An Improved Particle Swarm Optimization Algorithm for Globally Optimal Designs of Electromagnetic Devices

Hui Zhang[1], Man Nie[1], Shiyou Yang[1], Junwei Lu[2] and Guangzheng Ni[1]
[1] College of Electrical Engineering, Zhejiang University, China
[2] Griffith University, Australia

The design problems arising from electromagnetic devices usually involve optimizations of complex objective functions. It is almost impractical for the deterministic methods to find the global optimal solutions of this kind of design problems due to the multimodal nature of the objective functions. The Particle Swarm Optimization (PSO) method is a stochastic, population-based optimal technique which is a new entrant to the family of evolutionary algorithms. Based on a comprehensive analysis of available particle swarm optimization algorithms, an improved PSO (IPSO) is proposed in this paper. The IPSO algorithm is tested by using both mathematical functions and the TEAM Workshop problem 22. The numerical results of the proposed IPSO are compared with those of the standard PSO algorithm and an improved tabu search method, which reveals that the proposed IPSO outperforms its precursors in perspectives of convergence speed and global search ability.

Key Words: Particle swarm optimization, Optimal design, Evolutionary algorithm.

1. Introduction

The use of stochastic and heuristic algorithms for the optimal design of electromagnetic devices has significantly flourished in the last few decades because most of the practical design problems involve objective functions with numerous local optima which make it is impossible for deterministic optimal algorithms to converge to a global solution. Inspired by natural and physical phenomena, many evolutionary algorithms such as genetic algorithm, ant colony method, tabu search algorithm, are proposed and used successfully to solve typical electromagnetic design problems.

The particle swarm optimization was developed by Kenney and Eberhart on metaphor of bird flocking and fish schooling for searching food [1], [2]. Nowadays, the PSO algorithm has become a strong competitor to other evolutionary algorithms and has shown great potential for optimal designs.

The particle swarm optimization is initialized with a population (also called swarm) of candidate solutions which are called particles and then searches for optima by updating particles generation by generation. In every iteration, each particle keeps track of two best solutions. The first one is the best solution it has attained so far, denoted by p_{best}. And the second one is the best solution founded so far by all of its neighbour particles, denoted by g_{best}. Therefore, in a PSO algorithm, the individuals could profit from its discoveries and previous experiences of all other particles during the search process while it still has the ability to search a wide landscape around the better solutions.

However, PSO is a relatively new evolutionary algorithm and still in its infancy stage compared to its well-developed counterparts. In this regard, an Improved PSO (IPSO) is proposed in this paper.

2. Particle Swarm Optimization

The particle swarm optimization method works with a population of potential solution rather than a single individual. Each individual in PSO searches in the solution space with a velocity which is dynamically adjusted according to its own and its companions' experiences. This means that if a particle discovers a promising solution, other particles will move toward to it, exploring the region more extensively in the process.

Suppose that the problem space of an optimal problem is *D*-dimensional and the swarm size of a PSO algorithm is $N_{popsize}$. The position of the i^{th} particle of a swarm can be described by using a position vector, $x_i = (x_1^i, x_2^i, ..., x_D^i)$, which is a feasible solution of an optimal problem. In a PSO algorithm, the position of a particle is updated by adding an increment vector, which is called the

Correspondence: Shiyou Yang, College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China, email: shiyouyang@yahoo.com

velocity vector of this particle, $v_i = (v_1^{\,i}, v_2^{\,i}, ...v_D^{\,i})$, to its current position vector. Let the best previous position that particle i has ever found be denoted by $P_i = (p_1^{\,i}, p_2^{\,i}, ..., p_D^{\,i})$, and the best position that its neighbour particles has found be called g_{best}. The position vector of particle i at iterative step k+1 is updated according to the following equations:

$$v_d^{\,i}(k+1) = w \cdot v_d^{\,i}(k) + c_1 r_1 (p_d^{\,i} - x_d^i(k)) + c_2 r_2 (p_g^i - x_d^i(k)) \quad (1)$$

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

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

where, c_1 and c_2 are two positive constants, r_1 and r_2 are two random numbers in the range [0,1].

To reduce the likelihood of the particle leaving the search space, the particles' velocities along each dimension are clamped to a maximum velocity $v_d^{\max}$. The value of $v_d^{\max}$ is usually chosen to be $k \times x_d^{\max}$, with $0.1 \le k \le 1.0$ [3] ($x_d^{\max}$ denotes the domain of the search space in the d^{th} direction).

The inertia weight w in equation (1) represents the degree of the momentum of the particles, $w \cdot v_d^{\,i}(k)$ and the residual determine the global search and the local search abilities, respectively.

3. An Improved PSO Algorithm

In a particle swarm optimization method, it is very important to balance the exploration and exploitation. Exploration is the ability to search a good optimum, hopefully the global one. Exploitation is the ability to converge the search around a promising candidate solution in order to locate the optimum precisely [4]. However, during the process of the evolution, some particles become inactive when their locations are close to a g_{best} and their velocities will be close to zero. Consequently, the PSO algorithm would be trapped in an undesired state and loses its diversity, which leads the evolution to be stagnated. As a result, when g_{best} is a local optimum, the swarm becomes premature convergence and the search performance will be degraded. To address such problem, this paper introduced an improved PSO algorithm with the goal of enhancing its global search ability without destroying its fast convergence speed.

3.1 Velocity and position updating

Since the two parameters r_1 and r_2 in (1) are generated independently, there are cases in which the two parameters are weighting too much simultaneously which would drive the search away from the local optimum or inversely the cognitive and social experiences are not fully used which leads to undermining the convergence performance of the algorithm. To make a full usage of the empirical information which searched by the particle and its neighbours, the parameters r_1 and r_2 are set to be interrelated in the IPSO. Therefore, the particle updates its velocity in the IPSO using the following equation:

$$v_d^{\,i}(k+1) = w v_d^{\,i}(k) + c_1 r_1 (p_d^{\,i} - x_d^i(k)) + c_2 (1 - r_1)(p_g^i - x_d^i(k)) \quad (4)$$

where r_1 and r_2 are two random parameters which are uniformly out of interval [0,1].

3.2 Introduction of a recombination operator

When the particles are stagnant at a local optimum, it is possible that some components of the position vectors may have attained the values of the corresponding components of the global optima. Therefore, it is hopeful to re-combine position vectors to find the global optimum according to the searched experiences of the lately stagnant iterations. Moreover, through this recombination operation, it is possible for the algorithm to alleviate the stagnation phenomenon. In this regard, a recombination operator is introduced. For the proposed recombination operator to work, the position of every best solution in the lastly k consecutive swarms are memorized, and denoted by L_{best} as

$$L_{best} = \{l^k{}_{best}, l_{best}^{k-1}, ... l_{best}^1\} \quad (5)$$

with

$$l_{best}^j = (x_{best1,}^j x_{best2}^j, ..., x_{bestD}^j) \quad (6)$$

If the IPSO stagnates at a local optimum for a certain cycles of iterations, the particles will be re-initiated by randomly combining the position vectors of the L_{best} particles using

$$x_i = (x_{best1}{}^{r_1}, x_{best2}{}^{r_2}, ..., x_{bestD}{}^{r_D}) \quad (7)$$

where, r_1, r_2 ,... r_D are random integers which are uniformly out of interval [1, k].

3.3 Flocking exploration

As discussed previously, in the searching process of a PSO algorithm, the particles are inclined to be gravitated toward p_{best} and g_{best} solutions, which will inevitably reduce the diversity of the particles in the feasible space and result in a stagnation phenomenon. In this point of view, when the fitness value of the searched best objective function does not improve for a certain number of generations, $N\ (1 < N < N_{popsize})$ particles in the current swarm having the inferior fitness values will be selected to form a flock to explore the feasible space by means of recombination operations. The parents are selected in such a way that the longer of the distance between two particles, the more probable they are selected to recombine. This process will be repeated until a superior location is found or a fixed iterative number $K_1 + K_2$ is reached.

3.4 Termination criterion

In the proposed IPSO, two terminative criteria are used. The first one is that the iterative process will be terminated when the number of the total iterations exceeds a certain limit. The second one is that once the consecutively generated particles with no improvements in the best solution searched so far is larger than a threshold K_{hold}, the searching process will be stopped. Mathematically, the search process is stopped if

$$\left|f_k - f_{k-l}\right| \le e \quad (l = 1,2,\dots K_{hold}) \tag{8}$$

where e is a predetermined precision parameter.

3.4 Algorithm description

In summary, the proposed IPSO algorithm can be described as follows:

Step 1: Initialization: Set algorithm parameters;

Step 2: Randomly generate the initial position and velocity of each particle, calculate their fitness value and set the fitness as the local best p_{best}, and find the g_{best};

Step 3: Update the position of every particle using (4), (3), (2), calculate the fitness of each particle and find the particle with the best fitness, update p_{best}, g_{best};

Step 4: Termination criterion is satisfied? If yes, go to step 8; if no, go to step 5;

Step 5: Has the fitness improved? If yes, go to step 3; if no, does the unimproved generation equals K? If no, go to step 3; if yes, go to step 6;

Step 6: Select particle to form exploration flock, initiate the particles of the flock using (7), and update them by using the combination strategy, calculate the fitness;

Step 7: Have found a better solution? If yes, let the other particles join the flock and go to step 3, If no, repeat until a fixed iterative number $K_1 + K_2$ is reached, and then go to step 3;

Step 8: Stop the algorithm, output the searched results.

4. Numerical Examples

4.1 Mathematical function experiment

A mathematical function as defined below is used to test and compare the proposed IPSO with the original PSO method. The test function is:

$$F_1(x,y) = \frac{(\sin\sqrt{x^2+y^2})^2 - 0.5}{(1+0.001(x^2+y^2))^2} + 0.5 \quad (-100 \le x, y \le 100) \tag{9}$$

The global optimal point of this function is located at (0,0) with $f_{opt} = 0$. Although this function seems to be a simple two-dimensional one, it has a great number of local minima in a concentric circle around the global minimum.

In the numerical experiments, the parameters used by both the proposed and the standard PSO algorithms are set as: $c_1 = c_2 = 2$, $N_{popsize} = 10$, $N = 5$, K=10, $K_1 = K_2 = 100$, $K_{hold} = 150$, $v_d^{\max} = (b_d - a_d)/2$, $v_d^{\min} = (b_d - a_d)/1000$ $(d = 1,2,\dots,D)$ (a_d and b_d are the inferior and superior bounds for the d^{th} variable). Parameter w is reduced from 1 to 0.4 gradually as the generation increases. The maximum number of iterations allowed in each run is set to 1000. Averaged values of 100 runs for each algorithm are used to demonstrate their general performances. The comparison results of the two algorithms for this experiment are tabulated in Table 1. Obviously, these numerical results demonstrate that the proposed IPSO not only converges faster but also has a better global search ability than the original PSO does.

Table 1. Performance comparison of the IPSO and the original PSO on the test function for 100 runs

Algorithms	No. of averaged Iterations	No. of runs finding The global solutions
IPSO	500	100
Original PSO	1000	42

4.2 Application

The TEAM Workshop problem 22 of a Superconducting Magnetic Energy Storage (SMES) configuration with three free parameters, as shown in Fig. 1 [5], is selected as the numerical example of the proposed IPSO algorithm for solving engineering problems. This optimal problem is formulated as

$$\min f = w_1 \frac{B_{stray}^2}{B_{norm}^2} + w_2 \frac{|Energy - E_{ref}|}{E_{ref}}$$
$$s.t.\ J_i \le -6.4\,|(B_{\max})_i| + 56 \ \ (A/mm^2) \quad (i = 1,2) \qquad (10)$$

where; *Energy* is the stored energy in the SMES device; $E_{ref} = 180$ MJ; $B_{norm} = 3\times10^{-3}$ T; w_1 and w_2 are weighting factors; J_i and $(B_{\max})_i$ $(i = 1,2)$ are, respectively, the current density and the maximum field in the i^{th} coil; and B_{stray}^2 is a measure of the stray fields which is evaluated along 22 equidistant points of line A and line B of Fig.1 by

$$B_{stray}^2 = \sum_{i=1}^{22} (B_{stray})_i^2 \Big/ 22 \qquad (11)$$

This problem is solved by using, respectively, the original PSO, the proposed IPSO algorithms and an improved tabu search method [6]. Table 2 lists the final searched optimal solutions of different optimal methods for this case study as well as the best solutions reported so far. The field contours under the optimized geometry using the proposed IPSO algorithm is shown in Fig. 2. From these performance comparisons it is clear that: (1) the searched best solutions by using the proposed IPSO are slightly better than those of the so far searched as reported in literature, (2) the proposed IPSO is the most efficient one among the three optimal algorithms tested.

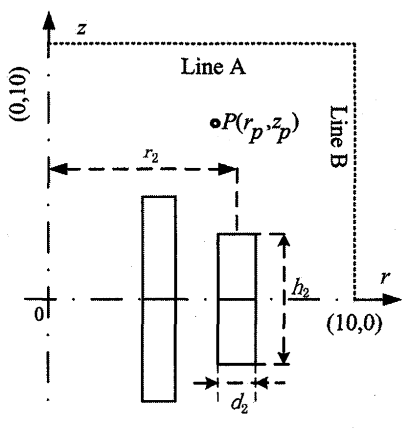

Fig. 1. The schematic diagram of a SMES

Table 2. Performances Comparison of different algorithms for Solving TEAM Workshop Problem 22

Method	R_2(m)	$h_2/2$ (m)	d_2(m)	No. of Iterations	Objective value
Proposed IPSO	3.085	0.2456	0.3815	1140	8.83×10^{-2}
PSO	3.10	0.246	0.379	1400	9.76×10^{-2}
ITS	3.10	0.240	0.388	1842	9.72×10^{-2}
Best ones	3.08	0.239	0.394	/	8.86×10^{-2}

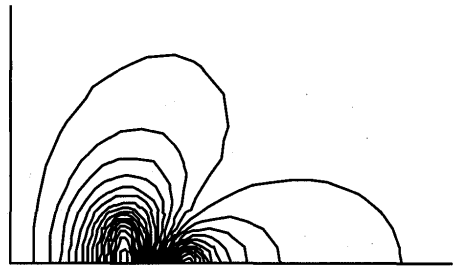

Fig. 2. The optimized field contours of the SMES

5. Conclusions

With the goal of developing a simple and efficient global optimizer for computationally heavy inverse problems, an improved PSO algorithm is proposed. The numerical results of both a mathematical function and a benchmark problem demonstrate that the global search ability and the convergence speed of the proposed IPSO are improved compared with those of the original PSO.

References

[1] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International conference on Neural Networks*, Perth, Australia, Vol. IV, pp. 1942-1948, 1995.
[2] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.
[3] R. C. Eberhart, P. K. Simpson, and R. W. Dobbins, *Computational Intelligence PC Tool*, Academic, ch.6, pp.212-226, 1996.
[4] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, Vol. 85, pp. 317-325, 2003.
[5] TEAM optimization benchmark problem 22 [online]. Available: *www.igte.tugraz.at.*
[6] S. L. Ho, S. Y. Yang, G. Z. Ni and H. C. Wong, "An improved tabu search for the global optimizations of electromagnetic devices," *IEEE Trans. Magn.*, Vol. 37, pp. 3570-3574, 2001.

Received: 20 July 2006/Revised: 31 January 2007