Application of Evolutionary Neural Networks in Prediction of Tool Wear in Machining Process[1)]

LI Xiang-Long YIN Guo-Fu LUO Hong-Bo
(*School of Manufacturing Science and Engineering, Sichuan University, Chengdu* 610065)
(E-mail: lxlnc@163.com)

Abstract An improved evolutionary method based on real-number encoding is presented to optimize the connection weights and the topology of neural networks. The algorithm could adaptively adjust magnitude of mutation according to individual fitness, and mutation rate will increase with evolving generations as soon as evolution gets into stagnancy. Experiments show that the evolutionary artificial neural network is efficient to predict tool wear in electrical discharge milling machining and the prediction results are better than the standard BP neural networks. The proposed prediction model can be used for tool compensation on-line in electrical discharge milling machining.

Key words Evolutionary algorithm, neural network, tool wear prediction, electrical discharge milling machining

1 Introduction

Evolutionary algorithms (EA) search from a population representing different sample points in the search space. So they could find the globally optimal (or near optimal) solution with a higher probability. The combination of ANN and EA produces evolutionary artificial neural networks (EANN). Evolution has been introduced into ANN at three different levels: connection weights, architectures, and learning rules[1]. Porto, V. W. *et al.* demonstrated that evolution and annealing approaches outperformed back propagation consistently[2]. Fogel indicated that evolutionary programming (EP), evolutionary strategy (ES) and genetic algorithm (GA) were similar, and there appeared to be no scientific rationale for discriminating between 'genetic' and 'evolutionary' computation[3].

Electrical discharge milling machining (EDMM), as a new kind of electrical discharge machining (EDM) technologies developed in the late 80's of 20th century, applies standard and simple shaped electrode which moves along certain tracks to different discharge place between electrode and workpiece for required shape[4]. Because the wear of electrode in EDMM is substantial, compensation for the wear is crucial to improve machining precision and productivity. General strategies to compensate tool wear are: replacing or dressing tool and compensation, tool wear measuring and compensation[5]. However, the methods are unsuitable for real production due to frequent interruption of machining process. A better method is to establish mathematical model for tool wear prediction according to wear rules obtained from vast processing experiments.

Because of the control problem of the stochastic, non-linear, multi-parameters and time-varying, it is difficult to accurately build a mathematical model for predicting tool wear. A prediction model of tool wear based on an improved EANN is proposed, which can be used for tool compensation on-line.

2 Design of EANN

An important task in the development of EANN is to design a set of genetic opera-

1) Supported by National Natural Science Foundation of P. R. China(50275100)
Received November 4, 2002; in revised form August 5, 2003
收稿日期 2002-11-04; 收修改稿件日期 2003-08-05

tors. The most widely used genetic operators are selection, crossover and mutation. Three genetic operations play different roles in EA. The selection acts as a guide, by which the initial random sampling of early generations is concentrated towards those areas of the search space demonstrating better-than-average performance. The crossover provides the necessary mechanism to exploit beneficial material originating from two different chromosomes. The mutation could bring new gene to recover diversity of population[6]. Therefore, the design of the mutation operator is crucial to improve EA local search ability and accuracy of the evolutionary training.

2.1 Encoding scheme

The most convenient representation of connection weights is binary strings, which are termed chromosomes. The advantages of the binary representation lie in its simplicity and generality. It is simple to apply evolving operation to binary strings. However, chromosomes representing large EANN will become very long and the evolution will become very inefficient. Real-number representation can overcome some shortcomings of the binary encoding scheme[6], and domain knowledge could be introduced into evolutionary search procedures to improve their performance. Therefore, Real-number representation is adopted in this paper.

The number of hidden layer nodes is randomly initialized in interval (0, 50), and all the weights initialized at random are a series of real numbers in interval $[-1,1]$.

2.2 Selection operator

Selection mechanisms have great influence on EA convergence rate. The choice based on individual fitness-proportionate will lead to premature convergence and stagnation. The linear rank for individuals order is adopted in this paper. Let P sorted individuals be numbered as 1, 2,···, P. The formula for computing the reproduction probability of the ith individual can be expressed as follows.

$$p_i = (K \times i + B) / \sum_{j=1}^{P} (K \times j + B) \tag{1}$$

where coefficient K affects on selection probability of individuals. The larger the value of K, the higher the selection pressure is. But the selection pressure will depend inversely on parameter B. To ensure an appropriate evolutionary rate and diversity of population, coefficients K and B have to be properly determined.

2.3 Crossover operator

Because of the changeability of network architectures which are mapped by individuals in population, crossover operators will be different from the traditional ones. Two parents are chosen from the population at random where the weights for the same number will be adjusted by crossover operation under the condition that the remaining weights are kept unchanged. Each individual is taken as a real connection weight vector. Let $\boldsymbol{G}_1 = (g_{11}, g_{12}, \cdots, g_{1m})$ and $\boldsymbol{G}_2 = (g_{21}, g_{22}, \cdots, g_{2n})$ denote two parents respectively, $m<n$. Let $\boldsymbol{R}_1 = (r_{11}, r_{12}, \cdots, r_{1m})$ and $\boldsymbol{R}_2 = (r_{21}, r_{22}, \cdots, r_{2n})$ denote two offspring. The components for offspring $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ which have been obtained by the crossover operation can be expressed as follows.

$$\begin{array}{ll} r_{1i} = \alpha_i \times g_{1i} + (1-\alpha_i) \times g_{2i}; & (i<=k) \\ r_{1i} = g_{1i}; & (i>k) \\ r_{2i} = (1-\alpha_i) \times g_{1i} + \alpha_i \times g_{2i}; & (i<=k) \\ r_{2i} = g_{2i}; & (i>k) \end{array}$$

where k denotes crossover point, $k<m$. $\alpha_1, \alpha_2, \cdots, \alpha_k$ are random numbers generated in interval (0, 1).

2.4 Mutation operator

Traditional binary mutation will not be used directly due to real-number encoding scheme, and special mutation operators have to be designed. The mutation operators could be divided into two parts: mutation rate and magnitude of mutation which can be used to adjust weights. The two main mutation operations for EANNs are as follows. 1) Hidden nodes and their corresponding weights are added (or deleted); 2) Some weights selected at random are adjusted.

2.4.1 Mutation rate

The mutation rate in common EA is usually a small constant or variant related to individual fitness, and the mutation rate bears no relation to the number of evolutionary generations. Consequently, population falls into premature convergence and stagnancy after a number of generations. The mutation rate which depends on the number of evolutionary generations can be calculated as follows

$$p_m = p_{m_0} + t \times C \tag{2}$$

where p_m denotes current generation mutation rate, p_{m_0} is an initial mutation rate, set to be a small value between (0, 1). C is user-defined small constant set in the range (0, 1), which influences the difference of mutation rates between two successive generations. The parameter t is the number of generations. If the fitness values for current individuals are larger than the ones for last generation, set $t=0$, otherwise $t=t+1$. (2) shows that mutation rate will increase with evolutionary generations as soon as evolution gets into stagnancy. If the number of generations t has reached the threshold determined by $p_m \geqslant 1$, which implies the population has trapped into premature convergence, then only mutation will be carried out until no prematurity occurs or the number of mutation reaches a specified number.

2.4.2 Magnitude of mutation

Refer to standard BP algorithm, EANN are optimized by adjusting weights whose increments Δw_{ij} depend on individual fitness. An algorithm similar to simulated annealing algorithm to compute the increment is proposed, in which the ith individual temperature wdt_i is introduced and defined as[6,7]

$$\max f = 1/(1+e_d) \tag{3}$$

$$wdt_i = 1 - f_i/\max f \tag{4}$$

where f_i is the fitness of ith individual, and

$$f_i = 1/(e_i + 1) \tag{5}$$

e_d denotes the desired square error, e_i denotes the actual square error of ith individual, $\max f$ denotes the biggest individual fitness value corresponding to e_d.

Then the weights can be calculated[7] as follows

$$\Delta w_{ij} = (1 \mp w_{ij}) \times (1 - r^{wdt_i^s}) \tag{6}$$

$$w_{ij} = w_{ij} \pm \Delta w_{ij} \tag{7}$$

where w_{ij} is the jth weight within ith individual, Δw_{ij} is the increment of the weight w_{ij}, r is a random number in interval (0, 1). s is a parameter within the range [2, 5][7], which plays the role of tuning local search region. '$\pm$' indicates increment signal chosen randomly with identical probability.

2.5 Evolution procedure

For the purpose of improving EANN convergence rate and prediction precision, the neural network model is divided into two sub-models which can predict removal rate and tool wear ratio respectively. Training procedure for the two sub-models is listed below:

Step 1. $k \leftarrow 0$. Randomly generate an initial population of P individuals ($P=100$).

Each individual corresponds to an ANN with 5 inputs, 1 output and h hidden neurons, where h is initialized in interval (0, 50). The number of hidden nodes and the initial connection weights for each individual are generated at random. Initialize all the hidden layer weights using a series of random values of a closed interval $[-1, 1]$.

Step 2. $i \leftarrow 0$.

Step 3. Feed training patterns (i. e. samples) to the ith ANN in sequence. Use sigmoid activation function for hidden layer and output layer. Set the number of training patterns to be M. Let m change in range $[0, M-1]$. Compute the output square error e_i of ith individual using the following equation

$$e_i = \frac{1}{M}\sum_{m=0}^{M-1}(y_i(m) - d(m))^2 \tag{8}$$

where $y_i(m)$ and $d(m)$ denote respectively actual output and target output for ith ANN model and mth training pattern.

Step 4. $i \leftarrow i+1$. If $i<P$, return to Step 3; otherwise go to next step.

Step 5. Order individuals based on their fitness values, and then perform reproduction according to the method in Section 2. 2. Set reproduction probability to be 0. 95.

Step 6. Recombine two individuals selected randomly by the crossover operation in Section 2. 3.

Step 7. The first mutation operation will cause greater behavioral changes on individuals than the second one. Therefore less mutation probability is used for the first one (set to be 0. 01). For the second mutation operation, mutation rates and magnitude of mutation are calculated according to the method in Section 2. 4, and set $p_{m0}=0.001$, $C=0.005$. If the number of generations t has reached the threshold, then only mutation will be carried out until the number of mutation operation reaches the number of connection weights for the individual at least.

Step 8. Select the best individual from the population. Calculate square error according to (8). If the error is less than desired error e_d, go to the next step, or else $k \leftarrow k+1$ and return to Step 2.

Step 9. Preserve the best individual mapping prediction network which can be used to calculate removal rate or tool wear ratio.

3 Tool wear prediction

There are very large number of factors influencing EDMM process, such as peak-current, pulse duration, duty cycle, reference voltage, free voltage, tool rotating speed, tool discharge area, feed speed, electrode polarity, electrode material, workpiece material, dielectric fluid, and so forth. By theoretical analysis and experiments, five main parameters, namely peak-current, pulse duration, duty cycle, tool discharge area and feed speed, are chosen as input parameters of the proposed EANN. The other parameters are set below: tool rotating speed—300r/min, free voltage—80V, electrode material—copper, workpiece material—No. 45 quenched steel, dielectric fluid—kerosene, tool polarity-positive.

The outputs of the EANN are tool wear ratio and material removal rate. A three-layer network is considered in our experiment, which can distinguish arbitrarily complex decision regions[8]. The above improved evolutionary algorithm was used to optimize the number of hidden nodes and connection weights. Table 1 shows some prediction results of the proposed EANN.

Table 1 Experimental results

Machining parameters					Target results		Prediction results	
Pulse duration (μs)	Tool discharge area(mm^2)	Duty cycle	Peak- current (A)	Feed speed (mm/min)	Material removal rate (mm^3/min)	Tool wear ratio(%)	Material removal rate (mm^3/min)	Tool wear ratio(%)
40	10	1	5	30	5.21	5.2	5.68	5.4
80	20	2	10	60	12.26	6	11.44	5.71
240	30	3	15	90	17.71	3.8	19.92	3.36
400	40	4	20	120	23.14	2.3	20.83	2.36
240	10	2	5	90	7.08	1.8	7.93	1.98
40	30	4	10	30	10.43	10.8	9.68	11.26
80	40	3	15	120	18.86	9.0	18.67	9.12
400	20	1	20	60	22.85	2.2	24.02	2.24
400	10	4	5	90	7.17	1.0	7.33	0.86
240	30	3	10	60	13.95	2.8	13.09	2.85
40	20	1	15	120	14.87	12.2	14.68	12.5
80	40	2	20	30	25.57	17.8	25.97	18.56

It is clear that the maximum prediction error rate is 14% and the minimum is 1.0% calculated from Table 1, which are less than the errors 17.8% and 2.1% obtained from BP network respectively. Results indicate that the improved EANN is efficient and the experiment results are better than the standard BP neural networks'.

4 Conclusions

An improved evolutionary method based on real-number encoding is presented to optimize neural network's connection weights and its topology. The algorithm could adaptively adjust mutation rate and magnitude of mutation according to individual fitness. Mutation rate will increase with evolving generations as soon as evolution gets into stagnancy. In addition, because architectural mutation causes greater behavioral change, weights adjustment is always considered first before architectural mutation. Experiments show that the EANN is feasible to predict EDMM process and the prediction results are better than that of the standard BP neural networks. The EANN lays a foundation for tool compensation on-line in EDMM.

The maximum prediction error rate is 14%, which is mainly caused by machining stability. Better results could be expected by improving the performance of pulse power and the performance of the experimental device developed by the authors, and by perfecting training patterns.

References

1 Yao Xin. Evolving artificial neural networks. *Proceedings of the IEEE*, 1999, **87**(9):1423~1447

2 Porto V W, Fogel D B, Fogel L J. Alternative neural network training methods. *IEEE Expert*, 1998, **10**(3):16~22

3 Fogel D B. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 2nd Ed. Piscataway, NY: IEEE Press, 2000

4 Kaneko T, Tsuchiya M, Kazama A. Improvement of 3-D NC contouring EDM using cylindrical electrodes optical measurement of deformation and machining of free Curves. In: Proceedings of the 10th International Symposium on Electro-Machining (ISEM-X). Germany: Magdeburg Technical University, 1992. 364~367

5 Kaneto T, Tsuchiya M. Three-dimensionally numerically controlled contouring by electrical discharge machining with compensation for the deformation of cylindrical tool electrode. *Precision Engineering*, 1988, **10**(3):157~163

6 Pan Zheng-Jun, Kang Li-Shan, Chen Ping. Evolutionary Computation. Beijing: Tsinghua University Press, 2001. 1~47 (in Chinese)

7 Michalewicz Z. Generic Algorithms+Data Structure=Evolution Programs. Berlin: Springer-Verlag, 1992

8 He Yu-Bing, LI Xin-Zhong. Neural Network Control Technology and Its Application. Beijing: Science Press, 2001. 28~29(in Chinese)

8 YIN Guo-Fu, LIU Dong, SHEN Xin-Wei. Research on control method of bittern density based on fuzzy neural network in salt solution mining. *Acta Automatica Sinica*, 2000, **26**(4):519～522(in Chinese)

LI Xiang-Long Ph. D. candidate in the School of Manufacturing Science and Engineering at Sichuan University. His research interests include intelligent control and intelligent CAD/CAM.

YIN Guo-Fu Professor in the School of Manufacturing Science and Engineering at Sichuan University. His research interests include intelligent control, and intelligent CAD/CAM.

LUO Hong-Bo Associate professor in the School of Manufacturing Science and Engineering at Sichuan University. His research interests include numerical control and CAM.

进化神经网络在机床工具损耗预测中的应用

李翔龙　殷国富　罗红波

（四川大学制造科学与工程学院　成都　610065）

（E-mail：lxlnc@163. com）

摘　要　设计了一个基于实数编码的改进进化算法优化神经网络的连接权和网络结构.该算法可以根据种群停止进化代数自适应调节变异率、根据个体适应度调节变异量.加工实验表明采用进化神经网络可以较准确预测出电火花铣削加工工具损耗,所提出的进化算法是有效的,预测结果较标准 BP 神经网络高.该预测模型为电火花铣削加工工具在线自动补偿打下基础.

关键词　进化算法，神经网络，工具损耗预测，电火花铣削加工

中图分类号　TP18