A Population-Based Incremental Learning Vector Algorithm for Multiobjective Optimal Designs

S. L. Ho[1], Shiyou Yang[2], and W. N. Fu[1]

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

To alleviate the deficiency of crossover and mutation operations in standard genetic algorithms, the population-based incremental learning (PBIL) method is extended for multiobjective designs of inverse problems. To quantitatively measure the number of improvements in the whole objective functions and to quantify the amount of improvements in a specific objective function, a novel metric is proposed to "penalize" the fitness of a solution. Moreover, a selecting strategy for the best solutions of the latest iterations of an individual is introduced. Furthermore, multiple probability vectors are employed to enhance the diversity of the found solutions. Numerical experiments on low- and high-frequency inverse problems are carried out to demonstrate the feasibility of the proposed vector PBIL algorithm for hard multiobjective engineering inverse problems.

Index Terms—**Genetic algorithm (GA), inverse problem, multiobjective design, population based incremental learning (PBIL) method.**

I. Introduction

In ENGINEERING synthesis, it is not uncommon to ask a designer to fulfill several seemingly conflicting objectives simultaneously to satisfy a set of tradeoffs among different objectives. To help a decision maker make the best informed judgment according to the operating conditions and environments of the system or device, an acceptable multiobjective optimizer must have the ability to find as many Pareto optimal solutions as possible, and these solutions must also be as uniformly distributed as possible (i.e., a designer needs to minimize the distance between the found solutions to the true Pareto front), and maximize the diversity among the found Pareto solutions in objective and parameter spaces. To realize these goals, increasing endeavors have now been devoted to develop vector genetic algorithms (GAs). Consequently, the vector GAs have now become the standards for multiobjective optimal designs. While the three key operators of selection, crossover, and mutation in GAs are very complex in terms of theory and numerical implementation, there has been, however, only lukewarm attention devoted to reducing the difficulties of crossover and mutation operations. Hence, it is preferable to design a genetic-based optimal algorithm that inherits the searching power of available GAs and excludes the use of, at least partly, the aforementioned operators. In this regard, the population-based incremental learning (PBIL) evolution algorithm is a worthy candidate deserving further attention [1].

In general, the dominance concept is commonly used to assign fitness values in available multiobjective optimal algorithms. This approach determines only qualitatively the dominant relationships among different solutions; it cannot quantitatively measure the "level" of dominations due to the nature in which Pareto optimality is defined [2]. Moreover, the following two aspects are generally not properly addressed in available vector algorithms which are, namely: 1) the number of improved objectives and 2) the relevance of these improvements [3]. Also, very few efforts are given to assess the balance between conflicts in terms of convergence toward the Pareto front, and the requirement to maintain diversity in the discovered Pareto optimal solutions [4]. In this respect, existing multiobjective optimal algorithms are having difficulties in finding the best tradeoffs to distribute the computational resources uniformly while accomplishing the aforementioned two ultimate goals.

In the study being reported in this paper, the PBIL method is extended to solve multiobjective design problems to alleviate the deficiency in implementing the three aforementioned GA operations. Also, some improved approaches are incorporated to try alleviating the shortcomings of vector optimal algorithms to make the PBIL algorithm a common and standard vector optimizer.

II. Vector PBIL Algorithm

The PBIL method is developed by combining GA and competitive learning, which is often used in artificial neural networks (ANNs), in order to reduce the difficulties on the crossover and mutation operations in GA, yet retaining its stochastic search nature. The PBIL method therefore is similar to GA in using a binary-encoded representation of an optimal problem. Due to space limitations, please refer to [1] and [5] for details about the scalar PBIL algorithms. Hereafter, only the approaches and methodologies to extend PBILs to a vector optimizer are described. To help explain our proposed algorithm, a minimizing problem is described as an example in this paper.

A. Taking Account of Dominations and Improvements Into Considerations

To produce a set of Pareto solutions, the ranking approach is extended [6]. However, this approach only qualitatively determines the relationship of dominances and cannot quantitatively measure the number of improvements in the objective functions and cannot quantify the amount of improvements in a specified objective function. To address this issue, a metric

Manuscript received May 29, 2010; revised October 05, 2010; accepted November 08, 2010. Date of current version April 22, 2011. Corresponding author: S. L. Ho (e-mail: eeslho@polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2010.2093571

which measures the improvements in these two aspects is proposed for each individual. Precisely, for a specific individual x_i, the proposed metric is given by

$$\Delta_{improve}(x_i)=\sum_{\substack{j=1\\j\neq i}}^{N_P}\sum_{l=1}^{k} sign\left[f_l(x_j)-f_l(x_i)\right]\left[f_l(x_j)-f_l(x_i)\right] \tag{1}$$

where N_P is the size of the population, k is the number of objective functions, and $sign(x)$ is a sign function defined as

$$sign(x)=\begin{cases}1 & (x>0)\\ 0 & \text{(otherwise)}\end{cases}. \tag{2}$$

Incorporating this metric to "penalize" the fitness value of individual x_i yields the following formula:

$$f_{fit}(x_i)=wf_{fit}^{nor}(x_i) +(1-w)\left[\Delta_{improve}(x_i)\Big/\sum_{j=1}^{N_P}\Delta_{improve}(x_j)\right] \tag{3}$$

where $f_{fit}^{nor}(x_i)$ is the normally defined fitness of x_i, and w is a weighting constant between [0, 1].

B. Introduction of Multiple Probability Vectors

In existing scalar PBIL methods, a single probability vector is used to generate the whole population to enhance the searching efficiency of the algorithms. Inevitably, the employment of a single probability vector will reduce the diversity and global search ability of the algorithm. In the context of multiobjective optimization, this will reduce the diversity of the finally found Pareto solutions. In this regard, multiple probability vectors are proposed (i.e., every individual uses a different probability vector to generate its own children). The probability vector of the jth individual p_{j} is updated at the end of each generation by using

$$p^j(i)=(1.0-LR^j)\cdot p^j(i)+LR^j\cdot best^j(i) \tag{4}$$

where LR^{j} is the learning rate of individual j, and $best^j(i)$ is the value of the ith bit of the binary encoded string of the best solutions found by individual j in the latest N_L generations.

C. Selection of the Latest Best Solution of Each Individual

The Pareto optimal solutions of a multiobjective design problem are not unique, they are a set of tradeoffs among different objectives. Hence, there are difficulties in selecting the best solution in (4). Also, the fitness value of different Pareto optimal solutions may be the same even if the dominances and improvements are considered by using (3) in the proposed algorithm. Moreover, only the so far found Pareto solutions are identified in the context of the latest explored feasible solutions and the distances of these "Pareto solutions" to the true Pareto front are unknown. In other words, for those found "Pareto solutions," some are closer to the true Pareto front than others. In addition, the best solution found by an individual may remain unchanged for many generations, and this will reduce the diversities of the algorithm, giving rise to difficulties when the entire Pareto front are uniformly sampled.

To address the aforementioned issues and, especially, to use some *a priori* knowledge about the unexplored space to guide the searches toward finding the exact Pareto solutions, some preference functions are first defined by using the concept of the gradient balance approach [7]. For the ith objective, f_{i} and the jth objective f_{j}, a pair-wise-defined preference function of a specific solution x_{i} is formulated as

$$p_{ij}(x_i)=\frac{\nabla f_i(x_i)}{\|\nabla f_j(x_i)\|}\bullet\frac{\nabla f_j(x_i)}{\|\nabla f_i(x_i)\|}+1. \tag{5}$$

Mathematically, the function $p_{\mathrm{ij}}(x)$ would reach its minimum value of zero at all Pareto optimal points. In other words, this function provides a "gauging factor" that measures the "closeness" of a feasible point to the exact Pareto optimal solutions. Therefore, if one selects the latest best solutions using a Roulette wheel selection scheme with probabilities which are inversely proportional to the function values as defined in (6) in case where more than one best solutions have the same fitness value, the subsequent generation will be guided toward the intensification of exploitations around those points which are closer to the exact Pareto solutions, thereby equipping the algorithm with an ability to find better Pareto solutions. Moreover

$$\mathrm{Prob}(x_i)=\frac{1}{\sum\limits_{i\neq j}(p_{ij}(x_i)+\alpha)^{\beta}}\Bigg/\sum_{k=1}^{N_B}\frac{1}{\sum\limits_{i\neq j}(p_{ij}(x_k)+\alpha)^{\beta}} \tag{6}$$

where N_B is the number of the total best solutions found by x_{i} in the latest N_L generations; and α and β are two positive constants.

It should be pointed out that the proposed approach is not applicable when the objective functions are nondifferentiable and the gradient information is not available even by exploiting design points already visited. However, the proposed algorithm can still work if a random selection mechanism is used.

D. Scaling of Different Objectives

For an engineering synthesis problem, different criteria or objectives have significant differences in magnitude, resulting in an unevenly distributed Pareto frontier [8]. To eliminate this problem, the objective functions are normalized to

$$\hat{f}_i=\left(f_i-f_i^{\min}\right)/\left(f_i^{\max}-f_i^{\min}\right) \tag{7}$$

where $f_i^{\min}$ and $f_i^{\max}$ are, respectively, the minimum and maximum values of the ith objective function.

It should be pointed that (7) is a simple, albeit not highly effective procedure, and a more effective, yet more complex normalization can be realized by using the utopia and nadir vectors to scale the objective functions.

III. Numerical Examples

A. Case Study One—High-Frequency Inverse Problem

The proposed vector PBIL algorithm is first used to synthesize a nonuniformly spaced antenna array, as shown in Fig. 1, to illustrate its advantages and disadvantages. The array factor of the array can be computed by using

$$F_M(\theta)=\sum_{i=1}^{M}R_ie^{jkd_i\cos\theta} \tag{8}$$

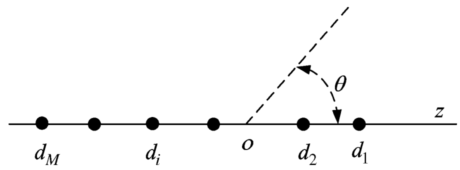

Fig. 1. Configuration of an M-element array placed on the z axis.

where R_i is the complex excitation coefficient of the ith element located at $z = d_i$ along the array direction z, and $k = (2\pi/\lambda)$ is the spatial wave number.

In this case study, the desired field pattern of a shaped beam with a cosecant variation is reconstructed by using a nonuniformly spaced antenna array. The desired pattern is defined as [9]

$$F_{\text{desired}}(\cos\theta) = \begin{cases} \text{cosecant}(cos\theta) & (0.1 \le cos\theta \le 0.5) \\ 0 & (\text{elsewhere}) \end{cases}$$
$$MSLL_{\text{Desired}} \le -22\,\text{dB} \tag{9}$$

where MSLL is the maximum sidelobe level.

To produce a radiation pattern which is as close as possible to the desired one, one will optimize a completely nonuniform antenna array with a minimal number of elements with respect to the following objective function:

$$\min f_1 = \sqrt{\frac{\sum_{i=1}^{N}\left[f_{\text{desired}}^{\text{norm}}(\theta_i) - f_{\text{designed}}^{\text{norm}}(\theta_i)\right]^2}{\sum_{i=1}^{N}\left[f_{\text{desired}}^{\text{norm}}(\theta_i)\right]^2}} \tag{10}$$

where $f_{\text{desired}}^{\text{norm}}(\theta_i)$ is the value of the normalized desired radiation pattern at the sampling point θ_i, $f_{\text{designed}}^{\text{norm}}(\theta_i)$ is the value of the normalized radiation pattern produced by a designed array of M elements, and N is the number of total sampling points set as 2001 in this case study.

To produce a field pattern with a minimum possible sidelobe level, the second objective of the proposed study is to minimize the MSLL. Consequently, the two-objective synthesis problem becomes

$$\min \begin{cases} f_1 \\ f_2 = \text{MSLLdesired} \end{cases}. \tag{11}$$

To quantitatively evaluate the performance of a vector optimizer, the *convergence* measure γ is used as the metric to measure the convergence of the found solution set to a known set of Pareto solutions, while the displacement metric is employed to gauge the uniformity or diversity performance of the found solutions over the non-dominated front [2]. If the set consisting of uniformly spaced true Pareto solutions, which are obtained using an exhaustive approach in this case study, is P^*, and the set of the final solution of a vector optimizer is Q, the proposed metrics are defined as

$$\gamma = \sum_{i=1}^{|Q|}\left(\min_{j\in[1,|P^*|]}(d(i,j))\right)\Big/|Q| \tag{12}$$

$$displacement = \sum_{i=1}^{|P^*|}\left(\min_{j\in[1,|Q|]}(d(i,j))\right)\Big/|P^*| \tag{13}$$

where $d(\cdot,\cdot)$ is the Euclidean distance of the two solutions.

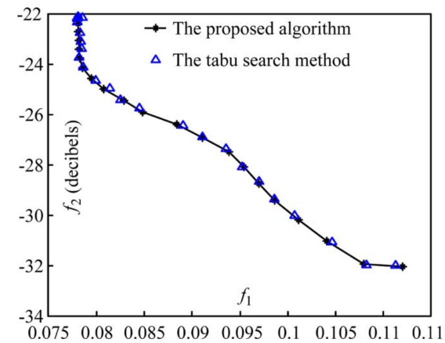

Fig. 2. The final solutions of the proposed and the tabu search algorithm.

To produce a field pattern which is close enough to the desired one, the proposed PBIL and tabu search [10] algorithms are used to optimize a 19-element nonuniform antenna array to find the Pareto front of (11). After a run of 29698 iterations, the proposed algorithm is found to have derived 119 Pareto solutions in the Pareto front as depicted in Fig. 2, which is compared to 34568 iterations of the tabu search algorithm with 121 Pareto optimals. It should be pointed out that for the sake of clarity, Fig. 2 shows only one in every four of the total Pareto solutions. The parameters γ and *displacement* of the final solution for the proposed algorithm are, respectively, 0.000016 and 0.00025; while the metrics γ and *displacement* of the final solution for the tabu search method are, respectively, 0.000022 and 0.00031. Since the value of the displacement metric of the final solutions for the proposed algorithm is considerably smaller than that for the tabu search method, it can be inferred that the diversity of the final solutions of the former is better than that of the latter. Moreover, Fig. 2 shows intuitively that most of the final solutions of the proposed PBIL method are better than those of the tabu algorithm. From these primary numerical results it can be seen that even there are slightly fewer finally found Pareto solutions obtained by the proposed algorithm when compared to that of the tabu search method, both the searching efficiency and the quality of the final solutions, in terms of the predefined metrics of the former, are significantly superior to those of the latter, for this case study being reported.

B. Case Study Two—A Low Frequency Inverse Problem

To demonstrate the feasibility and advantages of the proposed algorithm on hard inverse problems, a low frequency inverse electromagnetic problem, the design optimization of the multisectional pole arcs of a large hydrogenerator is studied [11]. This problem can be expressed mathematically as

$$\begin{array}{ll} \max & B_{f1}(X) \\ \min & (e_v, THF) \\ \text{s.t} & SCR - SCR_0 \ge 0 \\ & X_d' - X_{d0}' \le 0 \end{array} \tag{14}$$

where, B_{f1} is the amplitude of the fundamental component of the magnetic flux density in the air gap, e_v is the distortion factor of a sinusoidal voltage of the generator at no-load conditions, THF is the abbreviation of Telephone Harmonic Factor, SCR is the abbreviation of the Short Circuit Ratio, X_d' is the direct axis transient reactance of the generator.

For performance comparisons, this case study is also solved by using a vector particle swarm optimization (PSO) optimizer [11]. In the numerical implementation, the magnetic fields of the

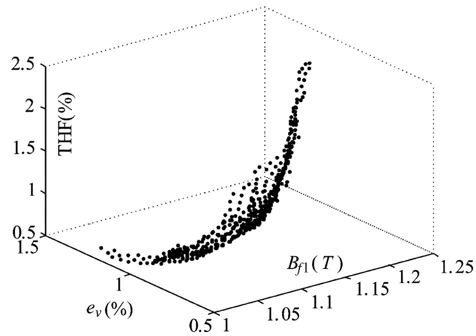

Fig. 3. Found Pareto solutions of the proposed method for example 2.

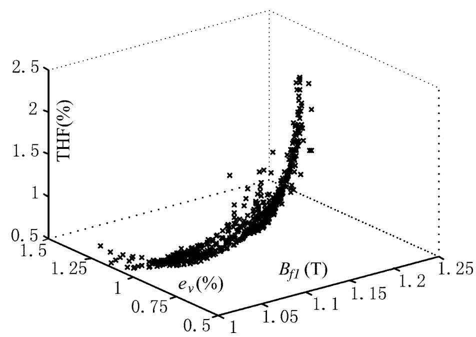

Fig. 4. Found Pareto solutions of the PSO algorithm for example 2.

generator at no load are computed using finite-element methods, and the performance parameters as required in (14) are then determined based on these finite-element solutions. Moreover, a practical 300-MW large hydrogenerator is used as the prototype machine in the numerical experiments. After 1865 iterations, the proposed algorithm identified 502 Pareto optimal solutions, which is compared with 594 Pareto solutions of 1672 iterations using the PSO method. To evaluate the quality of the final solutions, the two sets of Pareto solutions are extensively compared. To remove the influence of the computational error on the dominant relationships when comparing the two Pareto sets, a minimum error tolerance of 0.05 in the relative value is used hereafter. It is found that 126 out of the total 594 Pareto solutions of the PSO are dominated by at least one of the Pareto solutions of the proposed algorithm, whereas only 2 out of the total 502 Pareto solutions of the proposed algorithm are dominated by at least one of the Pareto solutions of the PSO. Figs. 3 and 4, depict, respectively, the finally found Pareto solutions of the two algorithms.

Moreover, to compare the performances of the two algorithms under the condition of equal computational costs, the iteration number of the proposed algorithm is limited to 1672, which is the same as that of the PSO method. Under such a restriction, the proposed algorithm has found 496 Pareto optimal solutions. To evaluate the quality of these 496 Pareto solutions, they are compared to the 594 solutions of the PSO algorithm. It is found that 120 out of the total 594 Pareto solutions of the latter are dominated by at least one of the Pareto solutions of the former, whereas only 4 out of the total 496 Pareto solutions of the former are dominated by at least one of the Pareto solutions of the latter.

From these numerical results, it can be observed that for the case studies being reported: 1) in terms of the quality of the final solutions, the proposed algorithm outperforms the PSO method since almost one-fifth of the finally found "Pareto" solutions of the latter are not "exact" Pareto solutions since they are dominated by solutions of the former while that of the final solutions of the former, which are dominated by the final solutions of the latter, are negligible; 2) indeed, the proposed algorithm can find more Pareto solutions compared with the PSO optimizer, although the number of final solutions of the latter is larger than that of the former; 3) in terms of diversity of the final solutions, the proposed algorithm is superior to the PSO algorithm since the distribution of the final solutions of the former is more uniform, especially in the neighborhood of the extreme solutions; and 4) in terms of convergence speed, the PSO algorithm outperforms the proposed method.

IV. Conclusion

This paper presents a feasible and robust vector optimizer based on the PBIL algorithm for the multiobjective optimal design of inverse problems. The numerical results have demonstrated that, when compared to an available vector optimizer for the case studies being reported in this paper, the proposed one has improved performance in realizing the two ultimate goals of a vector optimizer in terms of minimizing the distance between the found solutions to the true Pareto front, and maximizing the diversity among the found Pareto solutions in objective and parameter spaces. However, the most obvious disadvantage of the proposed algorithm is its complication in numerical implementations and theories.

Acknowledgment

This work was supported by the Research Grant Council of the Hong Kong SAR Government under Project PolyU 5166/07E.

References

[1] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *Proc. ICML*, 1995, pp. 38–46.
[2] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.
[3] A. Benedetti, M. Farina, and M. Gobbi, "Evolutionary multiobjective industrial design: The case of a racing car tire-suspension system," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 230–244, Jun. 2006.
[4] S. F. Adra and P. J. Fleming, "A diversity management operator for evolutionary many-objective optimization," *Lecture Notes Comput. Sci.*, vol. 5467, pp. 81–94, 2009.
[5] M. Gallagher and M. Frean, "Population-based continuous optimization, probabilistic modelling and mean shift," *Evol. Comput.*, vol. 13, pp. 29–42, 2005.
[6] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[7] P. Di Barba, "Multiobjective design optimization: A microeconomics-inspired strategy applied to electromagnetcis," *Int. J. Appl. Electromagn. Mechan.*, vol. 21, pp. 101–117, 2005.
[8] A. Messac and C. Mattson, "Generating well-distributed sets of Pareto points for engineering design using physical programming," *Optimiz. Eng.*, vol. 3, pp. 431–450, 2002.
[9] A. Akdagli and K. Guney, "Shaped-beam pattern synthesis of equally and unequally spaced linear antenna arrays using a modified tabu search algorithm," *Microw. Opt. Technol. Lett.*, vol. 36, pp. 16–20, 2003.
[10] S. L. Ho and S. Yang, "Multiobjective synthesis of antenna arrays using a vector tabu search algorithm," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 947–950, 2009.
[11] S. L. Ho, S. Yang, G. Ni, E. W. C. Lo, and H. C. Wong, "A particle swarm optimization-based method for multiobjective design optimizations," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1756–1759, May 2005.