

An Improved PSO Method With Application to Multimodal Functions of Inverse Problems

S. L. Ho¹, S. Y. Yang², G. Z. Ni², and K. F. Wong¹

¹Department of Electrical Engineering, the Hong Kong Polytechnic University, Hong Kong

²College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

Improvements that use new formulas and strategies for updating the particles' velocity and position, craziness, adaptive velocity adjustment scheme, and the use of a simple refinement searching scheme and so on are proposed on particle swarm optimization (PSO) methods to enhance their performances. Numerical results are given to validate the feasibility and to demonstrate the merits of the proposed method.

Index Terms—Design optimization, evolutionary algorithm, global optimal method, particle swarm optimization (PSO) method.

I. INTRODUCTION

MOST of the inverse problems in electrical engineering involve optimizations of a multimodal objective function. Due to the inability of deterministic methods to find the global solutions for these kinds of problems, a lot of efforts have been devoted to the study of stochastic and heuristic algorithms in the last couple of decades. Correspondingly, a wealth of stochastic optimal methods including, among others, simulated annealing (SA) method, evolutionary algorithms (EA) such as genetic algorithm (GA), and tabu search method have been proposed and used successfully to solve typical electromagnetic design problems. However, hitherto there is no universal stochastic algorithm which can be applied successfully to all problems and one has to use available stochastic optimal methods instead.

The particle swarm optimization (PSO) method is a new entrant to the family of EA. It was developed by Kenney and Eberhart based on a metaphor of the social behavior of birds flocking and fish schooling in their search for food [1], [2]. Thus, the PSO is similar to an EA in that it works with a population, where the population is called a swarm and each individual is called a particle. In a typical PSO, each particle "flies" over the search space to look for promising regions according to the experiences of both its own and those of the group's. Thus the social sharing of information takes place and individuals profit from the discoveries and previous experiences of all other particles in a wide landscape during their search process around the better solutions.

Considering a D -dimensional optimal problem, the time evolution of a swarm of N_{popsize} particles can be represented by those of a position vector, $x_i = (x_1^i, x_2^i, \dots, x_D^i)$, and a velocity vector, $v_i = (v_1^i, v_2^i, \dots, v_D^i)$. At iteration step $k + 1$, the position and velocity of particle i are updated by using

$$v_d^i(k+1) = v_d^i(k) + c_1 r_1 (p_d^i - x_d^i(k)) + c_2 r_2 (g_d^i - x_d^i(k)) \quad (1)$$

$$v_d^i(k+1) = \frac{v_d^i(k+1) \cdot v_d^{\text{max}}}{|v_d^i(k+1)|} \quad (\text{if } |v_d^i(k+1)| > v_d^{\text{max}}) \quad (2)$$

$$x_d^i(k+1) = x_d^i(k) + v_d^i(k+1) \quad (3)$$

where $p_i = (p_1^i, p_2^i, \dots, p_D^i)$ is the best position that particle i has ever found, $g_i = (g_1^i, g_2^i, \dots, g_D^i)$ is the group's best position that the neighborhood particles of the i^{th} particle have ever found, c_1 and c_2 are two positive constants, r_1 and r_2 are two random parameters chosen uniformly within the interval $[0,1]$, v_d^{max} is a parameter that limits the velocity of the particle in the d^{th} coordinate direction.

Obviously, the PSO method is very simple in concept and can be implemented in a few lines of computer codes. Since its discovery, the PSO algorithm has attracted the attention of many researchers in different disciplines and has indeed been used successfully to solve a wide range of engineering design problems. However, as a newly emerging optimal method, the PSO algorithm is still in its development infancy, when compared to its well developed counterparts such as SA and GA, and there are still many problems or issues that require further study. For example, the original PSO algorithm has difficulties in striking a balance between exploration (global investigation of the parameter space) and exploitation (the refinement of searches around a local optimum) [3]. Also, the algorithm cannot adaptively adjust its parameters according to the characteristics of the objective function to be optimized during the optimization process. In this regard, this paper is proposing some improvements on available PSO algorithms to enable them to become a topical and robust global optimizers.

II. A NEW PSO ALGORITHM

For the convenience of explanation of a PSO algorithm, one defines, on the right hand of (1), the second term as the cognitive part where the particle changes its velocity based on its own thinking and memory, and the third term is the social part where the particle modifies its velocity based on the social-psychological adaptation of knowledge.

A. Velocity and Position Updating

Original PSO algorithm has, as mentioned earlier, difficulties in controlling the balance between exploration and exploitation. Consequently, a second random parameter is proposed to modify the velocity updating formula of (1) as

$$v_d^i(k+1) = r_3 v_d^i(k) + c_1 r_1 (p_d^i - x_d^i(k)) + c_2 r_2 (g_d^i - x_d^i(k)) \quad (4)$$

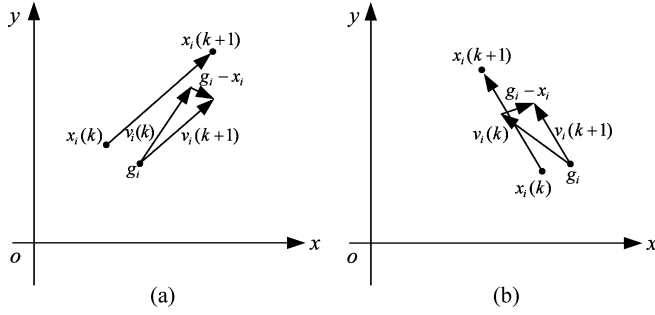


Fig. 1. Two specific moving patterns of the i^{th} particle for the velocity and position updating using (5) and (3) sequentially.

where r_3 is a random parameter which is uniformly chosen from within the interval $[0,1]$.

While the updated formula of (4) is effective in many optimization problems, it does not work in some numerical examples according to the numerical experimentation experiences of the authors. For notional simplicity, one rewrites (4) and considers only the group's best position as follows:

$$v_d^i(k+1) = r_3 v_d^i(k) + c_2 r_2 (g_d^i - x_d^i(k)). \quad (5)$$

Assume there is a two-decision parameter (x, y) optimal problem with the position of the i^{th} particle at time step $k+1$ following the moving patterns as shown in Fig. 1. The velocity and position of that particle are updated using (5) and (3) sequentially. Here the moving pattern (a) is dominated by the inertia of a flying bird, and the moving pattern (b) is due to mis-matching of the algorithm parameters. For both moving patterns, in the step that follows, the direction of its velocity is in the opposite direction of that of its exact one to allow the bird to fly back into the possible promising regions. Modeling this fact into the improved PSO algorithm, one amends (4) to

$$v_d^i(k+1) = (2r_3 - 1)v_d^i(k) + c_1 r_1 (p_d^i - x_d^i(k)) + c_2 r_2 (g_d^i - x_d^i(k)). \quad (6)$$

On the other hand, the utilization parameters, say r_1 and r_2 , for both the personal and group's knowledge in (6), are generated independently and randomly. Therefore, there will be cases in which the two random parameters are both too large or too small. In the former cases, both the personal and social experiences accumulated so far are over used, resulting in the particle being trapped quickly onto a local optimal point. In the latter case, both the personal and social experiences are under-used when guiding the search toward an optimal point, thereby slowing down the convergence speed of the algorithm. To optimize the use of cognitive knowledge of a particle and the social experience of its neighbors, the reasoning ability of an intelligent community is modeled in the proposed algorithm using one random parameter that combines the cognitive knowledge of the particle as well as the group's experiences.

In addition, the inferior limits are also proposed in the improved PSO algorithm to avoid the generation of meaningless or trivial particles.

After taking into consideration of all the aforementioned improvement aspects, and re-arranging the numbers of the subscripts of the random parameters, the new formulas for velocity updating in the proposed algorithm are

$$v_d^i(k+1) = (2r_2 - 1)v_d^i(k) + r_3 [c_1 r_1 (p_d^i - x_d^i(k)) + c_2 (1 - r_1)(g_d^i - x_d^i(k))] \quad (7)$$

$$v_d^i(k+1) = \frac{v_d^i(k+1) \cdot v_d^{\max}}{|v_d^i(k+1)|} \quad (if |v_d^i(k+1)| > v_d^{\max})$$

$$v_d^i(k+1) = \frac{v_d^i(k+1) \cdot v_d^{\min}}{|v_d^i(k+1)|} \quad (if |v_d^i(k+1)| < v_d^{\min}). \quad (8)$$

One can balance exploration and exploitation by exploiting the position updating strategy and thus (3) becomes

$$x_d^i(k+1) = r_4 x_d^i(k) + (1 - r_4) v_d^i(k+1). \quad (9)$$

B. Memory and Utilization of the Worst Solution

To utilize fully the latest information accumulated from the searched history to guide the subsequent search towards promising solutions, the position of the particle which gives the worst objective function value in the current iterative swarm is used in the proposed algorithm in order to shift the new particles away from the particle in issue. On the other hand, in the advance stage of a search, the difference between the best and worst particles will diminish, therefore the moving away from the worst particle could be counterproductive, as such move will also shift the search away from the best one. Therefore, an updating formula for velocity using the worst particles is proposed as

$$v_d^i(k+1) = \begin{cases} v_d^i(k+1) & (if |p_i^{\text{worst}} - p_i^{\text{best}}| \leq \varepsilon) \\ v_d^i(k+1) + c_3 r_4 [x_d^i - (p_d^i)^{\text{worst}}] & (\text{otherwise}) \end{cases} \quad (10)$$

where p_i^{worst} is the position of the particles that gives the worst objective function for the current iterative swarm; $p_i^{\text{best}} = p_i$ or g_i ; c_3 is a positive constant; r_4 is a random parameter which is chosen uniformly from the interval $[0,1]$; ε is a precision parameter defined by the user.

C. Craziness

In birds flocking or fish schooling, a bird or a fish often change directions suddenly. This 'craziness' of a bird or fish is modeled in the primary algorithm using a craziness variable [2]. However, this operator is eliminated in algorithms subsequently developed by the introduction of a cornfield vector. To maintain the diversity of the particles, it is however necessary to retain the craziness variable in the PSO algorithm. Hence a simple craziness operator is re-introduced to assure that the particle has a predefined probability of craziness to maintain the diversity of the particles in the proposed algorithm. To this end, before updating position using (7)–(10), the velocity of the particle is crazed by

$$v_d^i(k+1) = \begin{cases} v_d^i(k+1)(r_5 \leq P_{cr}) \\ Rand_{d,d}(r_5)(r_5 > P_{cr}) \end{cases} \quad (11)$$

where r_5 is a random parameter which is taken uniformly from within the interval $[0,1]$, $Rand_d(\bullet)$ is a function which is used to randomly generate the d^{th} component of the velocity of the particle i , P_{cr} is a predefined probability of craziness.

D. Adaptive Regulation of Velocity

Although the initial particles are generated in such a way that they are confined in the allowable space for a boundedly constrained optimal problem, it is very common for new particles to move out of the boundaries of the feasible space during the search process. Traditionally, this problem is tackled by either taking the bounds as the corresponding coordinates of the new particles, or keeping the coordinates of the particle unchanged but to assign that particle with an extremely poor objective function value. However, both treatments will reduce the diversity of the particles in the search process, thereby reducing the global search ability of the algorithm correspondingly. In the improved PSO algorithm, an adaptive regulation scheme is proposed that adjusts the velocity of every particle, according to the number of moves which are outside the feasible spaces in some specified iteration cycles, in order to prevent the particles to migrate to areas outside the boundaries. Mathematically, assume that the number of moves that are outside the boundary in the d -coordinate direction since the last velocity adjustment is N_d^{out} , and the number of iterative cycles between velocity adjustment is N_T , the velocity of particle i can be adjusted automatically, after every N_T successive iterations, using

$$v_d^i(k+1) = (1 + \beta)^\alpha v_d^i(k) \quad (\text{if } N_d^{\text{out}} < 1) \quad (12)$$

$$v_d^i(k+1) = \frac{v_d^i(k)}{(1 + N_d^{\text{out}}/N_T)^\gamma} \quad (\text{if } N_d^{\text{out}} \geq 1) \quad (13)$$

where α , β , and γ are positive constants.

A useful byproduct of this velocity adjustment scheme is that it will increase the diversity of particles to prevent them from being trapped onto a local optimum. For example, when a particle is confined in a local optimal point which is in the feasible space, the velocities of the particle in the successive searches that follow will eventually reach their maximum values if they are updated using (12), (13), which will equip the particle with an enhanced ability to move away from the specific local optimal point.

E. Exploitation Search

To enable the algorithm to refine searches around a new searched best solution in an attempt to find improved solutions, after a particle is being identified as a new best solution, an intensifying search around the newly identified particle is activated. The intensifying search proposed in this paper has characteristics that are reminiscent of a tabu search method in that a limited number, N_h , of new particles are generated randomly in the small neighborhood of the best solution. The objective function values of these neighborhood particles together with that of the normally generated ones are then evaluated, and the best one is selected as the new current particle, regardless of whether its objective function value is better or worse than that of the current particle, to maintain the ‘climb up’ property of the proposed algorithm.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT OPTIMAL METHODS
ON THE TEST FUNCTION FOR 100 INDEPENDENT RUNS

Algorithms	No. of averaged iterations	No. of runs finding the global solutions
Original PSO	1248	1
Proposed PSO	1464	100
Utabu	1629	100
CS	13478	100

III. NUMERICAL EXAMPLES

A. Benchmark Experiment

To validate the proposed improved PSO algorithm and to compare its performances with the original PSO method, numerical experiments have been done on many well designed mathematical functions, and only the results on a typical one are reported here, due to space limitation, to highlight the global search ability enhancement of the proposed algorithm. Essentially, the function being studied is

$$f(x) = k_3 \left\{ \sin^2((\pi k_4 x_1) + \sum_{i=1}^{n-1} (x_i - k_5)^2) \times [1 + k_6 \sin^2(\pi k_4 x_{i+1})] + (x_n - k_5)^2 [1 + k_6 \sin^2(\pi k_7 x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4) \quad (14)$$

$$X = \{x \in R^5 \mid -5 \leq x_i \leq 5, (i = 1, 2, \dots, 5)\}$$

$$k_3 = 0.1, k_4 = 3, k_5 = k_6 = 1, k_7 = 2$$

where u is a penalty function defined as

$$u(y, a, k, m) = \begin{cases} k(y - a)^m, & (\forall y > a) \\ 0, & (\forall -a \leq y \leq a) \\ k(-y - a)^m & (\forall y < -a). \end{cases} \quad (15)$$

This function has roughly 15^5 local optima, and the global one is at $X_{min} = (1, 1, 1, 1, 1)$ and $f_{opt} = 0$.

In the numerical experiments, the parameters used by both the proposed and the original PSO algorithms (if applicable) are: $c_1 = c_2 = 2$, $c_3 = 1$, $N_{popsize} = 16$, $v_d^{\text{max}} = (b_d - a_d)/2$, $N_h = 3$, $v_d^{\text{min}} = (b_d - a_d)/1000$ ($d = 1, 2, \dots, D$) (a_d and b_d are, respectively, the inferior and superior bounds for the d^{th} variable), $P_{cr} = 0.02$, $\alpha = \beta = \gamma = 1.01$. For this test function, every algorithm is independently run 100 times, and the performance comparison results are given in Table I. In order to demonstrate the merits of the proposed algorithm, this test function is also solved by using a combined strategy (CS) of the SA and GA [4] and a universal tabu search (Utabu) [5].

From these numerical results, it is obvious that:

- 1) in terms of global search ability, the proposed one is extremely superior when compared to its precursor, since even for a difficult function having nearly infinite (15^5) local optima, the 100 independent runs of the proposed improved PSO algorithm can always find the global optimal solution;

- 2) in terms of convergence speed, the proposed one is marginally inefficient compared with its original counterpart;
- 3) however, the proposed improved PSO algorithm is the most efficient one among the three algorithms, i.e., the CS, the Tabu search method and the proposed one, which can find the global solution with a 100% probability.

It should also be pointed out that when the population size is set to 8, the smallest number, i.e., a total of 884 iterations, is observed for the proposed improved PSO algorithm, yet the proposed algorithm can still find the global optimal solution of the test function.

B. Application

The geometry optimization of the multisectional arcs of the pole shoe of a large salient pole hydro-generator [5] is solved, using the proposed PSO algorithm, to elucidate its viability in the study of practical inverse problems. The problem is formulated as

$$\begin{aligned} \max \quad & B_{f1}(X) \\ \text{s.t.} \quad & e_v - e_{v0} \leq 0, THF - THF_0 \leq 0 \\ & SCR - SCR_0 \geq 0, X'_d - X'_{d0} \leq 0 \end{aligned} \quad (16)$$

where B_{f1} is the amplitude of the fundamental component of the flux density in the air gap; e_v is the distortion factor of the sinusoidal voltage of the machine on no-load; THF is the telephone harmonic factor; X'_d is the direct axis transient reactance of the generator; SCR is the short circuit ratio.

The decision parameters are the center positions and radii of the multisectional arcs of the pole shoes as shown in Fig. 2. In the numerical implementation, B_{f1} is directly computed from the finite element solution of the no-load electromagnetic field of the machine, and other performances of (16) are derived based on these finite element solutions. This problem is solved using, respectively, the proposed improved PSO algorithm and an universal tabu search method [5]. The parameters used in this application for the proposed PSO algorithm is the same as those used in the previous section of the benchmark experiment. For the universal tabu search method, the parameters used are the same as those reported in [5]. Table II lists the final searched optimal solutions as well as the performance comparison results of different optimal methods for a 300 MW hydro-generator. The objective function value and the number of function evaluations for the proposed algorithm are, respectively, 1.10 (relative value) and 1742; which are compared, respectively, to 1.10 and 1768 of the tabu search method. Again, these numerical results confirm that the proposed improved PSO algorithm can virtually find the same optimal solution as that found by a well designed tabu search method.

IV. CONCLUSION

In the search for a robust global optimizer for computationally heavy inverse problems, an improved PSO method based on a comprehensive simulation of birds flocking or fish schooling

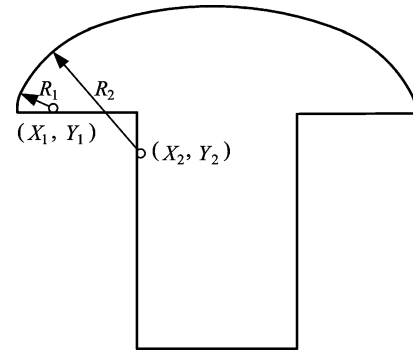


Fig. 2. The schematic diagram of the multisectional pole shoes.

TABLE II
GEOMETRY OPTIMIZATION RESULTS OF A 300 MW HYDROGENERATOR

Algorithms	R_1 (m)	R_2 (m)	X_1 (m)	Y_1 (m)	X_2 (m)	Y_2 (m)
Proposed	0.0532	1.7366	0.2570	5.6515	0.1950	5.1299
Tabu	0.0522	1.7376	0.2571	5.6510	0.1955	5.1302

is proposed and tested on a well designed mathematical function as well as a practical inverse problem. The primary numerical results as reported are good validation of the proposed algorithm. The future work of the authors will focus on a comprehensive study of the importance of the algorithm parameters and their effects upon the algorithm performances. Also, a theoretical convergence analysis of the algorithm would be carried out to make sure the PSO algorithm will become a powerful and widely recognized global optimizer in the study of optimization problems in different engineering disciplines.

ACKNOWLEDGMENT

The authors acknowledges the funding support of the Research Grant Council of the Hong Kong Special Administrative Region, China, under Project PolyU 5242/04E.

REFERENCES

- [1] J. Kennedy, "The PSO: social adaptation of knowledge," in *Proc IEEE Int. Conf. on Evolutionary Computation*, 1997, pp. 303–308.
- [2] J. Kennedy and R. Eberhart, "PSO optimization," in *Proc IEEE Int. Conf. on Neural Networks*, 1995, vol. IV, pp. 1941–1948.
- [3] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimization," *Lecture Notes in Computer Sci.*, vol. 2464, pp. 103–110, 2002.
- [4] R. Y. Tang, S. Y. Yang, Y. Li, G. Wen, and T. M. Mei, "Combined strategy of improved simulated annealing and genetic algorithm for inverse problems," *IEEE Trans. Magn.*, vol. 32, pp. 1326–1329, 1996.
- [5] S. Y. Yang, G. Z. Ni, and Y. Li, "An universal tabu search algorithm for global optimization of multimodal functions with continuous variables in electromagnetics," *IEEE Trans. Magn.*, vol. 34, pp. 2901–2904, 1998.