The effect of human factors on production performance in car body manufacturing

Péter Németh Ph.D. (nemethp@sze.hu)
Assistant professor
Széchenyi István University H-9026 Győr, Egyetem tér 1. Hungary

Ferenc Antreter
CFO
Erbslöh Hungaria Ltd. H-9027 Győr, Nyírfa sor Hungary

Abstract
Increasing pressure on companies in the modern automobile industry creates a need for constant development and cost effectiveness. One major part of this process is the management of human factors. In this paper we discuss the effects of human factors on production performance with artificial neural networks.

Keywords: Neural Networks, Production logistics, Manufacturing processes

Introduction
The modern automobile industry demands high quality for all actors in the supply chain. The rapidly changing business environment, the shorter life cycle of products, the reduced batch-sizes and constantly changing customer expectations create an increased emphasis on manufacturing flexibility. Increasing pressure on companies also creates a need for constant development of internal processes. Part of this development and an important task of companies is the consideration of human factors.

The company in our case study employs a performance measurement system. A properly designed enterprise performance measurement system can – besides diagnostic functions – help all stakeholders (employees also) by matching understandable, clearly identified process indicators to strategic plans. Part of this system is the constant observation of human factors that affect performance: the effect of pay rise and fringe benefit for example.

In our paper we describe the development of a tool that is able to predict the number of defect products of manufacturing processes based on human factors. With the use of this method the company will be able to predict the amount of rework needed and the cost associated to defect products in planned manufacturing processes.

Manufacturing process
The company in our case study produces body panels (aluminum decorative and

functional items) for premium cars, complying with the quality standards imposed by the industry. It is part of a supply chain with a large number of upstream and downstream connections.

The products are passing through a number of processes to achieve the final product state. The production process involves the processing of raw materials and manufacturing the finished product. The processing of raw materials in this case means the slicing and cutting to size according to specifications. The manufacturing processes are warping, bending and die cutting. The final process is a heat treatment that results in material hardening. Many differences and imperfections arise in the machining processes that result in not scheduled rework.

Almost the entire production process – suitable for performance measurement and assessment system - is recorded, together with quality indicators. Known is the production process, the sequence and number of assembling of steps, batch size and the technological order. Predetermined measurements are carried out at each step of the process. This measurement is mainly intended as documenting the processes and results without the use of data to plan future production processes. The planning of rework and the possible number of manufacturing defects is not based on these measurements. Our aim is to create a method for predicting manufacturing defects so the amount of rework needed and the associated costs can be calculated in advance and in specific cases preventive measures can be applied to lower rework cost.

Artificial neural networks

The effect of human factors on production is one example where traditional mathematical tools are not adequate. After considering the problem to be solved and based on the previous research work at the department our proposal was to use artificial neural networks as a computational model for predicting manufacturing defects.

Artificial neural networks (ANN) are inspired by the structure and functional aspects of biological neural networks (Hecht-Nielsen 1990). A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. This learning phase is important for the use of the model. With enough data from the past the ANN is able to learn the connection between the input and output data. Parallel with the learning phase a test phase is taking place where the network is testing itself with other data from the same problem. As a result after the learning the ANN can be used to predict output data from input data.

Lots of different types of artificial neural networks are proposed in the literature (Hecht-Nielsen 1990). In our application we use Multi-layer Perceptron (MLP) which is one of the most widely known types of ANNs. The topological structure of the MLP type neural network is illustrated in Figure 1.

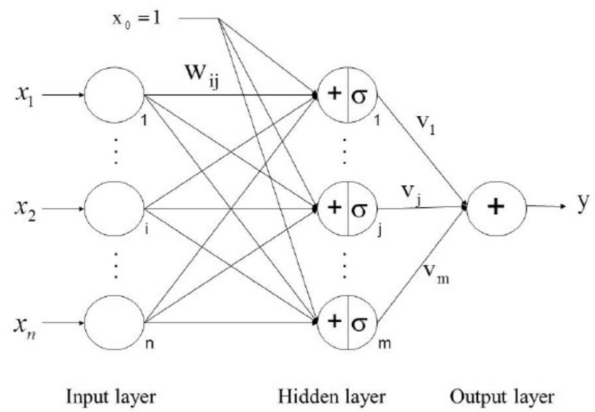

Figure 1: Topological structure of artificial neural network (Németh 2011a)

The network contains three layers. The input layer contains the inputs x_i. In the hidden layer m hidden neurons are applied. Between the input and hidden layers there are connection weights w_{ij} connecting the i-th input and j-th hidden neuron.

There is also a bias input $x_0 = 1$ with weight w_{0j} to the j-th hidden neuron. There is an output layer with output y. Between the hidden and output layers there are weights v_j connecting the j-th hidden neuron with the output (Földesi and Botzheim, 2012).
The output of the neural network can be computed as:

$$y = \sum_{j=1}^{m} v_j \cdot \sigma\left(\sum_{i=0}^{n} x_i \cdot w_{ij}\right) \tag{1}$$

In Equation (1) s is the sigmoid function:

$$\sigma(s) = \frac{1}{1+e^{-K \cdot s}} \tag{2}$$

where K is the slope parameter of the sigmoid function.

Training or learning is the method of modifying the parameters (e.g. the weights) of the neural network in order to reach a desired goal. In this paper the artificial neural network is trained in supervised, off-line manner by bacterial memetic algorithm (Botzheim 2005).

Nature inspired evolutionary optimization algorithms are often suitable for global optimization of even non-linear, high-dimensional, multi-modal, and discontinuous problems. Bacterial Evolutionary Algorithm (BEA) (Nawa 1999) is one of these techniques. BEA uses two operators, the bacterial mutation and the gene transfer operation. These operators are based on the microbial evolution phenomenon. The bacterial mutation operation optimizes the chromosome of one bacterium; the gene transfer operation allows the transfer of information between the bacteria in the population.

Evolutionary algorithms are global searchers, however in most cases they give only a quasi-optimal solution to the problem, because their convergence speed is low. Local search approaches can give a more accurate solution, however they are searching for the solution only in the neighborhood of the search space. Local search approaches might be useful in improving the performance of the basic evolutionary algorithm, which may find the global optimum with sufficient precision in this combined way. Combinations of evolutionary and local-search methods are usually referred to as memetic algorithms (Moscato 1989).

A new kind of memetic algorithm based on the bacterial approach is the bacterial memetic algorithm (BMA) proposed in (Botzheim 2005). The algorithm consists of four steps. First, a random initial population with N_{ind} individuals has to be created. Then, bacterial mutation, a local search and gene transfer are applied, until a stopping criterion (number of generations, N_{gen}) is fulfilled. BMA can be applied for training neural networks (Botzheim 2011). In this case the parameters to be optimized which are encoded in the bacterium are the w_{ij} and v_j weights of the neural network (see Figure 1). The details of BMA can be found in (Botzheim 2011).

Case study: Manufacturing defect prediction

First task of the creation of the model was the decision of inputs to be used. The output of the model is the number of defect products produced by a given worker in November 2011. After consulting with the company and based on the available data we have chosen following data as inputs:

- premium received in 2011
- years spent at the company
- evaluation of the worker from three aspects: quality of work; quality of data management and work performance (1:worst and 6:best)

We assumed that the higher the received premium in 2011 is the less manufacturing defects will be produced. Regarding the years spent at the company: a worker with more experience will produce less manufacturing defects. And finally we assumed that there will be connection between the evaluation of the workers and their level of work.

For the case study we collected the data of all workers employed at the company in November 2011. The number of defect products was 2000 pieces. We divided the data from 64 workers in two parts: 32 for training and 32 for testing the model.

For example the first line contained the following data:

Table 1: Data from the first worker

Premium (HUF)	Years spent at the company	Quality of work	Quality of data management	Performance	Defect products (November 2011)
98.800	9	3	4	4	1

In the experiments we performed 10 simulations using 6 hidden neurons and the mean relative error (MRE) of the train and test set was investigated which is defined as:

$$MRE = \frac{1}{p}\sum_{i=1}^{p}\left|\frac{y_i - t_i}{y_i}\right| \quad (3)$$

where y_i is the output of the network for the i-th pattern, t_i is the desired output for the i-th pattern, and p is the number of patterns.

The best result of the 10 simulations for the test set was 17%. This MRE percentage is similar to the results in our previous papers (Németh 2011b) so after evaluation we decided to use it for future work. This future work will probably add more type of inputs to the model thus allowing us to refine the model to achieve an even lower level of MRE.

Evaluation of results

An important part of the work was the evaluation of the results and the first use of the trained algorithm in company environment.

Three scenarios were created for the initial evaluation with the trained model with sequential worsening inputs (see Table 2). The best case scenario in this case described a person with high amount of premium received and good evaluation values and long experience at the company.

Table 2: Scenarios for the evaluation

Scenario	Premium (HUF	Years	Quality of work	Quality of data management	Performance	Number of predicted defect products per month
Best Case	200.000	10	6	6	6	1.27
Middle	100.000	5	4	4	4	8.36
Worst Case	0	0	2	2	2	12.5

The number of defect products in the three scenarios predicted by the model can be seen in Figure 2.

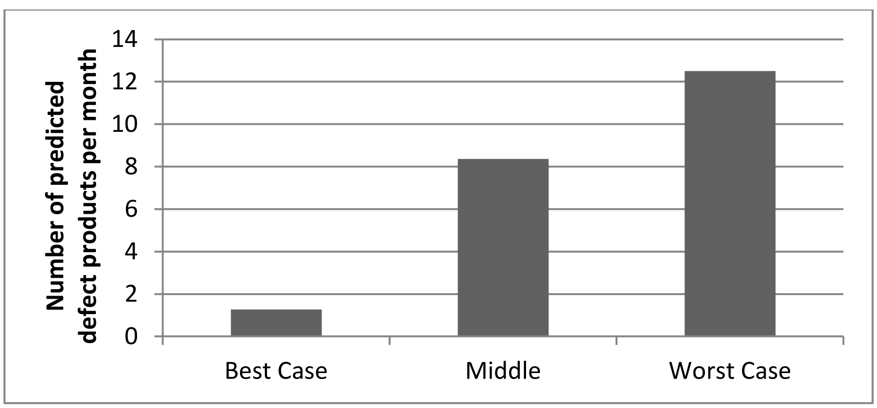

Figure 2: Defect products in the three scenarios

For further evaluation of the model we created more scenarios based on the best case mentioned above but with changing one input. The results showed that experience of the worker (the years spent at the company) is not as important as previously thought. The low level of other inputs however may cause high number of defect products. The results of these scenarios can be seen in Figure 3.

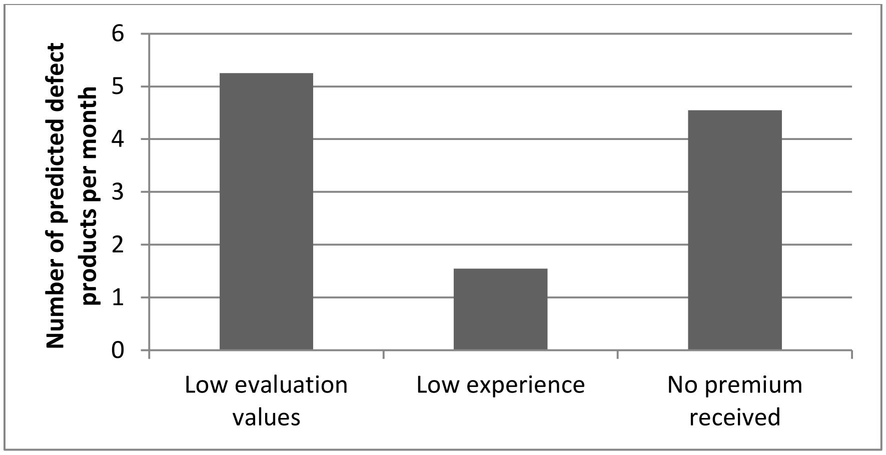

Figure 3: Defect products in three possible scenarios

These tests and scenarios showed that a usable tool was created that can be further used for other outputs. Other models can be created for further important indicators with completely different input parameters and the present model can also be refined by adding more suitable inputs and removing less significant ones (in our case the experience).

Acknowledgement
This work was supported by OTKA Fund (K-105529) of the Government of Hungary.

References

Botzheim J, Földesi P, 2011. Fuzzy neural network with novel computation of fuzzy exponent in the sigmoid functions. *Proceedings of the 8th International Symposium on Management Engineering*, Taipei, Taiwan, Pages 285–291

Botzheim J, Cabrita C, Kóczy LT, Ruano AE, 2005. Fuzzy rule extraction by bacterial memetic algorithms. *Proceedings of the 11th World Congress of International Fuzzy Systems Association*, IFSA 2005, Beijing, China, Pages 1563–1568

Földesi P, Botzheim J, 2012. Parametric Approximation of Fuzzy Exponent for Computationally Intensive Problem. *International Journal of Innovative Computing Information and Control* Volume 8, Number 8, Pages 5725-5744

Hecht-Nielsen R, 1990 *Neurocomputing*. Addison-Wesley, Michigan

Moscato P, 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Tech. Rep. Caltech Concurrent Computation Program*, Report. 826, California Institute of Technology, Pasadena, California, USA

Nawa NE, Furuhashi T 1999. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems* 7(5) (1999), pp. 608–616

Németh P, Földesi P, Botzheim J, 2011a. Enhancing warehouse performance at a global company *Proceedings of Knowledge Globalization Conference*, Boston, Massachusetts, USA, Pages 7-19,

Németh P, Földesi P, Csík Á, 2011b. The concept of logistic space in the modelling of supply chain performance, *Proceedings of the 22nd Annual Production and Operations Management Society Conference,* Reno, Nevada, USA