A New Implementation of Population Based Incremental Learning Method for Optimizations in Electromagnetics

S. Y. Yang[1], S. L. Ho[2], G. Z. Ni[1], José Márcio Machado[3], and K. F. Wong[2]

[1]College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
[2]Department of Electrical Engineering, the Hong Kong Polytechnic University, Hong Kong
[3]São Paulo State University (UNESP), São José do Rio Preto, Brazil

To enhance the global search ability of population based incremental learning (PBIL) methods, it is proposed that multiple probability vectors are to be included on available PBIL algorithms. The strategy for updating those probability vectors and the negative learning and mutation operators are thus re-defined correspondingly. Moreover, to strike the best tradeoff between exploration and exploitation searches, an adaptive updating strategy for the learning rate is designed. Numerical examples are reported to demonstrate the pros and cons of the newly implemented algorithm.

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

I. INTRODUCTION

GENETIC algorithm (GA) is commonly entrusted as a standard algorithm for function optimizations. In GA, the three key operators are selection, crossover and mutation. However, these operators are very complex in terms of both theory and numerical implementation. It is thus 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 attentions [1]. The PBIL method is developed by combining GA and the competitive learning, which is often used in artificial neural network, so as to reduce the difficulties on the crossover and mutation operations in a GA, and yet retaining the stochastic search nature of the GA. PBIL therefore is similar to GA in using a binary encoded representation of an optimal problem.

The salient feature of this algorithm is the introduction of a real valued probability vector. The length of this real valued probability vector is identical to that of the encoded chromosome of a feasible solution in optimal problems. The value of each element of the vector is the probability of having a 1 in that particular bit position of the encoded chromosome. In every generation, this probability vector is used to generate a new population in such a way that the probability for the ith bit of a chromosome to become 1 is proportional to the value of the ith element of this probability vector. After evaluating the objective functions of the new population, this probability vector is updated by using only the best individual of the current population to help it shifting towards the chromosome of this new individual. Initially, all the values of the probability vector are set to 0.5 and sampling from this vector will thus produce a uniform distribution of the initial population on the feasible parameter space, as there is equal likelihood in the generation of 1 or 0 for each binary bit of the solution chromosome in this case. As the search progresses, the probability vector is expected to shift gradually to correspond to solutions with the highest fitness values.

The evolutionary procedure of a basic PBIL algorithm can be described by using the pseudo-Pascal as

Begin

Initialize the probability vector, p, to 0.5 s

$\mathrm{stopcriterion} := \mathrm{false}$

While $\mathrm{stopcriterion} := \mathrm{false}$ do

Generate a new population of individuals from the probability vector, p, where the probability of a 1 in each bit position of a chromosome is determined by the value of the corresponding element of the current probability vector;

Find the individual with the best function value among the individuals of the new generation. Denote the chromosome of this solution as best;

Update the vector p towards the best solution by using

$$p^j(i) = (1.0 - LR) \cdot p^j(i) + LR \cdot \mathrm{best}(i) \tag{1}$$

Mutate the probability vector p;

If the stop criterion is satisfied then

$\mathrm{stopcriterion} := \mathrm{true};$

Endif

Enddo

End

Digital Object Identifier 10.1109/TMAG.2006.892112

In (1), LR is the (positive) learning rate from the best solutions. Conceptually, the PBIL method is very simple when compared with GA, and can be implemented easily. It requires only primitive mathematical operators and very few algorithm parameters need tuning. By virtue of its straight-forward design philosophy and implementation simplicity, PBIL has attracted the attention of many researchers and has been used very successfully in solving a wide range of optimal problems in many disciplines [2]–[4].

However, PBIL is still in its development infancy when compared to its mature counterparts of common GA. For example, while the use of probability vector is considered a blessing because the algorithm will be less dependent on the selection and crossover operators, it can also be a disadvantage as the collective knowledge accumulated from other searched individuals are not used properly, and the probability vector is only updated using a limited number of best solutions. This may degrade the diversity of the populations or reduce the computational efficiency of the algorithm. Also, the use of only one probability vector to represent the whole population will inevitably reduce diversity, thereby degrading the global search abilities of the algorithm. Hence many PBIL methods are often trapped onto local optima. To enhance the global search ability while retaining the advantages of available PBIL algorithms, such as their conceptual simplicity and implemental easiness, some innovative ideas as detailed below are proposed in this study.

II. A New Implementation of PBIL Method

A. Introduction of Multiple Probability Vectors

One of the reasons for common PBIL methods to have low global search ability is because only one probability vector is used to represent the whole population. Consequently, multiple probability vectors are proposed, i.e., every individual uses different probability vectors to generate its own children. The probability vector of the jth individual, p^j, is thus updated at the end of each generation by using

$$p^j(i) = (1.0 - LR^j) \cdot p^j(i) + LR^j \cdot \mathrm{best}^j(i) \tag{2}$$

where LR^j is the learning rate of the individual j; $\mathrm{best}^j(i)$ is the value of the ith bit of the binary encoded string of the best solutions so far searched by individual j.

B. Utilization of Community Knowledge

To use fully the community's knowledge to guide the search towards promising regions of the feasible parameter space, it is proposed that the information about the best solutions explored by the neighborhood individuals of a solution is integrated with the updating of the probability vector of the specific solution. Moreover, to strike a good balance in exploiting an individual's knowledge and its neighbors' experiences, an interrelated random weighting parameter, based on modeling the reasoning ability of an "intelligent" society, is introduced, and (1) then becomes

$$\begin{aligned} p^j(i) = (1.0 - LR^j) \cdot p^j(i) + LR^j \cdot r \cdot \mathrm{best}^j(i) \\ + LR^j \cdot (1.0 - r) \cdot \mathrm{neighbor}_{\mathrm{best}^j}(i) \end{aligned} \tag{3}$$

where $\mathrm{neighbor}_{\mathrm{best}^j}(i)$ is the value of the ith bit of the binary encoded string of the best solution so far searched by the neighborhood individuals of individual j; r is a random parameter chosen from within the interval [0, 1].

It should be pointed out that the neighborhood is defined in a topological sense rather than a physical one in this paper, and such concept is similar to that used in particle swarm optimization (PSO) methods.

C. Shift Away From Worst Individuals

To move away from the worst individuals, the negative learning concept is also employed to update the proposed multiple probability vectors. This again is similar to that used in PSO. To avoid shifting away from either the best or worst solutions, when the difference between the two are becoming increasingly small because most individuals have converged to the same solution towards the final search stage, the formulae for amending the probability vector of individual j using the worst solution is proposed as

$$p^j(i) = \begin{cases} p^j(i) & (\mathrm{best}(i) = \mathrm{worst}(i)) \\ p^j(i) \cdot (1 - NLR^j) \\ \quad + \mathrm{best}(i) \cdot NLR^j & (\mathrm{best}(i) \neq \mathrm{worst}(i)) \end{cases} \tag{4}$$

where NLR^j is the negative learning rate of individual j; $\mathrm{worst}(i)$ and $\mathrm{best}(i)$ are, respectively, the values of the ith bits of the binary encoded strings of the worst solution searched by the current population and the best solution searched so far by the algorithm.

D. Mutation Operations

To further enhance the diversity of the populations of the proposed algorithm, two different mutation operators, one acts directly on the generated individuals and one on the probability vectors, are introduced. Here the first mutation operator is the same as that which is commonly used in a GA. To mutate the probability vector of the jth individual, the following equation is used

$$p^j(i) = \begin{cases} p^j(i) & (\text{if } \mathrm{random}(0,1) \geq \mathrm{Mut\_probability}) \\ p^j(i) + \mathrm{sign}\,(\mathrm{random}(0,1)) \times \Delta_{\mathrm{Mut}} & (\text{otherwise}) \end{cases} \tag{5}$$

where Mut_Probability is the mutation probability; Δ_{Mut} is the amount for mutation to affect the probability vector; $sign(\bullet)$ is a sign function defined as

$$\mathrm{sign}(r) = \begin{cases} 1 & (r \geq 0.5) \\ -1 & (r < 0.5). \end{cases} \tag{6}$$

E. Adaptive Updating of Learning Rates

On the right hand side of (3), the first term determines the global search ability that forces the search to generate individuals uniformly from the entire feasible space. The second and third terms dominate the local search ability that compel an individual to gravitate toward a stochastically weighted average of the previous best position of its own and the best position found by any of its neighbors, on the assumption that the initial probability vectors of 0.5 are being used.

Therefore, to strike the best balance between solution quality (global solution) and speed, the learning rates should be as small as possible at the beginning and be as large as possible at the end of the search process. Ideally, this increase from minimum to maximum should be realized automatically during the entire search process. For a specific period, the learning rates should also vary according to the characteristics of the specific search process in order to strike a good balance among the exploration (global) and exploitation (local) searches.

As an illustration, one considers the ratio of N_I^j/N_T^j, where N_I^j is the successive iterations with improvements in the objective function in the most recent N_T^j iterations of individual j. A high value of this ratio means that it is possible to locate better solutions using its current probability vector, and consequently its learning rate should be small; Conversely, a low value of such ratio implies the chance to find better solutions is very slim unless the current probability vector is modified, and consequently the learning rate should be increased. Thus the variations of the learning rates should satisfy both requirements, one in local and one in global sense as mentioned earlier. Consequently, an adaptive updating strategy is designed to adjust the learning rates of every individual after certain number of iterations in order to achieve these two goals. For example, for the jth individual, its probability vector will be updated after every N_T^j iterations using

$$LR^j = LR^j_{\min} + \left(LR^j_{\max} - LR^j_{\min}\right)\exp\left(-\frac{N_I^j}{N_T^j}\right) \quad (7)$$

where $LR^j_{\min}$ and $LR^j_{\max}$ are the minimum and maximum learning rates of the jth individual, respectively.

F. Stop Criterion

To reduce the iterative number effectively, a dynamic stop criterion is used in the proposed PBIL algorithm, i.e., the searching process will be automatically terminated once the number of successive iterations without improvements in the objective function exceeds a threshold value N_s.

III. Numerical Examples

To evaluate the performances of the proposed PBIL method, it is tested on both mathematical functions and engineering design problems. In these numerical experiments, the parameters used for each probability vector of the proposed and common PBIL methods are the same, and are set as: population size: 20, mutation probability: 0.02, maximum learning rate: 0.15, minimum learning rate: 0.05, negative learning rate: 0.05, $N_{\mathrm{T}} = 100$, $\Delta_{\mathrm{Mut}} = 0.04$. Every optimal method will stop its iterative process once the number of successive iterations without improvements in the best objective function value searched so far reaches 100.

A. Validation

A multimodal mathematical function is solved by using the proposed and the original PBIL algorithms in order to compare their performances. The function is defined as

$$\begin{aligned} \min \quad & f(x) = \frac{\pi}{n}\Big\{10\sin^2(\pi x_1) \\ & \quad + \sum_{i=1}^{n-1}\left[(x_i-1)^2\cdot\left(1+10\sin^2(\pi x_i)\right)\right] \\ & \quad + (x_n-1)^2\Big\} \\ \text{subject to} \quad & -10 \le x_i \le 10 \qquad (i=1,2,\cdots,5). \end{aligned} \quad (8)$$

This function has roughly 10^5 local optima, and the global one is at $x_i = 1$ $(i = 1, 2, \ldots, 5)$ with $f_{\mathrm{opt}} = 0$. To extensively assess the computational efforts of the proposed PBIL algorithm, two variants of it are also investigated in this study. The first, PBIL_Variant_A, uses a constant learning rate of 0.15, and the second, PBIL_Variant_B, has a constant learning rate of 0.1. All other parameters are the same as those set earlier. For the sake of completeness, a tabu method [5] is also used to study this problem for purpose of performance comparisons.

TABLE I
Performance Comparison of Different Optimal Methods on the Mathematical Function for 100 Independent Runs

Algorithms	Averaged iterations	Chance of finding global solution
Common PBIL	2895	39/100
Proposed PBIL	3425	98/100
PBIL_Variant_A	2544	41/100
PBIL_Variant_B	3668	95/100
Tabu	3616	100/100

In the numerical implementations, every algorithm is run independently 100 times and the "averaged" performance comparison results are given in Table I. From these numerical results, it is obvious that:

1) the global search ability of the proposed PBIL method in solving this extremely challenging mathematical function, which has 10^5 local optima, is increased from 39% to 98%;
2) the search efficiency of the proposed PBIL method, without the proposed updating scheme for the learning rates, is degraded, comparatively, while its global search ability is also degraded slightly if a low positive learning rate, as shown by the performances of PBIL Variant_B, is used;
3) increasing the learning rates will reduce the global search ability, although it can increase the computation efficiency of a PBIL algorithm, as demonstrated by the performances of the PBIL_Variant_A;
4) even if a relatively high constant learning rate is used, the global search ability of the proposed PBIL method is still superior (albeit marginally only) to that of available PBIL ones while the "averaged" iterations of the former are less than 88% of those of the latter, thereby validating the effectiveness of the improvements in terms of performance enhancement of the PBIL methods using the proposed algorithm;
5) considering both solution speed and global search ability collectively, the overall performance of the proposed one is comparable to those of the tabu method.

B. Application

The proposed PBIL algorithm is used to study the geometric optimal design of the end region of a power transformer to compare its performances with other well designed optimizers [5]. The goal of this case study is to reduce the maximum electric fields to avoid unnecessary flashovers in the transformer [5]. Therefore, this design problem is formulated as

$$\begin{aligned} \min \quad & E_{\max}(x) \\ \text{subject to} \quad & x_{ib} \le x_i \le x_{ia} \qquad (i = 1, 2, \cdots, 4) \end{aligned} \quad (9)$$

where $E_{\max}$ is the maximum value of the electric field in the end region; x_1 is the width of the electrostatic rings of the winding; x_i $(i = 2, 3, 4)$ is the width of the spacer i (Fig. 1).

The end fields are computed by using finite element method. The boundary value problem is

$$\begin{aligned} & \varepsilon\frac{\partial^2\varphi}{\partial x^2} + \varepsilon\frac{\partial^2\varphi}{\partial y^2} = 0 \\ & \varphi|_{S_1=0}, \quad \varphi|_{S_3=1}, \quad \partial\varphi/\partial n|_{S_2} = 0. \end{aligned} \quad (10)$$

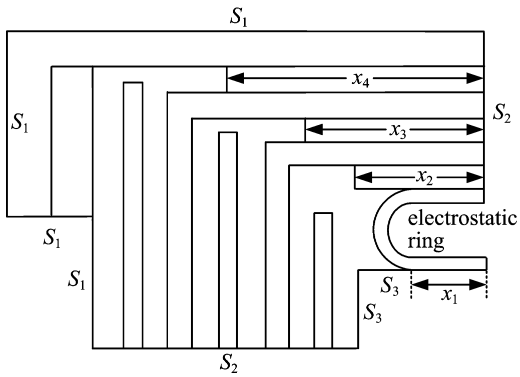

Fig. 1. Schematic diagram of the end region of the transformer being studied.

TABLE II
OPTIMIZED RESULTS OF A 63 MVA 110 kV POWER TRANSFORMER

Algorithms	x_1(mm)	x_2(mm)	x_3(mm)	x_4(mm)	Iterations	objective
Initial value	42.0	62.0	88.0	112.0	/	1.00
Proposed	39.9	40.3	85.8	120.5	2032	0.90
Tabu	41.1	39.6	86.1	113.1	1836	0.90
SL-SA	40.5	39.8	85.0	112.2	3475	0.91

In the numerical experiment, the proposed PBIL, a self-learning simulated annealing (SL-SA) [6], and a tabu search methods [5] are used to study this problem, and every optimal method will stop its iterative process once the number of successive iterations without improvements in the best objective function value searched so far reaches 50.

The geometry optimization results of the end region of a 63 MVA, 110 kV transformer using the aforementioned three optimal methods are given in Table II. The iteration numbers for different optimal methods are the average values of their 10 respective runs. It should also be noted that the values of the objective function given in the table are the results of one typical run, and measured in relative sense, using the maximum electric field of the solution region for the un-optimized end geometry as the base value. From these numerical results it can be seen that:

1) in terms of the solution quality, the proposed PBIL method is virtually the same as that of the tabu search algorithm, and is slightly more superior to the self-learning SA algorithm; however, the differences between the optimal results obtained using the three methods are negligibly small, in the range of about 1.1% only;
2) in terms of computation efficiency, the proposed PBIL method is comparable to that of the tabu algorithm, and significantly outperforms the self-learning simulated annealing algorithms because the average iteration number used by the former is less than 60% of that by the latter.

In summary, the robustness and effectiveness of the proposed algorithm for solving general electromagnetic inverse problems are validated using the primary numerical results as experimented in this case study.

IV. CONCLUSION

A new implementation of the PBIL method is proposed in this paper. The numerical results on both mathematical test functions and engineering design problems show:

1) The introduction of multiple probability vectors has increased the diversity of the population significantly, resulting in an enhancement in the global search ability of the algorithm;
2) The proposed adaptive updating strategy of the learning rates has equipped the algorithm with the ability to strike the best balance between exploration and exploitation searches, resulting in a reduction in iteration numbers while preserving the desired global search ability.

Therefore, the presented work provides an attractive alternative for both academicians and engineers when they have to deal with very challenging global optimization problems.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 50277036.

REFERENCES

[1] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *Proc. ICML95*, 1995, pp. 38–46.
[2] M. Gallagher and M. Frean, "Population-based continuous optimization, probabilistic modelling and mean shift," *Evol. Comput.*, vol. 13, pp. 29–42, 2005.
[3] S. Ghebreab, C. Carl Jaffe, and A. W. M. Smeulders, "Population-based incremental interactive concept learning for image retrieval by stochastic string segmentations," *IEEE Trans. Med. Imag.*, vol. 23, pp. 676–689, 2004.
[4] S. Bureerat and J. Limtragool, "Performance enhancement of evolutionary search for structural topology optimization," *Finite Elements Anal. Des.*, vol. 42, pp. 547–566, 2006.
[5] J. M. Machado, S. Y. Yang, and S. L. Ho, "A common tabu search algorithm for the global optimization of engineering problems," *Comput. Methods Appl. Mech. Eng.*, vol. 190, pp. 3501–3510, 2001.
[6] S. Y. Yang, J. M. Machado, G. Z. Ni, S. L. Ho, and P. Zhou, "A self-learning simulated annealing algorithm for global optimizations of electromagnetic devices," *IEEE Trans. Magn.*, vol. 36, pp. 1004–1008, 2000.

Manuscript received April 24, 2006 (e-mail: eeslho@polyu.edu.hk).