INT. J. NUM. METH. HEAT FLUID FLOW, VOL. 6, NO. 8, 19–29 (1996)

USAGE OF ARTIFICIAL INTELLIGENCE METHODS IN INVERSE PROBLEMS FOR ESTIMATION OF MATERIAL PARAMETERS

M. RAUDENSKÝ, J. HORSKÝ, J. KREJSA AND L. SLÁMA
Technical University of Brno, Technická 2, 616 69 Brno, Czech Republic

ABSTRACT

Inverse problems deal with determining the causes on the basis of knowing their effects. The object of the inverse parameter estimation problem is to fix the thermal material parameters (the cause) on the strength of a given observation of the temperature history at one or more interior points (the effect). This paper demonstrates two novel approaches to the inverse problems. These approaches use two artificial intelligence mechanisms: neural network and genetic algorithm. Examples shown in this paper give a comparison of results obtained by both of these methods. The numerical technique of neural networks evolved from the effort to model the function of the human brain and the genetic algorithms model the evolutional process of nature. Both of the presented approaches can lead to a solution without having problems with the stability of the inverse task. Both methods are suitable for parallel processing and are advantageous for a multi-processor computer architecture.

KEYWORDS Inverse problems Artificial intelligence Material parameters

NOMENCLATURE

B	= boundary conditions
c	= thermal capacity
HTC	= heat transfer coefficient
T	= temperature
t	= time
K_1, K_2, K_3 and K_4	= unknown constants
M	= mapping function
P	= genetic population
S	= genetic strings
x	= spatial co-ordinate
X	= vector of input signals
Y	= vector of output signals
W	= vector of weights of learned neural network
λ	= thermal conductivity
ρ	= density

INVERSE TASK – PROBLEM FORMULATION

The inverse parameter estimation problem deals with determining the material properties (the cause) on the strength of a given observation of the temperature history at one or more interior points (the effects). This contrasts with the usual forward problem in heat conduction that determines the internal temperature field for given thermal material properties and a set of boundary conditions.

The "classical" approach to the thermal inverse problem is based on the minimization technique[1,2]. Some new concepts of the inverse technique based on the usage of artificial intelligence methods have recently been published[3,4]. These new approaches eliminate stability problems typical for inverse tasks and offer a relatively simple technique where skills and experience necessary for classical methods are replaced by the capability of a computer.

0961–5539/96

Received November 1994
Revised October 1995

The one-dimensional direct problem is equal to the solution of the partial differential equation:

$$\frac{\partial}{\partial x}\left(l\frac{\partial T}{\partial x}\right)ep\frac{\partial T}{\partial t} \tag{1}$$

Boundary and initial conditions must be known for the equation (1). The boundary conditions are in our case described by the ambient temperature and the heat transfer coefficient, and are supposed to be known in addition to the initial temperature field. The computation of temperature fields from equation (1) is a routine problem. The presented inverse techniques are not limited to any particular numerical method used for equation (1). The developed methods will be described on a one-dimensional problem. But the dimensionality of the problem does not play any role. The same approach can be easily extended for 2D and 3D inverse problems. 3D problems are not limited by the inverse algorithm but by the time that is consumed to find a solution of the direct 3D computations.

The nature of the problem changes when inner temperature and boundary conditions are known (usually from experiment) whereas the thermal conductivity and product of density and thermal capacity have to be found. This is the inverse problem. This problem is incorrect from the mathematical point of view and ill-posed from the numerical point of view. The incorrectness of the problem is associated with the lack of the proof of the uniqueness and the ill-posedness with the fact that small changes in input data (error in measured temperature or heat transfer coefficient) can cause large deviations in output. The complexity of the problem increases with the demand to find material parameters as a function of the temperature.

It is supposed that the thermal material parameters are described by the function:

$$\lambda = K_1 + K_2 T, \; c\rho = K_3 + K_4 T. \tag{2}$$

The goal of this inverse problem is to find constants K_1, K_2, K_3 and K_4 by knowing the temperature history in an inner point of a solid body and knowing the initial and boundary conditions.

PRINCIPLES OF USAGE OF GENETIC ALGORITHM IN INVERSE TASK

Genetic algorithms are stochastic optimization algorithms based on the method of selecting the best members from a wide set of statistically generated and updated vectors[5]. Genetic algorithms are inspired by the theory of evolution, based on the Darwinian evolution theory. The mathematical model uses terms similar to those used in nature. A set of generated solutions is called the population, solutions that fit best the optimization criteria are called the best members of the population. New members of the population are derived from these best members by an apparatus of recombination. As the method of recombination of existing solutions cannot reach the global optimum, stochastic changes, called mutations, are made in the population to get probably better solutions, that is members that fit better the optimization criteria. The worst members are destroyed during the computation to leave space for better solutions.

Each member of the population is compounded from a set of elements similar to natural genes. These genes form a genetic string which works as a member of the population. These genes can be elements of the input vector that produces an input to the optimization function. The initial population can be created as a random set of genes in the members of the population.

The genetic algorithm is then divided into four steps that are repeated until an optimum solution to the optimization problem is found:

(1) *Evaluation*: the computation of the optimization criteria (cost function) for all members of the population. The cost function must give a simple value that indicates whether a member of the population is good (or bad) and how good (or bad) it is.

(2) *Selection*: the finding and ordering of the subset of the best members in the processed population. The population is reordered according to the result of the cost function for each

member, the order is descending, i.e. from the best member at the beginning to the worst member at the end of the population.

(3) *Recombination* of the best members are subsets combined to get new members of the population. The genetic strings from good members are paired and combined to build new members that replace the worst members of the previous population.

(4) *Mutation*: the method of producing slightly different members in the population. Some genes in the newly recombined strings are changed to get probably better results from the cost function. The mutation in fact searches the neighbourhood of the population's new members for better solutions. The size of the neighbourhood (radius of the searched space) is called the size of mutation.

The heat conduction problem can be schematically written as

$$T = H(K,B) \tag{3}$$

where operator H gives temperatures T by knowing material parameters K and boundary conditions B (here supposed to be known). An inverse operator H^{-1} is not known. To solve a parameter inverse problem means to find such K for given T that $T = H(K,B)$ exploring some suitable feature of a dependency of T on K.

The genetic algorithm can be used as an engine for solving this problem. Given an initial set of genetic strings S from which a population P is derived and a mapping M that describes the predefined context of the input vectors K_i and genetic strings S_i so that

$$T_i = H(M(S_i),B). \tag{4}$$

Given a desired value T_G and a fitness mapping that evaluates differences between determined solution T and the desired values (fitness or cost function C) based on population P

$$C = \sum_{i=1}^{N} (T_i - T_{Gi})^2 \tag{5}$$

Using the denomination from the previous paragraphs we can write for our problem:

- the input vector K contains values of material parameters (these are estimated at the beginning or selected as random values);
- the response vector T contains values of the temperatures;
- the transfer function H is the solution of the Fourier partial differential equation of heat conduction.

The vector T_G is for the pattern temperatures obtained from the experiment (here by numerical simulation of the experiment). The inverse task is looking for K_{opt} which would suit the equation $T_G = H(K_{opt}, B)$. This demand is practically unattainable. The goal of the genetic algorithm is to find the vector K that produces the response T for which the fitness function C reaches the global maximum.

The genetic algorithm will find an approximation of K for the desired value T by recombining and slightly changing an initial set of vectors K. The best fitting vectors are used to create a new population of vectors until a satisfying vector K is found (this is controlled using observations of mapping C magnitude). Direct computing with the mapping H minimizes the problems of numerical stability when trying to solve inverse mapping by numerical methods.

PRINCIPLES OF USAGE OF NEURAL NETWORK IN INVERSE TASK

The study of neural network simulation originates from the effort to understand the process of thinking in the human brain. The structure and ability of the human brain and neural network, which is used for the present computations, can hardly be compared. Even if the human brain is extremely complicated it can be stated that the overwhelming structure is layered.

Neural networks (NN) are information processing systems based on the connectionist theory. A neural net is a complex system which consists of neurons (units) and connections between them. The structure of a network (the way neurons are connected to other neurons) is called the network topology. Depending on the topology, recurrent networks (with feedback connections) and feed-forward networks are realized. Only feed-forward network is considered in the following text.

From the mathematical point of view, a feed-forward network simply maps an input vector *Xn* of *n* real values to an output vector *Ym* of *m* real values. Each connection has a parameter called a weight. The network's performance can be changed by adjusting these weights. There are two stages of using the NN: the training phase and the run phase. In the training phase, weights of the network are adjusted to find the best performance in consideration of the training data which consists of many known input/output vector pairs. In the run phase, the weights are fixed and the network calculates its response (the unknown output) on the basis of a given input vector.

The adjustment of the weights in the training phase is controlled by a learning algorithm. There are two main groups of learning algorithms: supervised and unsupervised. Unsupervised learning uses only information contained in the input vectors, while supervised algorithms use pairs of input and output vectors (such a set of facts is called the training set) and calculate the error of the network as a difference between the calculated output vector and the corresponding output vector from the training set. This error is used to adjust the network's weights. Supervised learning algorithm called back-propagation is used in our case.

The elements and connections which form a back-propagation neural network are organized into layers. Figure 1 shows the commonly used structure. The structure consists of one input layer, one output layer and several intermediate (hidden) layers. The input layer has one element for each element of the input data pattern. The input layer processing elements only distribute the input signal to the elements of the first hidden layer. The output layer has one element for each of the desired outputs.

Considering the input signal

$$X = (x_1, x_2, \dots, x_n), \tag{6}$$

and the network responds with the output signal

$$Y = (y_1, y_2, \dots, y_m) \tag{7}$$

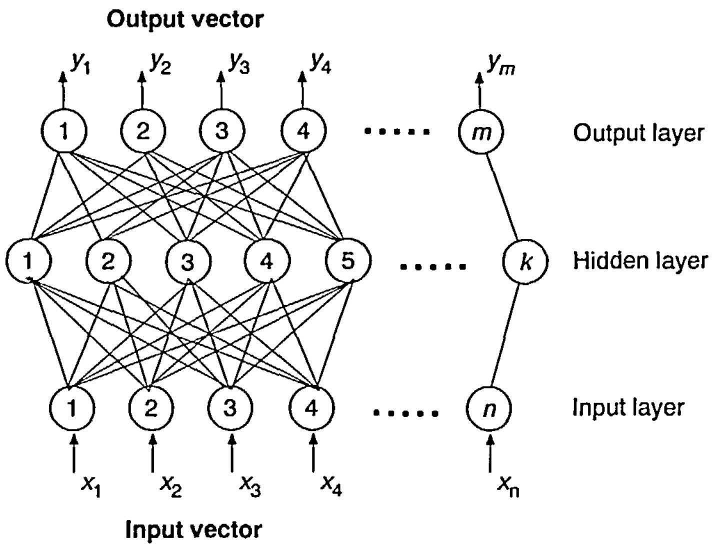

Figure 1 Neural network structure

the transformation of the signal can be written in the form:

$$Y = f(X, W), \tag{8}$$

where W is the vector of weights.

The input to the neuron is the sum of activations of the preceding neurons to which it is connected, multiplied by the weight value of each connection. This can be written in the form:

$$I_j = \sum_{i=1}^{n} W_{i,j} X_i \tag{9}$$

where I_j is the input to the neuron j, $W_{i,j}$ is the weight value of the connection from the neuron i to neuron j and X_i is the output from the previous neuron i. The output of the neuron j is given by transforming the input I_j by the activation (transformation) function.

Any activation function can be used with the back-propagation network. The only restriction is that there exists the first derivative. In the following examples a sigmoidal function was used:

As mentioned before, in dealing with our problem there are two distinct phases in the use of the back propagation network: training and application.

$$f = \frac{1}{(1 + e^{-I})} \tag{10}$$

In the training the network initially begins with random weight values connecting the various neurons. A set of values from the input data file is assigned to the input units. These values are propagated forward through the network by summing neuron inputs and calculating neuron outputs from the first layer to the last. At the output layer the calculated values are compared with the desired values. The difference is then propagated back towards the input units as the error signal. As this signal reaches each individual connection in the network the weight is modified so as to reduce the overall modelling error.

The process of fixing the input/output values and changing the weights is repeated for a number of data sets. With each iteration the weights are modified so that the network produces more precisely the correct output from the given input. The learning process is continued until the prescribed accuracy of response or a prescribed number or learning runs is reached.

The principle of neural networks usage in inverse problems can be describe as follows: an inverse problem is formulated as searching for the input data knowing the output. Probably all inverse problems are based on models which can describe the response of a system to its input signal. Here is the model described by equation (1) or (3). The necessary condition for this approach to the inverse problems is to have an estimation of the space to which the solution belongs. Now, samples from the space of solution are selected (for example, randomly) and are used for the model to generate outputs. This way the pairs of training vectors are prepared.

The inverse task searches for an input vector to the model knowing the output vector. When the input vector to the model is named Y and the output vector from the model X (see equations 6 and 7) the idea becomes clear. The neural network reverses the function of the model.

DESCRIPTION OF THE NUMERICAL TEST

The inverse problem is used for the computation of material parameters described by equation (2) from a quenching test. The real laboratory test is simulated numerically so that all parameters are known for the study of results of the inverse task.

A steel plate is uniformly heated to an initial temperature of 500°C. At an initial time instant, the quenching of the plate surface by a water spray starts. The body is cooled at $x = 0$ and the surface is adiabatic at $x = L$. The water temperature is known. The temperature history inside the plate in a position of $x = 0.1.L$ is recorded. The time step of temperature reading is 1s. The duration of the spray impulse is 12 seconds. The distribution of HTC in a time period of 50 time steps (counted from the beginning of spraying) is known and is shown in Figure 2. It is supposed that

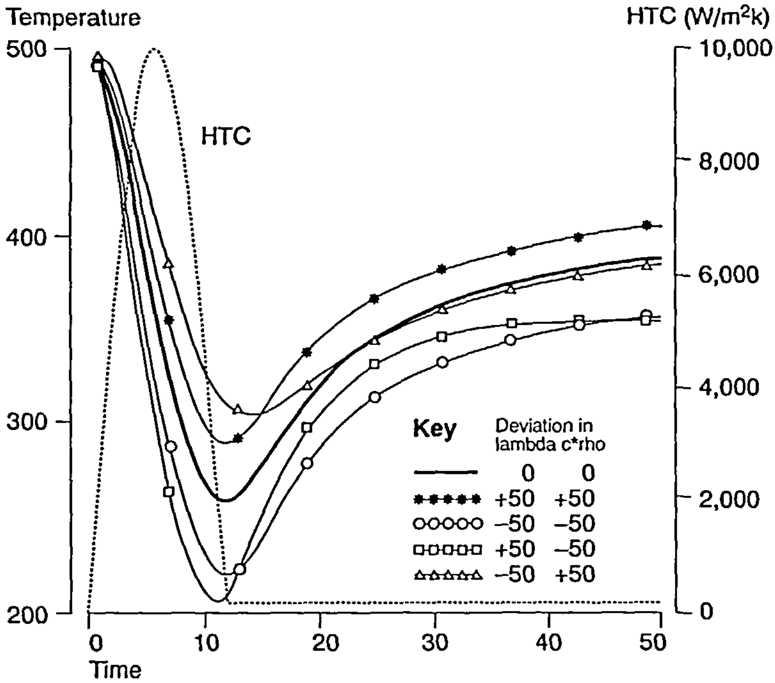

Figure 2 Heat transfer coefficient (HTC) and temperature histories used in numerical test

the boundary conditions can be determined by a well designed experiment (for example, fluid flow around a surface where an analytical solution for Nusselt number is known).

The following parameters are used for the numerical simulation of the laboratory test: K_1 = 15.0, K_2 = 0.013, K_3 = 4.1.10^6 and K_4 = 900 (see equation (2)). It is supposed that the initial estimations of these parameters defined before the start of the inverse task can vary in a range of ±50 per cent of the above mentioned values.

The material temperature history in sensor location can be seen in Figure 2. Combinations of the maximum and minimum values of λ and product $c\rho$ are used for the temperature histories presented in Figure 2. The temperature field from equation (1) was obtained by a 1D direct task based on the control volume method.

RESULTS OF THE NUMERICAL TEST

Genetic algorithm approach

The four unknown material parameters are modelled by four genes in a genetic string. A population of 32 genetic strings is used. The quality of each string is judged using equation (5). The 16 worst genetic strings are replaced by new strings in each genetic step.

The convergence process can be seen in Figure 3. Errors for the best member of population plotted here are computed for each genetic step from the following equation:

$$e=\left(\frac{1}{50}\sum_{i=1}^{50}(T_i-T_{Gi})^2\right)^{1/2} \tag{11}$$

The sharp increase in the quality of population can be seen in the early steps of the genetic algorithm. The final content of each gen after 500 steps is presented in Table 1.

The comparison of the exact and determined values of material parameters from equation (2) for a temperature range from 0 to 500°C is shown in Figures 4 and 5. Figure 6 gives a good idea about the quality of these results. An error in temperatures computed using the determined material parameters (compare with the result obtained using the exact material parameters) is plotted here for 50 time steps of the experiment. It can be observed that the maximum error in temperature does not exceed 0.1 per cent.

Neural network approach

The topology of the network used for the test was as follows:

- Input layer: 50 neurons.
- One hidden layer: 20 neurons.
- Output layer: 4 neurons.

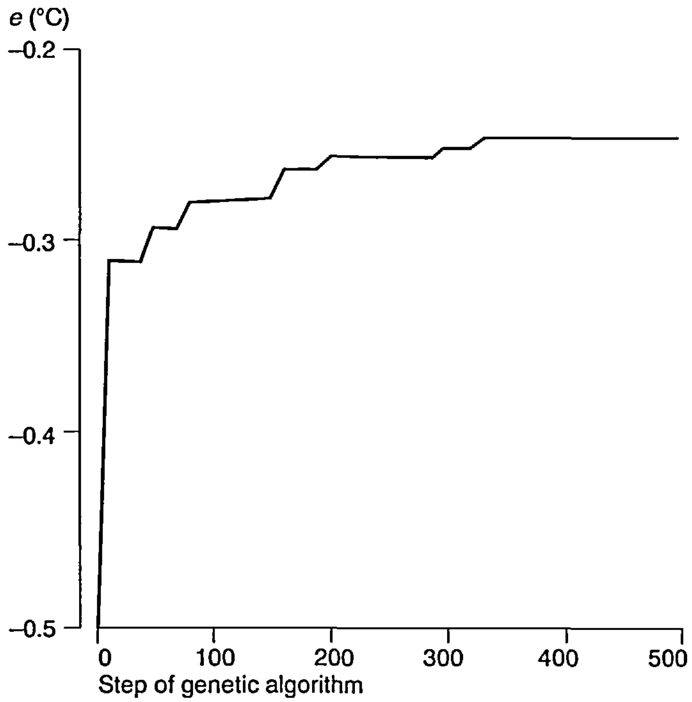

Figure 3 Quality of population in genetic algorithm

Table 1 Content of each gen after 500 steps

	Gen (parameter)			
	K_1	K_2	K_3	K_4
Exact value	15.0	0.013	4.10 E6	900
Value found by genetic algorithm	17.7	0.0109	4.34 E6	603

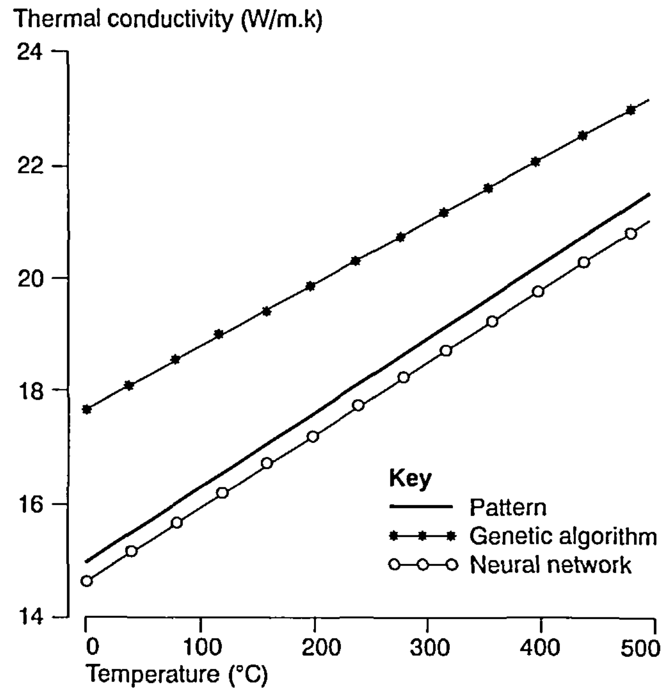

Figure 4 Results of numerical tests of estimating the thermal conductivity

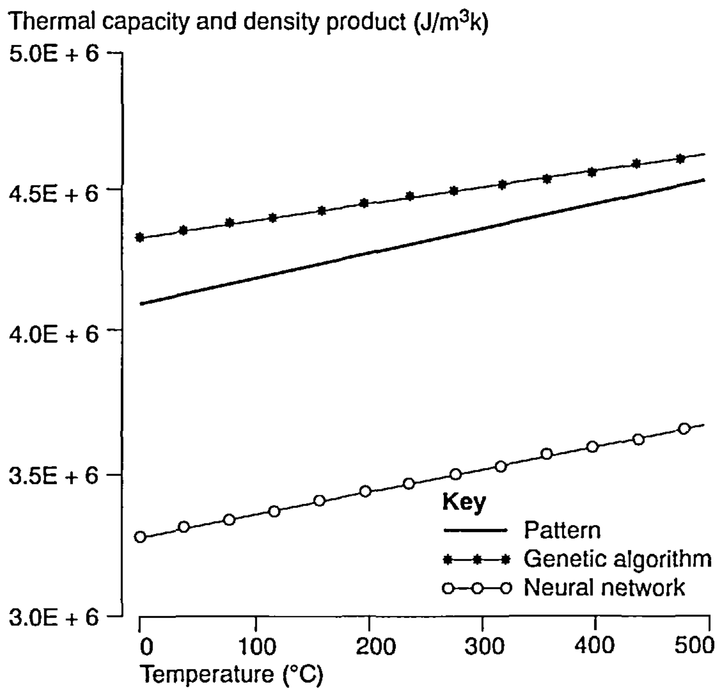

Figure 5 Results of numerical tests of estimating the product of density and thermal capacity

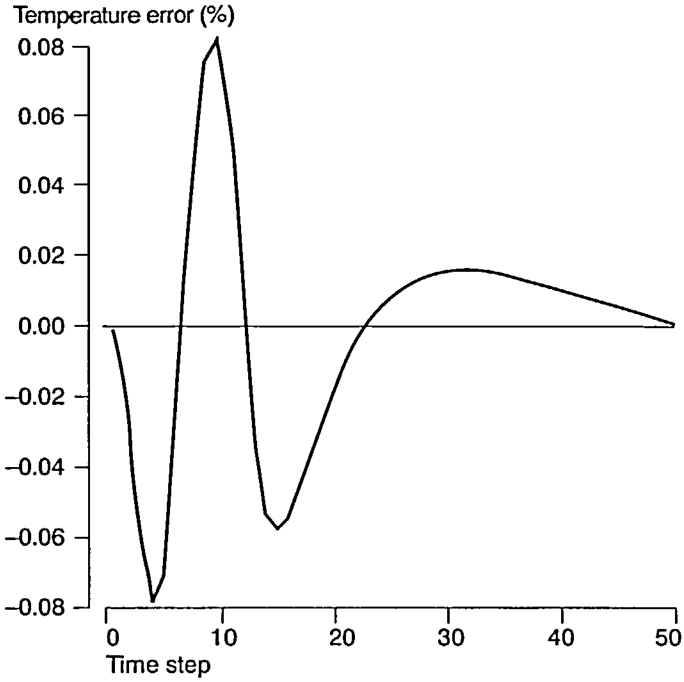

Figure 6 Error in temperature for material data obtained from genetic algorithm

The training data were formed by a set of 50 values of temperatures (input vector) and four values of thermal material parameters (output vector); 250 pairs of input and output vectors (found using the direct task) are used for the training process.

The following strategy is adopted for the generation of the training data: the area of uncertainty in the first estimation of the constants K_i was defined above as ±50 per cent of the exact value. The values of the constants K_1, K_2, K_3 and K_4 (output vector) are found as random numbers in the area of uncertainty. These constants are used in equation 2 for material parameters used in the direct task. As the boundary conditions are known the direct task can easily compute the temperatures in sensor location in 50 time steps (input vector). The area of 250 temperature histories used for training is shown in Figure 7.

The training process is stopped after 2,367 runs because the network error (the difference between the network output and the training pattern) has remained constant. Complete set of 250 training pairs is used in each run. The response of the trained net to the temperature vector T_G is shown in Table 2.

The comparison of the results can be seen in Figures 4 and 5. The error in temperatures is presented in Figure 8. This can be compared to Figure 6 for the genetic algorithm which shows that errors in neural network method are substantially higher.

The network was tested on PC (486, 8MB RAM), the training took 68 minutes.

Table 2 Response of trained net to temperature vector

	Net output (parameter)			
	K_1	K_2	K_3	K_4
Exact value	15.0	0.013	4.10 E6	900
Value found by neural network	14.7	0.0127	3.28 E6	799

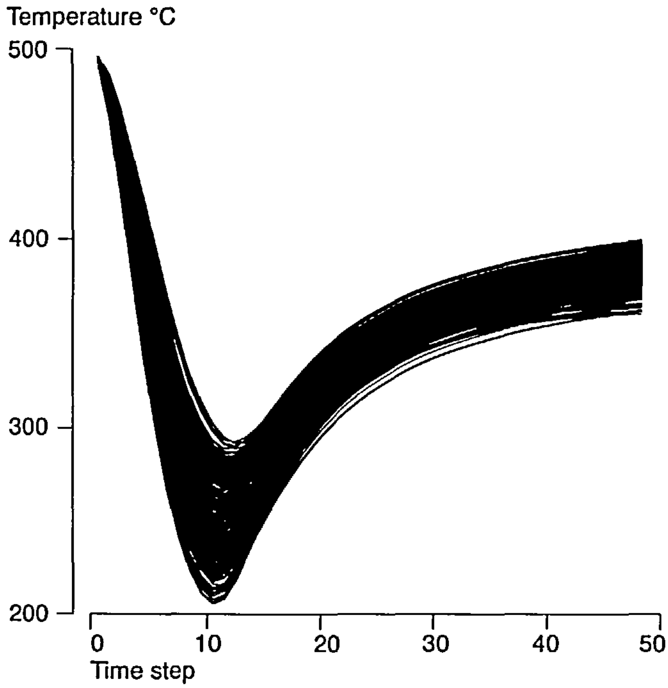

Figure 7 Area of 250 temperature histories used for training

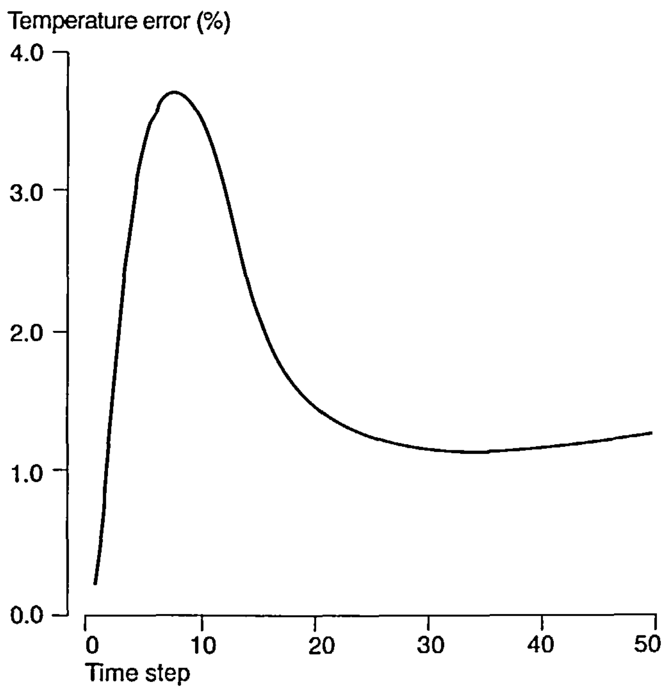

Figure 8 Error in temperature for material data obtained from neural network

CONCLUSIONS

Several advantages of the present inverse tasks based on the artificial intelligence methods can be pointed out when compared to the classical approaches. The mentioned techniques:

- can be used for a very wide class of inverse problems;
- exclude any problems with stability which are otherwise typical for inverse problems;
- are easy to program and are ideal for parallel computer architecture;
- make no difference between the 3D and 1D inverse task;
- require no special skills or experience.

Each of the two methods presented has certainly some special features and disadvantages.

The genetic algorithm approach gives relatively very precise results but is quite computer-time-consuming as a lot of direct computations are necessary.

The inverse technique based on the neural network concept requires some preliminary estimations of the area of possible solutions. This estimation is necessary for the preparation of the training pairs. The precision of the method is usually not very high. It depends on the quality of the training data, topology of the net and training process. 3D inverse task would probably require a bigger number of training runs and also a bigger number of patterns (training pairs). The training time in such case would increase rapidly. This method can be used in combination with another inverse method to enable the determination of the first estimation of results. The major advantage is that the trained net gives a solution almost immediately. It enables the use of this inverse method in realtime problems.

The same quality of results as that given by the genetic algorithm cannot be reached with even a very large number of training runs. A combination of both methods can be used – the neural network could give the first estimate of parameters and the genetic algorithm "fits" the results.

ACKNOWLEDGEMENT

The work presented in this paper was conducted under the support of the Government of the Czech Republic through project COST 512.20

REFERENCES

1 Beck, J.V. and Arnold, K.J., *Parameter Estimation in Engineering and Science*, John Wiley and Sons, New York (1977)

2 Beck, J.V., Blackwell, B. and Clair, C. St, *Inverse Heat Conduction: Ill-posed Problems*, Wiley-Interscience, New York (1985)

3 Dumek, V., Druckmüller M., Raudenský, M. and Woodbury, K., Novel approaches to the IHCP: neural network and expert systems, *First Int. Conf. on Inverse Problems in Engineering*, Palm Coast (1993)

4 Dumek, V., Raudenský, M., Grove, T. and Krejsa, J., Novel approaches to the IHCP: neural networks, *ISIP'94, Int. Symp. on Inverse Problems in Engineering Mechanics*, Paris (1994)

5 Golberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA and Wokingham (1989)