Springer

KSME

Journal of Mechanical Science and Technology 27 (5) (2013) 1469~1477
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-013-0327-0

# Application of artificial neural network and optimization algorithms for optimizing surface roughness, tool life and cutting forces in turning operation †

F. Jafarian[1,*], M. Taghipour[2] and H. Amirabadi[1]

[1]*Department of Mechanical engineering, University of Birjand, Birjand, Iran*
[2]*Faculty of Electrical and computer engineering, University of Birjand, Birjand, Iran*

(Manuscript Received June 18, 2011; Revised March 26, 2012; Accepted January 12, 2013)

**Abstract**

Our goal is to propose a useful and effective method to determine optimal machining parameters in order to minimize surface roughness, resultant cutting forces and maximize tool life in the turning process. At first, three separate neural networks were used to estimate outputs of the process by varying input machining parameters. Then, these networks were used as optimization objective functions. Moreover, the proposed algorithm, namely, GA and PSO were utilized to optimize each of the outputs, while the other outputs would also be kept in the suitable range. The obtained results showed that by using trained neural networks with genetic algorithms as optimization objective functions, a powerful model would be obtained with high accuracy to analyze the effect of each parameter on the output(s) and optimally estimate machining conditions to reach minimum machining outputs.

*Keywords*: Neural network (ANN); Surface roughness; Genetic algorithm (GA); Cutting forces; Particle swarm optimization (PSO); Tool life

## 1. Introduction

In recent years, industrial producers and manufacturers have attempted to increase the efficiency, performance and accuracy of machining operations. These attempts involve some activities such as increasing production rate, decreasing operation costs and enhancing the quality of production. These activities can be affected by a number of factors, such as machine tool conditions, tool geometry, workpiece material and also machining parameters [1]. Among these, machining parameters such as cutting speed, depth of cut and feed rate play a significant role in machining quality as parameters that are controlled by the user. Therefore, suitable selection of these parameters is necessary to reach optimal machining conditions in order to improve production efficiency.

So far, several researchers have performed experimental investigations about the machining operations and evaluated the effect of machining parameters on the outputs of the process [2, 3]. In fact, they have attempted to find suitable machining parameters in order to determine optimal conditions of the process. But, implementing numerous experimental tests for finding mentioned conditions is very time consuming and costly. To solve this problem, some of researchers have attempted to model the machining processes by various methods, such as statistic, intelligent and analytical methods [4, 5]. Among them, predictive models are capable of estimating complex relationships between machining input parameters and corresponding outputs [6, 7]. Therefore, these methods significantly reduce the required experimental tests for prediction outputs of the process. Artificial neural network (ANN), fuzzy logic (FL) and regression models are some instances of these methods [8, 9]. Among them, ANN is one of the most well known methods which has utilized widely for presenting a predictive model of machining processes. In this regard, several researchers have succeeded in estimating various outputs of the machining processes at specified range of input parameters by using artificial neural networks [6, 10]. Hence, this method can be used as an important and useful tool for reducing time and cost in industrial projects.

Owing to the fact that, in industrial applications, producers and engineers are interested in determining optimal input machining parameters in order to minimize or maximize output(s) of the process, using predictive model can not only eliminate requirements of industries. To solve this problem, in addition to the predictive methods, some researchers have utilized optimization techniques in their investigations [8, 11]. Evolutionary algorithms such as genetic algorithm (GA) and particle swarm optimization (PSO) are among the most well known techniques which are properly used for optimization of machining processes [11]. Of course, these techniques are applied to the problems with one objective function of optimi-

*Corresponding author. Tel.: +98 09357578764, Fax.: +98 5612502160
E-mail address: farshid.jafarian@ymail.com
† Recommended by Associate Editor Song Min Yoo

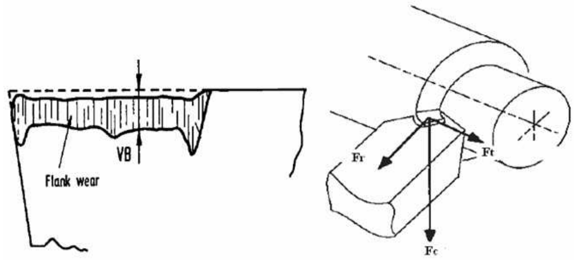
zation. In this regard, some researchers have determined suitable machining parameters for optimizing one output of the process by employing integration of predictive and optimizing methods [8, 12]. It should be noted that although in some cases, optimizing one specified output of the process is requirement of industries, in several cases machining operation is evaluated from several aspects and not limited to one purpose. Therefore, some researchers are employing multi objective optimization (MOO) for simultaneous optimization of two or more outputs of the process [4, 13].

As mentioned above, some advantages and requirements for using the aforementioned methods have been explained. But, implementing predictive and optimization methods in machining applications involves some limitations and deficiencies. More in detail, as far as predictive models (such as ANNs) are concerned, numerous data samples are necessary for better training and increasing performance of the neural networks. Owing to the fact that there are few data samples at the machining operation because of high expense of experimental tests, the procedure of network training plays an important role in accuracy of the predicted results. So far, several researchers have obtained satisfactory results for estimating output(s) of the machining operation by using ANNs which are trained by conventional methods (such as back propagation) [14]. But developing a novel method for training the ANNs which yields more accuracy of predicted results in spite of few data samples is necessary, especially for machining applications.

On the other hand, as far as optimization methods are concerned, although MOO techniques are employed to optimize machining outputs at the same time, a problem arises when the goal involves optimizing one output of the process while the other outputs are maintained in a suitable range. This method (which is not implementable properly by MOO technique) can be applied as a useful tool for obviating some requirements and limitations of process engineers. Therefore, considering a novel and innovative technique for implementing the mentioned method is favorable for engineers and deserves to be developed further.

According to the reasons mentioned above, our main goal is specialized to presentation of a novel intelligent method (including predictive and optimization method) which is more efficient than other conventional methods for machining applications.

Hence, in the first part of the paper, three artificial neural networks were employed to present a predictive model for the turning process. Surface roughness of the machined workpiece, resultant cutting forces and tool wear were predicted by ANNs at the specified range of input parameters (including cutting speed, feed rate and depth of cut). In this regard, a novel method (in addition to the conventional methods) was introduced and implemented for training the ANNs. Based on this, evolutionary algorithms (GA and PSO) were utilized for training the ANNs. In the rest of the paper, functions implemented by ANNs are employed as objective functions of the optimization algorithms. Finally, in addition to optimizing each of the mentioned outputs separately, a new method is implemented for optimizing each of specified outputs while other outputs are kept in the desired range.




(a) (b)

Fig. 1. (a) Views of tool flank wear; (b) View of main cutting forces in turning process.

## 2. Overview of machining characteristics

Surface roughness of machined workpiece is an important factor in evaluating the quality of manufactured products. It has a significant impact on characteristics such as fatigue, corrosion resistance and creep strength. Wear at the tool edge (flank wear) is also considered as an important factor which directly affects the surface roughness of the machined part. Increasing flank wear and reaching critical level leads to reducing the quality of the surface roughness [15]. Therefore, to increase optimal production, it is necessary to predict the flank wear and estimate suitable durability of the tool before reaching the critical state. A schematic view of the flank wear is shown in Fig. 1(a). Machining forces also play a significant role in the development of stresses and temperature in the machined part. Furthermore, increasing the forces may lead to unstable state and development of chatter during the machining process. Since all of these features may affect the machining surface characteristics, cutting forces are also considered as another factor for evaluating performance of the machining process and must be controlled. Fig. 1(b) is a schematic view of the main cutting forces during the turning process. Due to the importance of the mentioned points, researchers have done studies in this regard to determine suitable conditions of the machining processes. As mentioned above, applying efficient intelligent methods for prediction and optimization of machining processes contains several advantages, such as reducing cost and time of the production, and process engineers will be able to determine optimal cutting conditions in order to increase performance of the process. Hence, using intelligent methods is favorable for industrial applications.

On the other hand, several researchers have performed experimental investigations and obtained valuable data samples, but they have not analyzed their data samples properly. Therefore, it seems necessary to perform researches for developing a more suitable analysis of the aforementioned investigations by using several techniques such as novel intelligent methods. Based on this, in the performed study, 27 turning experiments using tungsten carbide with the grade of P-10 for machining of S45C steel have been utilized from the Ref. [16]. Three cutting parameters including cutting speeds in the range of 135-285 m/min, depth of cut in the range of 0.6-1.6 mm and finally feed rate in the range of 0.08-0.32 mm/rev have been investigated. Also, three outputs of the turning process including average surface roughness (measured by 3D-hommelewerk), resultant cutting forces (measured by Kisler5257A) and also tool life (based on the factor of reaching to the average flank wear of the tool $V_B = 0.3$mm or maximum flank wear $V_{B\,(max)} = 0.6$mm) have been used [16].

## 3. Overview of population-based algorithms

### 3.1 Genetic algorithm

The principle of natural evolution is the main idea of evolutionary algorithms (EAs). According to evolution theory, particles of population evolve themselves to obtain more adaptation with their environment. Therefore, the particles that can adapt themselves have a greater chance to survive. Such algorithms are stochastic optimization techniques; in these techniques, information of each generation is transferred to the next generation by chromosomes. Each chromosome consists of some genes, and each of the genes illustrates a special feature or behavior [17].

Genetic algorithm (GA) is one of the most well known evolutionary algorithms. In GA's process, first of all, an initial population is created based on the requirement of problem and after that, the objective function is evaluated in order to achieve the best solution. Off springs are also created from parents in the reproduction step. In this step, some actions happen such as crossover and mutation. Consequently, the best solution is obtained during predetermined iterations.

### 3.2 Particle swarm optimization (PSO) with constraint coefficient

PSO simulates the behavior of bird populations. In PSO, each single solution is a ‘‘bird’’ in the search space and it’s called a ‘‘particle’’. For all of the “p” particles, a fitness value is evaluated by the fitness function to be optimized. The *p* particles are ‘‘flown’’ through the problem space by following the current optimum *p* particles. PSO is initialized with random p particles (solutions) and then searches for optimization by updating generations. In each iteration, each *p* particle is updated by following two ‘‘best’’ values. The first one that is obtained so far by any particle is called "personal best (pbest)". Another ‘‘best’’ value is the best value among all personal bests and all iterations. This best value is a global best which is called "$g_{best}$". After finding the two best values, the particle updates its velocity and positions based on Eqs. (1) and (2) [18]:

$$V_{new} = W \times V_{old} + C_1 \times r \times \left(P_{pb} - X_{cs}\right) + C_2 \times r \times (P_{gb} - X_{cs}) \tag{1}$$

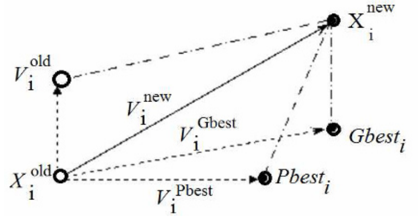



Fig. 2. Typical movement of one particle in solution space.

$$W_{new} = W_{old} + W_{new} \tag{2}$$

where $W$ is the inertial weight, $V_{new}$ is the particle velocity, $X_{cs}$ is the current particle (solution) of each particle, $P_{pb}$ and $P_{gb}$ are $p_{best}$ and $g_{best}$ respectively, $r$ is a random number between (0, 1) and $C_1$, $C_2$ are learning factors. Particle velocities on each dimension are clamped to a maximum velocity $V_{max}$. If the sum of old velocity and impact of particles recorded experiences (see Eq. (1)) would cause the velocity on that dimension to exceed ($V_{max}$ is a parameter specified by the user), the velocity on that dimension will be limited to $V_{max}$. In Fig. 2, typical movement of one particle in solution space is shown.

To improve the performance of the basic PSO one of the most powerful versions is presented in Ref. [19]. In this version of PSO, the updating step of velocity is changed in comparison with basic PSO. To do this, a constant coefficient is utilized instead of inertial weight and learning factors are replaced by two random values [20]. In this approach, the updating step is calculated by the following equations:

$$V_{new} = \chi \times (V_{old} + \varphi_1 \times (P_{pb} - X_{cs}) + \varphi_2 \times (P_{gb} - X_{cs})) \tag{3}$$

$$W_{new} = W_{old} + W_{new} \tag{4}$$

$$\begin{cases} \varphi > 4 \\ \chi = \dfrac{2k}{\varphi - 2 + \sqrt{\varphi^2} - 4\varphi} \end{cases} \tag{5}$$

where $\varphi_1$ and $\varphi_2$ are random numbers uniformly distributed in the range $(0, \varphi_2)$, and $\chi$ is the constant coefficient.

## 4. Overview of artificial neural network (ANN)

ANNs simulate the simplified model of human brain application. Learning ability is the most important characteristic of the human brain. Therefore, artificial intelligence engineers try to build a useful tool or software that can use experiences in features and can give the best decision in special conditions.

In general, ANNs based on the learning algorithm are divided into two major groups: supervised and unsupervised learning. In supervised learning, the input and desired output are available as learning or pattern data and ANN is trained by using them. In unsupervised learning target, outputs are not available and the network can only cluster input data, and

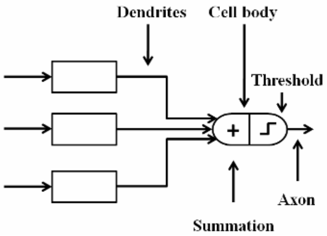



Fig. 3. A simple structure of ANN.

whenever new data is entered, unsupervised ANN can assign it into a corresponding cluster.

### *4.1 Performance of ANN*

As can be seen in Fig. 3 a simple model of an artificial neural network is created by connecting neurons. The relative position of cells in a network (number of neurons, number of layers and all connections between cells) is the network topology. Indeed, a topology is a hardware connection system of neurons that by using corresponding software (namely, mathematical procedure of information flow and weight's calculation) determines the kind of neural network application. In this topology, there is one input layer that accepts information; also there are some hidden layers that give information from a previous layer, and finally one output layer that accepts results of calculations and presents output.

Each cell is connected to all cells of the next layer. Self-connection, connecting to previous layer and jumping from one layer are forbidden. This is so-called "feed forward", because information always flows from input to output. About 90-95 percent of ANN's applications use this topology. At first, synapse weights are random values that will be corrected during especial special iterative training method [21].

## 5. Training neural network by genetic algorithm

As mentioned in the previous section, different methods are used to train (or in other words, update weights and biases), which most of them use analytical and mathematical based methods, like back propagation and gradient descent. In this paper, a novel method has been introduced to train ANN [22]. In the proposed method, genetic algorithm has been used to train ANN, as the optimization method. Fig. 4 shows the flowchart of training ANN using GA. Also, in the continuation of this section the proposed method for training ANN is explained in detail.

### *5.1 Preparation of the genetic algorithm*

The genetic algorithm has some parameters which should be set in the initialization step, such as population size, elite count, migration fraction, migration interval, crossover and mutation.

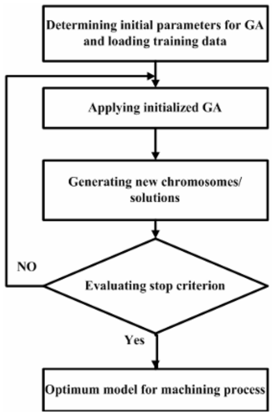



Fig. 4. Flowchart of training ANN with GA or PSO.

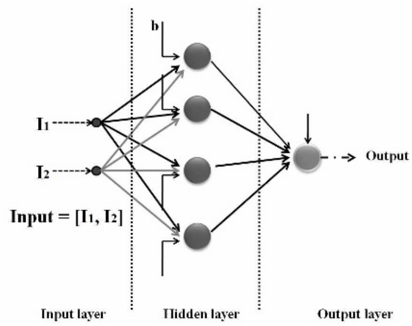



Fig. 5. A simple structure of ANN.

### *5.2 Administration of the neural network structure*

The neural network used in this paper is a multilayer perception neural network in which none of the conventional training methods have been considered. For example, as can be seen in Fig. 5, a two layer neural network, which has two inputs in the input layer, four neurons in the middle layer and one neuron in the output layer has been drawn. In this paper, the weights and biases of the neural network are the variables of the optimization algorithm.

Based on the example mentioned above, the number of variables to be updated can be calculated through Eq. (6).

$$Number\quad of\quad Variable = I \times N_{Lh} + B_{Lh} + N_{Lh} \times N_{Lo} + B_{Lo}\ . \tag{6}$$

In which $I$ is the number of the inputs, $N_{Lh}$ is the number of neurons of the hidden layer, $B_{Lh}$ is the number of biases of the hidden layer, $N_{Lo}$ as the number of neurons in the output layer and $B_{Lo}$ as the number of biases of the output layer. The number of dimensions of the optimization problem in training the neural network is therefore equal to the variables calculated through the above equation.

At first, input data and target matrix are loaded. Then, the matrix of weights that expresses the initial position of chromosomes should be loaded. This matrix is 1 × i, as follows:

$$Weight = [P_1, P_2 .... P_i] \tag{7}$$

where weights are the position of the first chromosome in solution space. In other words, it states the weights of ANN for first chromosome. Then this matrix should be divided into submatrixes which express weights of one layer and its biases, e.g., for the mentioned example, $W_1 = [P_1\text{-}P_8]$ belongs to synaptic weights between input data and first layer and $b_1 = [P_9\text{-}P_{12}]$ belongs to biases of first layer's neurons. After separating these matrixes, the structure of ANN can be constructed, as follows:

$$S_1 = W_1 \times [I] + b_1 \tag{8}$$
$$S_2 = \log sig(S_1) \tag{9}$$
$$Z_1 = W_2 \times S_2 + b_2 \tag{10}$$
$$Z_2 = \log sig(Z_1) \tag{11}$$

where $I$ is input data. $W_1$ and $W_2$ are weights between input and hidden layer, between hidden layer and output layer, respectively. Matrices $b_1$ and $b_2$ are biases related to hidden layer, output layer, respectively.

Finally, MSE value of subtraction of actual output and target output, $Z_2$, is calculated. This value is the fitness value of MLFFNN for GA and the proposed algorithm will change weights until this value becomes a minimum. The fitness function can be found in Eq. (12).

$$Fitness \text{ function } = \frac{1}{\text{n}} \sum_{i=1}^{n} (\text{target}_i - actual_i)^2 \tag{12}$$

where targets are true values, actuals are prediction values and "n" is the number of training data.

### *5.3 Optimization of weights and biases by GA*

In this method the neural network is used as an objective function in optimization. Each of the chromosomes of the genetic algorithm represents a set of weights and biases of one neural network, and each of them may separately be an answer to the optimization problem. Depending on the adjustments made, these chromosomes transform into new chromosomes, which are evaluated based on the objective function, or the neural network, in other words.

### *5.4 Evaluation of stop criterion*

To stop the optimization process by genetic algorithms a number of criteria have been used. Two of these criteria are the number of repetitions of the algorithm and the amount of error in training the neural network (less than 2%), which have been considered in this paper. The optimization process continues until one of the above conditions is reached. After training the neural network with the help of genetic algorithms, the six remaining data are used as test data to evaluate the accuracy of the trained network.

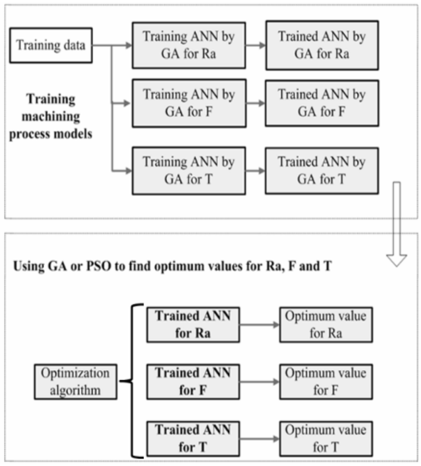



Fig. 6. Flowchart of the proposed method for optimization of machining outputs.

## 6. Proposed method

Neural networks can be used as a suitable model for complex processes. Since machining processes often involve nonlinear and complicated features, correct prediction of the behavior of such processes is of great significance. The proposed method has two major steps: in the first step, trained ANN with GA has been used to present an efficient and optimum model for machining process; in second step, designed models for optimization of the outputs of the machining process are then used.

Fig. 6 shows a block diagram of the whole method proposed for the optimization of the machining process.

### *6.1 Modeling the machining process*

Different methods have been proposed for training neural networks, all of which make an effort to update weights and biases of neural networks in a way that minimum error can be seen in the output of the network. Therefore, such a process can fall into the form of an optimization problem, in which the goal is to minimize the error in the output of the neural network with the most optimal weights and biases. Genetic algorithms can make this task possible as one of the most powerful optimization methods used by the operator for training the neural network. In this step, an optimal machining process model has been created using an artificial neural network which is trained by genetic algorithm and experimental data borrowed from Ref. [16] that are listed in Table 1.

In Table 1, 21data out of the 27 existing data have been randomly used as training data, which are applied in each stage of evaluation so that in the end an accurate model is obtained.

### *6.2 Optimization of the outputs of the turning process*

In this section, the designed models (the trained network)

Table 1. Utilized experimental data which are presented in Ref. [16].

| Vc (m/mm) | f (mm/rev) | d (mm) | Ra (μm) | F (N) | T.L (s) |
|---|---|---|---|---|---|
| 135 | 0.08 | 0.6 | 1.24 | 263.0 | 2645 |
| 135 | 0.20 | 0.6 | 5.34 | 403.0 | 2379 |
| 135 | 0.32 | 0.6 | 9.49 | 550.0 | 2233 |
| 135 | 0.08 | 1.1 | 1.68 | 454.0 | 2604 |
| 135 | 0.20 | 1.1 | 1.92 | 704.0 | 2060 |
| 135 | 0.32 | 1.1 | 4.06 | 889.0 | 1870 |
| 135 | 0.08 | 1.6 | 1.86 | 628.0 | 2563 |
| 135 | 0.20 | 1.6 | 4.12 | 924.0 | 2032 |
| 135 | 0.32 | 1.6 | 9.44 | 1198.0 | 1733 |
| 210 | 0.08 | 0.6 | 2.61 | 212.0 | 1605 |
| 210 | 0.20 | 0.6 | 4.51 | 389.0 | 1198 |
| 210 | 0.32 | 0.6 | 11.05 | 502.0 | 802 |
| 210 | 0.08 | 1.1 | 1.01 | 377.0 | 1350 |
| 210 | 0.20 | 1.1 | 2.75 | 622.0 | 1059 |
| 210 | 0.32 | 1.1 | 7.49 | 853.8 | 734 |
| 210 | 0.08 | 1.6 | 2.64 | 593.0 | 1310 |
| 210 | 0.20 | 1.6 | 6.06 | 952.0 | 1031 |
| 210 | 0.32 | 1.6 | 14.37 | 1169.7 | 602 |
| 285 | 0.08 | 0.6 | 0.56 | 203.0 | 860 |
| 285 | 0.20 | 0.6 | 2.84 | 363.8 | 847 |
| 285 | 0.32 | 0.6 | 9.70 | 464.2 | 216 |
| 285 | 0.08 | 1.1 | 0.91 | 335.0 | 854 |
| 285 | 0.20 | 1.1 | 2.74 | 573.1 | 846 |
| 285 | 0.32 | 1.1 | 6.12 | 812.7 | 212 |
| 285 | 0.08 | 1.6 | 1.25 | 443.0 | 810 |
| 285 | 0.20 | 1.6 | 4.18 | 856.8 | 765 |
| 285 | 0.32 | 1.6 | 10.17 | 1099.3 | 203 |

are used as objective functions for each of the outputs of the machining process in the previous section for the optimization of the outputs of the machining process. GA and PSO optimization algorithm are used for the optimization of the determined output. Machining parameters are determined in a way so that each time one of the outputs is optimized the other outputs are maintained in a suitable range.

## 7. Results and discussion

### *7.1 Creating predictive model of machining process*

Our first aim was presentation of an optimum machining process model. Using experimental results, three artificial neural networks have been trained by GA for average machined surface (*Ra*), resultant forces (*F*) and tool life (*T.L*). Table 2 illustrates the obtained mean error for training data and testing data by using trained ANN with GA; also, in order to confirm the ability of the proposed method, the obtained results with trained ANN by GA have been compared with two conventional ANN: feed forward back propagation neural network (FFBPNN) and cascade forward back propagation neural network (CFBPNN).

According to Eq. (12), minimum value of mean error shows that ANN is trained better. Table 1 shows trained ANN by GA presents better results than FFBPNN and CFBPNN. Therefore, models created by GA and ANN are better choices for the machining process as compared to those by FFBPNN and CFBPNN. Also, Fig. 7 shows the convergence curves of training procedure for F, T.L and Ra. It should be noted that tool

Table 2. Mean error value obtained by trained ANN by GA, FFBPNN and CFBPNN.

| Trained ANN | GA | | FFBPNN | | CFBPNN | |
|---|---|---|---|---|---|---|
| | Mean error (%) | | Mean error (%) | | Mean error (%) | |
| | Train | Test | Train | Test | Train | Test |
| (F) | 2.98 | 4.36 | 3.81 | 4.06 | 11.13 | 5.08 |
| (T.L) | 6.75 | 5.17 | 12.19 | 9.7 | 11.25 | 22.88 |
| (Ra) | 2.49 | 6.72 | 13.96 | 27.05 | 17.61 | 30.14 |

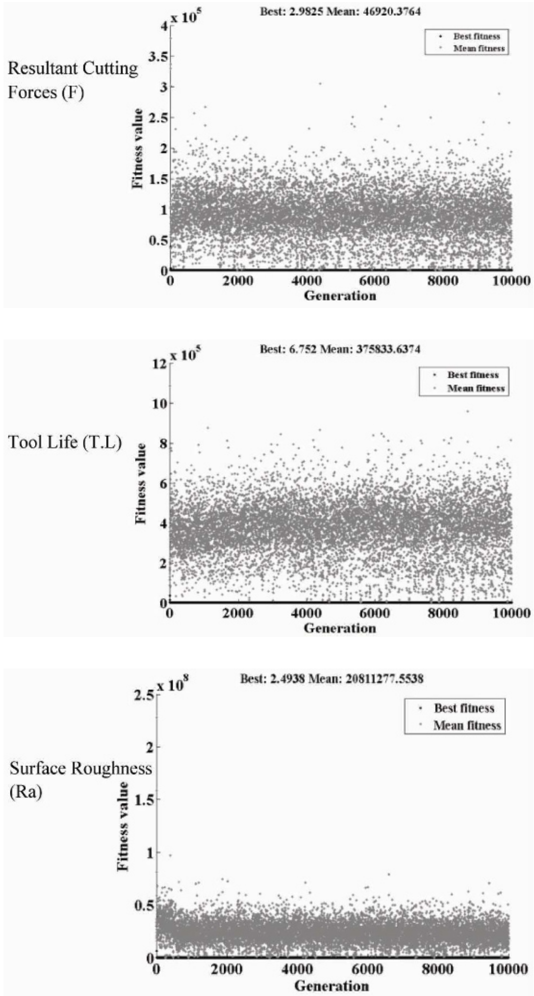



Fig. 7. Convergence curves of training procedure for F, T and Ra.

Table 3. Estimating optimum value of machining process using GA without constraint.

| Machining output | Input machining parameters | | | Optimum value |
|---|---|---|---|---|
| | d (mm) | f (mm/rev) | Vc (m/min) | |
| (Ra) | 1.1601 | 0.089 | 278.68 | 0.447 |
| (F) | 0.60 | 0.08 | 262.670 | 201.282 |
| (T.L) | 0.701 | 0.081 | 135 | 2666.6 |

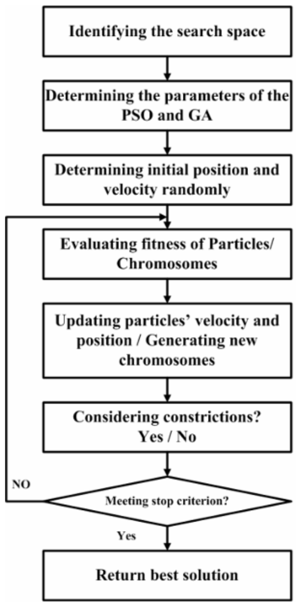



Fig. 8. Flowchart of optimization of machining process using PSO and GA.

life should be maximized; however, in this paper we use 1/T that this term is being minimized.

### 7.2 Optimization of machining outputs

In this section, obtained results have been presented in two forms. In the first form, optimum results have been obtained without any constrains and in the second form, one output is being optimized, whereas some constraints have been considered for other outputs. Fig. 8 shows a flowchart of the optimization process using PSO and GA.

#### 7.2.1 Optimum outputs without any constraints

As shown in Fig. 8, PSO and GA have been used in this optimization problem without any consideration. Obtained results using GA and PSO have been presented in Tables 3 and 4, respectively. In addition, convergence curves of the resultant forces using GA and PSO have been shown in Figs. 9 and 10, respectively.

#### 7.2.2 Optimum outputs with constrains

In many cases, obtaining optimum values for one output is not sufficient, and according to machining condition and type of machining device, evaluating and controlling more than

Table 4. Estimating optimum value of machining process using PSO without constraint.

| Machining output | Input machining parameters | | | Optimum value |
|---|---|---|---|---|
| | d (mm) | f (mm/rev) | Vc (m/min) | |
| (Ra) | 1.192 | 0.086 | 268.31 | 0.4539 |
| (F) | 0.60 | 0.080 | 250.16 | 201.7742 |
| (T.L) | 0.64 | 0.085 | 135.03 | 2665.032 |

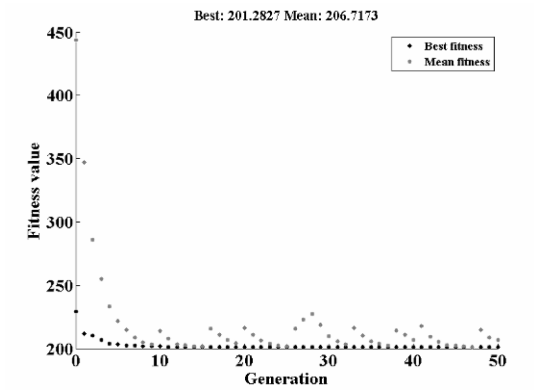



Fig. 9. Convergence curve of cutting forces (F) using GA.

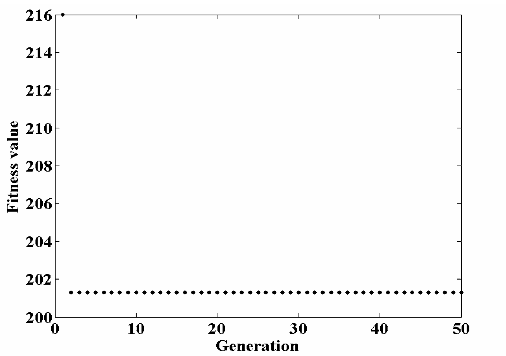



Fig. 10. Convergence curve of cutting forces (F) using PSO.

one output simultaneously may be required. Therefore, in this section, each output has been optimized, whereas other outputs have been considered in the given constraint. For example, when we want to determine optimum value for Ra by using GA, *T* and *F* are considered in the given range as shown in Table 5. Tables 5 and 6 present the obtained optimum values machining process outputs using GA and PSO, respectively. In these tables, limitations are based on minimum possible values for them.

## 8. Conclusion

The main goal of this paper was to develop a new intelligent method for presenting a predictive and optimizing model of the turning process. Hence, surface roughness of the machined workpiece, resultant cutting forces and tool wear were predicted by three artificial neural networks. A novel method was proposed for training the ANNs by using evolutionary algo-

Table 5. Obtained optimum values for machining process outputs using GA (by employing constraint).

| Output | Input parameters | | | Lim1 | Lim2 | Opt value |
|---|---|---|---|---|---|---|
| | Vc m/min | f mm/rev | d mm | | | |
| **Rough-ness** | --------- | ------ | ------ | F (N) | T.L (s) | --------- |
| Ra(μm) | 156.82 | 0.08 | 0.6 | 280 | 2100 | 1.34 |
| | 206.94 | 0.10 | 1.097 | 600 | 1200 | 0.897 |
| | 235.20 | 0.08 | 1.048 | 800 | 900 | 0.63 |
| **Forces** | --------- | ------ | ------ | Ra (μm) | T.L (s) | --------- |
| F(N) | 166.54 | 0.08 | 0.6 | 2200 | 1.5 | 235.437 |
| | 253.58 | 0.08 | 1.034 | 1000 | 0.8 | 332.788 |
| | 262.87 | 0.08 | 1.062 | 700 | 0.6 | 343.739 |
| **Tool life** | --------- | ------ | ------ | Ra (μm) | F (N) | --------- |
| T.L(s) | 236.21 | 0.08 | 1.416 | 0.85 | 450 | 1098.03 |
| | 149.07 | 0.08 | 0.6 | 1.5 | 250 | 2457.12 |
| | 228.69 | 0.08 | 1.048 | 0.9 | 350 | 1194.58 |

Table 6. Obtained optimum values for machining process outputs using PSO (by employing constraint).

| Output | Input parameters | | | Lim1 | Lim2 | Opt value |
|---|---|---|---|---|---|---|
| | Vc m/min | f mm/rev | d mm | | | |
| **Rough-ness** | --------- | ------ | ------ | F (N) | T.L (s) | --------- |
| Ra(μm) | 156.82 | 0.08 | 0.6 | 280 | 2100 | 1.34 |
| | 159.34 | 0.08 | 1.128 | 600 | 1200 | 0.903 |
| | 170.15 | 0.11 | 1.116 | 800 | 900 | 0.67 |
| **Force** | --------- | ------ | ------ | Ra (μm) | T.L (s) | --------- |
| F(N) | 165.94 | 0.08 | 0.6 | 1.5 | 2200 | 235.697 |
| | 251.91 | 0.08 | 1.034 | 0.8 | 1000 | 333.561 |
| | 284.06 | 0.08 | 1.0318 | 0.6 | 700 | 340.600 |
| **Tool life** | --------- | ------ | ------ | Ra (μm) | F (N) | --------- |
| T.L(s) | 235 | 0.08 | 1.0307 | 0.85 | 450 | 1136.27 |
| | 149.07 | 0.08 | 0.6 | 1.5 | 250 | 2457.18 |
| | 235.77 | 0.08 | 1.031 | 0.9 | 350 | 1129.13 |

rithms. Comparison between predicted results of the mentioned method by other NNs which were trained by conventional methods (FFBPNN and CFBPNN which use back propagation method) showed that the proposed method is more suitable than other implemented methods and can be properly utilized for predicting outputs of the turning process (see Table 2). Also, functions implemented by ANNs were employed as objective functions of evolutionary algorithms (GA and PSO) to determine suitable input parameters (including cutting speed, depth of cut and feed rate) for optimizing outputs of the process. In this regard, in addition to optimizing each of the aforementioned outputs separately (see Tables 3 and 4), a new method was implemented for optimizing one specified output, while other outputs were kept in the desired range (see Tables 5 and 6).

Finally, we hope that the proposed novel intelligent method in this research can be employed by other branches of manufacturing processes instead of other conventional methods.

## Nomenclature

| | |
|---|---|
| *Ra* | : Surface roughness |
| *T.L* | : Tool life |
| *F* | : Resultants cutting forces |
| *Vc* | : Cutting speed |
| *f* | : Feed rate |
| *d* | : Depth of cut |
| *PSO* | : Particle swarm optimization |
| *GA* | : Genetic algorithm |
| *ANN* | : Artificial neural network |
| *Lim* | : Limitation |
| *Opt* | : Optimum |
| *FFBPNN* | : Feed forward back propagation neural network |
| *CFBPNN* | : Cascade forward back propagation neural network |

## References

[1] T. Childs, K. Maekawa, T. Obikawa and Y. Yamane, *metal cutting theory and application*, New York, USA (2000).

[2] A. Javidi, U. Riegger and W. Eichlseder, The effect of machining on the surface integrity and fatigue life, *International journal of Fatigue*, 30 (2008) 2050-2055.

[3] D. I. Lalwani, N. K. Mehta and P. K. Jain, Experimental investigations of cutting parameters influence on cutting forces and surface roughness in finish hard turning of MDN250 steel, *Journal of Materials Processing Technology*, 206 (2008) 167-179.

[4] I. Mukherjee and P. K. Ray, A review optimizations techniques in metal cutting processes, *Computers & Industrial Engineering*, 50 (2006) 15-34.

[5] C. X. J. Feng and X. Wang, Development of empirical models for surface roughness prediction in finish turning, *international journal of Advanced Manufacturing Technology*, 20 (2002) 348-356.

[6] A. M. Zain, H. Haron and S. Sharif, Prediction of surface roughness in the end milling machining using artificial neural network, *journal of Expert Systems with Applications*, 37 (2010) 1755-1768.

[7] V. S. Sharma, S. Dhiman, R. Sehgal and S. K. Sharma, Estimation of cutting forces and surface roughness for hard turning using neural networks, *Journal of Intelligent Manufacturing*, 19 (2008) 473-483.

[8] A. M. Zain, H. Haron and S. Sharif, Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process, *Expert Systems with Applications*, 37 (2010) 4650-4659.

[9] Y. Jiao, S. Lei, Z. J. Pei and E. S. Lee, Fuzzy adaptive net-

works in machining process modeling: surface roughness prediction for turning operations, *International Journal of Machine Tools & Manufacture*, 44 (2004) 1643-1651.
[10] T. Ozel, Y. Karpat, L. Figueira and J. P. Davim, Modeling of surface finish and tool flank wear in turning of AISI D2 steel with ceramic wiper inserts, *Journal of Materials Processing Technology*, 189 (2007) 192-198.
[11] I. N. Tansel, B. Ozcelik, W. Y. Bao, P. Chen, D. Rincon, S. Y. Yang and A. Yenilmez, Selection of optimal cutting conditions by using GONNS, *International Journal of Machine Tools & Manufacture*, 46 (2006) 26-35.
[12] I. N. Tansel, S. Gulmez, M. Demetgul and S. Aykut, Taguchi Method-GONNS integration: Complete procedure covering from experimental design to complex optimization, *Expert Systems with Applications*, 38 (2011) 4780-4789.
[13] D. Mandal, S. K. Pal and P. Saha, Modeling of electrical discharge machining process using back propagation neural network and multi-objective optimization using non-dominating sorting genetic algorithm-II, *Journal of Materials Processing Technology*, 186 (2007) 154-162.
[14] T. Ozel and A. Nadgir, Prediction of flank wear by using back propagation neural network modeling when cutting hardened H-13 steel with chamfered and honed CBN tools, *International Journal of Machine Tools & Manufacture*, 42 (2002) 287-297.
[15] A. Ebrahimi and M. M. Moshksar, Evaluation of machinability in turning of micro-alloyed and quenched-tempered steels: Tool wear, statistical analysis, chip morphology, *journal of materials processing technology*, 209 (2009) 910-921.
[16] B. Y. Lee and Y. S. Tarng, Cutting-parameter selection for maximization production rate or minimizing production cost in multistage turning operations, *journal of materials processing technology*, 105 (2000) 61-66.
[17] C. H. Antunes, D. F. Pires, C. Barrico, Á. Gomes and A. G. Martins, A multi-objective evolutionary algorithm for reactive power compensation in distribution networks, *Applied Energy*, 86 (2009) 977-984.
[18] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *In Proc. IEEE International Conference on Neural Networks IV* (1995) 1942-1948.
[19] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transaction on Evolutionary Computation*, 6 (2002) 58-73.
[20] S. H. Zahiri, *Swarm intelligence and fuzzy systems*, Nova Publisher, USA (2010).
[21] Ch. M. Bishop, *Pattern recognition and machine learning*, Springer, USA (2006).
[22] M. Taghipour, A. R. Moradi and M. Yazdani-Asrami, Identification of magnetizing inrush current in power transformers using GSA trained ANN for educational purposes, *IEEE Conference on Open Systems (ICOS)*, Kuala Lumpur, Malaysia (2010) 23-27.

**Farshid Jafarian** received his B.Sc. in Manufacturing Engineering from Azad University of Najafabad, Iran, and his M.Sc. degree in Manufacturing Engineering from University of Birjand, Birjand, Iran. He is pursuing the Ph.D. in Manufacturing Engineering at University of Birjand, Birjand, Iran. His research interests include: traditional and nontraditional machining processes, surface integrity of machined workpiece, FE simulation of machining process and application of intelligent methods at the machining processes.

**Mehran Taghipour-Gorjikolaie** received his B.Sc. in Electrical Engineering from University of Mazandaran, Babol, Iran in 2008, and his M.Sc. degree in Electrical Engineering from University of Birjand, Birjand, Iran in 2011. He is pursuing the Ph.D. in Electrical Engineering at University of Birjand, Birjand, Iran. His research interests include machine learning, pattern recognition and artificial intelligence, design of fault detection systems and modeling and simulation of electrical machines.

**Hossein Amirabadi** received his B.Sc. in Mechanical Engineering (Solid Mechanics) from Ferdowsi University, Mashhad, Iran in 1998, and his M.Sc. in Manufacturing Engineering from IUT University, Isfahan, Iran in 2002. He received his Ph.D. in Manufacturing Engineering from University of Mazandaran, Babol, Iran in 2006. After that he joined the University of Birjand as an Assistant Professor. His research interests include Machine tools, Metal Cutting, Machining Simulation, surface Finishing, Metrology and CAD/CAM.