Menoufia J. of Electronic Engineering Research (MJEER), Vol. 31 , No. 2, 2022

An Adaptive Particle Swarm Based Compressive Sensing Technique

Asmaa Bakheet Mostafa*
Department of Electrical and Electronics Engineering
Faculty of Engineering, Assiut University
Assiut, Egypt
asmaabakheet@gmail.com

Osama Elnahas
Department of Electrical and Electronics Engineering
Faculty of Engineering, Assiut University
Assiut, Egypt
osama.hussien@eng.aun.edu.eg

Mostaf Mekky
Department of Electrical and Electronics Engineering
Faculty of Engineering, Assiut University
Assiut, Egypt
mymakkey@aun.edu.eg

Abstract—Compressive sensing (CS) has recently gained a lot of attention in the domains of applied mathematics, computer science, and electrical engineering by offering compression of data below the Nyquist rate. The particle swarm optimization (PSO) reconstruction algorithm is considered one of the most widely used evolutionary optimization techniques in CS. The self-tuned PSO parameters control can greatly improve its performance. In this paper, we propose a self-tuned PSO parameter control based on a sigmoid function in the CS framework. In the proposed approach, PSO parameters are adjusted by the evaluation at each iteration. The proposed self-tuned PSO parameter control approach involves two PSO parameters. First, acceleration coefficients, which are considered very effective parameters in enhancing the performance of the algorithm, second, inertia weight, which is used to accelerate the movement of particles towards the optimum point or slow down the particles so that they converge to the optimum. In contrast to conventional PSO, the proposed self-tuned PSO parameters control algorithm governs the convergence rate, resulting in a fast convergence to an optimal solution and very precise recovery of the original signal. A simulation study validates the effectiveness of the proposed method as compared to the conventional PSO algorithm.

Keywords—Sigmoid function, PSO, Reconstruction Algorithms, Compressive Sensing, Cognitive-IOT

I. Introduction

Compressive sensing (CS) is a signal processing technique that allows signals to be acquired with fewer samples [1, 2]. CS is based on the idea that we can represent many signals using only a few non-zero coefficients in a suitable basis or dictionary. Sparsity and incoherence are two key principles of CS. Sparsity is fundamental in the theory of CS, which works against the principle of the conventional Nyquist sampling theorem. CS exploits the fact that the sparse signal has less information relative to its length, so it can be recovered accurately from a small number of incoherent measurements. Mathematically, a signal $x \in R^N$ is k-sparse when $\|\boldsymbol{x}\|_0 \le k \le N$. Where $\|\boldsymbol{x}\|$ is l_0 norm defined as $\|\boldsymbol{x}\| = \sum_{i=0}^{n}|x_i|$ = number of non-zeros component of $\boldsymbol{x}$. The compressed signal can be represented by the following equation:

$$\mathbf{y} = \Phi \boldsymbol{x} \tag{1}$$

where the sensing matrix $\Phi=\{\varphi 1, \varphi 2, \ldots. \varphi N\}$ is M × N (M << N) and $y \in R^M$ is a measurement vector. The sensing matrix Φ must satisfy the restricted isometric property (RIP) of order k so that the sensing process does not damage the information stored in the original signal.

$$(1-\delta)\ \|\boldsymbol{x}\|_2^2 \le \ \|\Phi\boldsymbol{x}\|_2^2 \ \le \ (1+\delta)\ \|\boldsymbol{x}\|_2^2 \tag{2}$$

where: δ € (0,1) is the isometric constant. The goal of compressive sensing is to design the matrix Φ and a reconstruction algorithm so that for k-sparse signals we require only a "small" number of measurements. The underdetermined system of linear equations described by (1) can have infinitely many solutions indicating that very different signals may lead to the same measurements. So for a given sensing matrix Φ ∈ RM×N and measurement vector y ∈ RM, finding the maximally sparse solution is an ill-conditioned problem. The sparse reconstruction problem can be modeled as

$$\boldsymbol{x}' = \min_x \|\mathbf{y} - \Phi\boldsymbol{x}\|_2^2 \quad \text{subject to} \ \|\boldsymbol{x}\| \le k, \tag{3}$$

where $\boldsymbol{x}'$ is the sparsest possible solution. Sparse signal recovery such as l_0 minimization use sparsity constraint as regularized in order to find an estimated solution to (1) with limited non-zero entries. Using the non-convex problem defined by (3) to find the sparsest possible solution is computationally impossible as it implicates an exhaustive search for k non-zero values of the recovered vector.

In the CS framework, recovering the original signal from the compressed data represents a significant challenge. In the literature, many sparse signal reconstruction methods have already been presented [12, 13]. They can be classified into three major categories: convex relaxation, non-convex relaxation, and greedy algorithms. Using the non-convex problem defined by (3) to find the sparsest possible solution is hard to solve exactly in a reasonable time. Linearization techniques, in which the objectives and constraints are linearized, are another alternative to minimizing approaches. Some variables are kept fixed in a cyclical manner when using cyclical minimization procedures (or approximated by a convex function). Particle swarm optimization (PSO) is among the most evolutionary optimization methods.

Swarm intelligence algorithms are extremely helpful and effective for solving optimization problems such as non-convex problems. Adaptation of PSO parameters has been developed to prevent local optimization trapping. It is widely known that control parameters balance between global and local searches throughout the searching process, and therefore they play a vital role in successfully finding the optimal solution [9], [10]. The acceleration coefficients and inertia weight parameters act as the main two control parameters in the PSO algorithm, which have a substantial impact on the performance of swarm intelligence algorithms. Until now, some PSO variations have concentrated on changing the three control parameters described above. In [5] and [6], a linear-decreasing-inertia-weight-based PSO (PSO-LDIW) algorithm has been proposed where the inertia weight is updated in a time-varying manner. The PSO-LDIW algorithm concentrates on the selection of the inertia weight, where the updating equation of the inertia weight w at the each iteration is given as follows:

$$\mathrm{w} = w_{max} - (w_{max} - w_{min}) + \frac{k}{\text{maxiter}} \tag{4}$$

In [7], a random inertia weight value (PSO-RIW) is used to enable the PSO to track the optima in a dynamic environment such as the inertia weight is given by

$$\mathrm{w} = 0.5 + \frac{rand()}{2} \tag{5}$$

where rand() is a random number in [0,1]. Based on (5), the inertia weight w is a uniform random variable in the range [0.5, 1].

In this paper, we present an adaptive PSO parameter control technique that relies on self-tuned parameter control for the two main effective parameters in the PSO algorithm, the inertia weight and the acceleration coefficients by using sigmoid functions of the current iterative generation for each parameter. The proposed adaptive parameter control is iteratively optimized until it finds the optimal solution with improved convergence rate. During each iteration, we evaluate the particle swarm by comparing the current best particle solution and the previous best particle solution to a certain threshold. Once the threshold is reached, we can stop the process and choose the optimum solution, which means we will not see any updates to the best solution in a number of iterations. If the algorithm does not reach a threshold through iterations, we choose the optimum solution when we reach the maximum number of iterations. The proposed algorithm can maintain the population diversity by adaptively adjusting the parameter control through the optimization process. Finally, the efficiency of the proposed adaptive PSO parameter control is validated experimentally by exactly recovering a K-sparse signal from only a small number of compressed measurements

II. Reconstruction Algorithms

In the CS framework, the main purpose is to reconstruct a sparsely sampled compressed signal by solving an undetermined set of linear equations with a defined set of solutions. Recovering the original signal from compressed data has been one of the most challenging tasks in CS research. Furthermore, choosing the right reconstruction algorithm for every application and improving the reconstruction algorithm's performance with fewer computational steps are considered the primary objectives for researchers today. Based on the CS method, the algorithms for the reconstruction of the original sparse signal can be broadly classified into three main categories: convex relaxation, non-convex relaxation, and greedy algorithms.

A. Convex Relaxation

These class algorithms use linear programming [14] to solve a convex optimization problem to obtain reconstruction. Some examples of such algorithms include Basis Pursuit [15], Basis Pursuit De-Noising [15], Least Absolute Shrinkage, and Selection Operator (LASSO) [16]. Basis pursuit is the mathematical optimization of a problem in the form of

$$\min_x \ \|\boldsymbol{x}\|_1 \ \text{ subject to } \ \mathbf{y} = \boldsymbol{\Phi x} \tag{6}$$

These types are most commonly utilized when the signal of the linear equation $\mathbf{y} = \boldsymbol{\Phi x}$ is underdetermined. Because this approach is more complicated and time-consuming, it cannot be used in time-sensitive reconstruction applications.

B. Non- Convex Relaxation

There are many practical problems of importance that are not convex, and they can be challenging (if not impossible) to solve exactly in a reasonable amount of time. Hence the idea of using heuristic algorithms, which may or may not produce desired solutions. Optimization is carried out with some variables held fixed in a cyclical manner, and linearization techniques are used when objectives and constraints are linearized (or approximated by convex functions). There are many algorithms proposed in literature that use this technique like Focal Underdetermined System Solution (FOCUSS) [17], Sparse Baysian Learning algorithms [18], which are some examples of such algorithms. A search algorithm (such as a genetic algorithm and PSO) is another technique that relies on simple rules for updating the solution.

C. Greedy Algorithm

A Greedy Algorithm is a problem-solving method based on making the most local optimal decision at each step and aiming for a global optimal response by finding the answer, step by step, iteratively. Sensing matrix $\boldsymbol{\Phi}$ columns are selected in a greedy manner. A column of the sensing matrix $\boldsymbol{\Phi}$ that correlates the most with it is chosen at each iteration. Additionally, the least square error is minimized with each iteration. The most widely utilized greedy algorithms are Matching Pursuit [19] and Orthogonal Matching Pursuits (OMP) [20]. Improved versions of (OMP) such as Regularized OMP (ROMP) [21], Stagewise OMP (StOMP) [22], Compressive Sampling Matching Pursuits (CoSaMP) [23], Subspace Pursuits [24], Gradient Pursuits (GP) [25], Orthogonal Multiple Matching Pursuit (26), and Reducing Iteration OMP (RIOMP) [27].

III. Particle Swarm Optimization

PSO is a population based algorithm which was originally introduced by Kennedy and Eberhart [1]. This algorithm is a kind of optimized global search algorithm and is a relatively new, modern, and powerful optimization method that has been experimentally shown to perform well on numerous of these optimization problems. It is widely employed to find the global optimum solution in a complex search, the main benefits of PSO are that it is easy to implement and requires few parameters to be adjusted. PSO is derived from the study of bird foraging behavior: a group of birds are looking for food randomly. If there is only one piece of food in this area, the most basic and effective search strategy is to look for food in the closest location to the food.

The basic idea behind PSO is that when it is used to solve optimization problems, the solution corresponds to the position of a bird (particle) in the search space. Not only does the particle have its own position and velocity, but it also has a suitable value determined by the objective function. Each particle remembers and follows the most recent optimal particle while searching the solution space: Each search (iteration) includes some random variables. The particle updates its position in each iteration by tracking two "extreme points": The first is the best solution discovered by the particle itself (i.e., the individual extreme point (Pbest)) and the second is the extreme point (Gbest) of the entire particle swarm.

A. Basic Model of PSO Algorithm

PSO treats each potential solution to an optimization problem as a bird, also known as a particle. The set of particles, also known as a swarm, each particle has memory and can adjust its own motion trajectory based on its current position, information sharing among peers, and the best position experienced in memory, the particle continues to approach the best position until it reaches the optimal position. At each iteration, the position and velocity of the particles are updated according to the following equations:

$$v_i(t+1) = w \times v_i(t) + c_1 r_1 \left(p_i(\mathrm{t}) - x_i(t)\right) + c_2 r_2 \left(p_g(\mathrm{t}) - x_i(t)\right) \quad (7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

Where t indicates the iteration number, w denotes the inertia weight, p_i represents the Pbest of particle i, p_g is the Gbest found by the entire swarm also is known as global best ,c1 and c2 are acceleration coefficients, also known as convergence factors and r1 and r2 are random numbers distributed in the interval [0 , 1]. Fig. 1 shows the PSO algorithm flow with details of each step.

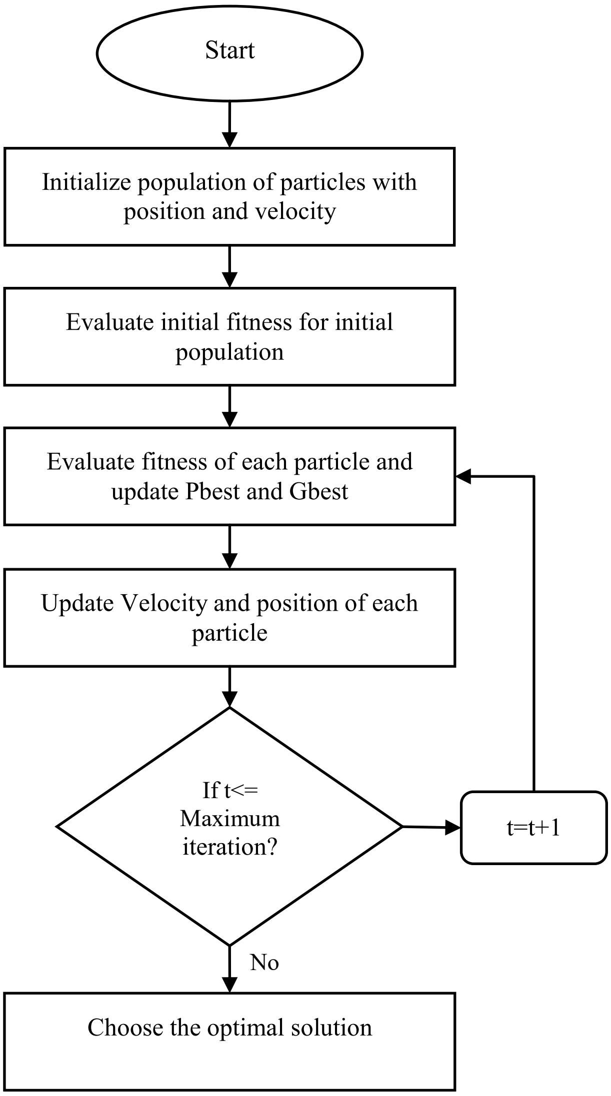

Fig. 1 Block diagram for PSO algorithm flow

IV. Proposed sigmoid function based adaptive PSO algorithm

Parameter adjustment for swarm intelligence algorithms is difficult due to the following challenges. First, the parameter control in swarm intelligence algorithms not only varies according to different optimization problems. Second, while it may be obvious whether a parameter should be increased or decreased based on the state of evolution, determining the extent to which the parameter should be adjusted is a difficult task. Third, avoiding introducing new parameters while increasing the time and space complexity is an interesting problem. In this paper, we present an adaptive PSO parameter control technique that has a significant impact on the PSO algorithm's enhancement and performance. We do that by applying a sigmoid function to both of the primary PSO parameters: inertia weight and acceleration coefficients. These parameters balance global and local searches throughout the searching process, which plays a vital role in successfully finding the optimal solution. Acceleration coefficients and inertia weight are adaptively adjusted as the iteration goes. In general, balancing between local and global search processes will improve PSO performance. This trade-off between local and global is influenced by modifying and tuning some parameters, namely current motion, inertia weight, cognitive and social coefficients.

A. Sigmoid function-based adaptive acceleration coefficients

The sigmoid-based acceleration coefficients are designed to strike a balance between early-stage global search ability and late-stage global convergence. Most of the existing PSO variants only tune the acceleration coefficients in a time-varying manner without taking the information of the population evolution into account [3,4]. As shown (7) and (8), the velocity of the particle updates is affected by the distances between the particles and their own Pbest and Gbest. In this case, it is reasonable to tune the acceleration coefficient based on the distances between each individual particle and its Pbest and Gbest, our main motivation is to accelerate the particles so that they can find the optimal solution as quickly as possible, thereby increasing the convergence rate. Unlike the time-varying updating strategy, the acceleration coefficients are adjusted based on the particle's distance from its Gbest and Pbest. If the particle is far away from its Pbest and Gbest, an acceleration coefficient with a relatively large value is used to accelerate it. However, the value of the acceleration coefficient is limited in an appropriate range to avoid premature convergence, which means that the velocity should be bounded to guarantee the searching capability of the algorithm.

Motivated by the previous discussions, we believe that an adaptive weighting updating function is appropriate to describe the relationship between the acceleration coefficient and the distances (from the particle to its Pbest and Gbest). In other words, the updates to the former acceleration coefficients should be adaptive to the latter distances, fully justifying the velocity of the particle movements toward the global optimum. From a mathematical standpoint, the proposed adaptive weighting updating rule is as follows:

$$C_{pi}(t) = f\left(G_{pi}(t)\right) \tag{9}$$

$$C_{gi}(t) = f\left(G_{gi}(t)\right) \tag{10}$$

Where the function $f(.)$ in (9) and (10) represents the adaptive PSO parameter updating function, Specifically, $G_{pi}(t)$ is the distance between the particle and its Pbest for the cognitive acceleration coefficient. For the social acceleration coefficient, $G_{gi}(t)$ indicates the distance between the particle and the Gbest. Remark; $G_{pi}(t)$ and $G_{gi}(t)$ are defined by

$$G_{pi}(t) = p_i(t) - x_i(t) \tag{11}$$

$$G_{gi}(t) = p_g(t) - x_i(t) \tag{12}$$

which denote the distances from the particle i to its Pbest and Gbest at the t th iteration, respectively, so we can calculate C_{pi}, C_{gi} At each iteration as follows:

$$\begin{aligned} C_{pi}(t) &= \frac{b}{1+e^{-\left(a * G_{pi}(t)-c\right)}} + d \\ C_{gi}(t) &= \frac{b}{1+e^{-\left(a * G_{gi}(t)-c\right)}} + d \end{aligned} \tag{13}$$

where a denotes the steepness of the curve which is a constant value, b represents the peak value of the curve, c represents the abscissa value of the central point of the curve, d is a positive constant value.

In search of suitable updating functions that are monotonically increasing, uniformly bounded, and inspired by neural network activation functions, we found some popular activation functions for neural networks, such as step functions, sigmoid functions and others, among which we decided to choose the sigmoid function as the adaptive parameter updating function for the following reasons:

- The sigmoid function is monotonic and bounded.
- The sigmoid function curve is S-shaped, avoiding undesirable abrupt changes in the parameter control.
- The sigmoid function is smooth and differentiable, reflecting the adaptive/dynamic nature of the weight updating iteration by iteration.

According to the above discussion in this paper, a sigmoid function is employed to tune the acceleration coefficients as follows:

$$f(D) = \frac{b}{1+e^{-(a * D-c)}} + d \tag{14}$$

Where a, b, c and d are constants, D is the distance between the particle and it's Pbest for the cognitive acceleration coefficient. For the social acceleration coefficient, D indicates the distance between the particle and the Gbest.

B. Sigmoid function-based adaptive inertia weight

The inertia weight (w) adjusts the particle's momentum by weighting the contribution of the previous velocity. The velocity and position update equations with an inertia weight are described in Equations (7) and (8). There are two different methods for calculating the inertia weight value: decreasing and increasing. In order to decrease, a large initial value of inertia weight is reduced linearly or nonlinearly to a small value. A large inertia weight facilitates in a global search, whereas a small inertia weight facilitates in a local search. In order to decrease, a small inertia weight increase linearly or non-linearly to a larger value. A large inertia weight has a greater chance of convergence, which implies that a larger inertia weight at the end of the search will foster the ability to converge. A large inertia weight has a greater chance of convergence, which implies that a larger inertia weight at the end of the search will foster the ability to converge. There are numerous methods in nonlinear approaches such as tracking and dynamic system [9] and constriction factor [8].Shi suggested that a linear decrease in inertia weight value from 0.9 to 0.4 [5, 6]. In [9], Y. Zheng, et. al. suggested that an inertia weight value beginning from 0.4 linearly increasing to 0.9.

In this paper, we propose a new inertia weight modulated with a sigmoid function for improving PSO performance. This work was inspired by the excellent performance shown by the sigmoid decreasing inertia weight based on detailed observation and analysis. We employed sigmoid decreasing inertia weight ensure faster convergence ability and a near optimum solution. Afterwards, a small inertia weight is retained to facilitate a local search in the final part of the PSO process. The basic of sigmoid function is given as:

$$f = \frac{1}{1+e^{-t}} \tag{15}$$

$$w_t = \frac{w_{min} - w_{max}}{1+e^{-s(t-n*gen)}} + w_{max} \tag{16}$$

$$s = 10^{(\log(gen)-2)} \tag{17}$$

Where: w_t is inertia weight at t iteration, w_{min} and w_{max} are inertia weight at the start and inertia weight at the end of a given run, respectively. Furthermore, s is the constant to adjust sharpness of the function, gen is the maximum number of generations to run and n is the constant to set partition of sigmoid function. We set w_{min}= 0.4, w_{max} = 0.9, According to the characteristics of the sigmoid function and experimental experience

V. SIMULATION RESULTS AND DISCUSSION

In this paper, the proposed adaptive PSO algorithm is compared with some currently popular PSO algorithms including the conventional PSO algorithm, the PSO-LDIW algorithm [5, 6] and the PSO-RIW algorithm [7]. For all of the algorithms, numerical simulations are performed using random measurement matrix Φ ∈ R256 x 512 with Gaussian distribution. The rows represent measurements and its columns are equal to the size of sparse signal. The k sparse test signal x_0∈ R256, with k = 85 having random magnitudes and random non-zeros elements indices is utilized to perform sparse signal reconstruction. This test signal is compressively sampled to get observation vector $\mathbf{y} = \Phi x_0$ ∈ R256. For acceleration coefficients PSO parameter when using sigmoid function defined by (13), according to the characteristics of the sigmoid function and experimental experience we found that best values for parameters in (13) are when setting the follows values b=0.5; c=0; d=1.5; a = 0.000035. For inertia weight PSO parameter when using sigmoid function defined by (16) we set w_{min}= 0.4, w_{max}= 0.9. The proposed adaptive PSO algorithm performance is measured by mean square error (MSE) at every iteration and compared to conventional PSO.

$$MSE = \frac{\| x' - x_0 \|}{\| x_0 \|} \tag{18}$$

Where x' is the recovered signal and x_0 is the original signal. The stopping criterion is set as the algorithm finds the globally optimal solution within the threshold condition.

Our goal is to arrive at the optimal global solution once one of the following two conditions is met: 1) we have reached the maximum number of iterations. 2) The threshold condition has been met. The threshold is calculated at each iteration by comparing the current best particle solution to the previous best particle solution. When the threshold is less than 10^{-12}, we can stop the process and choose the optimum solution. In this case, a smaller number of iterations indicate better convergence performance of the PSO algorithm. To avoid random phenomena, we repeated the experiment for 20 times.

Fig.2 shows the cost function of the proposed adaptive PSO algorithm and the conventional PSO algorithm versus number of iterations. From Fig. 2, initially both algorithms have the same cost function, then the proposed adaptive PSO algorithm rapidly decreases, whereas the conventional PSO objective function gets stuck in a local minimum.

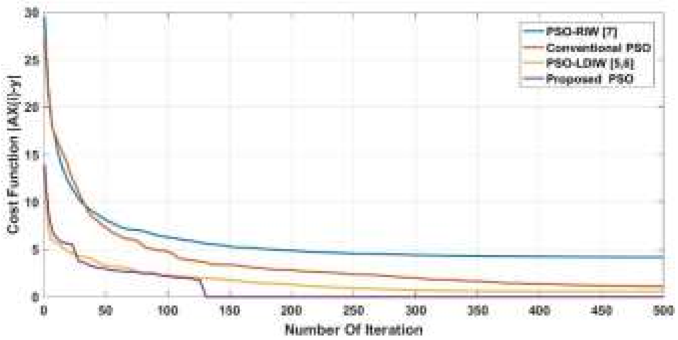

Fig. 2 Cost function of the proposed adaptive PSO algorithm, PSO-RIW, Conventional PSO and PSO-LDIW.

From Fig.3, it is obvious from the simulation results that the proposed adaptive PSO parameter control algorithm not only converges faster than PSO-LDIW, PSO-RIW and conventional PSO, but also achieves less MSE.

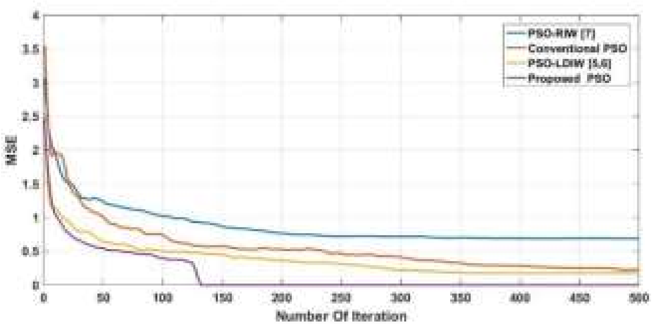

Fig. 3 MSE comparison between the proposed algorithm, PSO-RIW, Conventional PSO and PSO-LDIW.

Conventional PSO recovered signal is shown Fig. 4, demonstrating that there is large difference between the original signal and the reconstructed signal.

Fig. 5 shows sparse signal recovery using our proposed PSO adaptive parameter technique, demonstrating that the recovered signal is quite close to the original signal, implying that it was recovered almost exactly. The recovered sparse signal using the proposed PSO adaptive parameter not only recovers correct amplitudes of sparse signal but also is very precise in finding the support of the signal.

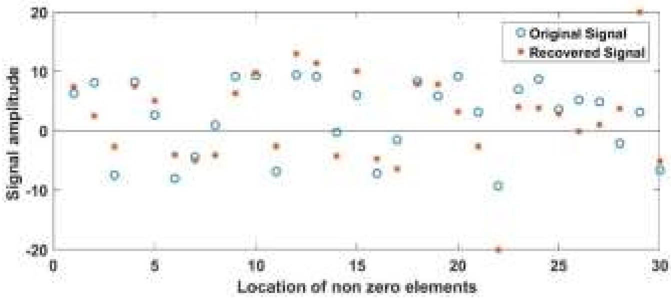

Fig. 4 Signal recovery using conventional PSO.

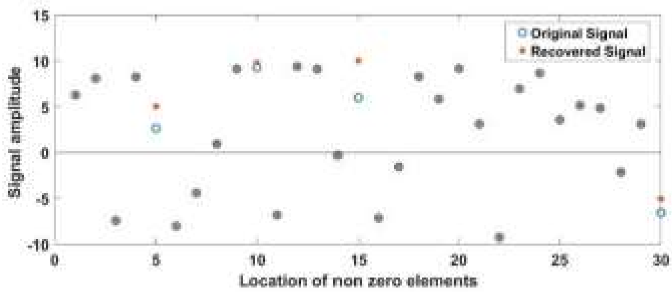

Fig. 5 Signal recovery using the proposed PSO parameters control algorithm.

From the simulation results, it is obvious that the proposed adaptive PSO algorithm not only converges faster than conventional PSO, also achieves less MSE and recovers k-sparse signals with great precision.

VI. CONCLUSION

In this paper, we proposed a self-tuned PSO parameters control approach based on using the sigmoid function. The proposed algorithm adaptively adjusts the PSO inertia weight and acceleration coefficient parameters by employing the sigmoid function of the current iterative generation for each parameter, allowing each parameter to tune its value at each iteration for quick finding an optimal solution. By employing the sigmoid function for the acceleration coefficients parameter. The acceleration coefficients parameter is adaptively adjusted based on the distance from the particle and its Pbest and Gbest values. For the inertia weight parameter, using a sigmoid function enabled more exploration and prevented a premature convergence.
for you.

Simulation results show that the proposed self-tuned PSO parameters control algorithm has high performance, less errors, and fast convergence to an optimal solution. In our future research directions, we plan to implement the proposed algorithm in a real application to further validate the results.

REFERENCES

[1] H. Haider, J. A. Shah, S. Ikram and I. A. Latif, "Sparse signal recovery from compressed measurements using hybrid particle swarm optimization,", In IEEE International Conference on Signal and Image Processing Applications, 2017.

[2] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone and X. Liu, "A Novel Sigmoid-Function-Based Adaptive Weighted Particle Swarm Optimizer,", In IEEE Transactions on Cybernetics, vol. 51, no. 2, pp. 1085-1093, Feb. 2021.

[3] A. Ratnaweera, S. K. Halgamuge and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,", In IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, June 2004.

[4] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator," Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, pp. 1958-1962, 1999.

[5] Shi, Y., Eberhart, R.C., "Parameter selection in particle swarm optimization", In: Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E. (eds) Evolutionary Programming VII. EP 1998. Lecture Notes in Computer Science, vol 1447. Springer, Berlin, Heidelberg, 1998.

[6] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, pp. 1945-1950, 1999.

[7] R. C. Eberhart and Yuhui Shi, "Tracking and optimizing dynamic systems with particle swarms," Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), pp. 94-100, 2001.

[8] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, pp. 1951-1957, 1999.

[9] Yong-ling Zheng, Long-hua Ma, Li-yan Zhang and Ji-xin Qian, "Empirical study of particle swarm optimizer with an increasing inertia weight," Proceeding of the IEEE Congress on Evolutionary Computation. CEC '03., pp. 221-226, 2003.

[10] D. L. Donoho, "Compressed sensing," in IEEE Transactions on Information Theory, pp. 1289-1306, April 2006,

[11] M. Davenport, M. Duarte, Y. C. Eldar, and G.Kutyniok, Compressed Sensing: Theory and Applications.Cambridge, UK: Cambridge Univ. Press, 2011.

[12] André Luiz Pilastri, Joao Manuel R. S. Tavares, "Reconstruction Algorithms in Compressive Sensing: An Overview" ,2016.

[13] R. Wang, J. Zhang, S. Ren and Q. Li, "A reducing iteration orthogonal matching pursuit algorithm for compressive sensing," in Tsinghua Science and Technology, pp. 71-79, Feb. 2016

[14] Candès, E.J., Recht, B., "Exact Matrix Completion via Convex Optimization". Foundations of Computational Mathematics 9, 717 (2009).

[15] Chen, S.S., Donoho, D.L., Saunders, M.A. "Atomic Decomposition by Basis Pursuit", SIAM Rev. 43(1), 129-159 (2001)

[16] Tibshirani, Robert. "Regression Shrinkage and Selection via the Lasso." Journal of the Royal Statistical Society. Series B (Methodological), pp. 267–88, 1996.

[17] J. F. Murray and K. Kreutz-Delgado, "An improved FOCUSS-based learning algorithm for solving sparse linear inverse problems," Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256), pp. 347-351, 2001.

[18] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," in IEEE Transactions on Signal Processing, vol. 52, no. 8, pp. 2153-2164, Aug. 2004.

[19] S. G. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," in IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[20] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," in IEEE Transactions on Information Theory, pp. 4655-4666, Dec. 2007

[21] Needell, D., Vershynin, R. "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit", Foundations of Computational Mathematics 9, 317-334, 2009.

[22] D. L. Donoho, Y. Tsaig, I. Drori and J. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," in IEEE Transactions on Information Theory, pp. 1094-1121, Feb. 2012.

[23] D. Needell, J.A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples", Applied and Computational Harmonic Analysis pp. 301-321, Feb. 2009.

[24] W. Dai and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," in IEEE Transactions on Information Theory, pp. 2230-2249, May 2009.

[25] M. A. T. Figueiredo, R. D. Nowak and S. J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," in IEEE Journal of Selected Topics in Signal Processing, pp. 586-597, Dec. 2007.

[26] E. Liu and V. N. Temlyakov, "The Orthogonal Super Greedy Algorithm and Applications in Compressed Sensing," in IEEE Transactions on Information Theory, pp. 2040-2047, April 2012.

[27] D. Baron, S. Sarvotham and R. G. Baraniuk, "Bayesian Compressive Sensing Via Belief Propagation," in IEEE Transactions on Signal Processing, pp. 269-280, Jan. 2010.

[28] Mohammed M. Abo-Zahhad, Aziza I.Hussein, Abdelfatah M. Mohamed,"Compressive Sensing Algorithms for Signal Processing Applications: A Survey ",Scientific Research Publishing Inc.,Nov. 2015