



# Neural Networks Approach to Optimization of Steel Alloys Composition

Petia Koprinkova-Hristova, Nikolay Tontchev, Silviya Popova

## ► To cite this version:

Petia Koprinkova-Hristova, Nikolay Tontchev, Silviya Popova. Neural Networks Approach to Optimization of Steel Alloys Composition. 12th Engineering Applications of Neural Networks (EANN 2011) and 7th Artificial Intelligence Applications and Innovations (AIAI), Sep 2011, Corfu, Greece. pp.315-324, 10.1007/978-3-642-23957-1\_36 . hal-01571366

**HAL Id: hal-01571366**

**<https://inria.hal.science/hal-01571366>**

Submitted on 2 Aug 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Neural Networks Approach to Optimization of Steel Alloys Composition

Petia Koprinkova-Hristova<sup>1</sup>, Nikolay Tontchev<sup>2</sup>, Silviya Popova<sup>1</sup>

<sup>1</sup> Institute of System Engineering and Robotics, Bulgarian Academy of Sciences,  
Acad. G. Bonchev str. bl.2, 1113 Sofia, Bulgaria  
{pkoprinkova, popova}@icsr.bas.bg

<sup>2</sup> University of Transport, 158 Geo Milev str., bl.2, 1574 Sofia, Bulgaria  
tontchev@vtu.bg

**Abstract.** The paper presents modeling of steels strength characteristics in dependence from their alloying components quantities using neural networks as nonlinear approximation functions. Further, for optimization purpose the neural network models are used. The gradient descent algorithm based on utility function backpropagation through the models is applied. The approach is aimed at synthesis of steel alloys compositions with improved strength characteristics by solving multi-criteria optimization task. The obtained optimal alloying compositions fall into martensite region of steels. They will be subject of further experimental testing in order to synthesize new steels with desired characteristics.

**Keywords:** steel alloys, neural networks, modeling, optimization.

## 1 Introduction

Production of high strength steel alloys is of big importance for the modern metallurgy. The main aim is to obtain high quality materials reducing quantity of used expensive compounds. Thus the production of steel alloys starts with optimization of number and content of used alloying components that improve their quality at reasonable price. Since the dependences between input and output variables in that case are strongly nonlinear, application of nonlinear modelling and optimization techniques must be applied.

The fact that neural networks are universal approximations of complex nonlinear dependences that apply “black-box” modeling approach [1, 3, 12] is well known. Therefore, they are proper candidates for modeling structure of such MIMO models. Another useful characteristic of neural networks are their training procedures that are in fact optimization algorithms aimed at minimization of error at neural network output with respect to network connection weights [12, 14]. The well-known backpropagation algorithm is procedure for propagating of derivatives of given function of network output backwards to the input [14]. Thus neural network training

procedures offer a common approach to optimization tasks in process optimization and control applications [11, 14, 15].

The application of intelligent modelling approaches such as neural networks is relatively new in this area but there are some examples in the literature concerning titanium alloys [9, 10] as well as several for different steel alloys compositions and different strength characteristics accounting [2, 4, 5].

In our previous works the approach described in [14] was successfully applied to dynamic optimization [6] as well as to optimization of cultural media composition and initial conditions [7] of two kinds of biotechnological process that are other examples of highly nonlinear systems. In [8, 13] we applied that approach to optimization of steel composition having 11 alloying elements aimed at simultaneous maximization of 6 steel strength characteristics; we also showed that neural networks outperform non-linear regression models. However, it appeared that having that much input and output variables do not allow obtaining neural network model able to fit with the same accuracy all output variables and solving of six criteria optimization task showed some controversial results.

In the present paper, we continue our work with increased data set and accounting for lower number of variables. Moreover, we have created separate neural network model for each one of the considered strength characteristic. The optimization criteria were simplified and multi-criteria optimization task from [8, 13] was restricted to two simplified sub-tasks. Comparison between optimized steel compositions with respect to their mechanical characteristics was made. The obtained results will be subject of experimental proof further.

## 2 Experimental Data Set

Here we used data base containing 91 steel alloys with their 8 alloying elements and the corresponding strength characteristics tested after thermal treatment of the steels: Rm – tensile strength; Re – yield strength; A – elongation; Z – reduction of area; HB – Brinell hardness. Tables 1 and 2 below summarize the minimal and maximal values of all considered variables.

**Table 1.** Alloying elements minimal and maximal values.

var. name	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>
element	C	Si	Mn	Ni	S (P)	Cr	Mo	V
min	0.120	0.100	0.300	0	0.015	0.150	0	0
max	0.520	1.400	1.750	4.220	0.035	3.250	1.500	0.150

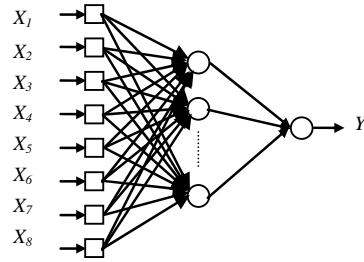
**Table 2.** Steel strength characteristics minimal and maximal values.

var. name	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>
characteristic	Rm	Re	A	Z	HB
min	500	300	7	30	179
max	1670	1375	26	55	300

For the sake of simplicity, names ( $X_i$  for alloying elements and  $Y_i$  for strength characteristics) are given to all considered variables in order to be easier to refer to them further in the text.

### 3 Neural Network Models of Steel Characteristics

First task of the present study was to train neural network models that approximate dependence between amounts of alloying elements in the steel for each of the five considered strength characteristics. For that purpose, the data are scaled in proper interval  $[0, 1]$ . The used neural network structure (presented on Figure 1) is multi-layered without feedback connections since the modeled dependences are static. The neurons transfer function is log sigmoid. Training procedure is resilient backpropagation. Different in number of layers and hidden neurons neural network structures were tested and the better one is chosen. Based on previous investigations [8, 13] the 8:40:1 structure was used here.



**Fig. 1.** Neural network models structure

We have relatively small database and we do not have guarantee that it includes all possible combinations between alloying elements. That is why for assessment of the generalization error of neural network models we apply k-fold cross-validation. In our case we split database into  $k=18$  data subsets each containing five data samples. Next, we train 18 neural network models for each strength characteristic using all except one data subset. The left subset is further used for generalization error assessment.

The obtained training and testing mean square errors (MSE) for all five neural network models are given in Table 3 below. Since for some of the strength characteristics there were no data for all alloying element combinations ( $Y_4$  and  $Y_5$ ) smaller number of subsets was created for them (17 and 16).

For further optimization purpose, we choose to use the models with smallest testing errors as follows: models number one for  $Y_1$  and  $Y_2$ , model number 13 for  $Y_3$ , model number 9 for  $Y_4$ , and model number 5 for  $Y_5$ .

**Table 3.** Training and testing errors for all NN models.

No	Y <sub>1</sub> train MSE	Y <sub>1</sub> test MSE	Y <sub>2</sub> train MSE	Y <sub>2</sub> test MSE	Y <sub>3</sub> train MSE	Y <sub>3</sub> test MSE	Y <sub>4</sub> train MSE	Y <sub>4</sub> test MSE	Y <sub>5</sub> train MSE	Y <sub>5</sub> test MSE
1	0.008	0.055	0.033	0.060	0.004	0.083	0.022	0.115	0.003	0.126
2	0.007	0.445	0.015	0.224	0.006	0.143	0.007	0.200	0.006	0.197
3	0.013	0.099	0.012	0.361	0.004	0.108	0.016	0.299	0.005	0.063
4	0.013	0.064	0.010	0.112	0.009	0.079	0.012	0.137	0.006	0.099
5	0.019	0.171	0.019	0.125	0.005	0.065	0.008	0.166	0.003	0.050
6	0.010	0.170	0.015	0.217	0.006	0.088	0.009	0.115	0.005	0.135
7	0.012	0.130	0.010	0.086	0.006	0.112	0.006	0.160	0.004	0.063
8	0.012	0.104	0.017	0.083	0.005	0.162	0.005	0.382	0.004	0.075
9	0.014	0.131	0.010	0.114	0.007	0.058	0.008	0.091	0.005	0.101
10	0.010	0.087	0.012	0.222	0.007	0.135	0.025	0.213	0.005	0.085
11	0.023	0.127	0.018	0.156	0.009	0.112	0.008	0.168	0.004	0.065
12	0.013	0.136	0.006	0.085	0.010	0.067	0.013	0.157	0.006	0.140
13	0.013	0.341	0.010	0.397	0.007	0.045	0.010	0.316	0.006	0.124
14	0.009	0.195	0.010	0.159	0.006	0.108	0.019	0.289	0.005	0.089
15	0.008	0.100	0.009	0.199	0.007	0.114	0.011	0.243	0.006	0.107
16	0.010	0.080	0.008	0.124	0.004	0.157	0.009	0.123	0.004	0.118
17	0.007	0.354	0.009	0.342	0.005	0.153	0.008	0.484		
18	0.009	0.091	0.019	0.089	0.007	0.104				

### 3 Optimization Procedure

Since the explored input/output space is multi-dimensional and the modeling function is highly nonlinear in order to find optimal values of input variables with respect to given quality criteria that comprises output variables, there is need to explore whole region of the input space. However, because of big number of possible combinations the exhaustive search on whole variables space will take too much time. Because of this, we applied gradient optimization technique starting from several different points of input variables surface and compare the obtained results.

Figure 2 below presents the optimization procedure scheme adopted from the so-called “backpropagation of utility” method [14]. The optimization task here is defined as follows: find values of input vector  $X$  that minimize/maximize the utility function:

$$J = J(X, Y) \quad (1)$$

Here  $Y$  is vector of output variables that are related to the input once by a given function (model)  $F$  as follows:

$$Y = F(X, p) \quad (2)$$

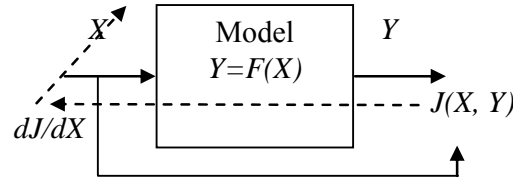
Here,  $p$  is model parameters vector.

The optimization procedure needs calculation of utility function gradients with respect to the optimized variables as follows:

$$dX = \frac{dJ}{dX} = \frac{\partial J}{\partial X} + \frac{\partial J}{\partial Y} * \frac{\partial F}{\partial X} \quad (3)$$

In case when  $J$  does not depend explicitly on  $X$ , the first term in equation (3) is zero and thus gradient depends only on function  $F$ .

The layered neural networks structure offers a convenient way for calculating derivatives from equation (3) because the backpropagation method [12, 14] was developed initially as procedure for error derivatives calculation and their “propagation” from the output to the input of the network. From a more common point of view it is method for a given function derivative calculations with respect to variables of an ordered system of equations [14]. Hence, it could be applied to any optimization problem that can be described in appropriate way. So application of collection of neural network models trained before as model function  $F$  within optimization scheme from Figure 2 allows easy gradients calculation using backpropagation method.



**Fig. 2.** Optimization procedure; dashed lines represent gradients calculation

Next, any gradient iterative optimization procedure could be applied to find optimal values of the variables  $X$  using calculated derivatives as follows:

$$X_i = X_{i-1} \pm \alpha * \Delta X_i \quad (5)$$

Here  $\alpha$  is parameter called learning speed and  $\Delta X_i$  is step-change of  $X$  for the  $i^{\text{th}}$  iteration calculated as follows:

$$\Delta X_i = g(dX_i) \quad (6)$$

Here  $g$  is some function of optimized variable derivative. Usually it is proportional to the derivative  $dX_i$  but it can also depend on the old values of  $\Delta X_i$ .

In present study, we used simple gradient optimization procedure with identity function for  $g$ , i.e.  $\Delta X_i = dX_i$ . The learning speed  $\alpha$  is set to a relatively small value and stopping criteria is very small change of performance function gradient.

## 4 Optimization of Steel Compositions

Two different optimization tasks were solved here. First, one is aimed at maximization of  $Y_2$  that is yield strength  $R_e$ . The second task includes two criteria maximization: maximal yield strength and maximal ratio between  $Y_1$  (tensile strength  $R_m$ ) and  $Y_2$ . In both cases, a restriction was added to obtain low carbon steels:  $X_1 \leq 0.3$ . The two tasks optimization results are presented and compared in the following sections.

### 4.1 First Optimization Task

First, we solved one criteria optimization task:

$$J = Y_2 \rightarrow \max \quad (7)$$

We explore 15 different initial steel compositions. The obtained after optimization new 15 alloying elements “recipes” are shown in Table 4 below.

**Table 4.** Optimized steel compositions from the first task.

No	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
1	0.1562	0.9180	0.4517	0.9215	0.0202	0.8730	0.2450	0.0410
2	0.1562	0.9137	0.4517	0.8663	0.0208	0.8740	0.2757	0.0423
3	0.1562	0.7997	0.7740	1.4991	0.0184	1.1572	0.1943	0.0087
4	0.1562	1.0861	0.8087	1.9735	0.0243	1.7111	0.1733	0.0087
5	0.1584	1.2125	0.8475	2.5041	0.0274	2.0026	0.1815	0.0465
6	0.1603	0.8917	1.3092	4.1876	0.0381	1.8100	0.6971	0.1025
7	0.1562	1.0171	1.4663	4.0238	0.0446	2.1378	1.0380	0.1023
8	0.1562	1.0466	1.4833	4.1469	0.0446	2.2158	1.0497	0.1146
9	0.1562	1.0566	1.4695	4.1876	0.0446	2.2936	1.0486	0.1263
10	0.1562	1.0589	1.4319	4.1876	0.0446	2.2193	0.9355	0.1440
11	0.1562	0.9196	0.4517	0.9113	0.0203	0.8721	0.2501	0.0416
12	0.1870	0.2352	0.6480	2.5490	0.0446	1.4195	0.9603	0.1583
13	0.1566	0.2352	0.7037	3.1023	0.0446	1.7221	0.9820	0.1583
14	0.2714	1.1062	1.0644	2.3624	0.0219	1.0496	0.1572	0.0087
15	0.2673	1.2441	1.1861	2.7854	0.0269	1.2693	0.2301	0.0218

The corresponding five strength characteristics of the new steel compositions and the needed number of iterations in each optimization run are presented in Table 5. The maximal value of criterion (7) was obtained in the case 4 (emphasized in bold in the table). The number of the iterations obviously depends on the starting steel composition position on the search surface; it varies between 160 and 530 iterations. In all cases, the obtained new compositions belong to the martensite steels subset (see Figure 3 below). All the obtained new steels have relatively close values of the optimization criterion. The final decision, which composition suits better for a given purpose, should be taken considering all strength characteristics from Table 5. For some of these characteristics (e.g. 4 and 5) differences are not so big but for characteristics 1 and 3 there are significant differences observed.

**Table 5.** Optimized steel characteristics from the first task, optimization criteria and iterations needed

No	$Y_1$	$J_1=Y_2$	$Y_3$	$Y_4$	$Y_5$	iterations
1	1643.6	1350.4	8.5	54.7	263.8	432
2	1640.2	1349.9	8.8	54.8	260.6	445
3	1407.0	1369.5	7.8	49.5	272.6	261
4	1638.0	<b>1372.6</b>	8.1	49.8	282.7	437
5	1621.3	1359.9	9.7	54.5	290.2	530
6	1122.8	1362.1	13.7	49.9	288.2	276
7	1394.7	1360.2	16.0	42.5	282.3	378
8	1294.9	1360.7	15.0	47.2	280.8	427
9	1227.7	1360.9	14.6	51.2	279.7	362
10	1246.9	1365.3	11.9	53.1	279.8	523
11	1642.9	1350.4	8.6	54.7	263.5	442
12	662.7	1341.0	24.5	55.0	208.9	312
13	841.0	1336.3	25.0	55.0	222.5	200
14	1666.9	1370.3	11.4	51.1	281.6	160
15	1647.5	1370.9	12.2	54.1	287.4	214

## 4.2 Second Optimization Task

Here in addition to the first criterion (7) the second one was added as follows:

$$J_1 = Y_2, \quad J_2 = \frac{Y_1}{Y_2} \quad (8)$$

$$J = a_1 J_1 + a_2 J_2 \rightarrow \max, \quad a_1 = a_2 = 1$$

**Table 6.** Optimized steel compositions from the second task.

No	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
1	0.1562	0.6937	0.5036	0.5673	0.0094	0.4832	0.3116	0.0301
2	0.1577	0.6915	0.5092	0.5807	0.0135	0.4832	0.2698	0.0225
3	0.1562	1.2231	0.7684	1.6164	0.0196	0.4832	0.1572	0.1255
4	0.1562	0.9408	0.7296	1.1316	0.0249	0.4832	0.1602	0.1459
5	0.1562	0.8331	1.7453	0.4564	0.0228	1.9982	0.5520	0.0632
6	0.1562	1.2004	1.7453	0.4564	0.0446	0.4832	1.4948	0.0087
7	0.1562	1.0013	1.6050	2.4654	0.0416	2.0609	1.4905	0.0570
8	0.2964	1.1273	1.4969	3.2602	0.0375	2.6033	1.2407	0.1245
9	0.2980	1.2624	1.6303	3.7373	0.0413	2.9228	1.3716	0.1427
10	0.3000	1.3968	1.7453	4.1874	0.0446	3.2287	1.4948	0.1583
11	0.1562	0.2352	0.4517	0.4569	0.0094	0.4832	0.1650	0.0093
12	0.1562	0.7446	1.4706	1.9032	0.0442	0.4832	0.9263	0.1096
13	0.1562	0.5792	1.1822	1.6692	0.0387	0.4832	1.1307	0.1441
14	0.3000	1.0140	1.1720	2.1100	0.0210	1.0760	0.2940	0.0087
15	0.2997	1.1352	1.3479	2.5514	0.0247	1.3766	0.4125	0.0246



**Table 7.** Optimized steel characteristics from the second task, optimization criteria and iterations needed

No	$Y_1$	$J_1=Y_2$	$Y_3$	$Y_4$	$Y_5$	$J_2=Y_1/Y_2$	iterations
1	1664.2	1233.3	7.6	52.5	261.0	1.3494	865
2	1665.5	1285.3	8.1	52.8	239.3	1.2958	157
3	1659.4	957.9	9.3	54.6	271.2	1.7323	266
4	1456.2	797.5	7.9	54.3	250.9	1.8260	605
5	499.1	300.1	11.4	40.7	185.3	1.6631	3
6	499.1	325.4	22.6	41.5	179.1	1.5338	6
7	735.5	609.3	23.3	52.5	289.3	1.2071	1353
8	1463.1	<b>681.1</b>	11.3	42.8	274.6	<b>2.1481</b>	462
9	1488.5	853.6	14.6	41.0	273.8	1.7438	839
10	1499.8	829.8	20.3	40.8	271.4	1.8074	2
11	1571.6	916.0	8.2	54.1	227.0	1.7157	289
12	1199.5	577.6	13.4	53.2	207.8	2.0767	607
13	760.0	615.1	12.2	54.7	282.1	1.2356	781
14	1635.4	<b>1326.0</b>	15.5	47.7	258.9	<b>1.2333</b>	3
15	1613.4	1308.0	15.6	53.4	267.2	1.2335	2516

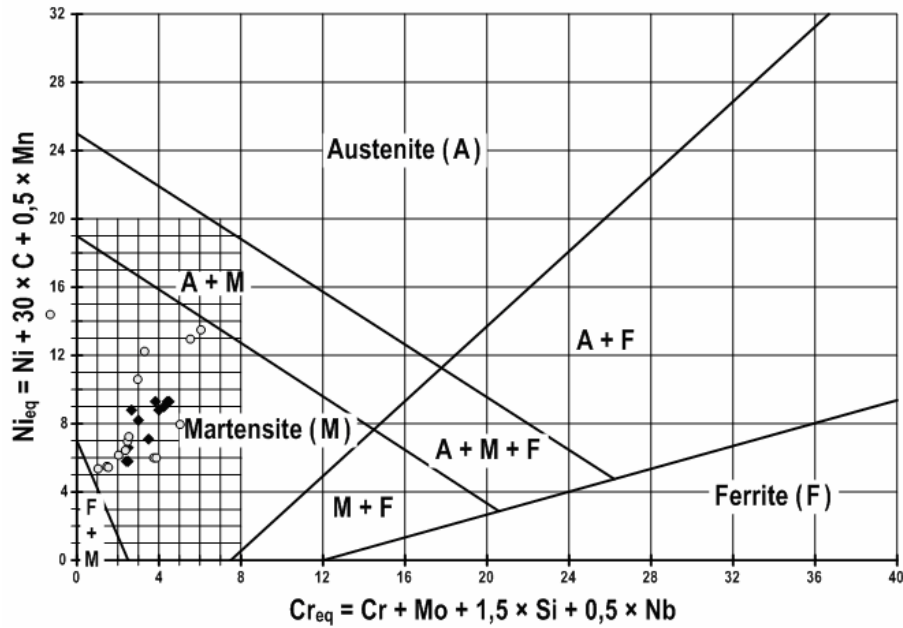
Here we start again with the same 15 initial steel compositions. The obtained after optimization new 15 “recipes” are shown in Table 6. The corresponding to them strength characteristics iterations needed and two optimization criteria values are presented in Table 7. Again, obtained new steels are in the martensite region on Figure 3 but there are bigger differences among them.

Because in this task we have two controversial criteria – one targeted towards increasing of yield strength and the other having  $Y_2$  in denominator, i.e. targeted towards decreasing it – the final decision about the best obtained “recipe” is not definitive. The best steel composition with respect to the first criterion is number 8 while with respect to the second – number 14 (all in bold in the table). This can explain the big variety of iterations number needed for individual optimization runs – from 3 to 2516. The variety of combinations between all strength characteristics is also bigger making final decision more complicated too.

### 4.3 Results Analysis

In order to compare two optimization tasks results we present them in the form of steel types diagram accepted in metallurgy – Figure 3. It consists of Cr and Ni equivalent values calculated according to the equation on diagram axis. Here Nb denotes element niobium that is not present in our recipes, i.e. Nb=0. According to the equivalent Cr and Ni values, steels are derived in several classes shown on the diagram. The bigger is the first of the optimized characteristics – yield strength – the bigger is the steel strength as a whole. From the other hand the ratio tensile strength/ yield strength (the second criterion) is related to local overloading steel resistance. Obtaining the good balance between these two major steel characteristics is difficult in practice. The imposed restriction in both optimization task (low carbon concentration) was also another characteristic that guarantees obtaining of high strength steels.

From the Figure 3 we conclude that all the obtained new steel compositions belong to the targeted martensite area. All the results from the first task are concentrated in the middle of that area while those from the second one diverge towards area borders. This result is expected having in mind controversial criteria of the second optimization task.



**Fig. 3.** Optimization results: squares represent the first optimization task solutions; circles represent the second optimization task solutions

## 5 Conclusions

The application of neural networks for modeling of the complex dependences of steel compositions characteristics from the relatively big number of alloying elements considered demonstrated their power as non-linear MIMO functions approximations. They allow us to carry out further optimization experiments by simulations in order to spent expensive laboratory synthesis and analysis of different steel compositions.

The results of the solved above optimization tasks are meaningful and will help further creation of high strength steel compositions for machine-building industry.

Our intensions for our future work are to include some other optimization criteria as well as to test the obtained steel “recipes” in practical experimental investigations in order to prove and refine their quality.

**Acknowledgments.** This work was partially supported by the Bulgarian National Science Fund under the Project No DDVU 02/11 “Characteristic modeling and composition optimization of iron-based alloys used in machine-building industry“.

## References

1. Ablameyko, S., Goras, L., Gori, M., Piuri, V. (eds.): Neural Networks for Instrumentation, Measurement and Related Industrial Applications. NATO Science series 185, IOS Press (2003)
2. Bhattacharyya, T., Singh, B.S., Das, S., Haldar, A., Bhattacharjee, D.: Development and Characterization of C-Mn-Al-Si-Nb TRIP Aided Steel. *Material Science and Engineering A* 528, 2394--2400 (2011)
3. Cichoski, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. John Wiley & Sons, New York (1993)
4. Dini, G., Najafizadeh, A., Monir-Vaghefi, S.M., Ebnonnasir, A.: Predicting of Mechanical Properties of Fe-Mn-(Al, Si) TRIP/TWIP Steels using Neural Network Modelling. *J. Computational Materials Science* 45, 959--965 (2009)
5. Esfahani, M.B., Toroghinejad, M.R., Yeganeh, A.R.K.: Modeling the Yield Strength of Hot Strip Low Carbon Steels by Artificial Neural Network. *Materials and Design* 30, 3653--3658 (2009)
6. Koprinkova, P., Petrova, M.: Optimal Control of Batch Biotechnological Processes using Neural Network Model. In: 9th Int. Conf. Systems for Automation of Engineering and Research, Sept. 24-26, Varna, Bulgaria, pp. 95--99 (1995)
7. Koprinkova-Hristova, P., Angelov, M., Kostov, G., Pandzharov, P.: Neural Network Optimization of Initial Conditions of Milk Starter Culture Cultivation. Special Issue on Innovations in Intelligent Systems and Applications of the International Journal of Reasoning-based Intelligent Systems 2(3/4), 285--292 (2010)
8. Koprinkova-Hristova, P., Tontchev, N., Popova, S.: Neural Networks for Mechanical Characteristics Modeling and Compositions Optimization of Steel Alloys. In: Int. Conf. Automatic and Informatics, Oct. 3-7, Sofia, Bulgaria, pp. 1-49--1-52 (2010)
9. Malinov, S., Sha, W.: Software Products for Modeling and Simulation in Material Science. *Computational Material Science* 28, 179--198 (2003)
10. Malinov, S., Sha, McKeown, J.J.: Modelling and Correlation between Processing Parameters and Properties of Titanium Alloys using Artificial Neural Network. *Computational Material Science* 21, 375--394 (2001)
11. Nguyen, D.H., Widrow, B.: Neural Networks for Self-Learning Control Systems. *Int. J. Control* 54(6), 1439--1451 (1991)
12. Rumelhart, D.E., McClelland, J.L.: *Parallel Distributed Processing* 1, MIT Press, Cambridge, MA (1986)
13. N. Tontchev, S. Popov, P. Koprinkova-Hristova, S. Popova, Y. Lukarski: Comparative Study on Intelligent and Classical Modeling and Composition Optimization of Steel Alloys (submitted paper)
14. Werbos, P.J.: Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE* 78(10), 1550--1560 (1990)
15. Werbos, P.J.: An Overview of Neural Networks for Control. *IEEE Control Systems Mag.* 11, 40--41 (1991)