Simulation Reduction Models Approach Using Neural Network
Philippe Thomas, Denise Choffel, André Thomas

To cite this version:
Philippe Thomas, Denise Choffel, André Thomas. Simulation Reduction Models Approach Using Neural Network. 10th International Conference on Computer Modelling and Simulation, EUROSIM'08, Apr 2008, Cambridge, United Kingdom. pp.679-684, 2008. <hal-00282804>

HAL Id: hal-00282804
https://hal.archives-ouvertes.fr/hal-00282804
Submitted on 28 May 2008

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.
Simulation Reduction Models Approach Using Neural Network

Philippe THOMAS, Denise CHOFFEL, André THOMAS
Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039),
Nancy-University, CNRS
ENSTIB 27 rue du Merle Blanc, B.P. 1041
88051 Epinal cedex 9 France
philippe.thomas@cran.uhp-nancy.fr

Abstract

Simulation is often used for the evaluation of a Master Production Schedule (MPS). Also, the goal of this paper is the study of the design of a simulation model by reducing its complexity. According to theory of constraints, we want to build reduced models composed exclusively by bottleneck and, in order to do that, a neural network, particularly a multilayer perceptron, is used. Moreover, the structure of the network is determined by using a pruning procedure. This approach is applied to a sawmill flow shop case.

1. Introduction

The planning or scheduling scenario evaluation by simulation is very useful for the decision makers. Indeed, simulation highlights the evolution of the machines states, the WIP (work in process), and the queues. This information is useful in order to perform a “Predictive scheduling” [13] which is the function that concerns the MPS initially established with the Manufacturing Planning and Control System (MPCS), as opposed to “Reactive scheduling” which gives a new MPS established after significant events occur during the concerned period. Effectively, once the MPS is released, daily events may require it to be brought into question: this is the problem of rescheduling. The real time systems performing manufacturing checks (production reporting) lead to have current follow up information very quickly into the management system [11]. However, it is difficult to use this huge amount of information in order to make decision [19, 20]. At this level of planning, load/capacity equilibrium is obtained via the “management of critical resource capacity” function or Rough-Cut Capacity Planning (RCCP) which essentially concerns bottlenecks [27]. Goldratt and Cox, in “The Goal” [6] put forward the Theory of Constraints (TOC), which proposes to manage the whole supply chain by bottlenecks control. For this, the use of dynamic discrete events simulation of material flow is helpful [22]. In fact, simulation models of actual industrial cases are often very complex and the modellers encounter problems of scale [17]. Also, many works have highlighted the interest to use simplest (reduced/aggregated) models of simulation [1, 3, 15, 18, 28]. In addition, neural networks have proved there abilities to extract performing models from experimental data [26]. Also the use of neural networks appears recently as an interesting approach within the framework of the supply chain [2, 21].

According to the theory of constraint concepts, the main goal of this paper is to prove the interest of using neural network in order to reduce simulation models useful for the decision making process in rescheduling problem. In the next part, the proposed approach of reduction model and the multilayer perceptron are presented. The third part will be devoted to the presentation of an industrial application which is a sawmill flow shop case. In order to evaluate the proposition, the obtained reduced simulation model of the sawmill flow shop case is compared to the complete one in the last part.

2. The model reduction

2.1. The algorithm

Amongst various authors, Zeigler was the first to deal with this problem [29]. In his view, the complexity of a model is relative to the number of elements, connections and model calculations. He
distinguished four ways of simplifying a discrete simulation model, in replacing part of the model by a random variable, coarsening the range of values taken by a variable and grouping parts of a model together. Innis et al. [10] first listed 17 simplification techniques for general modelling. Their approach was comprised of four steps: hypotheses (identifying the important parts of the system), formulation (specifying the model), coding (building the model) and experiments. Leachman [12] has proposed a model that considers cycle times in production planning models, especially for the semi-conductor industry which uses cycle time as an indicator. Brooks and Tobias [1] suggest a “simplification of models” approach for those cases where the indicators to be followed are the average throughput rates. Other cases have been studied [8, 9].

The reduction algorithm proposed is an extension of the works of Cybenko [4] and Funahashi [5] have proved that a multilayer neural network with only one hidden layer using a sigmoidal activation function and an output layer using a linear activation function can approximate all non linear functions with the desiderated accuracy. This result explains the great interest of this type of neural network which is called multilayer perceptron. In this research work, our hypothesis lies in the fact that a part of the modeled production system could be approximate by a non linear function obtained thanks to a MLP.

The structure of the multilayer perceptron is recalled here. Its architecture is shown in figure 1.

The i-th neuron in the hidden layer (figure 1) receives \( n_0 \) inputs \( \{x_0^i, \ldots, x_{n_0}^i\} \) with associated weights \( \{w_{0}^i, \ldots, w_{n_0}^i\} \). This neuron first computes the weighted sum of the \( n_0 \) inputs:

\[
z_i = \sum_{h=1}^{n_0} w_{ih}^i x_h^0 + b_i^1
\]

(1)

where \( b_i^1 \) is a bias or threshold term. The output of the neuron is given by a so called activation function of the sum in (1):

\[
x_i^1 = g(z_i^1)
\]

(2)

where \( g(.) \) is chosen as an hyperbolic tangent:

\[
g(x) = \frac{2}{1+e^{-2x}} - 1 = \frac{1-e^{-2x}}{1+e^{-2x}}
\]

(3)

The neuron in the last layer simply performs the following sum, its activation function being chosen linear:

\[
z = \sum_{i=1}^{n_1} w_i^2 x_i^1 + b^2
\]

(4)

where \( w_i^2 \) are the weights connecting the output of the hidden neurons with the output neuron and \( b^2 \) is the threshold of the output neuron.

Now, only the number of hidden neurons is always unknown. In order to determine it, the learning starts from an overparametrized structure. A weight elimination method is used to remove spurious parameters. The pruning algorithm used is the Optimal Brain Surgeon (OBS) [7].

The learning of the MLP is performed in three steps:

- Initialisation of the weights and biases of an oversized structure by using the Nguyen Widrow algorithm [16].
- Learning of the parameters by using Levenberg-Marquard algorithm with robust criterion [24].
- Weights elimination by using the Optimal Brain Surgeon algorithm with a robust criterion [25].
3. The overview of the sawmill

At the time of the study, the sawmill SIAT had a capacity of 270,000 m³/year, a 52 million euros turnover and 300 employees.

The physical industrial production system is composed of sequential work centers (kockums saw, trimmer, sorter,…) and queues or conveyors (RQM4, RQM5, RQM7, …). The log enters the system in RQM1 then is steered to RQM4 or 5 according to the cutting pattern. Following that it passes to the cutting machine (Canter). It then enters the edger. After this phase, the log is transformed into main and secondary products. The final operation is the cross cutting which consists in cutting up products to length.

4. The simulation models

4.1. The complete model

Preceding works [22, 23] have permit to construct the complete model of the sawmill process. This model is presented figure 2. It permits to simulate the log arrival. The second module, “input sorting” permits to drive the log into RQM4 or RQM5 in function of its characteristics. The conveyors RQM4, RQM5 and RQM7 serve of input inventory for the canter line.

![Figure 2. The complete model](image)

Two other modules permit to model the canter line and the passage on the RQM7 conveyor. The modelisation of the canter line needs two submodels which permit to differentiate main and secondary products. Two other modules permit to model the kockums saw and the trimmer.

4.2. The reduced model

As we can see, the design of a complete model for the simulation of a workshop is a difficult task which leads to a complex model. Moreover, the bottleneck of this line is the trimmer [22]. And industrial objectives are optimising the use of the bottleneck work station in order to maximise its utilisation [6, 14].

Consequently and within this framework, the modelisation of the functioning of inventories RQM4 RQM5 and RQM7, and of the canter line is unnecessary. Also, all the parts surrounded by grey dashed line on the figure 2 give no direct and useful information for the evaluation of a MPS. In fact, only the arrival times of the products in trimmer queue are useful in order to simulate the work load of this bottleneck. And this is why a multilayer perceptron is used to replace all the parts surrounded by grey dashed line on the figure 2. Then the neural network uses the available shop floor information.

For this, we need to collect the available data in input of the process. First, each log holds numerous information which are collected by a scanner in input of the canter line. This information are relative to the product dimension, as length (Lg) and three values for timber diameter (diaPB; diaGB; diaMOY). These variables serve to drive the log to RQM4 or RQM5 queues which is additional information (RQM).

In addition of this dimensional information, we have to characterise the process variables at the time of the log arrival. Particularly, the input stock of the trimmer (Q_trim), the utilisation rate of the trimmer (U_trim) and the number of log present in the process between the inputs of RQM4 and RQM5 and the output of the canter line (Q_RQM).

The last type of information is related to the cutting plan of the logs. In fact, each log will be cut into n main or secondary products. In our application, the cutting plan divides the log into 7 products:
- 2 secondary products resulting from the first step of the canter line cutting process on the saw CSMK,
- 2 secondary products resulting from the second step of the canter line cutting process on the saw CSMK after going through on the RQM7 queue,
- 3 main products resulting from the third step of the canter line cutting process on the saw MKV.

These two saws (CSMK and MKV) belong to the canter line. These 7 products can be classified into three categories according to the location (CSMK or MKV) and the time during the cutting process (first or second cutting). This information is given by the variable (T_piece). The last information is the thickness (in mm) of the product which is also the reference. In our case, we are taking into account only two references: main products 75; secondary products 25 (ref).

Consequently, the neural networks input variables are: Lg; diaGB; diaMOY; diaPB; ref; T_piece;
Our objective is to estimate the delay ($\Delta T$) corresponding to the throughput time for the 12775 products. $\Delta T$ is measured between the process input time and the trimmer queue input time. In practice $\Delta T$ is the output of the neural network:

$$\Delta T = \sum_{i=1}^{30} w_i^2 \cdot \left( \sum_{h=1}^{10} w_{ih}x_h + b_i \right) + b^2$$  \hspace{1cm} (5)$$

The learning of the network is supervised. So, it is necessary to divide the database into two datasets, the learning one and the validation one. The number of hidden neurons should be determined. In order to determine it, the learning starts from an overparametrized structure and a weight elimination method is used to remove spurious parameters [25].

So, the learning begins with a structure using 30 hidden neurons (5) which correspond to 361 parameters. Then, the spurious parameters are removed one after the other and the sum square of the error (SSE) is calculated for the two data sets. Figure 3 presents the SSE for the learning data set (grey circle) and for the validation data set (black cross) in function of the number of parameters remaining.

Different choices can be performed concerning the best structure. The structure with 161 parameters is the smallest which satisfies SSE. The structure with 225 parameters gives the lowest SSE for the learning dataset and those with 240 parameters gives the best SSE for the validation dataset. However, these three structures are very close together. And, in these three structures only 4 hidden neurons have been removed. The selected structure, with 240 parameters, uses 26 hidden neurons. It is presented figure 4.

Firstly, the results obtained on the learning data set are presented. Figure 5 shows the delay $\Delta T$ (ordinate) used by the 6365 products (abscissa) of the learning dataset. The black line presents the delay for the complete model and the grey line for the reduced model. The times are given in seconds. We can see that the major part of the pieces uses less than 1000 seconds in order to enter in the input stock of the trimmer. However, some of these products use more than 3000 seconds to perform the same course. Figure 5 shows that the times given by the two models are very similar except nine peaks where we can notice important differences. Figure 6 which presents the error performed between the two models corroborate these first remarks. This fact can be explained by the
lack of some variables which can have a particular incidence on the data, as, for example, the percentage of machine-gunned logs, or the repartition of logs into the different RQM queues, …

In the same way, the error mean performed on the validation data set remain small (9.4 seconds) and with an important standard deviation (448) for the same reasons as those mentioned above. If peaks are not considered, the mean error becomes 2.1 seconds.

The figure 8 shows the error performed on the validation data set and confirms the remarks performed during the study of the figure 6.

5. Conclusion

A new approach of simulation model reduction is presented here. This approach uses a neural network, and more particularly, a multilayer perceptron. The aim is to model the functioning of the part of the process which is not constrained in capacity. This approach has been applied to a sawmill flow shop case. The results have shown that:
- the two sets of data present similar results,
- the average of the error (less than 10s) is small according to the process time scale (2000s).

That means that it is very interesting to use neural network to model a part of the process instead of computing with the complete model. It is quicker and easier. It allows the decision making system to focus on the management of the bottlenecks.

Our perspectives are to investigate the structure determination of the network, and particularly, the choice of the inputs of it, and the validation of this approach on different application cases.

6. References


