WILEY | Hindawi

Research Article

Modeling of Throughput in Production Lines Using Response Surface Methodology and Artificial Neural Networks

Federico Nuñez-Piña, Joselito Medina-Marin [ID], Juan Carlos Seck-Tuoh-Mora, Norberto Hernandez-Romero, and Eva Selene Hernandez-Gress

Engineering Department, Autonomous University of Hidalgo State, Carr. Pachuca-Tulancingo, Col. Carboneras, 42184 Mineral de la Reforma, HGO, Mexico

Correspondence should be addressed to Joselito Medina-Marin; jmedina@uaeh.edu.mx

The problem of assigning buffers in a production line to obtain an optimum production rate is a combinatorial problem of type NP-Hard and it is known as Buffer Allocation Problem. It is of great importance for designers of production systems due to the costs involved in terms of space requirements. In this work, the relationship among the number of buffer slots, the number of work stations, and the production rate is studied. Response surface methodology and artificial neural network were used to develop predictive models to find optimal throughput values. 360 production rate values for different number of buffer slots and workstations were used to obtain a fourth-order mathematical model and four hidden layers' artificial neural network. Both models have a good performance in predicting the throughput, although the artificial neural network model shows a better fit ($R = 1.0000$) against the response surface methodology ($R = 0.9996$). Moreover, the artificial neural network produces better predictions for data not utilized in the models construction. Finally, this study can be used as a guide to forecast the maximum or near maximum throughput of production lines taking into account the buffer size and the number of machines in the line.

1. Introduction

The Buffer Allocation Problem (BAP) can be defined in three different ways by means of different objective functions as presented in [1]. These objective functions are expressed taking into account the maximum performance rate in a production line, the minimum buffer size to obtain a fixed throughput rate, and minimizing the average inventory of work in process. The mathematical models of these versions of the problem are given in the following paragraphs [2].

Problem BAP-A (The Dual Problem). In this problem we assume that there are K machines and $K-1$ storage areas with N total integer buffer slots to be allocated. Possible solutions have the form $n = (N_1, N_2, \ldots, N_{K-1})$. The objective is to maximize the throughput of the production line subject to a fixed quantity of buffer slots (N) distributed among the K machines. The problem may be denoted as follows:

$$\max \quad X(\mathbf{n}) = \max X(N_1, \ldots, N_{K-1})$$
$$\text{s.t.} \quad \sum_{i=1}^{K-1} N_i = N \quad N_i \geq 0, \ \forall i = 1, \ldots, K-1. \tag{1}$$

Problem BAP-B (The Primal Model). This problem is focused to minimize the total number of buffer slots to be allocated in between every machine K, given a minimum throughput X_0. The problem may be stated as follows:

$$\min \quad N = \sum_{i=1}^{K-1} N_i$$
$$\text{s.t.} \quad X(\mathbf{n}) = X(N_1, \ldots, N_{K-1}) \geq X_0 \tag{2}$$
$$N_i \geq 0, \ \forall i = 1, \ldots, K-1.$$

Problem BAP-C (Minimize the Average Work in Process). In this case, the problem consists in minimizing the average work in process (WIP), given a predefined minimum

throughput X_0. The problem may be stated mathematically as follows:

$$\min \quad L(\mathbf{n}) = \min L(N_1, \ldots, N_{K-1})$$

$$\text{s.t.} \quad X(\mathbf{n}) = X(N_1, \ldots, N_{K-1}) \geq X_0$$

$$\sum_{j=1}^{K-1} N_j \leq N \quad N_j \geq 0, \; j = 1, \ldots, K-1, \tag{3}$$

where $L(\mathbf{n}) = L(N_1, \ldots, N_{K-1})$ indicates the average of the WIP inventory, as a function of the buffer size vector, and the desired throughput X_0.

Metaheuristic methods are widely employed for solving combinatorial optimization problems, and these have been applied successfully in many types of problems of production lines where it is impossible to find optimum solutions by exact methods in a short amount of time when the problem size increases. Our objective is to study the relationship among the production rate, buffer size, and the number of machines in production lines. Then, we can propose the use of an RSM or ANN that models the behavior of the algorithms used to calculate the throughput of serial production line to obtain similar results in a shorter time, by knowing the number of workstations K and the total space of the buffer N.

The reminder of this paper is organized as follows. In Section 2, we present previous work related to this topic. In Section 3, we describe the basic concepts of the RSM and ANNs, the fitting techniques applied in this study. In Section 4, we show the steps in the development of both models and the numerical experiments performed. In Section 5, we give some results and compare the performance of the RSM and ANN obtained. Finally, Section 6 concludes this paper.

2. Related Work

The BAP has been studied for over 50 years and numerous articles have been published. The first study associated with this subject was done by Koenigsberg [3], presenting an analysis and a review of the problems associated with the effective functioning of production systems.

The solution methods of the BAP has fallen in two big groups: generative and evolutionary methods. The use of these methods is combined in a closed configuration. Generative methods are focused on the search of the optimal size of the temporal storage for a better performance of the system. The simpler algorithm uses a method that considers a complete enumeration of the line. However, it is only applicable for small systems since the total number of feasible solutions grows exponentially. Thus, for big systems it is impossible to look for through the space of solutions. In the last years, many search methods and metaheuristics have been highly adapted by investigators to solve the combinatorial problem of the size of temporary storage.

In [4], a two-stage heuristic algorithm is developed to solve the problem of minimization of the total temporal storage. However, this method cannot always find the optimal solution and it also does not converge in all cases. Seong et al.

adopt the concept of a pseudo gradient and the projection of this method, to figure out the problem of maximization of the production rate for a line of fabrication [5]. A year later, Seong et al. employed a method of gradient for the maximization of the production rate for a problem with exponential fabrication [6]. Other researchers also employed the gradient, such as Gershwin and Schor, who faced the problem of the temporal storage capacity, minimizing the total capacity, based on the observation of whether the production rates are expanded in a first order; this may be formulated as a problem of integer linear programming [7].

Search algorithms tend to solve the exponential explosion of the number of solution vectors. In this case, some algorithms apply a primal-dual approach to minimize the total temporal storage, subject to the restriction of the production speed. The main problem is to reduce the total size of temporal storage in a defined production rate, while the dual formulation is maximizing the throughput of the production line, subject to a limitation of total space of storage. Vouros and Papadopoulos studied the maximization of the benefits of the production line through a nonlinear method that is fast and precise [8]. Nevertheless, the limitation of the production rate is not taken into account. The authors based their research on a system called ASBA2 that contains a knowledge based system. This system determines the optimal plans of storage capacity, whose objective is to maximize the performance of the production lines. In order to validate the ASBA2 results, the authors executed an exact algorithm to calculate the production rate and compare them.

In [9], Nahas et al. utilized a local search heuristic to obtain the allocation plan for a given number of buffers slots. Aksoy and Gupta developed a near optimal buffer allocation plan (NOBAP) for a cellular remanufacturing system with a certain number of buffer slots [10]. The algorithm that Aksoy and Gupta proposed uses an open queueing network, and it is based on the decomposition principle and expansion methodology.

There are primarily two disadvantages of the traditional search methods. The first one is that the traditional search sometimes cannot jump over local optimal solutions in the search of the global optimal solution. The second disadvantage is that, with these approximate methods, it is difficult to observe small changes in the buffer size that will affect the system.

Metaheuristics are search methods using strategies that guide the search process and explore the search space. The aim is to find optimal solutions or almost optimal. The algorithms are approximated and are generally nondeterministic. The typical methods applied in this area include Tabu Search [11–13], Simulated Annealing [14], Genetic Algorithms [15], and Ant Colony Optimization [16]. In order to search for the best solution space, a recent tendency is to hybridize metaheuristics with other methods like nested partitions [17], the Branch and Bound method [18], and the local search [15]. These hybrid search methods have an advantage over the traditional ones because they can jump over local optimal solutions in the search of the global optimal. The main disadvantage of these methods is that they only can be applied to resolve specific problems.

On the other hand, dynamic programming [19, 20], artificial neural networks [21], genetic programming [22], and immune algorithm systems [23] have been applied with success in the solution of BAP in production lines. Moreover, some studies utilize diverse experimental designs for the evaluation of the solutions for the BAP [24–28].

There are two methods mainly used to evaluate the results produced by the strategies described above: analysis and simulation methods. The exact analytical method shows results based on queueing theory methods, which are difficult to obtain and are only available for line systems of short productions. The methods of approximate evaluation are decomposition, aggregation, and expansion. These are the most utilized methods to resolve BAP; in particular, decomposition method is used by many researchers [7, 16, 29]. The main idea of this method is to decompose the original module through the analysis of a conjunction of smaller subsystems that are much easier to solve. The main advantages of the decomposition method are the computational efficiency and the precision to find a solution. However, the disadvantage is that it can be applied only under the assumption that the production rates are deterministic or they follow an exponential distribution and the rates for machine failure and repair are geometrical or exponential randomly distributed.

The expansion method takes up the method of queueing theory. It is a method of general expansion that is used under certain assumptions and can be used for service times in general. This method was applied to solve series and merge and split topologies of production lines with finite buffers [30, 31].

The evaluation by the aggregation method has been also used to solve the BAP [18, 32]. It was applied to evaluate the performance of the buffer allocation in production lines. The idea of this method is to define two stations and a buffer as a subline; then it will be replaced for just one equivalent station. Next, this station is combined with a buffer and station of the original line to form again a subline of two stations and one buffer. This new subline is aggregated into a new equivalent station. This process is repeated until all the stations are added.

On the other hand, simulation provides many advantages for modeling realistic and complex systems instead of analytical methods. Nevertheless, the development of simulation models is a time consuming task. Simulation is more suitable to analyze problems of production lines at a detailed level, when mathematical analysis is not able to be applied to these kinds of problems.

Several research works [15, 21] can be cited as applications of simulation models to search the solution of BAP. The study of the transference lines without restrictions in the stations and finite temporal intermediate buffer was performed by Hillier and So in 1991 and Hillier et al. in 1993 [33, 34]. In these research papers, the stations have exponential time in the process; both articles employ a complete evaluation of all the buffers for minimizing the space and maximizing the throughput. The authors employed an exact method to determine the production rate. The difficulty with this approach is that exact solutions can be only applied to small systems.

ANNs have been applied to model the behavior of many systems with a good acceptance. In medicine, there is a study where ANNs where used to identify patients with hard risk for dying after suffering an acute myocardial infarction [35]. Furthermore, Pesko et al. present a comparative study between the use of ANNs and support vector machines, which are used to estimate costs and duration time of the construction of urban roads [36].

Tsadiras et al. propose the application of an ANN in the development of a decision support system, in order to assist production line designers in making decisions related to buffer distribution in reliable production lines. The proposed ANN contains one hidden layer with 10 hidden neurons [37].

In this research work, a response surface model and an artificial neural network are proposed to represent the relationship among the number of buffer slots, the number of workstations, and the throughput of the production line. Furthermore, the ANN models were created with different combination of neurons, and with 1, 2, 3, and 4 hidden layers, which provide a wide range of possibilities instead of only one hidden layer of neurons.

3. Data Fitting Techniques

In this paper, two techniques have been applied to obtain models representing the relationship among the throughput of a production line and the parameters N (number of buffer slots) and K (number of machines in the production line).

3.1. Response Surface Methodology. Response surface methodology (RSM) is a group of mathematical and statistical techniques used to describe the relationships between a response of interest, y, and a number of associated control (or input) variables denoted by $x_1, x_2, \ldots, x_k$ [38]. The most extensive applications of RSM are in cases where several input variables potentially influence some performance measure or quality characteristic of a process. Thus, performance measure or quality characteristic is called the response. The objectives of RSM include the determination of variable settings, for which the mean response is optimized, and the estimation of the response surface. In general, such a relationship is unknown but can be approximated by a low degree polynomial model of the form

$$y = f'(x)\beta + \epsilon, \tag{4}$$

where $x = (x_1, x_2, \ldots, x_k)'$, $f(x)$ is a vector function of k elements, β is a vector of unknown constant coefficients referred to as parameters, and ϵ is a random experimental error assumed to have a zero mean. RSM has been widely applied in optimizing various processes in environmental studies for modeling and analysis of water and wastewater treatment processes [39].

Some stages in the application of RSM are [40] (1) the selection of independent variables of major effects on the system through screening studies; (2) the choice of the experimental design and carrying out the experiments according to the selected experimental matrix; (3) the mathematic and statistical treatment of the obtained experimental data through the fit of a polynomial function; (4) the evaluation

Representation of throughput of production lines
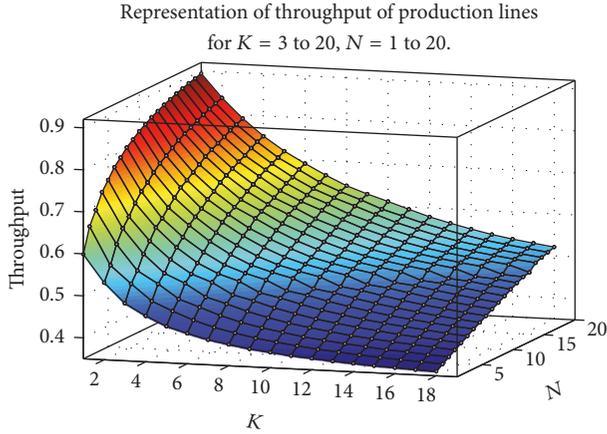for $K = 3$ to 20, $N = 1$ to 20.

FIGURE 1: Numerical experiments used for training and testing the ANN.

of the model and fitness; (5) the verification of the necessity and possibility of performing a displacement in direction to the optimal region; (6) obtaining the optimum values for each studied variable.

3.2. Artificial Neural Network. An artificial neural network (ANN) consists of a set of processing units, also called neurons, that are connected with each other. It can be described as a directed graph, and each neuron is a transfer function. A neuron is generally a nonlinear element of multiple inputs and a single output. The architecture of a neural network is determined by all the connections in the network and transfer functions of the neurons [41]

The backpropagation algorithm proposed by [42] is the most popular algorithm to train ANNs. Moreover, advanced methods like Marquardt [43–45], Quasi-Newton [46], or conjugating gradient algorithms [47, 48] are also very popular. Due to their application in dynamic environments, these classic learning methods have to be modified to fulfill three important requirements:

   (i) The capacity to work in online mode

   (ii) The capacity to adjust its control parameters

   (iii) The capacity to adapt its structures, all of them in accordance with the learning process.

4. Numerical Experiments

In this work, two sets of numerical experiments were considered. The first one consists of 360 experiments taken from [2], where $K = 3, 4, \ldots, 20$ and $N = 1, 2, \ldots, 20$ (Figure 1); it was considered for the development of the RSM and ANN models. The second one consists of 55 experiments (Table 1); these data were utilized for the validation process of the obtained models.

In the latter case, the experiments are classified in three categories: small, medium, and large lines (Table 2). In all cases, reliable exponential and balanced lines were considered with equal mean service rate $\mu_i = \mu = 1$

TABLE 1: Sets of numerical experiments used to validate the ANN.

ID	Problem size	K	N
(1)	Small	11	22
(2)	Medium	21	10
(3)	Medium	22	11
(4)	Medium	23	11
(5)	Medium	24	12
(6)	Medium	25	12
(7)	Medium	26	13
(8)	Medium	27	13
(9)	Medium	28	14
(10)	Medium	29	14
(11)	Medium	30	15
(12)	Medium	20	21
(13)	Medium	21	22
(14)	Medium	22	23
(15)	Medium	23	24
(16)	Medium	24	25
(17)	Medium	25	26
(18)	Medium	26	27
(19)	Medium	27	28
(20)	Medium	28	29
(21)	Medium	29	30
(22)	Medium	30	31
(23)	Medium	12	24
(24)	Medium	13	26
(25)	Medium	14	28
(26)	Medium	15	30
(27)	Medium	16	32
(28)	Medium	17	34
(29)	Medium	18	36
(30)	Medium	19	38
(31)	Medium	20	40
(32)	Medium	21	42
(33)	Medium	22	44
(34)	Medium	23	46
(35)	Medium	24	48
(36)	Medium	25	50
(37)	Medium	26	52
(38)	Medium	27	54
(39)	Medium	28	56
(40)	Medium	29	58
(41)	Medium	30	60
(42)	Large	40	41
(43)	Large	50	51
(44)	Large	60	61
(45)	Large	70	71
(46)	Large	80	81
(47)	Large	90	91
(48)	Large	100	101
(49)	Large	40	80
(50)	Large	50	100
(51)	Large	60	120
(52)	Large	70	140
(53)	Large	80	160
(54)	Large	90	180
(55)	Large	100	200

4.1. Statistical Analysis for Response Surface Methodology. As we mentioned above, the first dataset of numerical experiments composed of 360 data was employed for the

TABLE 2: Classification of lines.

Small lines	$K = 3, 4, \ldots, 11$
Medium lines	$K = 12, 13, \ldots, 30$
Large lines	$K = 40, 50, \ldots, 100$

construction of the RSM model. Two numerical factors, K (number of machines) and N (buffer size), designated as x and y, respectively, were considered in the experimental design. In addition, a fourth-order mathematical model was chosen for modeling the relationship among K, N, and the throughput, which is sufficiently complex to approximate the main features of the system. The data were analyzed using multiple regressions through the least squares method to fit the following:

$$\begin{aligned} \mathrm{Th}\,(x, y) = {} & \theta_0 + \theta_1 x + \theta_2 y + \theta_3 x^2 + \theta_4 xy + \theta_5 y^2 \\ & + \theta_6 x^3 + \theta_7 x^2 y + \theta_8 xy^2 + \theta_9 y^3 + \theta_{10} x^4 \\ & + \theta_{11} x^3 y + \theta_{12} x^2 y^2 + \theta_{13} xy^3, \end{aligned} \quad (5)$$

where Th is the throughput response; θ_0 is the intercept; $\theta_1, \ldots, \theta_{13}$ are the linear coefficients, and x, y are the coded independent variables.

For this task, a specialized Matlab toolbox was used to carry out the regression analysis.

4.2. Artificial Neural Network Development. A Matlab script was used to generate the ANN models using the Matlab Neural Network toolbox, starting with a three-layer model, where the first layer (input layer) contains two neurons (K and N). The number of hidden layers was varying from 1 to 4, and the output layer, composed of one neuron, represents the throughput of the production line.

Each hidden layer was composed of 5, 8, 10, 12, or 15 neurons. Thus, a total of 780 ANN configurations were created, and for each of them 50 iterations were executed in order to find the best model for every configuration. After the 780 best ANNs were generated and saved in a.mat file, another Matlab script was implemented to find the ANN with the lowest error and a coefficient of correlation near to 1.

The obtained ANNs were trained with a Levenberg-Marquardt backpropagation algorithm [49].

The design of the ANN is as follows:

(i) 2 input variables, K and N

(ii) One output variable, *Throughput*, which is the maximum or near maximum performance of the production line

(iii) The ANNs containing 1 to 4 hidden layers, which employ the hyperbolic tangent sigmoid transfer function as an activation function in every hidden layer:

$$f(x) = \frac{1}{1 + e^{-ax}}, \quad (6)$$

where a is a slope parameter

(iv) The ANNs contains an output layer of one neuron; it employs the linear transfer function.

$$f(x) = ax, \quad (7)$$

where a is a slope parameter.

All the experiments were run in a desktop PC with the following specifications: Intel(R) Core(TM) i5-2450 M CPU @2.50 GHz, 4.00 GB RAM.

4.3. Validation. The mean squared error (MSE) and the coefficient of correlation (R) were used as indicators to measure the performance of the RSM obtained and every ANN generated. These errors are defined as follows.

(a) The Mean Squared Error (MSE) [50]

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(\mathrm{Target}_i - \mathrm{Output}_i \right)^2, \quad (8)$$

where n is the number of the input data to the ANN and RSM, Target_i is the target output value of the ANN and RSM for Input_i, and Output_i is the predicted value of the ANN and RSM to Input_i.

(b) The Coefficient of Correlation [50]. It measures the correlation between the Output and Target values. It is defined as

$$R = \frac{\mathrm{Cov}\,(\mathrm{Target}, \mathrm{Output})}{\sigma_{\mathrm{Target}} \sigma_{\mathrm{Output}}} \quad -1 \leq R \leq 1, \quad (9)$$

where $\mathrm{Cov}(\mathrm{Target}, \mathrm{Output})$ is the covariance between the values of Target and the ANN and RSM Output. σ_{Target} and σ_{Output} are the standard deviations of Target and Output, respectively. The maximum value of $R = 1$ is reached when the linear relationship is perfect between the target and output values, whereas when $R = 0$ it means that a linear correlation does not exist between the output and target values.

5. Results

In the present study, the data utilized to develop the RSM and ANN models were gathered from a total of 360 experiments, varying the values of K and N.

The regression model in terms of the actual values that described the throughput is presented in (10). The results indicated that this quadratic model can be used to navigate the design space. The value of R for the model was 0.999603176, and the MSE = 1.22796×10^{-5}.

$$\begin{aligned} \mathrm{Th}\,(x, y) = {} & 0.8661 - 0.1342x + 0.04927y \\ & + 0.01331x^2 - 0.001985xy \\ & - 0.002671y^2 - 0.0005775x^3 - 8.568 \\ & \times 10^{-5}x^2 y + 0.0002081xy^2 + 4.734 \\ & \times 10^{-5}y^3 + 9.157 \times 10^{-6}x^4 + 4.064 \end{aligned}$$
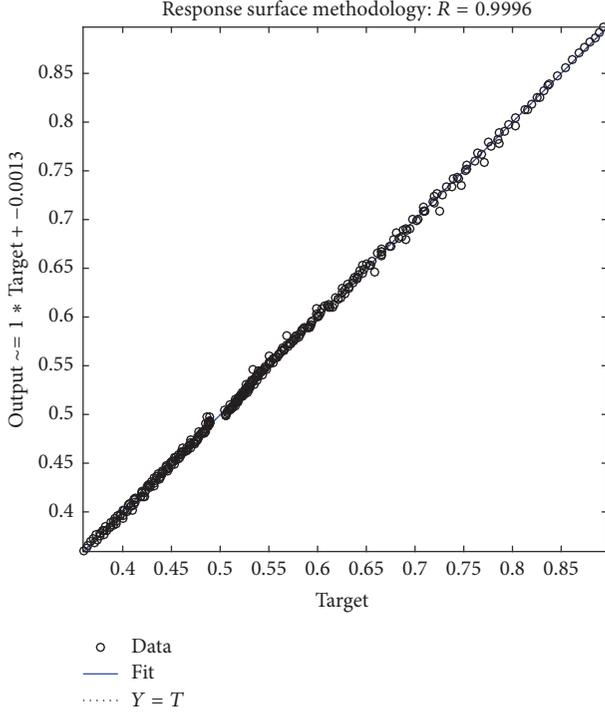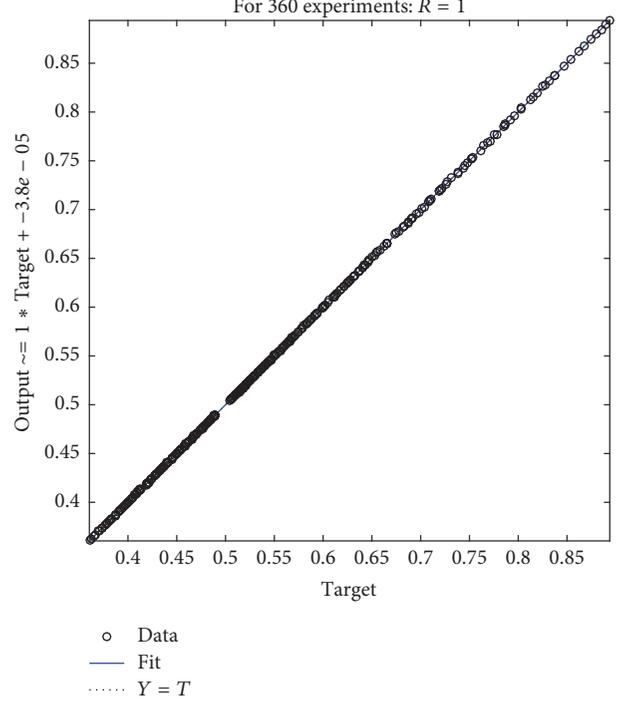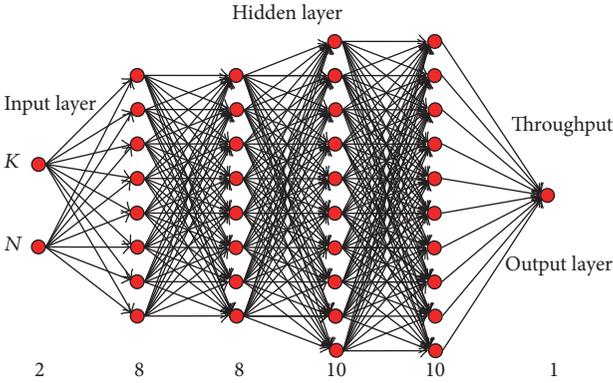
Figure 2: Performance indicator R for the RSM model.

Figure 4: Performance indicator R for the ANN model.

Figure 3: ANN with the best performance.

$$\times 10^{-6}x^3 y - 2.961 \times 10^{-6}x^2 y^2 - 3.058$$

$$\times 10^{-6}xy^3,$$

(10)

where Th is the throughput, x is the number of machines K, and y is the buffer size N.

The prediction of the model was validated for a correlation between experimental data and predicted throughput (Figure 2).

On the other hand, the ANN with the minimum error was one composed of four hidden layers and their corresponding neurons ($8:8:10:10$). It is shown in Figure 3.

Figure 4 shows the correlation between experimental values and predicted throughput in production lines for training dataset, validation dataset, and test dataset.

Most of the data are on the bisector or in its vicinity which represents a proper correlation between experimental data and predicted outputs. Figure 4 indicates the closeness among the experimental data and the predicted results using the ANN. The maximum error and MSE are 0.34% and 1.150×10^{-7}, respectively.

Furthermore, a second dataset was used to prove the effectiveness of the RSM and ANN to forecast the throughput of a production line from the K and N values. Table 3 shows experimental and predicted values, as well as the errors of the RSM and ANN models in the throughput forecasting. Indeed, experimental data and predicted outputs obtained with the ANN model have a good fit with each other. On the contrary, the RSM model cannot predict adequately the numerical experiments of the second dataset.

Values in both models were near those of the experimental data of the first dataset. But, the ANN showed a better performance with the second dataset (Table 4).

In the case of the ANN model, the correlation between experimental data and predicted throughput is illustrated in Figure 5. It can be observed that there is an appropriate correlation between experimental values and predicted data. The maximum error and MSE obtained from this dataset are 2.76% and 3.57×10^{-5}, respectively.

Figure 6 shows that the ANN is able to predict small lines ($K = 3, 4, \ldots, 11$) and medium lines ($k = 12, 13, \ldots, 30$) accurately. Moreover, errors in predicting the throughput in BAP for large lines ($K = 40, 50, \ldots, 100$) are small. However, the graphic of the RSM model is quite different to the graphic of the experimental values.

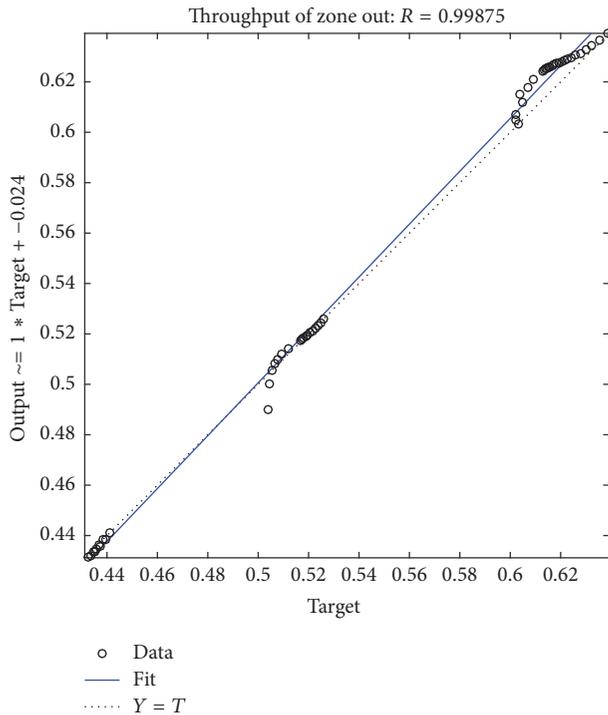TABLE 3: RSM and ANN errors for the 55 data samples.

ID	K	N	Th from experiments	Th from ANN	Th from RSM	% error ANN	% error RSM
(1)	11	22	0.63892	0.639236	0.647968	0.05	1.42
(2)	21	10	0.43869	0.438564	0.447065	0.03	1.91
(3)	22	11	0.44135	0.441200	0.463878	0.03	5.10
(4)	23	11	0.43656	0.436088	0.477474	0.11	9.37
(5)	24	12	0.43927	0.438463	0.512036	0.18	16.57
(6)	25	12	0.43494	0.433669	0.546250	0.29	25.59
(7)	26	13	0.43739	0.435921	0.608091	0.34	39.03
(8)	27	13	0.43345	0.432226	0.672812	0.28	55.22
(9)	28	14	0.43586	0.434505	0.773369	0.31	77.44
(10)	29	14	0.43234	0.431178	0.880447	0.27	103.65
(11)	30	15	0.43447	0.433623	1.033058	0.19	137.77
(12)	20	21	0.52608	0.525900	0.522454	0.03	0.69
(13)	21	22	0.52465	0.524538	0.524495	0.02	0.03
(14)	22	23	0.52361	0.523318	0.530127	0.06	1.24
(15)	23	24	0.52247	0.522231	0.540554	0.05	3.46
(16)	24	25	0.52146	0.521265	0.557152	0.04	6.84
(17)	25	26	0.52051	0.520410	0.581470	0.02	11.71
(18)	26	27	0.51962	0.519653	0.615230	0.01	18.40
(19)	27	28	0.51892	0.518983	0.660328	0.01	27.25
(20)	28	29	0.51815	0.518388	0.718831	0.05	38.73
(21)	29	30	0.51747	0.517859	0.792980	0.08	53.24
(22)	30	31	0.5168	0.517385	0.885189	0.11	71.28
(23)	12	24	0.63556	0.636344	0.648968	0.12	2.11
(24)	13	26	0.6322	0.634189	0.647653	0.31	2.44
(25)	14	28	0.63005	0.632574	0.641090	0.4	1.75
(26)	15	30	0.62789	0.631348	0.625888	0.55	0.32
(27)	16	32	0.626	0.630397	0.598202	0.7	4.44
(28)	17	34	0.62428	0.629639	0.553727	0.86	11.30
(29)	18	36	0.62277	0.629014	0.487706	1	21.69
(30)	19	38	0.62159	0.628481	0.394921	1.11	36.47
(31)	20	40	0.62033	0.628011	0.269700	1.24	56.52
(32)	21	42	0.61926	0.627583	0.105914	1.34	82.90
(33)	22	44	0.6183	0.627184	−0.103023	1.44	116.66
(34)	23	46	0.61743	0.626804	−0.364154	1.52	158.98
(35)	24	48	0.61666	0.626437	−0.684977	1.59	211.08
(36)	25	50	0.61583	0.626078	−1.073447	1.66	274.31
(37)	26	52	0.61517	0.625725	−1.537977	1.72	350.01
(38)	27	54	0.61458	0.625376	−2.087435	1.76	439.65
(39)	28	56	0.61399	0.625029	−2.731146	1.8	544.82
(40)	29	58	0.61345	0.624684	−3.478891	1.83	667.10
(41)	30	60	0.61288	0.624339	−4.340910	1.87	808.28
(42)	40	41	0.51223	0.514290	3.455109	0.4	574.52
(43)	50	51	0.50949	0.512086	11.769262	0.51	2210.01
(44)	60	61	0.50768	0.510035	31.093176	0.46	6024.56
(45)	70	71	0.50639	0.508186	68.420859	0.35	13411.49
(46)	80	81	0.50544	0.505767	132.474799	0.06	26109.80
(47)	90	91	0.5047	0.500121	233.705964	0.91	46205.92
(48)	100	101	0.50412	0.490193	384.293803	2.76	76130.62
(49)	40	80	0.60928	0.620941	−21.824940	1.91	3682.09
(50)	50	100	0.60714	0.617804	−65.388150	1.76	10869.86
(51)	60	120	0.6037	0.615156	−152.801820	1.9	25410.89

TABLE 3: Continued.

ID	K	N	Th from experiments	Th from ANN	Th from RSM	% error ANN	% error RSM
(52)	70	140	0.60475	0.611799	−306.402750	1.17	50766.02
(53)	80	160	0.60244	0.606839	−553.093260	0.73	91908.85
(54)	90	180	0.60205	0.604855	−924.341190	0.47	153632.30
(55)	100	200	0.60302	0.603219	−1456.179900	0.03	241581.19

TABLE 4: Comparison of errors from RSM and ANN models.

	ANN	RSM
Test set for 360 data samples		
MSE	$1.1505E − 07$	$1.22796E − 05$
R	1	0.999603176
Maximum error	0.34%	2.36%
Test set for the 55 data samples		
MSE	$3.57115E − 05$	66032.89926
R	0.99875	−0.225360965
Maximum error	2.70%	241581.1947%

FIGURE 6: Comparison of RSM and ANN models.

FIGURE 5: R value for predicted throughput with the ANN on the second dataset.

6. Conclusions and Further Research

The Buffer Allocation Problem is a combinatorial problem that requires a high amount of computational time for medium and large lines. It is necessary to have a model that could predict the production line throughput in a short time.
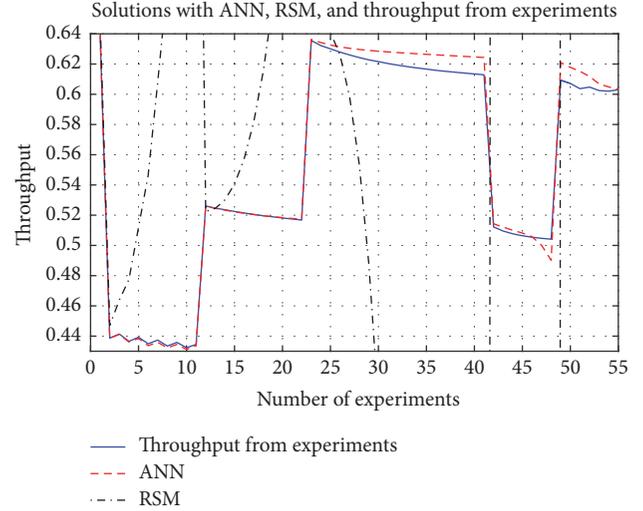
In this work, the relationship among the number of buffer slots (N), the number of workstations (K), and the throughput in production lines was accurately modeled with RSM and ANNs. In order to study this relationship, a total of 360 experimental data were used in the construction of both models.

A fourth-order mathematical model was obtained by applying the RSM, with a coefficient of correlation $R = 0.9996$. On the other hand, 780 ANNs models were created in order to obtain the ANN with a coefficient of correlation near to 1, with models from 1 to 4 hidden layers and 5, 8, 10, 12, or 15 neurons for each layer. The ANN with the best performance has 4 hidden layers, with 8, 8, 10, and 10 neurons in each hidden layer, respectively, obtaining a coefficient of correlation $R = 1$.

Both models have a good performance with the initial 360 experimental data; however, for a second dataset not considered in the model creation, composed of 55 experimental data, the ANN shows a higher performance ($R = 0.99875$) than the equation obtained by the RSM ($R = −0.22536$).

These results show that the ANN model provides a good fit and it can represent accurately the behavior of the throughput in production lines with different sizes, even for large lines.

As further work, this ANN model can be used to find optimal or near optimal values of the throughput for minimal number of buffer slots between each machine in the system, in order to minimize the total buffer size.
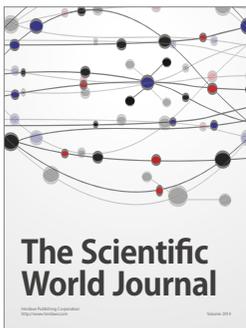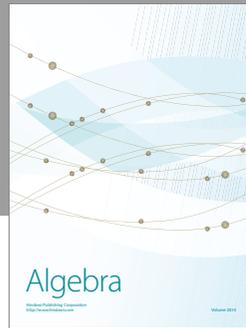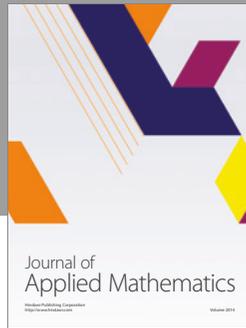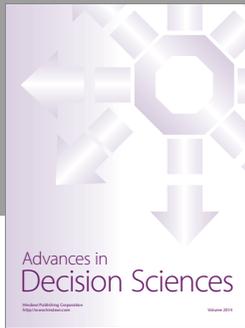
Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Demir, S. Tunali, and D. T. Eliiyi, "The state of the art on buffer allocation problem: A comprehensive survey," *Journal of Intelligent Manufacturing*, vol. 25, no. 3, pp. 371–392, 2014.

[2] C. T. Papadopoulos, M. E. J. Okelly, and A. K. Tsadiras, "A DSS for the buffer allocation of production lines based on a comparative evaluation of a set of search algorithms," *International Journal of Production Research*, vol. 51, no. 14, pp. 4175–4199, 2013.

[3] E. Koenigsberg, "Production lines and internal storage—A review," *Management Science*, vol. 5, no. 4, pp. 410–433, 1959.

[4] T. Park, "A two-phase heuristic algorithm for determining buffer sizes of production lines," *International Journal of Production Research*, vol. 31, no. 3, pp. 613–631, 1993.

[5] D. Seong, S. Y. Chang, and Y. Hong, "An algorithm for buffer allocation with linear resource constraints in a continuous-flow unreliable production line," *Asia-Pacific Journal of Operational Research*, vol. 17, no. 2, pp. 169–180, 2000.

[6] D. Seong, S. Y. Changs, and Y. Hongs, "Heuristic algorithms for buffer allocation in a production line with unreliable machines," *International Journal of Production Research*, vol. 33, no. 7, pp. 1989–2005, 1995.

[7] S. B. Gershwin and J. E. Schor, "Efficient algorithms for buffer space allocation," *Annals of Operations Research*, vol. 93, pp. 117–144, 2000.

[8] G. A. Vouros and H. T. Papadopoulos, "Buffer allocation in unreliable production lines using a knowledge based system," *Computers & Operations Research*, vol. 25, no. 12, pp. 1055–1067, 1998.

[9] N. Nahas, D. Ait-Kadi, and M. Nourelfath, "A new approach for buffer allocation in unreliable production lines," *International Journal of Production Economics*, vol. 103, no. 2, pp. 873–881, 2006.

[10] H. K. Aksoy and S. M. Gupta, "Near optimal buffer allocation in remanufacturing systems with n-policy," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 496–508, 2010.

[11] L. Demir, S. Tunali, and D. T. Eliiyi, "An adaptive tabu search approach for buffer allocation problem in unreliable production lines," in *Proceedings of the 24th Mini EURO Conference on Continuous Optimization and Information-Based Technologies in the Financial Sector, MEC EurOPT 2010*, pp. 207–212, tur, June 2010.

[12] L. Demir, S. Tunali, and A. Lkketangen, "A tabu search approach for buffer allocation in production lines with unreliable machines," *Engineering Optimization*, vol. 43, no. 2, pp. 213–231, 2011.

[13] L. Demir, S. Tunal, and D. T. Eliiyi, "An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines," *Computers & Operations Research*, vol. 39, no. 7, pp. 1477–1486, 2012.

[14] D. D. Spinellis and C. T. Papadopoulos, "A simulated annealing approach for buffer allocation in reliable production lines," *Annals of Operations Research*, vol. 93, pp. 373–384, 2000.

[15] M. Amiri and A. Mohtashami, "Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 62, no. 1-4, pp. 371–383, 2012.

[16] M. Nourelfath, N. Nahas, and D. Ait-Kadi, "Optimal design of series production lines with unreliable machines and finite buffers," *Journal of Quality in Maintenance Engineering*, vol. 11, no. 2, pp. 121–138, 2005.

[17] L. Shi and S. Men, "Optimal buffer allocation in production lines," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 35, no. 1, pp. 1–10, 2003.

[18] A. Dolgui, A. V. Eremeev, and V. S. Sigaev, "HBBA: Hybrid algorithm for buffer allocation in tandem production lines," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 411–420, 2007.

[19] A. C. Diamantidis and C. T. Papadopoulos, "A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines," *Mathematical Problems in Engineering*, no. 3, pp. 209–223, 2004.

[20] H. Yamashita and T. Altiok, "Buffer capacity allocation for a desired throughput in production lines," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 30, no. 10, pp. 883–891, 1998.

[21] F. Altiparmak, B. Dengiz, and A. A. Bulgak, "Buffer allocation and performance modeling in asynchronous assembly system operations: An artificial neural network metamodeling approach," *Applied Soft Computing*, vol. 7, no. 3, pp. 946–956, 2007.

[22] B. Can and C. Heavey, "Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 447–462, 2011.

[23] Y. Massim, F. Yalaoui, L. Amodeo, E. Chatelet, and A. Zeblah, "Efficient combined immune-decomposition algorithm for optimal buffer allocation in production lines for throughput and profit maximization," *Computers & Operations Research*, vol. 37, no. 4, pp. 611–620, 2010.

[24] I. Sabuncuoglu, E. Erel, and Y. Gocgun, "Analysis of serial production lines: Characterisation study and a new heuristic procedure for optimal buffer allocation," *International Journal of Production Research*, vol. 44, no. 13, pp. 2499–2523, 2006.

[25] I. Sabuncuoglu, E. Erel, and A. G. Kok, "Analysis of assembly systems for interdeparture time variability and throughput," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 34, no. 1, pp. 23–40, 2002.

[26] K.-C. Jeong and Y.-D. Kim, "Heuristics for selecting machines and determining buffer capacities in assembly systems," *Computers & Industrial Engineering*, vol. 38, no. 3, pp. 341–360, 2000.

[27] B. Can and C. Heavey, "A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models," *Computers & Operations Research*, vol. 39, no. 2, pp. 424–436, 2012.

[28] T. McNamara, S. Hudson, and S. Shaaban, "Unpaced production lines with three simultaneous imbalance sources," *Industrial Management & Data Systems*, vol. 111, no. 9, pp. 1356–1380, 2011.

[29] S. Helber, "Cash-flow-oriented buffer allocation in stochastic flow lines," *International Journal of Production Research*, vol. 39, no. 14, pp. 3061–3083, 2001.

[30] J. M. Smith and F. R. B. Cruz, "The buffer allocation problem for general finite buffer queueing networks," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 37, no. 4, pp. 343–365, 2005.

[31] J. M. Smith, F. R. Cruz, and T. van Woensel, "Topological network design of general, finite, multi-server queueing networks,"

European Journal of Operational Research, vol. 201, no. 2, pp. 427–441, 2010.

[32] H. Yamamoto, J. Abu Qudeiri, and E. Marui, "Definition of FTL with bypass lines and its simulator for buffer size decision," *International Journal of Production Economics*, vol. 112, no. 1, pp. 18–25, 2008.

[33] F. S. Hillier and K. C. So, "The effect of machine breakdowns and interstage storage on the performance of production line systems," *International Journal of Production Research*, vol. 29, no. 10, pp. 2043–2055, 1991.

[34] F. S. Hillier, K. C. So, and R. W. Boling, "Notes. Toward characterizing the optimal allocation of storage space in production line systems with variable processing times," *Management Science*, vol. 39, no. 1, pp. 126–133, 1993.

[35] T. Gligorijevic, Z. Sevarac, B. Milovanovic et al., "Follow-up and risk assessment in patients with myocardial infarction using artificial neural networks," *Complexity*, vol. 2017, Article ID 8953083, 8 pages, 2017.

[36] I. Pesko, V. Mucenski, M. Seslija et al., "Estimation of costs and durations of construction of urban roads using ann and svm," *Complexity*, vol. 2017, Article ID 2450370, 13 pages, 2017.

[37] A. Tsadiras, C. Papadopoulos, and M. O'Kelly, "An artificial neural network based decision support system for solving the buffer allocation problem in reliable production lines," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 1150–1162, 2013.

[38] A. I. Khuri and S. Mukhopadhyay, "Response surface methodology," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 2, pp. 128–149, 2010.

[39] A. T. Nair, A. R. Makwana, and M. M. Ahammed, "The use of response surface methodology for modelling and analysis of water and wastewater treatment processes: A review," *Water Science and Technology*, vol. 69, no. 3, pp. 464–478, 2014.

[40] M. A. Bezerra, R. E. Santelli, E. P. Oliveira, L. S. Villar, and L. A. Escaleira, "Response surface methodology (RSM) as a tool for optimization in analytical chemistry," *Talanta*, vol. 76, no. 5, pp. 965–977, 2008.

[41] S. F. Ding, H. Li, C. Y. Su, J. Z. Yu, and F. X. Jin, "Evolutionary artificial neural networks: a review," *Artificial Intelligence Review*, vol. 39, no. 3, pp. 251–260, 2013.

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[43] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 6, pp. 989–993, 1994.

[44] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[45] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[46] A. Likas and A. Stafylopatis, "Training the random neural network using quasi-Newton methods," *European Journal of Operational Research*, vol. 126, no. 2, pp. 331–339, 2000.

[47] L. M. Saini and M. K. Soni, "Artificial neural network-based peak load forecasting using conjugate gradient methods," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 907–912, 2002.

[48] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[49] L. M. Saini and M. K. Soni, "Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods," *IEE Proceedings Generation, Transmission and Distribution*, vol. 149, no. 5, pp. 578–584, 2002.

[50] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, Thomson/Brooks-Cole, USA, 2007.