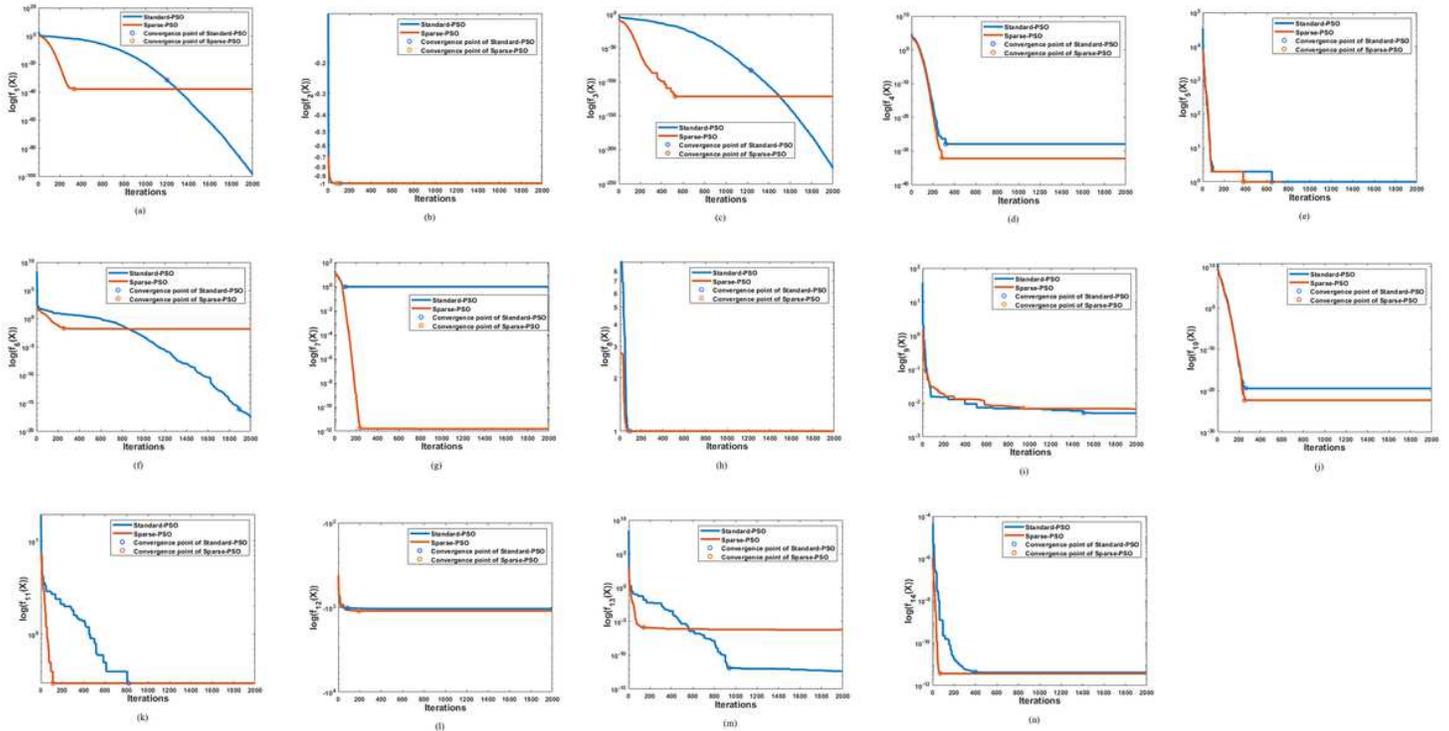


Figure 2

Convergence graphs of Sparse-PSO and Standard-PSO for fourteen test functions (a) Brown, (b) Exponential, (c) Powell Sum, (d) Sphere, (e) Step, (f) Zakharov, (g) Ackley, (h) Periodic, (i) Quartic, (j) Qing, (k) Salomon, (l) Styblinski-Tank, (m) Xin-She Yang, (n) Xin-She Yang N.2

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Tables.docx](#)