International Journal of the Physical Sciences Vol. 6(7), pp. 1803-1807, 4 April, 2011
Available online at http://www.academicjournals.org/IJPS
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

The experimental parameters optimization approach using a learning genetic algorithm

Lu Lu and Xiuxia Quan

School of Computer Science and Engineering, South China Technology, Guangzhou 510006, P. R. China.

Accepted 24 March, 2011

A learning genetic algorithm is proposed to solve the experimental parameters optimization problem. This method can not only enhance the efficiency of genetic algorithm through the pre-given user experience, but also improve the efficiency of genetic algorithm via learning the knowledge obtained from the optimization process. Experimental results suggest that the learning genetic algorithm can effectively optimize the experimental parameters.

Key words: Genetic algorithms, experimental parameters optimization, combinatorial optimization.

INTRODUCTION

As a result of the complexity and randomness of experimental parameters optimization and the difficulty in determining the optimal parameter combination, various researchers have done a lot of research works (Su et al., 2004; Lin et al., 2000; Marafona and Wykes, 2000), on neural networks, genetic algorithms and fuzzy logic to the effect of the experimental parameters optimization. In recent years, the experimental parameters optimization approach based on experimental design obtains a wide range of applications. Jia et al. (2003) executed the multi-objective process parameters optimization via a combination of the gray correlation analysis with orthogonal design. Huang et al. (2005) established the multiple linear regression equation to the electrical parameters on surface roughness and process time through the uniform design. However, the performance of these methods is not satisfied, more research works should be done to improve the efficiency of optimization algorithms.

Recently, more and more scholars have studied the applications of the interaction between evolution and learning (Xing et al., 2008a, b, 2010a). Normally, these approaches keep useful features of previous individuals to improve the performance of current individuals (Xing et al., 2006a, 2007, 2009). In fact, such approaches outperform traditional evolutionary algorithms on several benchmarks (e.g., flexible job shop scheduling problem, traveling salesman problem, and capacitated arc routing problem) (Xing et al., 2006b, 2010b; Ho et al., 2007; Louis and McDonnell, 2004). In a similar fashion, a Learning Genetic algorithm (LGA) is proposed in this work.

This paper is organized as follows. The experimental parameters optimization problem is formulated. The learning genetic algorithm is introduced in detailed. Also, computational experiments and comparison studies are reported. Finally, some concluding remarks are made.

PROBLEM FORMULATION

The experimental parameters optimization problem can be formulated as follows.

(1) Inputs: In the experimental parameters optimization problem, the inputs are the different parameters in the given experiments. The parameters can be the continuous variables, discrete variables and Boolean variables. The inputs of experimental parameters optimization problem can be displayed as

$$X=\{x_1,x_2,\cdots,x_a,x_{a+1},\cdots,x_{a+b},x_{a+b+1},\cdots,x_{a+b+c}\} \tag{1}$$

Where $x_1,x_2,\cdots,x_a$ denote the a continuous variables $x_{a+1},\cdots,x_{a+b}$ denote the b discrete variables, and $x_{a+b+1},\cdots,x_{a+b+c}$ denote the c Boolean variables.

(2) Outputs: The outputs of experimental parameters optimization problem can be displayed as

$$X^*=\{x_1^*,x_2^*,\cdots,x_{a+b+c}^*\} \tag{2}$$

*Corresponding author. E-mail: lul@scut.edu.cn.

(3) Constraints: To these continuous variables, the constraints can be displayed as

$$l_i \le x_i \le u_i, \forall i = 1, 2, \cdots, a$$

l_i and u_i denote the minimum value and the maximum value if variable x_i.

To these discrete variables, the constraints can be displayed as

$$x_i \in Z, l_i \le x_i \le u_i, \forall i = a+1, a+2, \cdots, a+b$$

l_i and u_i denote the minimum value and the maximum value if variable x_i.

To these Boolean variables, the constraints can be displayed as

$$x_i \in \{0,1\}, \forall i = a+b+1, a+b+2, \cdots, a+b+c$$

(4) Objectives: In most experiments, there are two kinds of objectives: one is the maximum objective, and another is the minimum objective. For this reason, the objectives of experimental parameters optimization problem can be displayed as

$$\begin{cases} \max\{J_1, J_2, \cdots, J_s\} \\ \min\{J_{s+1}, J_{s+2}, \cdots, J_{s+t}\} \end{cases} \quad (3)$$

Where $J_1, J_2, \cdots, J_s$ denote the s maximum objective, and $J_{s+1}, J_{s+2}, \cdots, J_{s+t}$ denote the t minimum objective.

THE LEARNABLE GENETIC ALGORITHM

To the experimental parameters optimization problem, a learning genetic algorithm was proposed in this paper. The learning genetic algorithm is characterized by the following points: it can not only enhance the efficiency of genetic algorithm through the pre-given user experience, but also improve the efficiency of genetic algorithm via learning the knowledge from the optimization process.

Population initialization

Chromosome, also called individual, is the encoded solution for specific problems. To the experimental parameters optimization problem, this paper adopts a matrix Pop with the $a+b+c$ size as one chromosome. Where Pop denotes Population, which is the set of chromosomes. In this paper, each chromosome is initialized using random generation.

Selection operation

In the learning genetic algorithm, the binary tournament is employed to implement the selection operation. First, two chromosomes are randomly selected, and then the least-cost one is kept.

To improve the performance of the learning genetic algorithm, the elitism among the current population is directly copy to the next population.

Crossover operation

In the genetic algorithms, crossover operator is a main method for producing new individuals. In order to inherit a set of building blocks from each parent, the crossover operator recombines the gene-codes of two parents and produces offspring. It is very pivotal to select a small, but representative sample of points as the potential offspring. Therefore the orthogonal crossover with quantization (Leung and Wang, 2001) is applied as the crossover operator of learning genetic algorithm.

Mutation operation

Mutation takes place on some newly formed children in order to prevent all solutions from converging to their particular local optima. According to traditional ways, to perform mutation on a chromosome, it randomly generates an integer $j \in [1, N]$ and a real number $z \in [l_j, u_j]$, and then replaces the j^{th} component of the chosen chromosome by z to get a new chromosome.

Termination conditions

The learning genetic algorithm is terminated when one of the following conditions are satisfied: the elitism is not improved in the successive SI generations, and the maximum MI generations are exhausted.

The user experience

In order to enhance the efficiency of genetic algorithm, we can apply these user experiences to guide the evolution of learning genetic algorithm. Normally, the users can learn some experiences from many experiments. For example, the influence of one variable to the final objective (Figure 1) is very important to deal with the practical problem.

As displayed in Figure 1(a), $f(x)$ is increased with the increasing of variable x while in Figure 1(b), $f(x)$ decreased variable x increases. Also, in Figure 1(c), the retaliation between the objective $f(x)$ and variable x is the U Curve while in Figure 1(d), the retaliation between the objective $f(x)$ and variable x is the reversed U Curve. As shown in Figure 1(e), the retaliation between the objective $f(x)$ and variable x is one line, and the influence of variable x to objective $f(x)$ is small. As displayed in Figure 1(f), the retaliation between the objective $f(x)$ and variable x is too complex. In fact, these user experiences are very important to enhance the efficiency of genetic algorithm.

Learning the knowledge from the optimization process

In this paper, the feasible space $[l_i, u_i]$ of variable x_i is divided

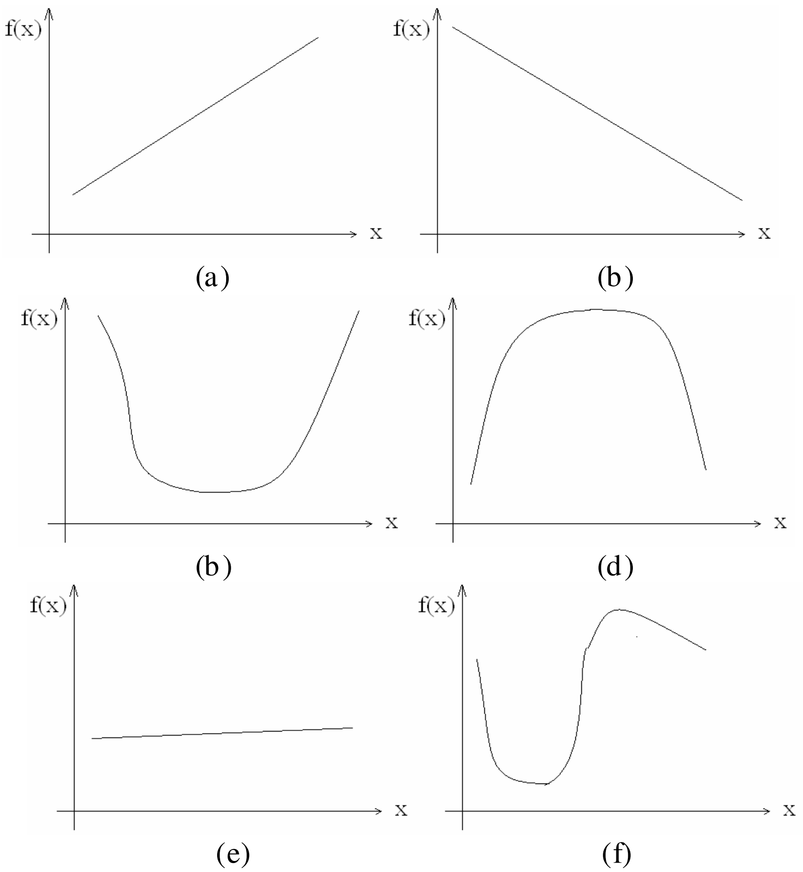

Figure 1. The user experience learning from many experiments.

Table 1. A favorable choice of parameters to learning genetic algorithm.

Name	Role	Value
P_S	Population size	100
P_X	Crossover rate	0.9
P_m	Mutation rate	0.05
SI	The elitism is not improved in the successive generations	200
MI	The maximum generations	2000

into 10 subintervals. To these near-optimal solutions, the times of variable x_i fall into each subinterval are recorded. We called this information as knowledge. This knowledge is very important to improve the performance of genetic algorithm. In this paper, the knowledge is employed to guide the sequential crossover and mutation of learning genetic algorithm.

EXPERIMENTAL RESULTS

The learning genetic algorithm was implemented using Visual C++ language, and executed on a personal computer with the 2 GHz processor and 2 GB memory. We established a favorable choice of parameters, as listed in Table 1, by means of systematic experimentation. In this paper, the final experimental results were averaged over 100 trials.

In order to validate the performance of learning genetic algorithm, the standard genetic algorithm (SGA), the intelligent genetic algorithm (IGA) (Xing et al., 2006a) and the multiprogramming genetic algorithm (MGA) (Xing et al., 2007) are applied to compare with the learning genetic

Table 2. The experimental results of learning genetic algorithm

Approach	Objectives	Experimental results
Standard genetic algorithm	J_1 (maximum)	210.98
	J_2 (maximum)	235.61
	J_3 (maximum)	198.18
	J_4 (minimum)	35.68
	J_5 (minimum)	33.75
Intelligent genetic algorithm	J_1 (maximum)	215.39
	J_2 (maximum)	241.52
	J_3 (maximum)	202.83
	J_4 (minimum)	32.06
	J_5 (minimum)	30.69
Multiprogramming genetic algorithm	J_1 (maximum)	221.13
	J_2 (maximum)	248.05
	J_3 (maximum)	209.11
	J_4 (minimum)	28.13
	J_5 (minimum)	26.05
Learning genetic algorithm	J_1 (maximum)	230.55
	J_2 (maximum)	253.67
	J_3 (maximum)	231.39
	J_4 (minimum)	21.05
	J_5 (minimum)	22.77

algorithm. These different versions of genetic algorithms were implemented using Visual C++ language in this research. The final experimental results were summarized in Table 2.

From the experimental results of Table 2, we can see that, to the maximum objectives, the objective obtained by learning genetic algorithm is larger than other three approaches. To these minimum objectives, the objective obtained by learning genetic algorithm is smaller than other three approaches. In terms of the optimal objective, the learning genetic algorithm is powerful than other different genetic algorithms. In summary, experimental results suggest that the learning genetic algorithm can effectively optimize the experimental parameters.

Conclusion

The contribution of this paper can be summarized thus: A learning genetic algorithm is proposed to solve the experimental parameters optimization problem. This method can not only enhance the efficiency of genetic algorithm through the pre-given user experience, but also improve the efficiency of genetic algorithm via learning the knowledge obtained from the optimization process.

ACKNOWLEDGMENTS

This paper is supported by the China National Science Fund and the Guangzhou Government Special Fund.

REFERENCES

Su JC, Kao JY, Tarng YS (2004). Optimization of the electrical discharge machining process using a GA-based neural network. Int. J. Adv. Manuf. Technol., 24: 81-90.

Lin JL, Wang KS, Yan BH (2000). Optimization of the electrical discharge machining process based on the Taguchi method with fuzzy logics [J]. J. Mater. Process. Technol., 102: 48-55.

Marafona J, Wykes C (2000). A new method of optimizing material removal rate using EDM with copper-tungsten electrodes. Int. J. Mach. Tools Manuf., 40(2): 153-164.

Jia ZY, Zhou M, Yang LW (2003). Study on the Optimization for Electrical Discharge Machining Micro and Small Hole. Chinese J. Mech. Eng., 39(2): 106-112.

Huang RN, Di SC, Chi GX (2005). Study of the Technology of MWEDM Based on Uniform Design. Electromachining Mould, (1): 12-15.

Xing LN, Philipp R, Chen YW (2010a). An Evolutionary Approach to the Multi-depot Capacitated Arc Routing Problem. IEEE Trans. Evol, Comput., 14(3): 356-374.

Xing LN, Chen YW, Yang KW (2008a). A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem. Eng. Appl. Artif. Intell., 21(8): 1370-1380.

Xing LN, Chen YW, Yang KW (2008b). Double Layer Ant Colony Optimization for Multi-objective Flexible Job Shop Scheduling Problems. New Generat. Comput., 26(4): 313-327.

Xing LN, Chen YW, Yang KW (2009). An Efficient Search Method for Multi-objective Flexible Job Shop Scheduling Problems. J. Intell. Manuf., 20(3): 283-293.

Xing LN, Chen YW, Shen XS (2007). Multiprogramming Genetic Algorithm for Optimization Problems with Permutation Property. Appl. Math. Comput., 185(1): 473-483.

Xing LN, Chen YW, Cai HP (2006a). An intelligent genetic algorithm designed for global optimization of multi-minima functions. Appl. Math. Comput., 178(2): 355-371.

Xing LN, Chen YW, Shen XS (2006b). A Constraint Satisfaction Adaptive Neural Network with Dynamic Model for Job-Shop Scheduling Problem. Lect. Notes Comput. Sci., 3973: 927-932.

Xing LN, Chen YW, Wang P (2010b). A Knowledge-based Ant Colony Optimization for Flexible Job Shop Scheduling Problems. Appl. Soft Comput., 10(3): 888-896.

Ho NB, Tay JC, Lai EMK (2007). An Effective Architecture for Learning and Evolving Flexible Job-Shop Schedules. Eur. J. Oper. Res., 179(2): 316-333.

Louis SJ, McDonnell J (2004). Learning with Case-Injected Genetic Algorithms. IEEE Trans. Evol. Comput., 8(4): 316-328.

Leung YW, Wang F (2001). An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. Evol. Comput., 5(1): 41-53.