Soft Computing Methods for Behaviour Based Control

***Olivér Hornyák*[1]*, Gábor Barthal*[2]**
PhD, senior lecturer[1], student[2]
University of Miskolc, Department of Information Engineering[1,2]

ABSTRACT

This paper deals with Soft Computing methods to be applied to classification problems for Behaviour Based Control. It belongs to the research work in the area of disturbances handling and uncertainty management in Manufacturing Execution Systems (MES). The paper proposes classification to determine the behaviour of production system. Neural Networks and Fuzzy logic are used to execute the classification. The performance of the aforementioned methods is also presented.

INTRODUCTION

The following section deals with a possible ways of uncertainty management of production processes. The model of the practical (multi task, multiple resources) problems is non-linear. Processes can be characterised by internal relationships and external constraints. The jobs follow each other in sequence and have limited technological intensity. This may result production bottlenecks, unstable states, unwanted delays of the tasks.
The production processes are assessed by certain *process indicators*. The primary goal of a production process is to satisfy production orders under the specified constraints. The goal can be met at different cost and time. The performance indicators are depending on predictive planning, scheduling and allocations. Systems working at the margin of overload have the tendency to have bottlenecks and critical processes. These systems may behave chaotically, i.e. a small change in the scheduling may result dramatic change of the performance indicators. The stochastic events increase the tendency for the deviation of the planned states. The disturbances are often impossible to avoid. The most common sources of disturbances of a production sytem are as follows:

- Tool breakage, interruption of operations.
- Machine breakdown, outage of resources.
- High rejects rate.
- Unexpectedly low production intensity.
- Human errors.
- Material problem, supply chain delays.
- Long set-up times.
- Outage of labour resources.
- High rate of demands.
- Change of the priority of the jobs.
- Appearance of urgent jobs.

The aim of uncertainty handling at the MES level is as follows:

1. Develop performance indicators from the local data which allows the global state of whole system to be determined.
2. Identify the most important situations based on the global indicators.
3. Classify the situations into appropriate number of classes to allow interactions to be done in real time.
4. Assess the state of the production and make decision on the behaviour of production control. This defines a Behaviour-Based Control whose interactions are assigned to behaviour classes.
5. Select the appropriate actions based on the selected behaviour. The possible actions and their parameters should be modelled beforehand. The interactions should direct the production processes towards a stable, planned state.
6. The interactions affect the whole schedule.
7. Following the interactions new situations arise.

In Behaviour-Based Control the possible and allowed interactions play important role. The interactions can be done at different hierarchical levels. The corrections should be initiated in up-down direction. A higher level correction may override the decisions of the lower hierarchical levels, lower level decisions can be even banned. The hierarchical BBC propagates new constraints form the upper level to down. The identifications of anomalies spread in bottom-up direction.

A POSSIBLE CLASSIFICATION OF BEHAVIOURS

Experiments show that a few numbers of classes are favoured in practical applications. For production processes the following general global states are suggested:

- *Normal,*
- *Deviated,*
- *Critical,*
- *Dangerous*.

Normal state requires no interaction. In deviated situation the process does not go as planned: readiness for delivery is decreasing, jobs late, waste rate increasing, etc. The situation is critical if the original schedule becomes unmaintable. Usually rescheduling is required. The situation is dangerous if the master production plan becomes unfeasible.

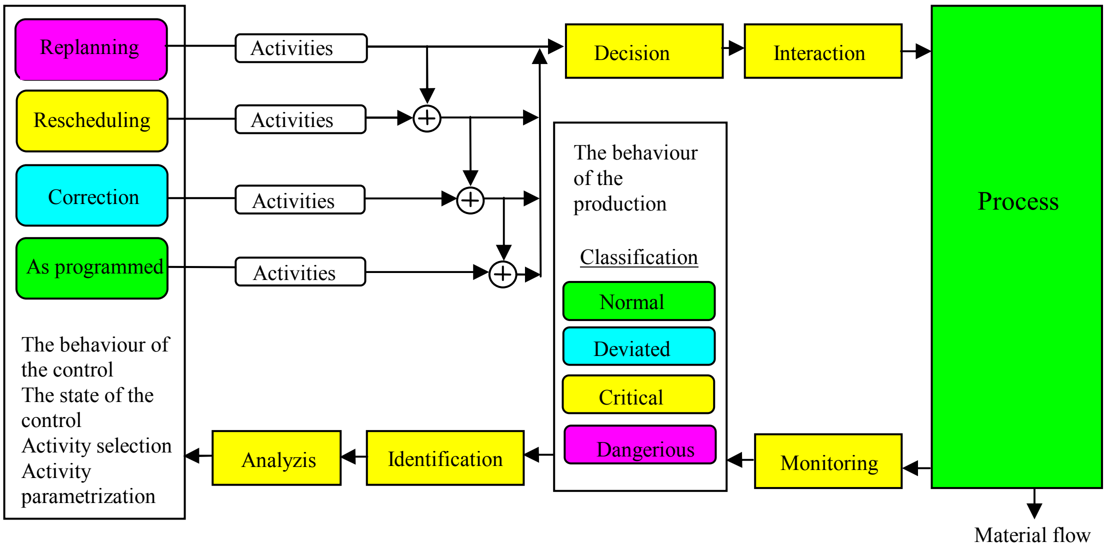

Fig 1. Behaviour based approach in production control

SOFTCOMPUTING METHODS FOR CLASSIFICATION

There are several ways to classify the system state into the aforementioned categories. Soft computing methods are widely accepted and used by researchers. In the scope of this research work the following methods were investigated:

- Backpropagation Neural Network,
- Radial Basis Neural Network,
- Probabilistic Neural Network,
- Fuzzy Logic.

A pilot application has been created for each of the methods to evaluate them. The classification problem was inspired by a case study explained in [8].

Backpropagation Neural Network (BPNN)

In the BPNN a hard-limit transfer function was used which is suitable for classification problems. Two classification regions are formed by the decision boundary line at $W.p + b = 0$, where W is the weight vector, p is the input value b is the bias. In this study one hidden layer network was used.

As there were 4 different statuses, it required 2 bits at the output. A binary code was assigned to the statuses. For learning the perceptron learning rule algorithm was used.

Radial Basis Function Neural Network (RBF)

Radial Basis Neural Networks consist of two layers: a hidden radial basis layer and an output linear layer. The first layer determines the distance of an input vector v. If the vector is *close* to the weight vector of the neuron then the output is close to 1. If the distance between the vectors is grater then the output is close to 0. The higher the output of the neurons of the first layer are the more importance they have in the second layer.

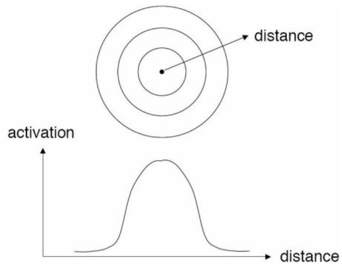

Fig 2 RBN activation

PROBABILISTIC NEURAL NETWORKS

Probabilistic networks perform classification where the target variable is categorical

PNN networks have advantages and disadvantages compared to Multilayer Perceptron networks:

- It is usually much faster to train a PNN network than a multilayer perceptron network.
- PNN networks often are more accurate than multilayer perceptron networks.
- PNN networks are relatively insensitive to outliers (wild points).
- PNN networks generate accurate predicted target probability scores.
- PNN networks approach Bayes optimal classification.
- PNN networks are slower than multilayer perceptron networks at classifying new cases.
- PNN networks require more memory space to store the model.

Fuzzy Logic

The initial project contained great number of input variables thus a performance issues was experienced. To cope with this a two level hierarchical processing was introduced. In the first step all production orders were classified individually. After conducting the PO statuses the classification of the whole system was evaluated. As it is well known the number of rules highly affects the computing time: the more rule we have the more processing time is required. It was expedient to reduce the number of rules to the minimum. At the same time the remaining rules should classify the systems without too much ambiguity.

During the fuzzy classification all the rules are evaluated one after the other by using the appropriate variables. A rule is a set of NOT, AND and OR connections. The variables are inserted into a rule resulting a numeric value in the range of [0..1]. This represents the degree of truth, in other words the extent to which a proposition is true.

Some sample rules used in the first pass of fuzzy processing:

1. IF *tardiness* is *small* AND *machine utilisation rate* is *big* THEN *status* is *normal*.
2. IF *tardiness* is *considerable* AND *machine utilisation rate* is big THEN *status* is *deviant.*
3. IF *quantity* is *many* AND *priority* is *high* AND *tardiness* is *considerable* THEN *category* is *critical.*
4. IF *tardiness* is *considerable* AND *machine utilisation rate* is *big* THEN *status* is *dangerous.*

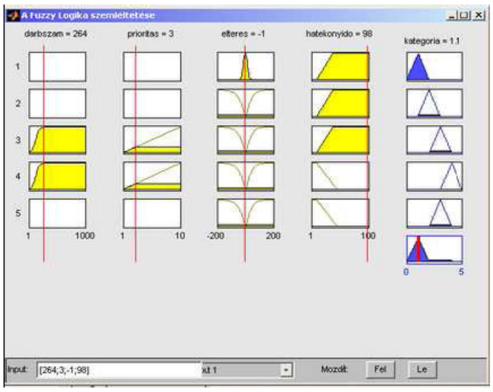

Fig 3 Screenshot of fuzzy classification

In the second processing phase the overall tardiness time was also calculated. The rules used in this phase were as follows:

1. IF *status of* PO_i is *normal* AND *tardiness* is *negligible* THEN *status* is *normal.*
2. IF *status of* PO_i is *deviant* AND *tardiness* is *small* THEN *status* is *deviant.*
3. IF *status of* PO_i is *critical* AND *tardiness* is *observable* THEN *status* is *critical.*
4. IF *status of* PO_i is *dangerous* AND *tardiness* is *observable* THEN *status* is *dangerious.*

EVALUATION OF THE SOFT COMPUTING METHODS

Two kinds of evaluation were applied: the method was tested against the data set used at the learning phase, and then the method was tested against a new data set. (Obviously, fuzzy processing had no learning data set, so that test case was skipped).

BPNN succeeded the learning phase but after certain time it showed no more progress in learning, there was a remaining error. The learning time was considerable.

RBF achieved a very small remaining error and produced better results than BPNN. The learning time was smaller. By increasing the size of the training set the network produced more reliable output.

PNN was characterised by the smaller learning time among the NNs used. By increasing the size of the training set the network produced more reliable output. However RBF was more reliable.

Fuzzy system was difficult to create good rules. After some experiments the rules were fine tuned. Its performance lagged behind the NNs. Very likely some more fine tuning is still required.

Some performance data can be found in Table 1.

NN type	Number of inputs	Number of correct classifications	Relative goodness [%]	Learning
BPNN	100	53	53	Set of 100 input data
RBF	100	23	23	
PNN	100	100	100	
BPNN	100	51	51	Set of 400 input data
RBF	100	89	89	
PNN	400	299	74.7	
BPNN	100	52	52	Set of 800 input data
RBF	400	380	95	
PNN	800	609	76.1	
Fuzzy	100	70	70	-
Fuzzy	800	544	68	

Table 1. Performance data

FUTURE WORK

The future work is to implement an *Action Generator* which – based on the classification results – selects the appropriate actions. The actions can be generated by a rule based system or an expert system or a neural network. Cockpit Task Management, in which the operators are supported by the classification and other relevant data, can be also suitable.

REFERENCES

[1] Hornyák, O., Erdélyi, F., Kulcsár, Gy. (2006): **Behaviour Based Control for Uncertainty Management in Manufacturing Execution System.** Proceedings of MITIP 8. International Conference, Budapest, pp. 73-81.

[2] Lengyel, A., Erdélyi, F. (2006): **Behaviour Based Combined Approaches to Uncertainty Management in Manufacturing Systems**. IWES 6th International Conference. Tokyo, pp. 77-83.

[3] Funk, K., Kim, J. N. (1995): **Agent-based aids to facilitate cockpit task management**. IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp. 1521-1526.

[4] Stenzel, R. (2000). **A behaviour-based control architecture**, in Proceedings of the 2000 IEEE International Conference on Systems, Man & Cybernetics, Vol. 5, pp. 3235-3240.

[5] Mataric, M. J. (1998): **Behavior Based Robotics as a Tool for Synthesis of Artificial Behavior and Analysis of Natural Behavior**, Trends in Cognitive Science. V. 2. No. 3. pp. 82-87.

[6] Tóth, T., Erdélyi, F. (1997): **The Role of Optimization and Robustness in Planning and Control of Discrete Manufacturing Processes**, Proceedings of the 2nd World Congress on Intelligent Manufacturing Processes & Systems. June 10-13. pp. 205-210, Budapest, Hungary, Springer.

[7] Hornyák, O. – Erdelyi, F. (2006), **Uncertainty Management in Manufacturing Execution Systems**, MicroCAD'2006, 16-17 March , pp. 295 - 303