

# A SIMULATION METAMODELLING BASED NEURAL NETWORKS FOR LOT-SIZING PROBLEM IN MTO SECTOR

Hachicha, W.

Unit of Mechanics, Modelling and Production (U2MP), Engineering School of Sfax,  
University of Sfax, B.P. 1173, 3038, Tunisia  
Higher Institute of Industrial Management of Sfax, B.P. 1164, 3018, Tunisia  
E-Mail: wafik.hachicha@isgi.rnu.tn

## Abstract

Simulation is essentially a trial-and-error approach, and is therefore, time-consuming and does not provide a method for optimization. Metamodelling techniques have been recently pursued in order to tackle these drawbacks. The main objective has been to provide robust, fast decision support aids to enhance the overall effectiveness of decision-making processes. This paper proposes an application of simulation metamodelling through artificial neural networks (ANNs). The building of the appropriate ANN model over second-order linear regression model and the reverse simulation metamodelling as simulation-optimization are assisted by the Neuro<sup>®</sup> Software. To validate the proposed approach, a case study which is adopted from literature, deals with a lot sizing problem in make-to-order supply chain. The optimal solution is to determine the fixed lot size for each manufacturing product type that will ensure order mean flow time target. The comparative results with others metamodels techniques; illustrate the efficiency and effectiveness of the proposed approach.

(Received in January 2011, accepted in June 2011. This paper was with the author 1 month for 1 revision.)

**Key Words:** Simulation Based Metamodel, Neural Network, Multiple Criteria Optimization, Lot-Sizing Problem, MTO, Supply Chain Management, Case Study

## 1. INTRODUCTION

Today, simulation is a popular tool for the analysis and/or design of existing or proposed complex systems. This popularity is partly due to its flexibility, and to its ability to model real-world systems in some detail, which, in turn, leads simulation to be used as a decision support tool in supervising and controlling the underlying system. Though, simulation models require fewer restrictive assumptions than mathematical models when representing complex, dynamic systems, these models themselves are usually fairly complex and of relatively high dimensionality. That is, the performance of a simulation model mostly depends on a large number of parameters or factors that act and interact in a complex manner. However, with simulation modelling, the relationships between the design parameters and their resulting performance measures are not explicitly known. Therefore, simulation modelling becomes a trial-and-error process in which a set of input factors is used to predict a set of output performance measures. If the desired performances are achieved, a good system design has been attained; otherwise the process is repeated until a satisfactory set of performance measures is obtained. Unfortunately, the iterative nature of this process can result in both high computing costs and difficulties in interpretation and prediction of the results. In order to overcome these problems, there is a significant body of literature devoted to the problem of simulation metamodelling. A metamodel, or model of the simulation model, simplifies the simulation optimization in two ways: the metamodel response is deterministic rather than stochastic, and the run times are generally much shorter than the original simulation.

Various modelling forms have been introduced for metamodelling [1], such as response surface metamodel, regression spline metamodels, spatial correlation (kriging) metamodels, radial basis function metamodels and neural network metamodels. An important issue in metamodelling is the selection of an appropriate metamodel type under a given condition. Many authors such as in [2-5] were found that in most cases, Artificial Neural networks (ANN) metamodels outperform the others regression metamodel techniques. In addition, two approach types have been used for developing simulation metamodels [2]: the direct simulation metamodelling approach used for prediction, and the reverse simulation metamodelling approach used for optimization. When building the metamodel using the direct approach, the inputs of the simulation (design parameters) are used as inputs for the metamodel, and the outputs of the simulation (performance measures) are used as desired outputs for the metamodel. When building a reverse simulation metamodel, the outputs of the simulation (performance measures) are used as inputs to the metamodel, and the inputs of the simulation (design parameters) are used as desired outputs of the metamodel. The decision-maker inputs the required performance measures and the reverse metamodel outputs the optimal (necessary) parameters to achieve those measures.

Moreover, The SO approaches can be classified in two classes [6]: The SO of a single performance measure, and the SO with multiple performance measure. Numerous researchers have investigated the SO problem under the existence of a single performance measure. Unfortunately, multiple performance measure SO has been an area of research fairly overlooked by the simulation community in contrast to the vast research performed on the single objective SO problem. Real-life problems often embody many characteristics of a multi-objective optimization. For instance, the simultaneous maximization of machine utilizations and minimization of the work-in-process while keeping system throughput is above a certain level. Use RSM in solving a multi-objective problem requires more treatments than in solving a single response problem. A variety of methods can be found in the literature such as in [6-7], devoted to solving multi-objective SO problems.

In this paper the use of metamodels to approximate the reverse of simulation models is explored using an ANN model over second-order linear regression model and with multiple performance measure. That is, given a set of desired performance measures, the metamodels output a design to meet management goals. To validate the proposed approach, a literature case study is adopted from Hachicha et al. [8-9] for which mathematical model is still extremely complex up to now (multi-stage, multi-product, multi-location, multi-resource with setup, capacity constraints and stochastic demand). The aim of this paper is to develop a simulation based-ANN metamodels for solving the Lot Sizing Problem (LSP) in make-to-order (MTO) supply chain. To the best of our knowledge, excluding [8-9], only methods based on queuing theory have been subject to a mathematical analysis of MTO problems.

The remainder of this paper is organized as follows: the ANN paradigm and ANNs as simulation metamodels are provided in Section 2. The details for the case study are presented in Section 3. The details of the proposed methodology and the case study empirical results are presented in Section 4. Finally, Section 5 presents the conclusions.

## **2. ARTIFICIAL NEURAL NETWORK**

### **2.1 The ANN paradigm**

Artificial intelligence is the generic name given to the field of computer science dedicated to the development of programs that attempt to replicate human intelligence. ANNs is one of the artificial intelligence techniques that have gained an important role in solving problems with extreme difficult or unknown analytical solutions. ANNs are composed of interconnected,

usually adaptive elements, which are intended to respond to stimulus in a manner not unlike the human nervous system. Perhaps the most attractive property of these structures is the ability to learn associations between and patterns within data. The teaching procedure commonly consists of repeatedly presenting known input and output patterns to the system as it restructures itself to recreate the patterns. There are several important resulting attributes. First, it is not necessary to develop difficult algorithms, statistical models and knowledge-based rules can be extracted from data, and the inherent parallelism speeds processing [10]. Another important advantage is the ANNs ability to generalize data and produce outputs for previously unseen inputs.

The most fundamental aspect of ANNs is their use of simple interconnected processing elements, which correspond loosely to neurons and synaptic connections in the brain. The strength of the connecting links determines the functionality of the ANN as a whole. The strength of each link is determined as the ANN learns. The most common applications of ANNs involve regression and classification. Regression models attempt to estimate input-output transformation functions, while classification involves using the known inputs to determine class membership [11].

There are three basic elements of an ANN model, which are illustrated in Fig. 1 [12]: (1) a set of connecting links or *synapses*, each characterized by a *weight* of its own. Specifically, an input signal to the synapse  $j$  connected to neuron  $k$  is multiplied by the synaptic weight  $w_{kj}$ ; (2) an *adder*, which sums the input signals weighted by the respective synapses of the neuron, and (3) an *activation function*, which limits the permissible amplitude range of the output signal to some finite value. It defines the output of the neuron in terms of the induced local field, which is formed by the *linear combiner output*  $u_k$  and the *bias*  $b_k$ . This externally applied bias is used to increase or lower the net input of the activation function.

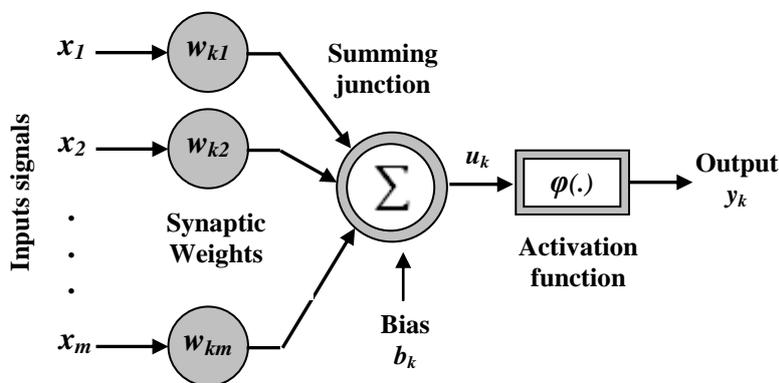


Figure 1: Non-linear model of a neuron.

The ability to learn and improve its performance from examples is the ANN's fundamental trait. Haykin [12] defines as learning, in the ANNs context, the process by which the free parameters of the ANN are adapted through stimulation provided by the environment where the network is embedded. Instead of following a set of rules, ANNs are able to learn underlying relationships from a collection of training examples. ANNs are usually classified into two main categories: recurrent networks, in which loops occur because of feedback connections, and feed-forward networks, in which the network structure has no loops. The choices of network architectures are intimately linked to the learning algorithm used in the training of the network.

Multilayered, feed-forward, non-linear network models are utilized for general-purpose and generalization applications. These types of networks are commonly known as Multilayer Perceptions which have been successfully applied to solve diverse problems by training them

in a supervised environment. The back-propagation network (BPN) was adopted for the proposed metamodel. The BPN is the well-known supervised ANN technique for adjusting randomized weights during the data-learning phase according to the steepest gradient along the error surface. The BPN algorithm learns to generate a mapping from the input space to the output space by minimizing the error between the desired output and the actual output produced by the network. At each an iteration, neurons slightly adjust their input connection weights in the direction that reduces their signal errors.

## **2.2 ANNs as simulation metamodels**

The main purpose of simulation metamodelling is to reduce the cost, time, and amount of effort required during a simulation analysis. A metamodel, or response surface, is an approximation of the input/output function implied by the underlying simulation model. It is usually a supplementary model that can be alternatively used to interpret a more detailed model. Regression has been for more than two decades the most popular method to perform metamodelling of simulations. However, many authors such as [8-10] rigorously demonstrated that multilayer ANNs were theoretically capable of approximating any measurable function to any desired degree of accuracy. They provided a fundamental basis to establish that these networks were “general universal approximators”.

Within the area of simulation and ANN two different areas of simulation metamodels have surfaced in the last decade: The direct simulation metamodelling such in [4] and [5] and the reverse simulation metamodelling such in [2] and [14]. In this paper, we will only discuss the reverse simulation metamodels which is applied in this study. The extent of the literature on ANN metamodelling is somewhat limited. The method of using ANN for metamodels was introduced by Hurriion [15]. Chryssolouris and Domroese [14] gave a general overview and discussion of potential uses of ANN for simulation modelling. They explored the use of ANN for identifying the relative importance of pertinent manufacturing criteria for given performance measures. Finally, the studies by Kilmer [16] and Kilmer et al. [17] exposed the use of supervised neural networks as a metamodelling technique for discrete, stochastic simulation. Kilmer [16] compares results to ordinary regression, and suggests the possibility of applying the metamodelling scheme and provides a foundation and methodological guidelines for developing ANN metamodels. Kilmer et al. [17] reports the creation of ANN metamodels for simulating a factory’s inventory warehouse. Recently, Mohamed Sultan et al. [18] use radial basis function ANN metamodel for controller parameters optimization for fluid mixing system. The reported studies are evidence that metamodelling through ANNs may overcome many of the disadvantages of traditional regression approaches. Regression is limited to approximate a subset of the simulation domain, is sensitive to deviations from statistical model assumptions such as constant error variance, and there exists the possibility of selecting an incorrect functional form. ANN-based models can achieve a global fit, remove the possibility of pre-selecting improper approximation functions, are able to accommodate together discrete and continuous variables, are insensitive to common deviations from statistical model assumptions, and are robust to incomplete, noisy, or inaccurate data [4].

## **3. PROBLEM DESCRIPTION**

An increasing number of companies have been shifting its production from the make-to-stock (MTS) to the MTO sector such as Dell, BMW, Compaq, and Gateway [19]. However, most of the literature research on Manufacturing Resources Planning (MRP) approaches concentrates on MTS systems. The MTO area has not received the same degree of attention. In particular, there are only a handful of research papers that explicitly talk about the LSP in MTO sector.

In this case, only methods based on queuing network models have been proposed in the literature. Though, these existing analytical models are not able to handle all the dynamically changes in Supply Chain (SC) and they require many simplification assumptions and constraints.

### 3.1 Process description

A detailed of process description of the SC is a mandatory step for understanding what is implemented inside and how the simulation model and metamodels works. The SC under study is composed of six locations (entities) as indicated in Fig. 2. Two locations, L1 and L2, serve a distribution or retail function and are exposed to independent customer order. Locations L3, L4, L5 and L6 serve a production function. These locations may be considered to be constrained because time delay operations are assumed. The SC under study belongs to multi-stage category. Each finished product (P1, P2 and P3) has component items (manufacturing products) as indicated by the Bill-of-Distribution (BOD) and the Bill-of-Material (BOM). Fig. 3 describes BOD and BOM dictating the flow of material. Note that the BOM also shows the quantity of items required by each finished part. The number in brackets beside the part number shows the requirements for all finished part order.

Part types P1 and P2 are both derived from P4. P3 is derived from P7. Since P7 is also a component of P4, the parts going directly to L2 could be considered spare or repair parts. At L3, P4 is assembled from three unit of P5, one unit of P7 and one P8 parts. At L5, P7 is produced from two units of P6. At L4, P8 is produced from one unit P6 and P5 is produced from one unit of raw material RM5. At L6, P6 is produced from one unit of raw material RM6. Supplies of RM5 and RM6 are assumed to be unlimited.

### 3.2 Model assumption

The current paper considers a situation in which several types of parts are produced at the same SC. If the production is changed from one type to another, a set-up is needed. For some reasons, such as a large assortment of parts which is subject to regular changes, a highly uncertain demand, or unique parts, no safety stocks can be kept and production scheduling are based on customer's specifications. For specificity, this research makes the following assumptions concerning the simulation parameters. For all part types, it was assumed that the required lot-size order quantity was not available from the upstream supplier before shipments could be released (i.e. lot splitting is allowed). Furthermore, for assembly operations it was assumed that the required lot-size quantities of all components were not also available before any components were released for shipment. A common transit time was then applied to all components so arrival at the assembly location was simultaneous. The transit times are shown along side the arrows in Fig. 2. Moreover, when there are a capacity constraints, lots arriving at manufacturing locations must undergo a setup time and a lot processing time. Mean setup and part processing times for processed and assembled parts are displayed in Table I.

The lot set-up times are stochastic and follow a normal distribution with a coefficient of variation of 0.3. The lot processing times are deterministic and based on multiplying the lot size times the fixed part processing time. Processing of all lots in queue is based on FCFS. Transit times, also shown in Table I, are defined as the time to move an available lot of inventory from an upstream location to a downstream location. The transit times for all part types were assumed to be stochastic and follow a gamma distribution with a coefficient of variation of 0.1. No capacity constraints were assumed for inventory transportation.

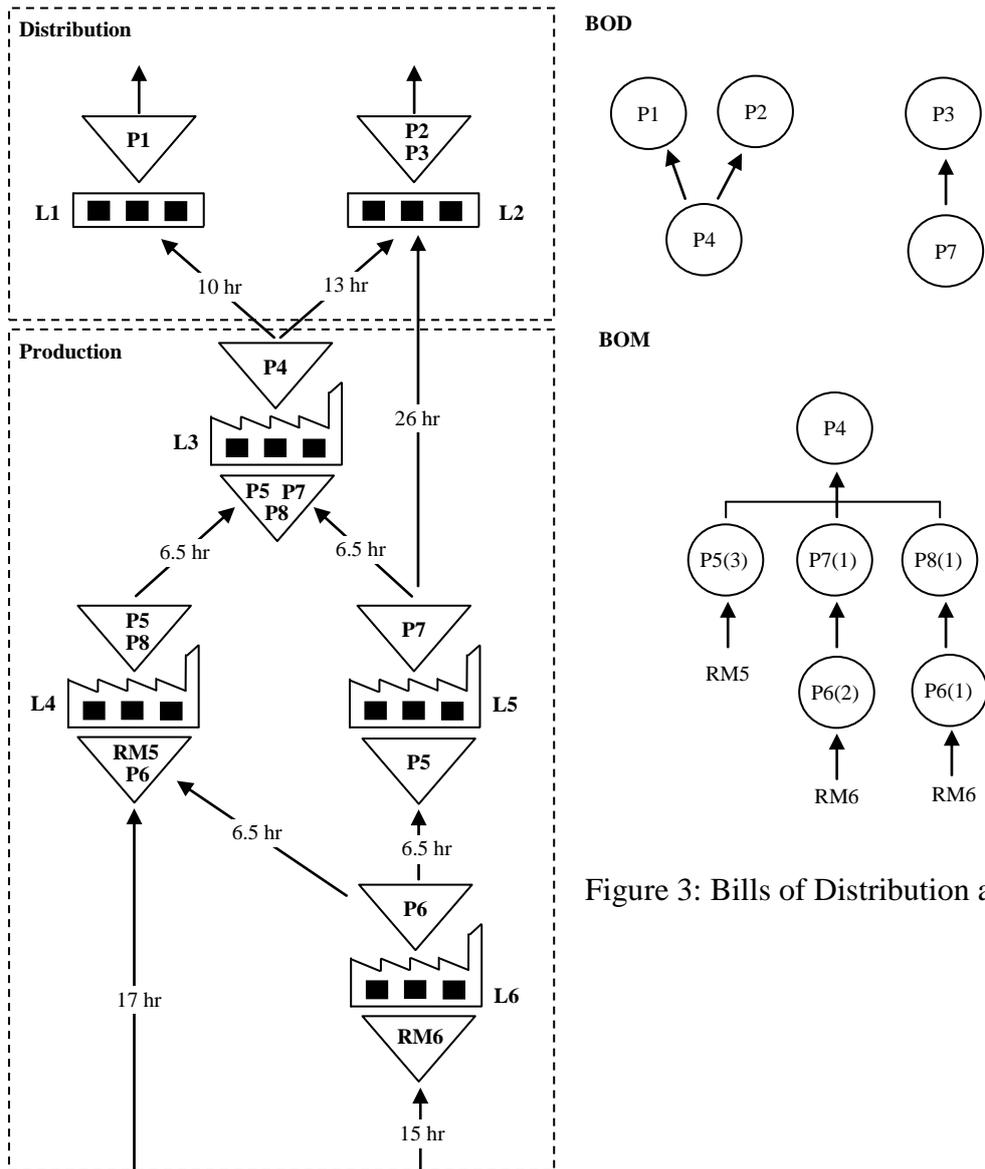


Figure 3: Bills of Distribution and Materials.

Figure 2: Configuration of the supply chain network.

Table I: Supply chain data.

Part type	Mean Setup time (hr)	Part Processing time (hr)	Mean Transit time (hr)	Lot size
P1	0.5	0.009	10.0	200
P2	0.5	0.011	13.0	200
P3	0.9	0.023	26.0	300
P4	1.2	0.014	6.5	Unknowns “to be optimized”
P5	0.24	0.002	17.0	
P6	0.50	0.003	15.0	
P7	0.8	0.006	6.5	
P8	1.6	0.007	6.5	

Furthermore, customer’s order which consists of fixed lot size of same product type follows a gamma distribution with a mean of 4000 units per week. One week is assumed to be equal to five working days. Times are given in hours, using the assumption there are 40 hours

per week (or 8 hours per day). Daily order variation is determined on the basis of having a week order coefficient of variation of 0.1. Based on customer requirements and specifications, lot sizes.

### 3.3 Performance measure

Many performance measures (metrics) can be considered for SC analysis such as work-in-process, mean tardiness, mean flow time and others. The main thing, the coherence between the criteria of performance guarantees the overall performance of the SC. In MTO production strategy, the main priority is to minimize mean lot tardiness in order to avoid associated penalties. On the other hand, it is also important to minimize mean lot earliness and it is related to extra storage costs. For this reason, the most appropriate performance measure would be the order mean flow time (OMFT). There is a fixed target value (delivery promise date) proposed for each customer order. This target value must be framed between the lower and the upper values which are fixed for reduce cost. The target OMFT for finished product types P1, P2 and P3 are fixed at 60 hr, 65 hr, and 60 hr, respectively.

## 4. EMPIRICAL RESULTS

The case study was solved by reverse simulation based ANN metamodel. The proposed solution approach is illustrated by the case study. The major steps are shown in Fig. 4.

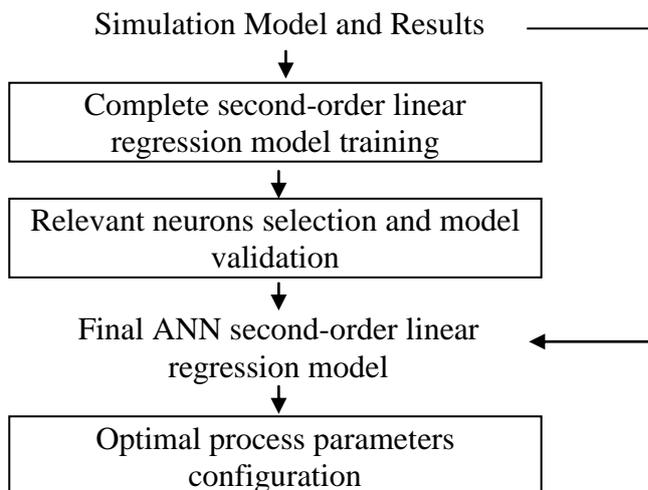


Figure 4: Architecture of the proposed methodology.

### 4.1 Simulation model and results

This step consists of creating the simulation model to be used as the baseline for the ANN metamodel. It involves the selection of relevant input and response variables, the construction of the basic simulation model for generating the training and testing facts, and the determination of the appropriate number of replications for each developed scenario in to ensure statistical confidence. The simulation model and more details were presented in [8]. The sample size decision separates the available samples, presented in Table II, into two groups as: training (for 1 to 40) and testing samples (for 41 to 52), which are used for the model training and for the model fitness validation, respectively.

Table II: Experimental design [8].

Exp.	Manufacturing part lot levels					MFT for each Finished parts (hr)			Exp.	Manufacturing part lot levels					MFT for each Finished parts (hr)		
	P4	P5	P6	P7	P8	P1	P2	P3		P4	P5	P6	P7	P8	P1	P2	P3
1	1	1	2	2	1	60.49	64.96	55.06	27	1	1	1	2	2	63.14	63.33	54.84
2	1	1	2	1	1	61.89	63.99	55.15	28	1.5	0.31	1.5	1.5	1.5	63.29	65.04	56.48
3	1	1	1	1	1	60.72	65.65	55.96	29	2	1	1	2	1	60.48	64.92	58.22
4	1.5	1.5	1.5	0.31	1.5	62.38	64.80	63.08	30	2	2	2	1	1	60.70	64.68	58.66
5	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55	31	1	2	2	2	1	60.03	64.20	55.21
6	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55	32	1.5	1.5	0.31	1.5	1.5	62.40	66.33	59.17
7	1	2	2	1	2	60.23	66.29	55.77	33	2	2	2	2	1	61.50	65.24	55.19
8	2	1	1	1	1	60.75	64.83	59.08	34	1	2	1	1	2	61.38	65.66	54.89
9	1	2	2	1	1	60.23	66.29	55.77	35	2	2	1	1	1	61.57	65.42	59.44
10	1	1	1	2	1	63.14	63.33	54.84	36	1	1	2	1	2	61.89	63.99	55.15
11	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55	37	1	2	1	2	2	60.30	65.25	55.25
12	1	2	1	1	1	61.38	65.66	54.89	38	1	1	2	2	2	60.49	64.96	55.06
13	2	1	2	1	1	60.35	64.34	56.18	39	1.5	1.5	1.5	1.5	0.31	60.98	63.34	56.55
14	2	1	1	1	2	60.75	64.83	59.08	40	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55
15	2.7	1.5	1.5	1.5	1.5	59.37	64.77	61.06	41	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55
16	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55	42	1.5	1.5	2.7	1.5	1.5	60.51	66.01	57.35
17	1	2	1	2	1	59.67	64.01	58.08	43	1.5	2.7	1.5	1.5	1.5	61.85	61.42	56.40
18	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55	44	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55
19	2	2	1	2	1	59.67	64.01	58.08	45	2	1	2	2	2	60.22	64.16	57.23
20	2	2	1	2	2	59.67	64.01	58.08	46	1.5	1.5	1.5	2.7	1.5	59.69	64.63	56.66
21	2	2	2	1	2	60.70	64.68	58.66	47	2	1	2	1	2	60.35	64.34	56.18
22	1	2	2	2	2	60.03	64.20	55.21	48	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55
23	2	2	1	1	2	61.57	65.42	59.44	49	1.5	1.5	1.5	1.5	1.5	60.98	63.34	56.55
24	0.31	1.5	1.5	1.5	1.5	63.54	72.14	56.24	50	2	1	1	2	2	60.48	64.92	58.22
25	2	2	2	2	2	61.50	65.24	55.19	51	1.5	1.5	1.5	1.5	2.7	60.98	63.34	56.55
26	2	1	2	2	1	60.22	64.16	57.23	52	1	1	1	1	2	60.72	65.65	55.96

#### 4.2 Complete second-order linear regression model training

Definition of an ANN architecture means determining the number of input, output, and hidden neurons, and the number of hidden layers. As indicated in Fig. 5, the configuration of the neural network second-order linear regression model has 5 input nodes (LP4, LP5, LP6, LP7, and LP8), three output nodes (OMFTP1, OMFTP2, and OMFTP3), and one hidden layers which is composed of 15 nodes. The second order polynomial model allows characterizing the influence of squares and cross products. It is a non-linear model towards the inputs variables, but linear towards the coefficients.

A commercial software package, ©Neuro One Standard 6.12 was adopted for the ANN modelling and optimization. The used software includes two mains modules: (1) The Neuro One module is the tool which allows the loading and the pre-treatment of data, the selection of the descriptors, the building of polynomial linear models and of most frequent neural networks, the supervision and analysis of the learning results, charts and statistic tables editing, results storage, and (2) the Neuro Fit module is a tool which simulates in direct mode,

reverse mode, input compensation mode, desirabilities about outputs of the models that have been saved from the Neuro One module.

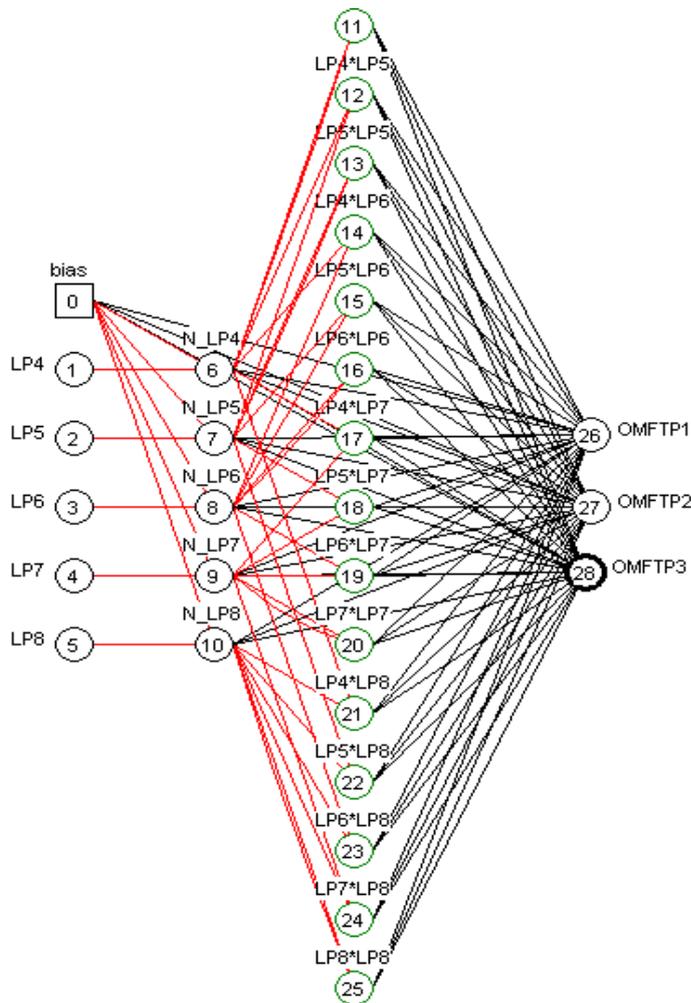


Figure 5: The complete second-order linear regression model.

### 4.3 Relevant neurons selection and model validation

The window presented in Fig. 6 gives an analysis of the relevance of each hidden neuron. The ©Neuro One software performs a Gramm-Schmidt analysis followed by a relevance computation by the random probe method on the network neurons directly connected to the output. If the output neuron is linear, this analysis is very relevant, for the output value is a linear combination of the hidden neuron contributions. Only 8 neurons of 20 are selected as relevant. The click on the button "Apply 50 % Selection" to withdraw the irrelevant neurons and provides the final neural network model.

Validation of the selected ANN models is a critical phase in any study. Validation involves determining whether the conceptual model is an accurate representation of the real process. The selected ANN models can be validated by executing a set of simulation situations different from those used in the training and testing phases. A practical procedure to validate situations where correct answers are unknown, as it occurs in numerous simulation studies, involves comparing the results of the ANN models with the results from a certain number of different simulation packages. Twelve data are used for testing the neural network simulation metamodel. The prediction time for all testing data was less than one millisecond.

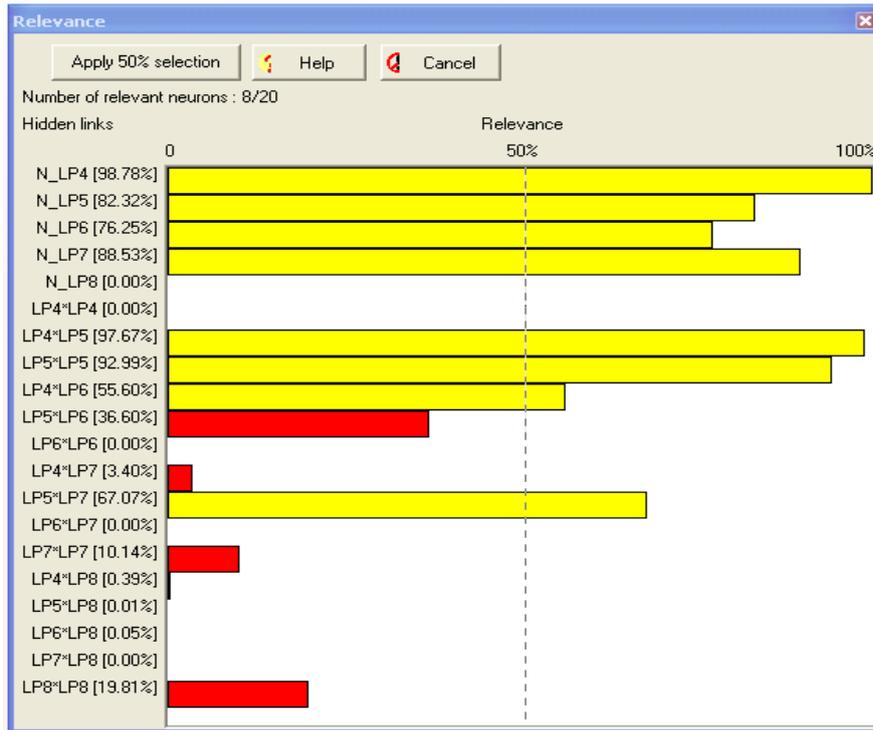


Figure 6: Relevant neurons selection.

Table III shows the average prediction maximum error of each performance measure is 1.18 %. There is no prediction error that exceeds 5 %. In addition the prediction time for OMFT vector is less than one millisecond, and it can predict a lot of configurations in a very short time.

Table III: Testing results.

Testing No.	Number in Table II	Prediction Results						Max error (%)
		Simulation Model			Metamodel			
		OMFTP1	OMFTP2	OMFTP3	OMFTP1	OMFTP2	OMFTP3	
1	41	60.98	63.34	56.55	60.91	63.32	56.42	0.23 %
2	42	60.51	66.01	57.35	60.46	65.93	56.13	2.13 %
3	43	61.85	61.42	56.40	61.56	63.02	55.72	2.61 %
4	44	60.98	63.34	56.55	60.91	63.32	56.42	0.23 %
5	45	60.22	64.16	57.23	60.21	64.57	56.41	1.43 %
6	46	59.69	64.63	56.66	59.95	63.96	57.57	1.61 %
7	47	60.35	64.34	56.18	60.33	63.76	57.17	1.76 %
8	48	60.98	63.34	56.55	60.91	63.32	56.42	0.23 %
9	49	60.98	63.34	56.55	60.91	63.32	56.42	0.23 %
10	50	60.48	64.92	58.22	60.53	64.61	58.26	0.48 %
11	51	60.98	63.34	56.55	60.52	63.26	55.45	1.95 %
12	52	60.72	65.65	55.96	61.47	65.90	55.74	1.24 %
<b>Average</b>								1.18 %

#### 4.4 Optimal process parameters configuration

Based on ANN model presented in Fig. 5, the second step of the proposed approach is the reverse simulation metamodelling that will be used as an optimization tool. The case study problem is to determine the fixed lot size for each manufacturing product type P1, P2 and P3, which will ensure order mean flow time target. The optimal configuration is indicated in Fig. 7. Lot size of product P4, P5, P6, P7 and P8 would be set at 472 units, 286 units, 207 units, 675 units and 234 units, respectively.

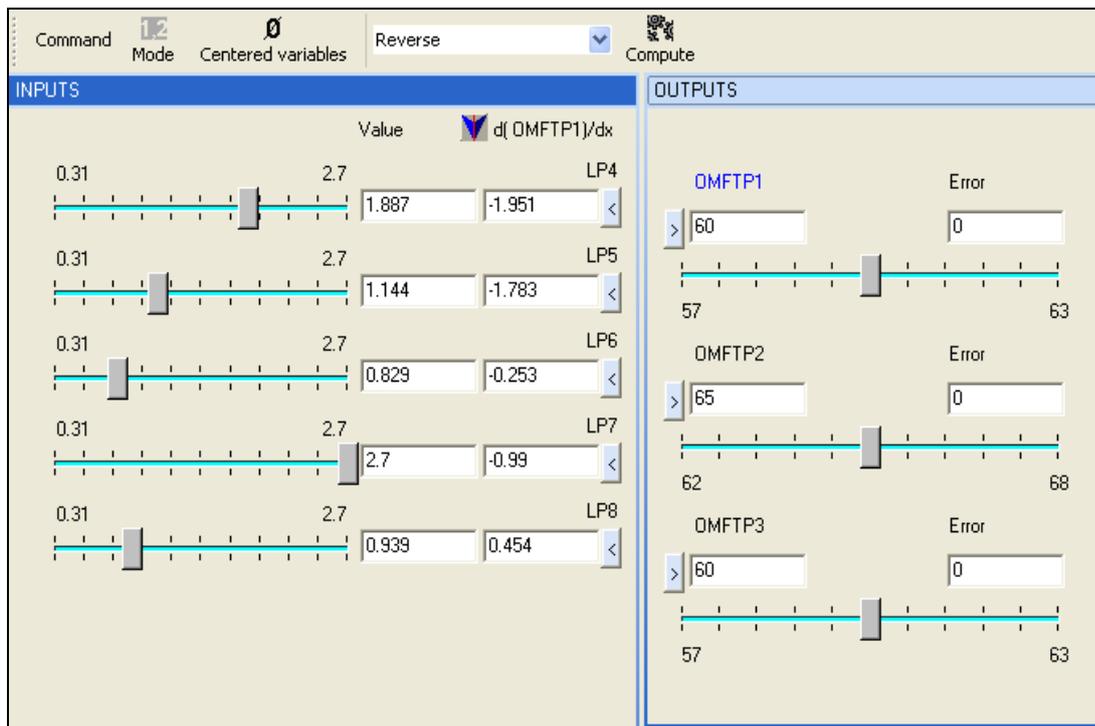


Figure 7: The main window of optimal configuration.

For the same SC configuration investigated in this research. Hachicha et al. [8] proposes an approach that suggests using simulation metamodels based on Response Surface Methodology (RSM) and the multi-objective desirability optimization. They obtain in optimal configuration that, lot size of P4 would be set at 562 units. lot size of P5 would be set at 77.5 units, lot size of P6 would be set at 274 units, lot size of P7 would be set at 212.5 units and lot size of P8 would be set at 675 units. Moreover, in another work, Hachicha et al. [9] presents a simulation metamodels based on classical Factorial Design of Experiment (FDOE). They obtain in optimal configuration that: lot size of each product P4, P5 and P8 would be set at 500 units, lot size of P6 would be set at 168 units and lot size of P7 would be set at 464 units. Table IV summarizes the optimal value of order mean flow time performance found by these approaches. In order to compare these different outcomes, Euclidian distance between the solution results and the target value are computed for each solution approaches. Euclidian distance examines the root of square differences between the obtained solution and the target value.

The results of the comparative study between the three methods show at first, all methods which are metamodel based approach, give an acceptable solution. The first one which is based on RSM realized by [8] and the proposed approach which is based on ANN are very effective while the second method which is based on classical FDOE [9] gives a good satisfactory solution.

Table IV: Comparative results.

	Metamodel based RSM [8]	Metamodel based FDOE [9]	Metamodel based Neural Network (this paper)
LP4	562	500	472
LP5	77	500	286
LP6	274	168	207
LP7	212	464	675
LP8	675	500	234
OMFTP1	60.00	61.08	60.00
OMFTP2	65.00	65.00	65.00
OMFTP3	60.00	59.99	60.00
Distance	0	1.08	0

Based on the results, the response of the metamodel should not be interpreted in an absolute sense. In solving the present case study problem, ANN applications appear simple and robust. Despite the constraints of the proposed ANN based metamodel, they are promising for practical application.

## **5. CONCLUSION**

This research deals with the LSP in supply chain solved by reverse simulation metamodelling through ANN methodologies. The supply chain under study which operates in MTO environment (no possibility for stock keeping and limited production capacity) is characterized by multi-product, multi stage, multi-location production planning with capacity constrained and stochastic parameters such as lot arrivals order, transit time, set-up time, processing time, etc.

The results of the case study confirm the effectiveness, flexibility and usability of the ANN method in practical applications. Moreover, ANNs proved to be a viable tool for stochastic simulation metamodelling. Despite the noise introduced by the stochastic nature of most simulation processes, the ANN-based simulations may be able to fairly capture underlying relationships between appropriately selected inputs and responses. In addition, the availability of ANN design and analysis in many commercial software packages such as Neuro® make the proposed method an attractive approach for real case applications.

This paper only developed a reverse simulation metamodel based on ANN for a case of lot-sizing problem optimization. It might be necessary to better specify the tolerance and repeatability, according to the ANN methods used in others case studies. It would be interesting to make results comparisons with some other simulation optimization approaches (evolutionary algorithms, artificial intelligence, etc.) to ensure a good repeatability and effectiveness of ANN methods. The direction based on this idea will be more considered in future work.

## **REFERENCES**

- [1] Barton, R. R.; Meckesheimer, M. (2006). Chapter 18: Metamodel-Based Simulation Optimization, Henderson, S. G. and Nelson, B. L. (Eds.), *Handbook in OR & MS*, Vol. 13, 535-574
- [2] Nasereddin, M.; Mansooreh, M. (1999). The development of a methodology for the use of neural networks and simulation modeling in system design, *Proceedings of the 31<sup>st</sup> Winter Simulation Conference*, 537-542

- [3] Hornik, K.; Stinchcombe, M.; White, H. (1989). Multilayer feed forward networks are universal approximators. *Neural Networks*, Vol. 2, No. 5, 359-366, [doi:10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [4] Fonseca, D. J.; Navarrese, D. O.; Moynihan, G. P. (2003). Simulation metamodeling through artificial neural networks, *Engineering Applications of Artificial Intelligence*, Vol. 16, No. 3, 177-183, [doi:10.1016/S0952-1976\(03\)00043-5](https://doi.org/10.1016/S0952-1976(03)00043-5)
- [5] Badiru, A. B.; Sieger, D. B. (1998). Neural Networks as Simulation metamodels in economic analysis of risky projects, *European Journal of Operations Research*, Vol. 105, No. 1, 130-142, [doi:10.1016/S0377-2217\(97\)00029-5](https://doi.org/10.1016/S0377-2217(97)00029-5)
- [6] Rosen, S. L.; Harmonosky, C. M.; Trabant, M. T. (2008). Optimization of systems with multiple performance measures via simulation: Survey and recommendations, *Computers & Industrial Engineering*, Vol. 54, No. 2, 327-339, [doi:10.1016/j.cie.2007.07.004](https://doi.org/10.1016/j.cie.2007.07.004)
- [7] Teleb, R.; Azadivar, F. (1994). A methodology for solving multi-objective simulation-optimization problems, *European Journal of Operational research*, Vol. 72, No. 1, 135-145, [doi:10.1016/0377-2217\(94\)90336-0](https://doi.org/10.1016/0377-2217(94)90336-0)
- [8] Hachicha, W.; Ammeri, A.; Masmoudi, F.; Chabchoub, H. (2010). A multi-product lot size in make-to order supply chain using discrete event simulation and response surface methodology, *International Journal of Services, Economics and Management*, Vol. 2: No. 3-4, 246-266
- [9] Hachicha, W.; Ammeri, A.; Masmoudi, F.; Chabchoub, H. (2010). Case study for lot-sizing problem in MTO supply chain based on Simulation Optimization approach, *Proceedings of the 8<sup>th</sup> International Conference of Modelling and Simulation - MOSIM'10*, paper No. 23, [www.enim.fr/mosim2010/articles/23.pdf](http://www.enim.fr/mosim2010/articles/23.pdf)
- [10] Prieto, A. (1991). Artificial Neural Networks, *International Workshop, IWANN '91*, Granada, Spain, *Proceedings in Lecture Notes in Computer Science*, Springer
- [11] Tsoukalas, L. H.; Uhrig, R. E. (1997). *Fuzzy and neural approaches in engineering*, Wiley, New York
- [12] Haykin, S. (1999). *Neural Networks*, Prentice- Hall, New Jersey, NY
- [13] Hurrion, R. D.; Birgil, S. (1999). A comparison of factorial and random experimental design methods for the development of regression and neural network simulation metamodels, *Journal of the operational research society*, Vol. 50, No. 10, 1018-1033
- [14] Chryssolouris, G.; Domroese, M. (1990). The use of neural networks in determining operational policies for manufacturing systems, *Journal of Manufacturing Systems*, Vol. 10, No. 2, 166-175, [doi:10.1016/0278-6125\(91\)90018-W](https://doi.org/10.1016/0278-6125(91)90018-W)
- [15] Hurrion, R. D. (1992). Using a neural network to enhance the decision making quality of a visual interactive simulation model, *Journal of the Operational Research Society*, Vol. 43, No. 4, 333-341
- [16] Kilmer, R. (1994). *Artificial neural network metamodels*, PhD Dissertation, University of Pittsburgh
- [17] Kilmer, R.; Smith, A.; Shuman, L. (1999). Computing confidence intervals for stochastic simulation using neural network metamodels, *Computer & Industrial Engineering*, Vol. 36, No. 2, 391-407, [doi:10.1016/S0360-8352\(99\)00139-4](https://doi.org/10.1016/S0360-8352(99)00139-4)
- [18] Mohamed Sultan, M.; Shahrum Shah, A.; Osman David, C. (2009). Controllers optimization for a fluid mixing using metamodelling approach, *International Journal of Simulation Modelling*, Vol. 8, No. 1, 48-59, [doi:10.2507/IJSIMM08\(1\)5.117](https://doi.org/10.2507/IJSIMM08(1)5.117)
- [19] Gunasekaran, A.; Ngai, E. W. T. (2009). Modeling and analysis of build-to-order supply chains, *European Journal of Operational Research*, Vol. 195, No. 2, 319-334, [doi:10.1016/j.ejor.2008.03.026](https://doi.org/10.1016/j.ejor.2008.03.026)