# Evaluation of ANN-BP and ANN-GA Models Performance in Predicting Mechanical Properties and Machinability of Cast Copper Alloys

*Miloš MADIĆ, Goran RADENKOVIĆ* and *Miroslav RADOVANOVIĆ*

Mašinski Fakultet, Univerzitet u Nišu
(Faculty of Mechanical Engineering,
University of Niš),
Aleksandra Medvedeva 14, 18000 Niš,
**Republic of Serbia**

madic@masfak.ni.ac.rs

Original scientific paper

In this paper artificial neural network (ANN) models were developed to predict the mechanical properties and machinability of Cu–Sn–Pb–Si–Ni–Fe–Zn–Al alloys on the basis of the chemical composition (wt%) of alloying elements. The multi-layer perceptron architecture was used for developing ANN models. Two ANN training approaches, namely, the classical gradient descent back propagation (BP) and genetic algorithm (GA), were applied and statistically compared. The statistical methods of root mean square error (RMSE), absolute fraction of variance ($r^2$) and mean absolute percent error (MAPE) were used for evaluating the performance of the developed ANN models. The results showed that training with GA improved the prediction performance of ANN models. By taking the full potential of GA through fine tuning of the GA parameters, the effectiveness of the approach could be further improved allowing for a wide application in the area of material engineering for the prediction of mechanical properties.

**Ocjena karakteristika ANN-BP i ANN-GA modela u predviđanju mehaničkih svojstava i obradivosti ljevačkih legura bakra**

Izvornoznanstveni članak

U ovom radu su razvijeni modeli umjetnih neuronskih mreža (UNM) za predviđanje mehaničkih svojstava i obradivost Cu-Sn-Pb-Si-Ni-Fe-Zn-Al legura na temelju kemijskog sastava (%) legirajućih elemenata. Za razvoj UNM modela korištena je arhitektura višeslojnog perceptrona. Dva pristupa u treniranju UNM, odnosno gradijentno opadajući algoritam širenja unatrag (BP) i genetski algoritam (GA), su primijenjena i statistički uspoređena. Za ocjenu karakteristika razvijenih modela UNM korištene su statističke metode korijen srednje kvadratne pogreške (RMSE), apsolutna frakcija varijance ($r^2$) i prosječna apsolutna postotna pogreška (MAPE). Dobiveni rezultati pokazuju poboljšanje karakteristika predviđanja UNM modela primjenom GA. Koristeći u potpunosti potencijal GA finim podešavanjem GA parametara, učinkovitost pristupa se može dodatno poboljšati što omogućuje široku primjenu u području inženjerstva materijala za predviđanje mehaničkih svojstava.

## 1. Introduction

The investigation the effects of alloying elements on mechanical properties of metals has been the subject of considerable research. The prediction of mechanical properties of engineering alloys is important for scientists and engineers and it can save not only cost but also time [1]. Due to the complex interconnections among chemical compositions and materials properties, conventional mathematical models have limited range of use [1,2]. Artificial neural networks (ANNs), on the other hand, have been proven to be able to model very complex relationships among a large number of variables. Recent papers [1-4] confirm the validity and effectiveness of using ANNs for developing predictive models in material science and engineering. ANNs were successfully applied for prediction of mechanical properties of magnesium alloys [1], structural steels after normalization process [2], constructional steels after heat treatment [3], Cu–Sn–Pb–Zn–Ni cast alloys [4]. Jančiková et al. [3] and Ozerdem and Kolukisa [4] demonstrated ANNs effectiveness in modeling multi input/output relationships. Dobrzański and Krol [1] showed that the prediction accuracy of the ANN model could be improved by refining pre-processing variables

**Symbols/Oznake**

| | | | | |
|---|---|---|---|---|
| $b$ | - biases of hidden neurons<br>- pragovi skrivenih neurona | $RMSE$ | - root mean square error<br>- korijen srednje kvadratne pogreške | |
| $b_{ok}$ | - bias of output neuron<br>- prag izlaznog neurona | $v$ | - input to hidden neurons weights<br>- težine između ulaznih i skrivenih neurona | |
| $d$ | - the desired output<br>- željeni izlaz | $w$ | - hidden to output neuron weights<br>- težine između skrivenih i izlaznog neurona | |
| $E$ | - global error<br>- ukupna pogreška | $x$ | - input variables<br>- ulazne varijable | |
| $f$ | - transfer function in hidden layer<br>- prijenosna funkcija u skrivenom sloju | $x_{max}$ | - maximal value of variable<br>- maksimalna vrijednost varijable | |
| $g$ | - transfer function in output layer<br>- prijenosna funkcija u izlaznom sloju | $x_{min}$ | - minimal value of variable<br>- minimalna vrijednost varijable | |
| $h$ | - the input value of hidden neuron<br>- vrijednost ulaza skrivenog neurona | $x^n$ | - variable normalized value<br>- normalizirana vrednost varijable | |
| $H$ | - the output value of hidden neuron<br>- vrijednost izlaza skrivenog neurona | $\hat{y}$ | - neural network output<br>- izlaz neuronske mreže | |
| $MAPE$ | - mean absolute percent error<br>- prosječna apsolutna postotna pogreška | | | |
| $N$ | - number of data points<br>- broj podataka | | **Greek letters/Grčka slova** | |
| $N_{tr}$ | - number of training data<br>- broj podataka za učenje | $\alpha$ | - learning coefficient<br>- koeficijent učenja | |
| $R$ | - correlation coefficient<br>- koeficijent korelacije | $\mu$ | - momentum<br>- moment | |
| $r^2$ | - absolute fraction of variance<br>- apsolutna frakcija varijance | | | |

and by using a more reasonable structure of hidden layers.

The mechanical properties of cast alloys mainly depend on the weight fractions of alloying elements, applied heat treatments, microstructures, morphologies of the various phases constituting [4]. Similarly, it has been widely reported that the machinability of a selected material, besides cutting conditions, depends on the chemical composition of the workpiece material. An approach based on genetic programming for the prediction of machinability of steels can be found in [5]. The aim of this paper is to develop ANN models for predicting mechanical properties, namely, tensile strength and yield strength, and machinability of Cu–Sn–Pb–Si–Ni–Fe–Zn–Al cast copper alloys on the basis of the chemical composition (wt%) of alloying elements. Two approaches were employed and statistically evaluated for ANN training. The first one uses a gradient descent with momentum back-propagation (BP) algorithm for ANN training, which is one of the most used training algorithm for ANNs because it is stable and easy to implement. Although this algorithm has proved efficient, its convergence tends to be very slow, and there is a possibility to get trapped in some undesired local minimum [6].

Therefore, in the second approach, real coded genetic algorithm (GA) was used for ANN training.

## 2. Artificial neural networks

Artificial neural networks (ANNs) are massive parallel systems made up of numerous simple processing units called neurons that are linked with weighted connections. ANNs are characterized by their architecture, weight vectors and biases, and activation functions that are used in hidden and output layers. Among various types of ANNs, multi-layer perceptron (MLP) with BP training algorithm, is the most commonly used. The BP MLP is designed to operate as a multilayer fully-connected feed-forward network, with a particular BP training algorithm for supervised learning.

The feed-forward ANN is composed of many interconnected neurons which are grouped into input, hidden, and output layer (Figure 1). The number of input neurons of the ANN is equal to the number of independent variables, while the number of output neurons is equal to the number of functions being approximated by the ANN. The number of hidden layers and the number of neurons in each of them is not defined in advance. The number of hidden neurons can

change during the ANN training until the optimum topology is defined, namely, the one that produces the best performances of ANN.
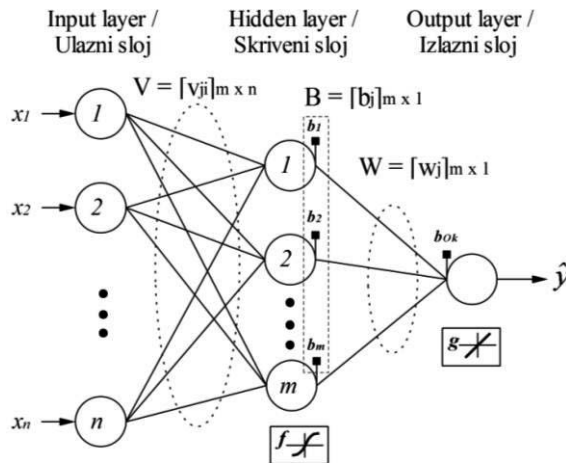


**Figure 1.**   Three-layer feed forward ANN

**Slika 1.**      Troslojna umjetna neuronska mreža

Each neuron in the ANN is interconnected with all neurons in the preceding and following layers. The input neurons are used to introduce the data in the ANN. Through neurons interconnections each input data is processed with weights to be used in the hidden layer. The *j*-th hidden neuron receives an activation signal which is the weighted sum from the neurons in input layer:

$$h_j = \sum_{i=1}^{n} v_{ji} \cdot x_i + b_j \, ; \, j = 1,...,m \,, \qquad (1)$$

where $v_{ji}$ is the weight between *j*-th hidden neuron and *i*-th neuron in input layer, and $b_j$ is the bias (threshold) of the *j*-th hidden neuron. This sum is then passed through an activation (transfer) function (*f*) to produce the neurons output ($H_j$). Transfer function in hidden layer is most commonly sigmoid function whose general form is:

$$H_j = f(h_j) = \frac{1}{1 + e^{-h_j}} \, ; \, j = 1,...,m \,, \qquad (2)$$

Finally, the output neuron receives the following signals from the hidden neurons:

$$\hat{y} = \sum_{j=1}^{m} w_j \cdot H_j + b_{ok} \,, \qquad (3)$$

where $w_j$ is the weight between *j*-th hidden neuron and output neuron, and $b_{ok}$ is the bias of output neuron. These activation signals can be transformed again, using the sigmoid transfer function to give the outputs of the ANN. However, for prediction, it is sufficient to use linear activation function (*g*) for output neuron. The ANN output is then as in equation (3) which are predicted values of the ANN.

Once the ANN architecture is developed, the ANN must be trained in order to learn the relationships between inputs and output(s). Essentially, training is the process of determining weights of connections and bias adjoined to every neuron which are initially assigned to random continuous (real) values. During the process of training the aim is to reduce global error given by:

$$E = \frac{1}{2} \cdot (d_i - \hat{y}_i)^2 \,, \qquad (4)$$

where $d_i$ is *i*-th desired output (target). To minimize this error, an iterative error reduction of the gradient-descent method with momentum is often used because it is stable and easy to implement. An alternative to classical training algorithms are GA where the problem of determining weights and biases is formulated as an optimization problem.

Prior to training process, all the weights and biases are initialized to small random numbers (e.g., between -0.5 and 0.5 or between -1 and 1). Weights initialization according to Nguyen-Widrow method is also one of the most popular methods.

In order to improve the converge speed of training process, the input and output data is usually normalized (scaled). The normalization range depends largely on the activation functions used. Normalizing to [-1, 1] range for the hyperbolic tangent transfer function and to [0, 1] for the sigmoid transfer function is often applied.

## 3. ANN predictive models for mechanical properties and machinability

### 3.1. Data collection

Chemical compositions, mechanical properties and machinability ratings of cast copper alloys were collected from were collected from the referential literature [8-10]. Cast copper alloys are classified into alloy groups according to chemical composition [9]. In the present study, 63 data were collected for Cu–Sn–Pb–Si–Ni–Fe–Zn–Al cast alloys.

Although the properties of copper materials are mainly determined by the composition of the alloy, categorizing materials machinability according to their composition is unsuitable, because alloys in the same alloy group often exhibit different machinability properties [10]. In order to provide practitioners with a basic overview of the machinability of the copper alloys, a machinability rating was proposed in the literature [8, 10]. The machinability rating of copper alloys goes from 20 (hard to machine) to 100 (excellent machinability), with step 10. According to [10], copper and copper alloys are classified into three main machinability groups:

- Machinability group I, with 70 < machinability rating < 100,
- Machinability group II, with 40 < machinability rating < 60,
- Machinability group III, with 20 < machinability rating < 30.

## 3.2. ANN training using BP

The ANN models were aimed to predict the mechanical properties (yield and tensile strength) and machinability. To this aim, three ANN models were developed:

- **Model 1** – which relates Cu–Sn–Pb–Si–Ni–Fe–Zn–Al (wt%) content and tensile strength,
- **Model 2** – which relates Cu–Sn–Pb–Si–Ni–Fe–Zn–Al (wt%) content and yield strength,
- **Model 3** – which relates Cu–Sn–Pb–Si–Ni–Fe–Zn–Al (wt%) content and machinability rating.

In developing ANN models, the available data set is divided into two sets, one to be used for training of the ANN model, and the remaining to be used to evaluate the generalization capability of the ANN model. Therefore, 51 data patterns were used for ANN model training, and 12 data patterns for ANN model testing. The selection of training and testing data was done by random method.

The upper limit of number of hidden neurons was determined knowing that the number of weights does not exceed the number of data for training. Though the ANN can still be trained, the case is mathematically undetermined [11]. Considering the available data for training, the number of hidden neurons was set to four.

Linear transfer function and hyperbolic tangent transfer function were used in the output and hidden layer, respectively. The entire data was normalized between 0.1 and 0.9 using the following equation:

$$x_i^n = 0.8 \cdot \frac{x_i - x_{min}}{x_{max} - x_{min}} + 0.1; i = 1,...,n, \qquad (5)$$

where $x_i^n$ is the normalized value for the variable, and $x_{min}$ and $x_{max}$ are the minimum and maximum of each variable $x_i$. Normalizing to this range also allows the representation of an alloy that does not contain a given chemical element.

Prior to ANN training, the initial values of weights were set according to Nguyen-Widrow method. The MATLAB software package was used for training and testing ANN models. Gradient descent with momentum BP algorithm was used for ANN training with a zero as a target error value. In order to ensure stable training, small value of learning coefficient (α=0.01) [7] was used in combination with momentum value of μ=0.9. The ANN training was stopped when no further improvement in ANN performance was achieved and by considering the well known bias-variance trade-off in model development [12].

## 3.3. ANN training using GA

GAs are probabilistic heuristic algorithms based on biological genetics and natural selection which is primarily Darwin's theory of survival of the fittest [13]. The solution of an optimization problem by GAs starts with a set of chromosomes (strings, individuals), encoding a potential solution in a given problem (search) space. GAs are well suited techniques that can find the global optimum solution by searching the space with a high probability [14]. By considering several points in the search space simultaneously, GA has a reduced chance of converging to local optimum. Further details of GA can be found elsewhere [13, 14].

The GA training of ANN can be formulated as an optimization problem aimed at determining weights ($V$, $W$) and biases ($B$, $b_{ok}$). Since the 8-4-1 ANN architecture for all ANN models was used (m=8, n=4, Figure 1), the optimization problem was reduced to determining 41 ANN parameter values, that is:

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{18} \\ \vdots & \vdots & & \vdots \\ v_{41} & v_{42} & \cdots & v_{48} \end{bmatrix};$$
$$W = [w_1 \cdots w_4]; \qquad (6)$$
$$B = [b_1 \cdots b_4]; b_{ok};$$

To determine the unknown values in equation (6), the code for the 8-4-1 ANN model was created using MATLAB and the fitness (objective) function to be minimized was formulated as:

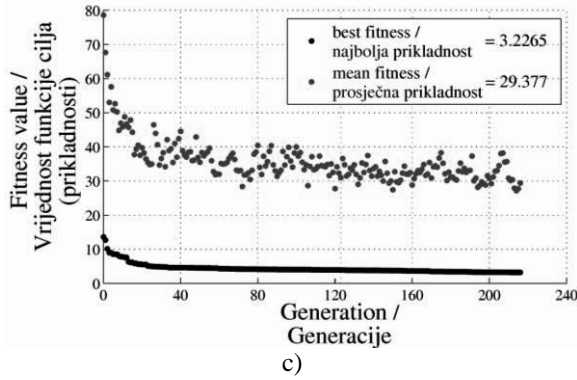$$E = \sum_{i=1}^{N_{tr}} |d_i - \hat{y}_i|, \qquad (7)$$

where $N_{tr}$ is the number of training data. Using the MATLAB optimization toolbox and by taking the default parameter settings for GA operators, the weights and biases for all three ANN models were determined. Figure 2 shows the plot functions of the best fitness obtained for the ANN models.
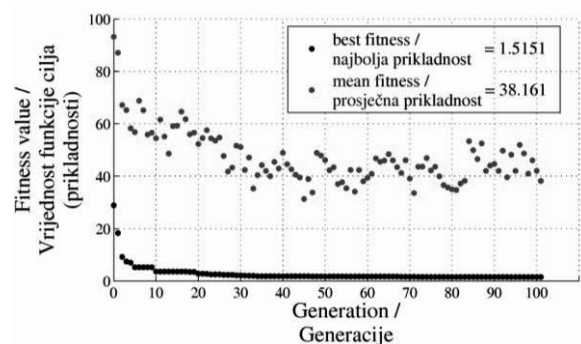


a)
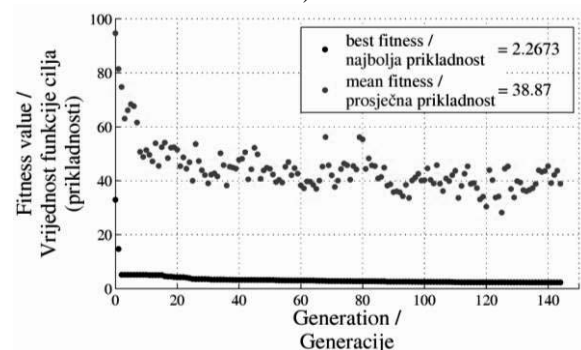


b)

**Figure 2.**    Objective function values with the number of generations for: a) Model 1, b) Model 2, and c) Model 3

**Slika 2.**    Vrijednosti funkcije cilja (prikladnosti) s brojem generacija za: a) Model 1, b) Model 2, i c) Model 3

After an iterative calculus, GA provided the (near) optimal values for weights (*V, W*) and biases (*B, $b_{ok}$*) for all three ANN models. With these values, the ANN models were able to calculate yield strength, tensile strength and machinability of the Cu–Sn–Pb–Si–Ni–Fe–Zn–Al cast copper alloys.

### 3.4. Evaluation of the results

In order to evaluate the performance of both training approaches, statistical methods of root mean square error (*RMSE*), absolute fraction of variance (*$r^2$*) and mean absolute percent error (*MAPE*) were used on both training and testing data. These values are mathematically defined by the following equation:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|d_i - \hat{y}_i\right|^2} \ , \qquad (8)$$

$$r^2 = 1 - \left(\frac{\sum_{i=1}^{N}(d_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(\hat{y}_i)^2}\right), \qquad (9)$$

$$MAPE = \frac{1}{N}\cdot\sum_{i=1}^{N}\left|\frac{d_i - \hat{y}_i}{d_i}\right|\cdot 100\% \ , \qquad (10)$$

where *N* is the number of data points.

Table 1 shows the comparison results between ANN-BP and ANN-GA training approaches in terms of these three statistical measures. The statistical *RMSE*, *$r^2$* and *MAPE* values indicate that the predication of the ANN-BP models appears satisfactory. On the other hand, the GA based approach gave better results than the BP method for training the ANN models. The GA based approach was proved able to improve the ANN modeling performance.

The comparisons of the predicted yield strength, tensile strength and machinability against the target values using GA trained ANN model are shown in Figure 3.

**Table 1.**    Evaluation of the ANN models performance

**Tablica 1.** Ocjena karakteristika modela UNM

|  | ANN-BP | | ANN-GA | |
|---|---|---|---|---|
|  | training data / podaci za učenje | testing data / podaci za testiranje | training data / podaci za učenje | testing data / podaci za testiranje |
| **Model 1** | | | | |
| *RMSE* | 40.42 | 46.93 | 25.39 | 30.30 |
| *$r^2$* | 0.987 | 0.984 | 0.995 | 0.993 |
| *MAPE* | 10.44 | 10.15 | 6.43 | 7.04 |
| **Model 2** | | | | |
| *RMSE* | 24.89 | 34.28 | 22.03 | 31.50 |
| *$r^2$* | 0.977 | 0.968 | 0.982 | 0.954 |
| *MAPE* | 12.78 | 13.21 | 9.61 | 12.95 |
| **Model 3** | | | | |
| *RMSE* | 8.41 | 16.23 | 9.46 | 9.25 |
| *$r^2$* | 0.980 | 0.929 | 0.975 | 0.977 |
| *MAPE* | 16.16 | 26.63 | 15.28 | 18.73 |



a)



b)

c)
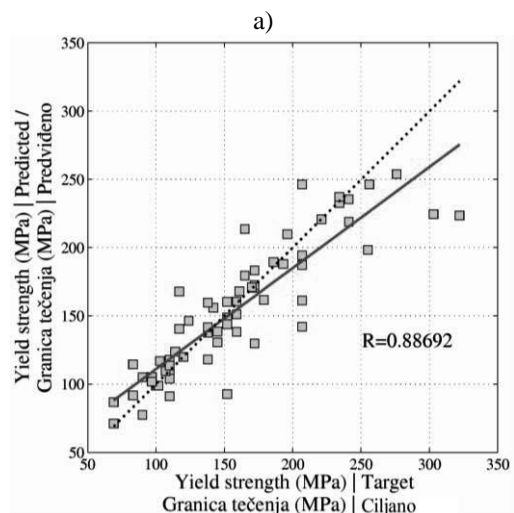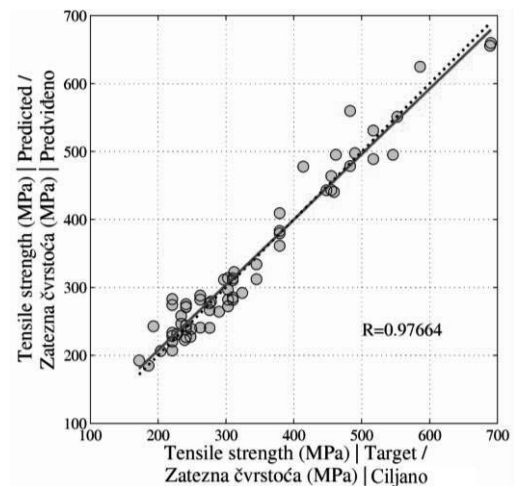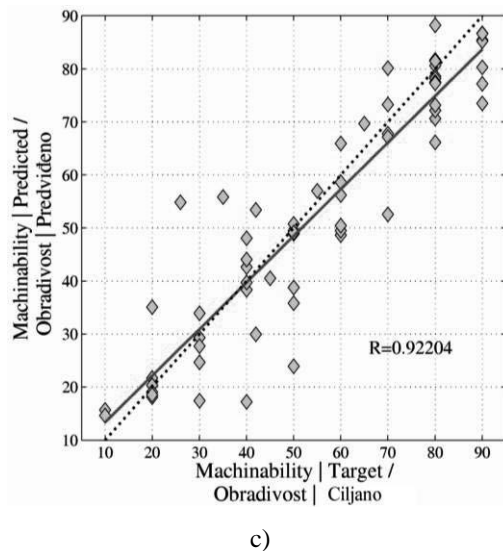
**Figure 3.** Comparison of target and predicted for entire data

**Slika 3.** Usporedba ciljanih i predviđenih vrijednosti na svim podacima

The results from Fig. 3 and Table 1 demonstrate that training ANN using GA is suitable for developing ANN prediction models for mechanical properties and machinability.

## 4. Conclusion

In this paper, ANN approach for predicting the mechanical properties (namely, yield strength and tensile strength) and machinability as a function of the chemical composition of Cu–Sn–Pb–Si–Ni–Fe–Zn–Al cast alloys was presented. The use of gradient descent with momentum BP and genetic algorithm for ANN training was employed. Although BP is by far the most popular method, the results showed that the GA is superior to BP for ANN training. The prediction accuracy of the ANN models was found to be quite satisfactory in terms of *RMSE*, $r^2$, and *MAPE* statistical values. Obtained results confirm the adequacy of using GA for ANN training for the application in the area of material engineering because of its ease-of-use and accurate results. The effectiveness of the approach could be increased by exploiting the full potential of GA through fine tuning of the GA parameters.

## Acknowledgements

## REFERENCES

[1] DOBRZAŃSKI, L.A.; KROL, M.: *Application of neural network for Mg-Al-Zn mechanical properties modelling*, Journal of Achievements in Materials and Manufacturing Engineering, 37 (2009) 2, 549-555.

[2] DOBRZAŃSKI, L.A.; HONYSZ, R: *Application of artificial neural networks in modelling of normalised structural steels mechanical properties*, Journal of Achievements in Materials and Manufacturing Engineering, 32 (2009) 1, 37-45.

[3] JANČIKOVÁ, Z.; ROUBIČEK, V.; JUCHELKOVA, D.: *Application of artificial intelligence methods for prediction of steel mechanical properties*, Metalurgija, 47 (2008) 4, 339-342.

[4] OZERDEM, M.S.; KOLUKISA, S.: *Artificial neural network approach to predict the mechanical properties of Cu–Sn–Pb–Zn–Ni cast alloys*, Materials and Design, 30 (2009) 3, 764-769.

[5] BREZOČNIK, M.; KOVAČIČ, M.; PŠENIČNIK, M.: *Prediction of steel machinability by genetic programming*, Journal of Achievements in Materials and Manufacturing Engineering, 16 (2006) 1-2, 107-113.

[6] GUPTA, J.N.D.; SEXTON, R.S: *Comparing backpropagation with a genetic algorithm for neural network training*, Omega 27 (1999) 6, 679-684.

[7] ŠIMUNOVIĆ, K.; ŠIMUNOVIĆ, G.; ŠARIĆ, T.: *Application of artificial neural networks to multiple criteria inventory classification*, Strojarstvo, 51 (2009) 4, 313-321.

[8] SCHMIDT, R.; SCHMIDT, D.; SAHOO, M.: *Copper and Copper Alloys, ASM Handbook: volume 15*, ASM International, 1988.

[9] TYLER, D.; BLACK, W.: *Introduction to Copper and Copper Alloys, ASM Handbook: volume 2*, ASM International, 1990.

[10] *Recommended Machining Parameters for Copper and Copper Alloys*, Deutsches Kupferinstitut, 2010.

[11] SHA, W.; EDWARDS K. L.; *The use of artificial neural networks in materials science based research*, Materials and design, 28 (2007) 6, 1747–1752.

[12] FENG, C.X.; YU, Z.G.; KUSIAK, A.: *Selection and validation of predictive regression and neural network models based on designed experiments*, IIE Transactions, 38 (2006) 1, 13-23.

[13] RAO, S.S.: *Engineering Optimization: Theory and Practice*, John Wiley & Sons, Inc., New Jersey, 2009.

[14] JURKOVIĆ, Z.; BREZOČNIK, M.; GRIZELJ, B.; MANDIĆ, V.: *Optimization of extrusion process by genetic algorithms and conventional techniques*, Technical Gazette, 16 (2009) 4, 27-33.