A Particle Swarm Optimization-Based Method for Multiobjective Design Optimizations

S. L. Ho[1], Shiyou Yang[2], Guangzheng Ni[2], Edward W. C. Lo[1], and H. C. Wong[3]

[1]Electrical Engineering Department, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[2]Electrical Engineering College of Zhejiang University, Hangzhou 310027, China
[3]Industrial Center, The Hong Kong Polytechnic University, Kowloon, Hong Kong

A particle swarm optimization (PSO) based algorithm for finding the Pareto solutions of multiobjective design problems is proposed. To enhance the global searching ability of the available PSOs, a novel formula for updating the particles' velocity and position, as well as the introduction of craziness, are reported. To handle a multiobjective design problem using the improved PSO, a new fitness assignment mechanism is proposed. Moreover, two repositories, together with the age variables for their members, are introduced for storing and selecting the previous best positions of the particle as well as that of its companions. Besides, the use of age variables to enhance the diversity of the solutions is also described. The proposed method is tested on two numerical examples with promising results.

***Index Terms*—Inverse problem, multiobjective optimal algorithm, optimal design, particle swarm optimization (PSO).**

I. Introduction

In finding the solutions of optimal designs of electromagnetic devices, a wealth of heuristic algorithms such as genetic (GA), evolution (EA), and simulated annealing (SA), as well as a new EA called the particle swarm optimization (PSO) algorithm, have all been used successfully to mimic the corresponding natural, or physical, or social phenomena. The PSO was developed by Kenney and Eberhart to model birds flocking and fish schooling for food [1]. As opposed to its well-developed counterparts, PSO is still in its infancy, and there are many associated problems that need further study. For example, the sharing of information among particles can be considered a blessing, in that the particles profit from the discoveries and previous experiences of all particles during the search process, to result in an enhancement of the convergence speed of the solver. However, such a feature is also a demerit in optimal problems involving multimodal objective functions, since the information sharing will also degrade the diversity of the algorithm and reduce the global searching ability of the algorithm. While the original PSO had difficulties in controlling the balance between explorations and exploitations [2], it has been successfully used in engineering optimizations studies by virtue of its simplicity [3]–[6]. Moreover, researchers are also seeing PSO as a very strong competitor to other algorithms in solving multiobjective optimal (MOP) problems, even though very few works have been reported [7]. In this study, a PSO-based vector optimal algorithm is proposed.

II. A PSO-Based Vector Optimal Algorithm

A. Brief Introduction of PSO Methods

The PSO method is a population-based one and is described by its developers as an optimization paradigm, which models the social behavior of birds flocking or fish schooling for food. Therefore, PSO works with a population of potential solutions rather than with a single individual. Unlike other population-based algorithms in which the evolutionary operators are used to manipulate the individuals, each individual in PSO flies in the parameter space with a velocity which is dynamically adjusted according to the flying experiences of its own and those of its companions. Therefore, every individual is gravitated toward a stochastically weighted average of the previous best point of its own and that of its neighborhood companions. Mathematically, given a swarm of N_{popsize} particles (hereafter, a "population" will be called a "swarm" and an "individual" a "particle" for terminology consistency), each particle i ($i \in \{1, 2, \ldots, N_{\text{popsize}}\}$) is associated with a position vector $x_i = (x_1^i, x_2^i, \ldots, x_D^i)$ (D is the number of decision parameters of an optimal problem), which is a feasible solution for an optimal problem; let the best previous position (P_{best}) (the position giving the best objective function value) that particle i has found in the parameter space be denoted by $p_i = (p_1^i, p_2^i, \ldots, p_D^i)$; the best position that the neighborhood particles of the ith particle have ever found is $g_i(g_{\text{best}})$, denoted using $g_i = (g_1^i, g_2^i, \ldots, g_D^i)$. At each iteration step $k+1$, the position vector of the ith particle, $x_i(k+1)$, is updated by adding an increment vector $\Delta x_i(k+1)$, denoted using the velocity $v_i(k+1)$. In the original PSO algorithm, the particles' positions are manipulated according to the following equations:

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

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

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

where c_1 and c_2 are two positive constants, r_1 and r_2 are two random parameters which are chosen uniformly within the interval [0, 1], and $v_d^{\max}$ is a parameter that limits the velocity of the particle in the dth coordinate direction.

This iterative process will continue swam by swarm until a stop criterion is satisfied, and this forms the basic iterative

Digital Object Identifier 10.1109/TMAG.2005.846033

process of a PSO algorithm. It is worth pointing out that in the right-hand side of (1), the second term represents the cognitive part of a PSO algorithm at which the particle changes its velocity based on its own thinking and memory, while the third term is the social part of a PSO algorithm at which the particle modifies its velocity based on the adaptation of the social-psychological knowledge. Essentially, the PSO algorithm is conceptually very simple, and can be implemented in a few lines of computer codes. Also, it requires only primitive mathematical operators and very few algorithm parameters need to be tuned.

B. Some Improvements on PSO Algorithms

The original PSO algorithm as given in (1)–(3) had difficulties in striking a balance between exploration and exploitation. Hence, the global search ability of PSO algorithm is restricted. To address this problem, some improvements are made on available PSOs as described below.

1) Velocity and Position Updating: In the original PSO algorithm as formulated in (1), a particle updates its velocity according to its own and its companion's flying experiences. The particle gravitates toward a stochastically weighted average of the previous best points of its own and its neighborhood. Since the two weighting parameters are independently and randomly generated, there are cases in which the two random parameters are both too large or too small. In the former case, both the personal and social experiences accumulated so far are over used and the particle is driven too far away from the local optimum. For the latter case, both the personal and social experiences are not used fully, and the convergence speed of the algorithm is reduced. However, in human social activities such as in hunting, most people do have the abstract reasoning ability to make the best use of his own knowledge and the group's knowledge collectively in the determination of the most promising regions to search. In other words, the two random weighting parameters (that of his own and that of his companions) are not completely independent, i.e., if one weighting parameter is large, the other should be small, and *vice versa*. By modeling this reasoning ability into the updating formula and noting the sum of the two interrelated weighting parameters is equal to 1, this paper is proposing to use only one random parameter to include the collective experiences of the individual particle and his neighbors when updating the velocity of the particle.

Moreover, to control the balance of global and local searches, another random parameter, r_2 in (4), is introduced in the proposed algorithm.

After including all the aforementioned improvement aspects, the new formula for velocity updating is

$$\begin{aligned} v_d^i(k+1) = r_2 v_d^i(k) + (1-r_2)c_1 r_1 \left(p_d^i - x_d^i(k)\right) \\ + (1-r_2)c_2(1-r_1)\left(g_d^i - x_d^i(k)\right) \end{aligned} \quad (4)$$

where r_1 and r_2 are two random parameters which are chosen uniformly within the interval [0, 1].

For birds flocking for food, there could be some rare cases that after the position of the particle is changed according to (3), a bird may not, due to inertia, fly toward a region at which it thinks is most promising for food. Instead, it may be heading toward a region which is in the opposite direction of what it should fly in order to reach the expected promising regions. As a consequence, in the step that follows, the direction of the bird's velocity should be reversed in order for it to fly back into the promising region. By modeling this fact in the proposed improved PSO algorithm, (4) is further modified to

$$\begin{aligned} v_d^i(k+1) = r_2 \mathrm{sign}(r_3) v_d^i(k) + (1-r_2)c_1 r_1 \left(p_d^i - x_d^i(k)\right) \\ + (1-r_2)c_2(1-r_1)\left(g_d^i - x_d^i(k)\right) \end{aligned} \quad (5)$$

where r_3 is a random parameter uniformly taken from the interval [0, 1], and *sign* is a sign function defined as

$$\mathrm{sign}(r) = \begin{cases} -1 & (r \leq 0.05) \\ 1 & (r > 0.05) \end{cases}. \quad (6)$$

2) Introduction of Craziness: In birds flocking or fish schooling, a bird or a fish often changes directions suddenly. This is described using a "craziness" factor and is modeled in the primary algorithm (paradigms) by using a craziness variable [1]. However, this operator is eliminated in subsequent paradigms by introducing a cornfield vector. To maintain the diversity of the particles in an optimization algorithm, it is necessary to retain the craziness operation in a PSO algorithm. Therefore, a craziness operator is reintroduced in the proposed algorithm to ensure that the particle would have a predefined craziness probability to maintain the diversity of the particles. Consequently, before updating its position using (3), the velocity of the particle is crazed by

$$v_d^i(k) = v_d^i(k) + P(r_4)\mathrm{sign}(r_4) v_d^{\mathrm{craziness}} \quad (7)$$

where r_4 is a random parameter which is chosen uniformly within the interval [0, 1]; $v_d^{\mathrm{craziness}}$ is a random parameter which is uniformly chosen from the interval $[v_d^{\min}, v_d^{\max}]$; and $P(r)$ and $\mathrm{sign}(r)$ are defined, respectively, as

$$p(r) = \begin{cases} 1, & (r \leq P_{\mathrm{cr}}) \\ 0, & (r > P_{\mathrm{cr}}) \end{cases} \quad (8)$$

$$\mathrm{sign}(r) = \begin{cases} 1, & (r \geq 0.5) \\ -1, & (r < 0.5) \end{cases} \quad (9)$$

where P_{cr} is a predefined probability of craziness.

C. Vector Optimal Method Based on the Improved PSO

To extend a scalar optimizer to solve a multiobjective optimal problem, the following two issues must be addressed: 1) means to accomplish the fitness assignment and selection in order to guide the search toward the Pareto-optimal set and 2) approaches to guarantee that the solutions obtained are not only the Pareto optimal, but are also uniformly distributed in the Pareto front. Due to the similarity of PSOs and other EAs, many multiobjective handling techniques available in EA can be incorporated into PSOs as reported in [7], [8]. However, some new ideas are proposed and used in this paper.

1) Introduction of Pareto Set: As similar to the global repository of [7], a Pareto set S_{Pareto} is introduced to report the searched Pareto solutions in the proposed algorithm for evaluating the fitness value of a particle in the iteration process.

2) Fitness Assignment: To favor the selection of individuals near the Pareto-optimal front and also to distribute the individuals uniformly along the tradeoff surface, other things being equal, the fitness assignment mechanism as proposed by Zitzler and Thiele [8] is extended and used in the proposed algorithm. Actually, the fitness value of a particle in a swarm is assigned in accordance to a two-stage process.

Step 1 Determine the strength of the individuals of S_{Pareto}.

For each individual i in S_{Pareto}, its strength s_i is proportional to the number of swarm members which are dominated by it, i.e.

$$s_i = \frac{n_i}{N+1} \tag{10}$$

where n_i is the number of swarm members which are dominated by individual i, N is the size of the swarm. The fitness value of the individual i is the inverse of its strength.

Step 2 Fitness assignment of particles in the swarm S.

The fitness value of a particle j in the swarm is computed by summing the strengths of all Pareto set members which dominate particle j. Mathematically

$$f_j = \frac{1}{1 + \sum_{i\,(i \succ j)} s_i} \tag{11}$$

where $i \in S_{\text{Pareto}}, j \in S, k \succ l$ means that solution k dominates solution l.

It should be pointed out that a promising byproduct of the proposed fitness assignment mechanism is that the densely populated individuals, other things being equal, are penalized due to the high strength value of their associated nondominated individuals. Therefore, there is no need for a fitness sharing in the proposed fitness assignment mechanism.

3) Memory and Selection of P_{best}*:* The solution of a multiobjective problem is not unique but is a set of tradeoffs of different objectives referred as the Pareto optimal. Therefore, the fitness value of two Pareto optimals may be the same, and, hence, it is difficult to select the P_{best} when a particle has found more than one Pareto solutions. Consequently, a fixed-length dynamic repository, one by one, for each particle, is introduced to memorize its latest found N_p Pareto solutions. To further maintain the diversity of the algorithm, an age variable, which will increase as the number of iteration increases, is assigned to each solution of these repositories. When selecting a P_{best} for a specific particle, the Roulette wheel selection mechanism is used to pick a solution among those of its repositories according to the weighted sums of their age variables and their fitness values. Once a solution is selected, its age variable is reassigned, the minimum value. Moreover, the age variable of the ith particle in the repository at iteration step k, age_i^k is updated using

$$\text{age}_i^k = 1.02\text{age}_i^{k-1}. \tag{12}$$

4) Selection of g_{best}*:* Since the "optimality" of all Pareto solutions is the same in a MOP, the same dilemma as that for selecting P_{best} may occur when one tries to evaluate and choose the g_{best} from S_{Pareto} if the fitness value is the only criterion. To overcome this dilemma, the age variable as described earlier is used. A byproduct of the introduction of the age variable is that it will further increase the diversity of the algorithm. The same selecting mechanism as that used for selecting P_{best} is employed to decide the g_{best} based on the weighted sums of their age variables and fitness values. Again, the age value of an individual in S_{Pareto} is increased swarm by swarm following the rule of (12).

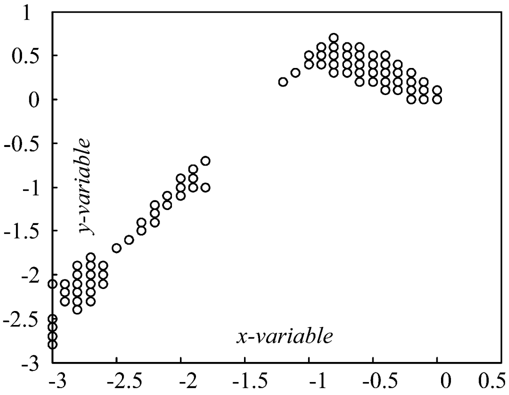

Fig. 1. Searched Pareto set of the function using the proposed algorithm.

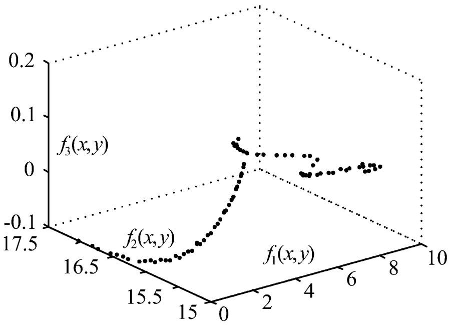

Fig. 2. Searched Pareto front of the function using the proposed algorithm.

III. Numerical Examples

To validate the proposed algorithm, it is firstly used to solve a deliberately designed mathematical function [9]

$$\begin{aligned} \min F &= [f_1(x,y)\ f_2(x,y)\ f_3(x,y)]^T \\ f_1(x,y) &= 0.5(x^2+y^2) + \sin(x^2+y^2) \\ f_2(x,y) &= (3x-2y+4)^2/8 + (x-y+1)^2/27 + 15 \\ f_3(x,y) &= 1/(x^2+y_2+1) - 1.1e^{(-x^2-y^2)} \\ &\quad \times (-3 \le x, y \le 3). \end{aligned} \tag{13}$$

This case study is very demanding for a MOP solver to obtain the desired Pareto solutions, since its Pareto front is a three-dimensional (3-D) curve following a convoluted path in the objective space. Figs. 1 and 2 depict, respectively, the searched Pareto set and Pareto front of this test function using the proposed algorithm after 1000 iterations. Comparing these results with Figs. 3 and 4 of [9] which are obtained by computing all possible variable combinations at a given computational resolution, it is clear that the proposed algorithm can produce nearly complete and uniform Pareto optimals.

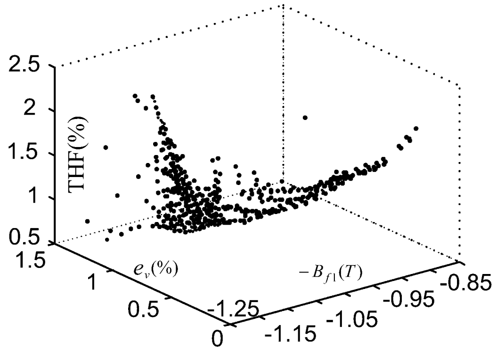

Fig. 3. Searched Pareto front of a 300-MW hydrogenerator.

After validating the proposed algorithm using a mathematical test function, it is also employed to optimize the geometrical design of the multisectional pole arcs of large hydro-generators [10] and the problem is formulated as

$$\begin{aligned} &\max B_{f1}(X) \\ &\min (e_v, \mathrm{THF}) \\ \text{s.t. } &\mathrm{SCR} - \mathrm{SCR}_0 \geq 0 \\ &X'_d - X'_{d0} \leq 0 \end{aligned} \tag{14}$$

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 a sinusoidal voltage of the machine at no-load conditions, THF is the abbreviation of the telephone harmonic factor, X'_d is the direct axis transient reactance of the generator, and SCR is the abbreviation of the short circuit ratio.

The decision parameters of this problem are the center positions and radii of the multisectional arcs of the pole shoes. In the numerical implementation, B_{f1} is directly computed from the finite element solution of the no-load electromagnetic field of the machine, and the other performances of (14) are derived based on these finite element solutions. After 1672 iterations, the proposed algorithm found 594 Pareto solutions (Fig. 3) for the 300-MW, 44-pole hydrogenerator. Obviously, the searched Pareto solutions of this case study using the proposed algorithm are uniformly distributed in the objective space. Comparing the performances of the proposed PSO with a similar SA-based algorithm which uses 1678 iterations to find 628 Pareto solutions of the same problem [10], the proposed algorithm can be considered a competitive algorithm, although not necessarily better than the other one in the reported case study.

IV. Conclusion

A PSO-based vector algorithm is proposed and tested on a standard mathematical function and an electromagnetic inverse problem. The numerical results on the mathematical function, which is ranked as a very demanding MOP problem for a vector optimizer by its designer, demonstrate the robustness of the proposed method in finding the Pareto solutions of extremely difficult MOP problems, while those on the inverse problem demonstrate the feasibility of applying the proposed algorithm in the optimization studies of electromagnetic devices. Moreover, the numerical results also show that the algorithm as reported is competitive to an SA-based one in our case study.

References

[1] J. Kennedy and R. Eberhart, "PSO optimization," in *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, 1995, pp. 1941–1948.

[2] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimization," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2002, vol. 2464, pp. 103–110.

[3] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability," in *Proc. Int. Conf. Intelligent System Application to Power Systems*, Rio de Janeiro, Brazil, 1999, pp. 117–121.

[4] P. C. Fourie and A. Groenword, "The particle swarm optimization algorithm in size and shape optimization," *Struct. Multidisc. Optim.*, vol. 23, pp. 259–267, 2002.

[5] A. R. Cockshott and A. Hartman, "Improving fermentation medium for Echinocandin B production part II: Particle swarm optimization," *Process Biochem.*, vol. 36, pp. 661–669, 2001.

[6] G. Ciuprina, D. Ioan, and I. Munteanu, "Use of intelligent-particle swarm optimization in electromagnetics," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 1037–1040, Mar. 2002.

[7] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, 2002, pp. 1051–1056.

[8] Z. Eckart and T. Lothar, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 3, pp. 257–271, Mar. 1999.

[9] A. David, V. Veldhuizen, and G. B. Lamont, "Multiobjective evolutionary algorithm test suites," in *Proc. ACM Symp. Applied Computing*, 1999, pp. 351–357.

[10] S. L. Ho, S. Y. Yang, H. C. Wong, and G. Z. Ni, "A simulated annealing algorithm for multiobjective optimizations of electromagnetic devices," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1285–1288, May 2003.

Manuscript received June 8, 2004.