International Journal of Production Research, 2014
Vol. 52, No. 19, 5597–5610, http://dx.doi.org/10.1080/00207543.2014.887232

A bi-objective stochastic programming model for optimising automated material handling systems with reliability considerations

Madjid Tavana[a,b]*, Hamed Fazlollahtabar[c] and Reza Hassanzadeh[d]

[a]*Business Systems and Analytics Department, Lindback Distinguished Chair of Information Systems and Decision Sciences, La Salle University, Philadelphia, PA, USA;* [b]*Faculty of Business Administration and Economics, Business Information Systems Department, University of Paderborn, Paderborn, Germany;* [c]*Faculty of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran;* [d]*Departement of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran*

(*Received 14 August 2013; accepted 18 January 2014*)

The optimisation of material handling systems (MHSs) can lead to substantial cost reductions in manufacturing systems. Choosing adequate and relevant performance measures is critical in accurately evaluating MHSs. The majority of performance measures used in MHSs are time-based. However, moving materials within a manufacturing system utilise time and cost. In this study, we consider both time and cost measures in an optimisation model used to evaluate an MHS with automated guided vehicles. We take into account the reliability of the MHSs because of the need for steadiness and stability in the automated manufacturing systems. Reliability is included in the model as a cost function. Furthermore, we consider bi-objective stochastic programming to optimise the time and cost objectives because of the uncertainties inherent in the optimisation parameters in real-world problems. We use perceptron neural networks to transform the bi-objective optimisation model into a single objective model. We use numerical experiments to demonstrate the applicability of the proposed model and exhibit the efficacy of the procedures and algorithms.

Keywords: material handling system; stochastic programming; automated guided vehicle; reliability; perceptron neural network

1. Introduction

The material handling system (MHS) in a manufacturing setting plays an important role in the performance of the entire system. Inadequately designed MHSs can interfere with the overall performance of the manufacturing system and lead to substantial losses in productivity and competitiveness, and to unacceptably long lead times. Among the advanced technologies available for MHSs, automated guided vehicles (AGVs) have found increasing applications because of their capability to transport a variety of part-types from point to point without human intervention.

Today's automated MHSs are technologically advanced and increasingly complex. Uncertainty is an inevitable consequence of the complexities generated by technological advancements. Jain et al. (2013) show that most of the automated manufacturing studies have used single-item measures. They argue that single-item measures are appropriate for relatively simple manufacturing systems. As the different dimensions of automated MHSs are complex in nature, single-item measures are inappropriate. Choosing adequate and relevant performance measures is critical in accurately analysing MHSs (Beamon 1998). A company also needs to address uncertainties in the manufacturing system to survive and compete in such an uncertain environment (Jain et al. 2013).

AGVs are the most flexible means for transporting pieces among workstations in an automated manufacturing system. An AGV is a driverless and programmed vehicle used to transfer the load from one part of a manufacturing facility to another part (Maniya and Bhatt 2011). Maxwell and Muckstadt (1982) first recognised the importance of AGV-based MHS design. They developed an optimisation model that minimised the total travel time and determined the maximum number of AGVs needed to efficiently transfer material from one shop to another. AGVs increase efficiency, reduce costs and improve flexibility by automating an MHS. Time-based performance measures are often used to evaluate MHSs with AGVs. However, moving materials from one part of the manufacturing floor to another part utilise time and incur costs. In this study, we consider both time and cost measures in an optimisation model and evaluate an MHS with AGVs.

*Corresponding author. Email: tavana@lasalle.edu

The remainder of this paper is organised as follows. In Section 2 we review the relevant literature on MHSs, AGVs, stochastic programming, reliability and perceptron neural networks. In Section 3 we introduce the problem and our assumptions. In Section 4 we present the mathematical representation of the problem and our proposed model. In Section 5 we present a numerical experiment to illustrate the effectiveness and applicability of the proposed model. In Section 6 we present some extended results for the numerical experiment. Finally, we conclude in Section 7 with our conclusions and future research directions.

2. Literature review

The use of AGVs increases flexibility and has a significant impact on the overall performance and reliability of MHSs (Sarker and Gurav 2005). As AGVs become larger and more complex, the traditional design requires more attention to issues such as control, cost, time, reliability, flexibility, etc. A number of different MHS design and evaluation methods (e.g. simulation, optimisation and the genetic algorithm) have been proposed in the literature. Simulation is an acceptable method for analysing manufacturing systems. However, simulation is often challenging and time-consuming (Law and Kelton 2000; Kuo et al. 2007), particularly, when it is used for modelling complex manufacturing systems such as MHSs with AGVs.

The problem of scheduling AGVs in an automated MHS has been studied extensively. Abdelmaguid et al. (2004) addressed the problem of simultaneous scheduling of machines and AGVs with the objective of minimising the makespan. This problem is composed of two interrelated decision problems: the scheduling of machines and the scheduling of AGVs. They showed that each problem is an NP-complete problem and a simultaneous consideration of the two problems results in a more complicated NP-complete problem. They proposed a hybrid genetic-algorithm/heuristic coding scheme to solve the problem. Deroussi, Gourgand, and Tchernev (2008) also studied this problem and proposed a solution based on vehicles rather than machines. Each solution was evaluated using a discrete event approach. Gnanavel Babu et al. (2010) studied this problem further and proposed a meta-heuristic differential evolution algorithm for solving it. They introduced an iterative algorithm that anticipated the complete set of flow requirements for a given machine schedule and made vehicle assignments accordingly. Le-Anh and De Koster (2006) have compiled a comprehensive review of the AGV design and control models and methods in the literature.

Farling, Mosier, and Mahmoodi (2001) used a simulation model to compare the performance of three AGV configurations under a variety of experimental conditions. They showed that system size, load/unload time and machine failure rate factors have significant impacts on the operation and reliability of MHSs. Smith (1993) defined reliability as the probability that an item will perform a required function, under stated conditions, for a specific period of time. A reliability measure is a metric for quantifying this probability.

A number of different reliability measures (i.e. availability, unavailability, failure rate and mean time between failures) have been proposed in the literature. For degradable systems, such as MHSs, the performance of the system during a specific period of time can be described by different levels of performance as a function of machine failures (Beamon 1998). Miriyala and Viswanadham (1989) developed several measures and algorithms for evaluating part-based reliability and system-based reliability for automated MHSs. Beamon (1995) proposed an analytical model for designing guide paths for automated MHSs as a function of reliability and quantified the reliability of the handling components.

In order to ensure an acceptable service level for each machine in each shop, we adopt and further extend the concept of reliability proposed by Ball and Lin (1993) in the model. We define reliability as the probability that the system is operational until time t. A failure is when a machine in a shop breaks down. A desired level of reliability can be achieved by limiting the failure probabilities. This approach for handling reliability is called the chance constraints method and was initially proposed by Charnes and Cooper (1959) in the context of mathematical programming. The use of chance constraints in the vehicle routing problem was illustrated in Stewart and Golden (1983). Carbone (1974) used chance constraints for selecting multiple facilities under normally distributed demand. The model minimised an upper bound on the total demand distance while ensuring that the constraints were satisfied with a specified chance or probability. Shiode and Drezner (2003) used a similar approach in a competitive location problem on a tree network.

In many real-life applications, the parameters in a manufacturing system may have varying values. This value variation may result from machine breakdowns, lack of training, unexpected delays, non-qualified operators or complex tasks, among others (Özcan 2010). Stochastic programming can provide an effective means for incorporating uncertainty in real-life MHSs (Birge and Louveaux 1997). Stochastic programming has been used frequently in the design and control of MHSs with AGVs (Sayarshad and Tavakkoli-Moghaddam 2010).

The presence of uncertainties in automated material handling and AGV systems has also motivated researchers to explore compensator design such as neural networks (Pamosoaji, Cat, and Hong 2013). Kuo et al. (2007) argues that simulation is very time-consuming for large MHSs and the process of collecting adequate sample data places limitations on any analysis. They proposed to overcome this problem by developing a neural network simulation metamodel that required only a comparably small training data-set. Artificial neural networks are composed of interconnected adaptive elements which are intended to respond to stimuli in a manner not unlike the human nervous system (Kohonen 1988). McCulloch and Pitts (1943) introduced one of the first artificial neuron models in 1943. The main feature of their neuron model is that a weighted sum of input signals is compared to a threshold to determine the neuron output. Unlike biological networks, the parameters of their networks had to be designed, as no training method was available. However, the perceived connection between biology and digital computers generated a great deal of interest. In the late 1950s, Rosenblatt (1958) and several other researchers developed a class of neural networks called perceptrons. The neurons in these networks were similar to those of McCulloch and Pitts (1943). Rosenblatt (1958) key contribution was the introduction of a learning rule for training perceptron networks to solve pattern recognition problems.

In this paper, we consider a manufacturing system with the following physical characteristics: (1) the manufacturing system is a job shop and (2) single-load AGVs perform the material handling job in the shop. We use a stochastic programming framework and propose a bi-objective optimisation model (which has not been used in the MHS evaluation studies) to determine the optimal production time and cost in a manufacturing system with an automated MHS and AGVs. The contribution of this paper is fivefold: we consider (1) a stochastic programming problem and decompose the optimisation process into manageable steps and integrate the results to arrive at a solution consistent with organisational goals and objectives; (2) stochastic parameters in multi-objective optimisation models in general and the proposed bi-objective optimisation model in particular; (3) machine reliability as an important component in the proposed optimisation model; (4) machine maintenance through the breakdown rate for a realistic representation of the system; and (5) the elements of uncertainty within the proposed structured framework by using a perceptron neural network to weigh the two objectives in the proposed bi-objective optimisation model.

3. Problem description and assumptions

In this problem, we consider a manufacturing system equipped with an AGV for material handling in a job-shop environment. The AGV moves a part-type from one shop to another to complete a production cycle. In each shop, the part-type is partially processed. The part-types completed in the final shop are moved to the warehouse. For example, let us consider a manufacturer of cylinder block in the automotive industry. The manufacturer of cylinder block can install an AGV system to supply the lines with parts and to transfer cylinder blocks between five shops. Generally, the sequence of operations on a typical machining line for cylinder blocks involves five shops as shown in Figure 1:

This machining line for cylinder blocks is comprised of the following 16 processes performed in each of the five job shops:

- Shop 1: Assuring a uniform wall thickness for the cylinder bores – qualifying (Process-1); rough mill pan and head faces (Process-2); and rough machining cylinder bores (Process 3).
- Shop 2: Milling bearing cap width and slots (Process 4) and finishing mill pan and bearing cap width (Process 5).
- Shop 3: Drilling oil holes – compound angles (Process 6); drilling, reaming, tapping – left and right, pan and head faces (Process 7); assembly of bearing caps (Process 8); finish front and rear end (Process 9); and drilling, reaming, tapping – end faces (Process 10).
- Shop 4: Line boring crankbore (Process 11); finish tappet bores (Process 12); assembly cam liners, finish line boring (Process 13); finish cylinder bores (Process 14); and finish mill/grind head face (Process 15).
- Shop 5: Hone and grade (Process 16).

The AGV waiting time in the job shop is not known precisely. It is also possible for a machine to breakdown while processing a job. The number of breakdowns (and the breakdown cost) is also not known precisely. The distributions for the stochastic parameters in the problem are assumed to be normal or can be estimated by the normal distribution. Additionally, the machines' reliability is also considered as a stochastic parameter following the exponential distribution. We present the mathematical details of the proposed model in the next section.

4. Mathematical representation of the problem

In this section, we propose a mathematical model for a simultaneous optimisation of the production times and costs.

4.1 *Reliability component*

We assume that the reliability of each machine type is independent according to exponential processes. Also, J is the total number of machine types (i.e. milling, drilling, turning, welding, etc.) We discuss the reliability-based model as follows:

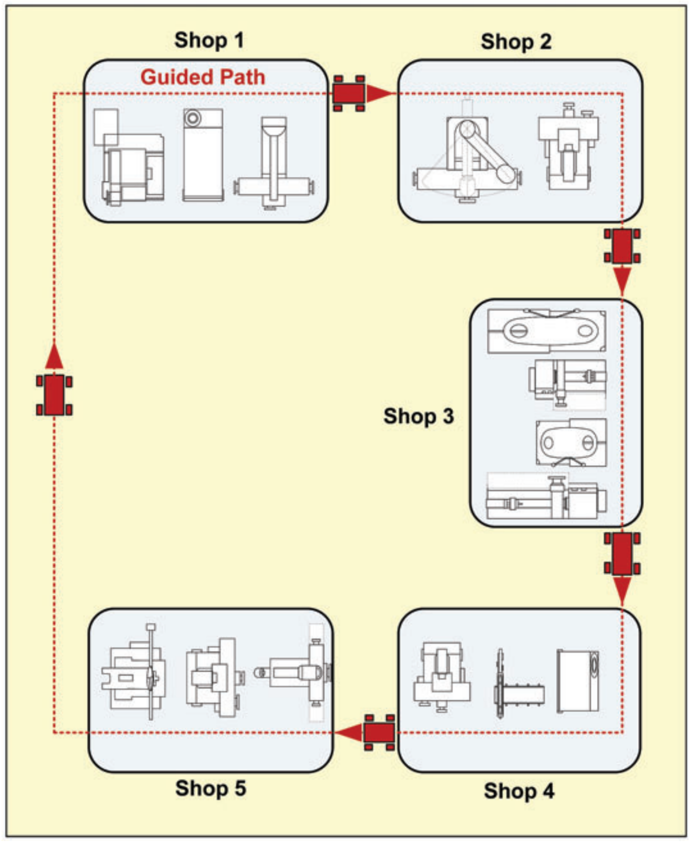

Figure 1. Cylinder block machining line example.

$$R(t)_{\text{system}} = \begin{cases} \left(1 - \prod_{j=1}^{J} \left(1 - R_j(t)\right)\right), & \text{when machines in each shop are in parallel} \\ \left(\prod_{j=1}^{J} R_j(t)\right), & \text{when machines in each shop are in series} \end{cases} \tag{1}$$

where $R_j(t)$ is the probability that the machine type j works for a period of t time units.

As stated earlier, the machine types in each shop are parallel and the shops are organised in series. Therefore, the reliability of the system can be measured as follows:

$$\left(1 - \prod_{j=1}^{J} \left(1 - R_j(t)\right)\right) \geq \alpha, \tag{2}$$

where α is the lower bound for a desirable system reliability during the time period t. As previously assumed the reliability of each machine type is independent and can be measured according to the following exponential distribution:

$$R_j(t) = e^{\frac{-t}{\theta_j}}, \tag{3}$$

where θ_j is the exponential parameter for machine type breakdown. Then,

$$\left(1 - \prod_{j=1}^{J} \left(1 - e^{\frac{-t}{\theta_j}}\right)\right) \geq \alpha \tag{4}$$

In order to obtain a higher level of reliability, more cost is incurred to the system. Hence, a cost function ($C_j(t)$) is defined to keep machine type j reliable for the time period t. The following represents the cost for the entire system:

$$\sum_{j=1}^{J} C_j(t) \tag{5}$$

In order to validate the stated cost function, we consider the costs as losses and use the minimum expected loss (or minimum risk) associated with the system. We represent these losses with a quadratic loss function which is mathematically more tractable than other loss functions because of its symmetric property (i.e. an error above the target causes the same loss as the same magnitude of error below the target). If the target value of the pre-planned exponential parameter is f, then a quadratic loss (cost) function is

$$C_j(t) = L|f - \theta_j|^2 \tag{6}$$

where L is a constant and its value could be set to 1 if the constant makes no difference to a decision.

4.2 *Mathematical formulation*

We replace our proposed stochastic parameters with a combination of the expected value and the variance for that parameter and the following non-linear deterministic mathematical model is derived. As indicated earlier, our goal is to simultaneously optimise production time and cost in the following bi-objective model:

Indices:

i	Number of shops $i = 1,2, \ldots, I$
j	Number of machines $j = 1,2, \ldots, J$

Parameters:

C_d	Stochastic cost per defective unit
C_o	AGV operational cost per unit produced
C_{t_i}	Tool cost for shop i
t_{p_j}	Job processing time on machine j
$t_{w_{ij}}$	Stochastic waiting time of machine j in shop i
t_m	Material handling time
t_c	Cycle time
β	Total daily cost of the defective items
B_1	Total AGV budget
B_2	Total tools budget
B_3	Total machines budget
D_i	Total demand for shop i
M_i	Number of machines in shop i
N	Number of jobs
Z_P	Standard normal Z value for percentile P

Decision variables:

X_{ij}	Number of units produced in shop i by machine j
θ_j	Exponential reliability function of machine j

Objective function 1 (cost minimisation):

$$\begin{aligned} \text{Min} \quad & \left(\sum_{j=1}^{J} \sum_{i=1}^{I} E(C_j(t)) \cdot X_{ij} \right) + Z_P \sqrt{\left(\sum_{j=1}^{J} \sum_{i=1}^{I} \mathrm{Var}(C_j(t)) X_{ij}^2 \right)} + \left(\left(\sum_{i=1}^{I} C_{t_i} + C_o \right) \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) \\ & + \left(E(C_d) \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) + Z_P \sqrt{\mathrm{Var}(C_d) \cdot \left(\sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij}^2 \right)} \end{aligned} \tag{7}$$

Objective function 2 (time minimszation):

$$\text{Min} \quad \left(\left(\sum_{j=1}^{J} t_{p_j} + t_m \right) . \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) + \left(\sum_{j=1}^{J} \sum_{i=1}^{I} E(t_{w_{ij}}) \cdot X_{ij} \right) + Z_P \sqrt{\left(\sum_{j=1}^{J} \sum_{i=1}^{I} \mathrm{Var}(t_{w_{ij}}) \cdot X_{ij}^2 \right)} \tag{8}$$

Subject to:

$$\left(\left(N\sum_{j=1}^{J}t_{p_j}\right)\cdot\left(\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}\right)\right)+\left(\sum_{j=1}^{J}\sum_{i=1}^{I}E(t_{w_{ij}})\cdot X_{ij}\right)+Z_P\sqrt{\left(\sum_{j=1}^{J}\sum_{i=1}^{I}\mathrm{Var}(t_{w_{ij}})\cdot X_{ij}^2\right)}+\left(t_m\cdot\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}\right)\le\left(t_c\cdot\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}\right) \tag{9}$$

$$C_o\cdot\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}\le B_1, \tag{10}$$

$$\left(E(C_d)\cdot\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}\right)+Z_P\sqrt{\mathrm{Var}(C_d)\cdot\left(\sum_{j=1}^{J}\sum_{i=1}^{I}X_{ij}^2\right)}\le\beta, \tag{11}$$

$$\left(1-\prod_{j=1}^{J}\left(1-e^{\frac{-t}{\theta_j}}\right)\right)\ge\alpha, \tag{12}$$

$$\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{C_{t_i}\cdot X_{ij}}{M_i}\le B_2, \tag{13}$$

$$\left[\sum_{j=1}^{J}C_j(t)\right]\cdot R_{ij}(t)_{\text{system}}\le B_3, \tag{14}$$

$$\sum_{j=1}^{J}X_{ij}\ge D_i,\quad i=1,2,\ldots,I, \tag{15}$$

$$X_{ij},\theta_j\ge 0,\quad i=1,2,\ldots,I;\quad j=1,2,\ldots,J. \tag{16}$$

Note that, *E()* and *Var()* are the expected value and the variance of the stochastic parameters, respectively. Equation (7) is the first objective function intended to minimise the total cost of production. Equation (8) is the second objective function and is intended to minimise the total time of production. Equation (9) shows that the total production time is limited to the cycle time. Equation (10) indicates the limitation of the operational budget for the AGVs. Equation (11) represents a constraint for the acceptable defect rate. Equation (12) shows that the reliability of the system is restricted to a lower-bound α. Equation (13) indicates that the total available budgets for the required tools in each shop are limited to a pre-specified upper-bound value. Equation (14) indicates that the total available budget for the reliability of the machines is limited. Equation (15) indicates that the demand at each shop must be satisfied. Equation (16) enforces the non-negativity of the variables. Finally, θ_j represents the stochastic breakdown of machine *j*. θ_j is one of our decision variables, and our goal is to obtain the θ_j values that can ensure no machine breakdown.

5. Numerical example

In this section, we illustrate the effectiveness and applicability of the proposed model with the cylinder block manufacturing example presented in Section 2. Let us revisit the five shops ($i = 5$) in the cylinder block machining line. The number of machines (M_i) in shops 1 through 5 is 3, 2, 4, 3 and 3, respectively. The defect cost is a stochastic parameter (C_d) with an expected value of 7.85 dollars and a variance of 1.9465. These defect costs are calculated based on the data collected by the maintenance department in one month (20 days of operations). The cost of operating the AGV (C_o) is expected to be 15 dollars per day. The tool costs in the five shops (C_{t_i}) are estimated to be \$11.00, \$12.00, \$14.00, \$17.00 and \$10.00 per day for shops 1 through 5, respectively. The daily processing times (t_{p_j}) for machines 1 through 4 are expected to be 35, 27, 45 and 31 min, respectively. The material handling time (t_m) is expected to be 11 min for the entire line in a day. The cycle time (t_c) (total time required to convert raw materials into finished goods) is 225 h (13,500 min). The total daily cost of the defective items (β) is expected to be 2400 dollars for the entire line. The available budget for the AGV (B_1) is \$5000, the available budget for the tools (B_2) is \$3000 and the available budget for the machine types (B_3) is \$2700. The standard normal Z value (Z_p) for the 0.95 percentile is 1.96.

The α for reliability is assumed to be 0.65. α is the lower bound for a desirable system reliability expressed as a percentage between 0 and 100%. Higher α values are used for highly reliable systems while lower α values are used for less reliable manufacturing systems. The value for α could be determined based on: (1) actual historical data; (2) expert opinion or management judgment; or (3) a combined approach. In the manufacturing system under consideration, the system reliability data were 54.4%. However, the operations managers believed that the new components in the manufacturing system should improve reliability by approximately 10%. Considering both the available data and our management judgments, we used 0.65 as the lower bound for system reliability in this study.

The total number of machines is 15 and numbers of machines in shops 1–5 are 3, 2, 4, 3 and 3, respectively. The demands for the shops are equal and assumed to be 5 units of part-types. Note that, the machine types are equal to 4 and all machine types do not operate in all shops. For example, machine type 3 does not operate in shop 1 because of the process plan and machine type specifications in this shop. The expected values and variances of the machine waiting times in different shops are summarised in Table 1. These values are computed based on 100 observations for each machine in each shop.

The multi-objective mathematical model is then transformed into a single-objective model by the perceptron weighing technique. Several conventional approaches such as point allocation, probability theory, swing weighting (Von Winterfeldt and Edwards 1986), paired comparisons (Saaty 1980), trade-off analysis (Keeney and Raiffa 1976), the simple multi-attribute rating technique (Edwards 1977; Von Winterfeldt and Edwards 1986) and regression estimates are used to specify weights in multi-objective optimisation (Kleindorfer, Kunreuther, and Schoemaker 1993; Pöyhönen and Hämäläinen 2001). In this study, we use perceptron neural networks to determine a suitable weight for the *time* and *cost* objectives. We cannot use the aforementioned conventional approaches because of the lack of precise or fuzzy data in this problem.

Neural networks have a remarkable ability to derive meaning from complicated or imprecise data. They can detect trends and extract patterns that are too complex to be noticed by either humans or other computer techniques. We use the perceptron weighing technique proposed in this study to extract meaning from our imprecise data with regards to the importance of the time and cost objectives. The proposed perceptron weighting technique has the ability to determine the necessary weights in our bi-objective model based on the data given for training or initial experience. Perception neural networks also do not make any assumption with regards to the underlying probability density functions about the pattern classes under consideration in comparison to other conventional probability-based methods (Khosravi et al. 2011). Perception neural networks are also both reliable and effective when applied to applications involving prediction, classification and clustering (Kros, Lin, and Brown 2006). In addition, they can process incomplete and imprecise data by approximating non-linear relations in data (Ultsch, Korus, and Kleine 1995).

5.1 *Perceptron neural network*

In this study, we apply perceptron to weight our time and cost objectives. To conceptualise a perceptron structure for our objectives, consider a single-neuron perceptron with two-input and one neuron as shown in Figure 2.

In this figure, p_1 and p_2 are the inputs and w_{11} and w_{12} are their corresponding weights, respectively. Also, $w_p^{(T)}$ is the weighted input corresponding to the target output. The output of this network is determined by:

$$a = \text{hardlim}(n) = \text{hardlim}(w_p^{(T)} + b) = \text{hardlim}(w_{11} \cdot p_1 + w_{12} \cdot p_2 + b). \tag{17}$$

The decision boundary is determined by the input vectors for which the net input n is zero; that is,

$$n = w_p^{(T)} + b = w_{11} \cdot p_1 + w_{12} \cdot p_2 + b = 0. \tag{18}$$

Table 1. Expected values and variances of the machine waiting times in different shops.

Expected Values	$E(t_{w_{11}})$	$E(t_{w_{12}})$	$E(t_{w_{14}})$	$E(t_{w_{21}})$	$E(t_{w_{24}})$	$E(t_{w_{31}})$	$E(t_{w_{32}})$	–
	29.6	41.4	29.7	15.8	31.6	29.7	15.8	–
	$E(t_{w_{33}})$	$E(t_{w_{34}})$	$E(t_{w_{41}})$	$E(t_{w_{42}})$	$E(t_{w_{44}})$	$E(t_{w_{52}})$	$E(t_{w_{53}})$	$E(t_{w_{54}})$
	23.6	16.4	41.4	21.5	29.7	23.6	16.4	30.4
Variances	$V(t_{w_{11}})$	$V(t_{w_{12}})$	$V(t_{w_{14}})$	$V(t_{w_{21}})$	$V(t_{w_{24}})$	$V(t_{w_{31}})$	$V(t_{w_{32}})$	–
	3.44	20.84	3.61	12.36	16.64	3.61	12.36	–
	$V(t_{w_{33}})$	$V(t_{w_{34}})$	$V(t_{w_{41}})$	$V(t_{w_{42}})$	$V(t_{w_{44}})$	$V(t_{w_{52}})$	$V(t_{w_{53}})$	$V(t_{w_{54}})$
	5.44	8.44	20.84	9.45	3.61	5.44	8.44	18.44

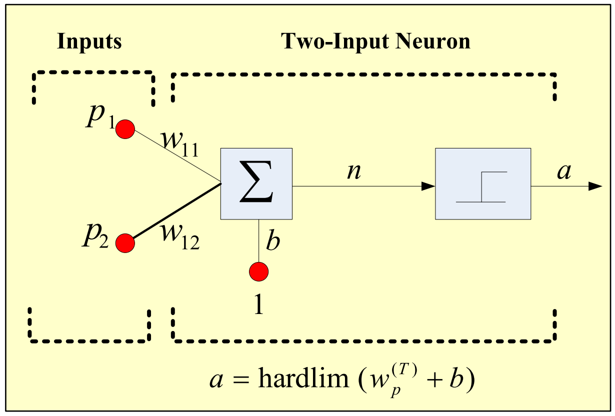

Figure 2. Two-input/single output perceptron.

The weights are the learning rates used in the proposed mathematical model. The summation of the learning weights is used as a single learning rate ξ. In order to configure our perceptron network, we consider the two factors of machine and operation as the inputs for time and the two factors of material and operator as the inputs for cost. The goal here is to find the final weights of the time and cost outputs. The perceptron network is run using the data observed in the manufacturing system and the weights of W_1 (for time) and W_2 (for cost) are obtained after the convergence. The perceptron computations are implemented in the MATLAB 7.0 package. These weights are used to unify the objective function. The weights are 0.2 for time and 0.1 for cost. The perceptron convergence process is illustrated in Table 2.

This table includes several columns where each column computes a stage of the proposed perceptron algorithm for weighting the conflicting objective functions for the purpose of integration. In the first iteration (first row of this table), the input and the initial weights are given by the operator and the calculation begins by computing the 'calculated output' and 'sum' columns. The entry in the sum column (S) is then compared with the threshold network (N) and if the sum is larger than the threshold network, then, the positive error term is considered for correction purpose. The correction column (R) is computed using the learning rate and the error term. The updated weights are then given in the final weights column. The updated weights are used as the initial weights in the next iteration and the process continues until convergence is reached. More specifically, three sets of inputs (X_0, X_1, X_2) are mapped based on (b, p_1, p_2), where machine and operation are inputs for time, and material and operator are inputs for cost (see Figure 2). Then, using the initial weights, the computation begins iteratively to obtain outputs based on the desired outputs and the convergence is verified with a pre-specified threshold (TH = 0.5). In addition, corrections are performed in each iteration using the learning rate of 0.1. This process is repeated until all the weights converge. Convergence is reached when the final weights are stabilised. The information about the proposed perceptron is as follows:

input: X_1, X_2,
$X_0 * W_0 = b$,
TH = 0.5,
learning rate (LR) = 0.1.

The training set consists of four samples: {((0,0),1), ((0,1),1), ((1,0),1), ((1,1),0)}. In the following, the final weights of one iteration become the initial weights in the next iteration. Each cycle over all the samples in the training set is demarcated with heavy lines. The pseudo code implemented in MATLAB 7.0 is given in Figure 3.

LINGO 9.0 was used to facilitate the computations. Note that, while the model is non-linear, we applied the successive linear programming (SLP) tool box. This tool box computes the linear approximation of the model using the SLP algorithm and solves the problem. The values of θ for the different types of machines are as follows:

$$\theta_1 = 13.744, \quad \theta_2 = 14.25, \quad \theta_3 = 11.144, \quad \theta_4 = 15.71.$$

The manufacturing system should set the breakdown of the machines according to the obtained θ for different machine types. The cost function is $C_j(t) = |7 - \theta_j|^2$, and the corresponding expected value and variance are 14.02 and 0, respectively. The value of e and t in the exponential distribution is set to be 2.71 and 6, respectively.

The numbers of produced part-types in different shops and by various machine types are reported in Table 3. Note that the other values of X are zero. The objective function value is 4558.676.

We are also able to present some comparative results to show the advantages of the proposed stochastic bi-objective mathematical model. Here, we consider a deterministic case with separate cost and time objectives:

Table 2. Perceptron convergence process for the proposed problem.

Inputs: Sensor values X_0	Inputs: Sensor values X_1	Inputs: Sensor values X_2	Inputs: Desired output Z	Initial weights w_0	Initial weights w_1	Initial weights w_2	Calculated outputs C_0 $X_0 \cdot w_0$	Calculated outputs C_1 $X_1 \cdot w_1$	Calculated outputs C_2 $X_2 \cdot w_2$	Sum S $C_0 + C_1 + C_2$	Network threshold N IF($S >$ TH,1,0)	Error E $Z - N$	Correction R $LR \cdot E$	Final weights W_0	Final weights W_1	Final weights W_2
1	0	0	1	0	0	0	0	0	0	0	0	1	+0.1	0.1	0	0
1	0	1	1	0.1	0	0	0.1	0	0	0.1	0	1	+0.1	0.2	0	0.1
1	1	0	1	0.2	0	0.1	0.2	0	0	0.2	0	1	+0.1	0.3	0.1	0.1
1	1	1	0	0.3	0.1	0.1	0.3	0.1	0.1	0.5	0	0	0	0.3	0.1	0.1
1	0	0	1	0.3	0.1	0.1	0.3	0	0	0.3	0	1	+0.1	0.4	0.1	0.1
1	0	1	1	0.4	0.1	0.1	0.4	0	0.1	0.5	0	1	+0.1	0.5	0.1	0.2
1	1	0	1	0.5	0.1	0.2	0.5	0.1	0	0.6	1	0	0	0.5	0.1	0.2
1	1	1	0	0.5	0.1	0.2	0.5	0.1	0.2	0.8	1	−1	−0.1	0.4	0	0.1
1	0	0	1	0.4	0	0.1	0.4	0	0	0.4	0	1	+0.1	0.5	0	0.1
1	0	1	1	0.5	0	0.1	0.5	0	0.1	0.6	1	0	0	0.5	0	0.1
1	1	0	1	0.5	0	0.1	0.5	0	0	0.5	0	1	+0.1	0.6	0.1	0.1
1	1	1	0	0.6	0.1	0.1	0.6	0.1	0.1	0.8	1	−1	−0.1	0.5	0	0
1	0	0	1	0.5	0	0	0.5	0	0	0.5	0	1	+0.1	0.6	0	0
1	0	1	1	0.6	0	0	0.6	0	0	0.6	1	0	0	0.6	0	0
1	1	0	1	0.6	0	0	0.6	0	0	0.6	1	0	0	0.6	0	0
1	1	1	0	0.6	0	0	0.6	0	0	0.6	1	−1	−0.1	0.5	0.1	0.1
1	0	0	1	0.5	−0.1	−0.1	0.5	0	0	0.5	0	1	+0.1	0.6	−0.1	−0.1
1	0	1	1	0.6	−0.1	−0.1	0.6	0	−0.1	0.5	0	1	+0.1	0.7	−0.1	0
1	1	0	1	0.7	−0.1	0	0.7	−0.1	0	0.6	1	0	0	0.7	−0.1	0
1	1	1	0	0.7	−0.1	0	0.7	−0.1	0	0.6	1	−1	−0.1	0.6	−0.2	−0.1
1	0	0	1	0.6	−0.2	−0.1	0.6	0	0	0.6	1	0	0	0.6	−0.2	−0.1
1	0	1	1	0.6	−0.2	−0.1	0.6	0	−0.1	0.5	0	1	+0.1	0.7	−0.2	0
1	1	0	1	0.7	−0.2	0	0.7	−0.2	0	0.5	0	1	+0.1	0.8	−0.1	0
1	1	1	0	0.8	−0.1	0	0.8	−0.1	0	0.7	1	−1	−0.1	0.7	−0.2	−0.1
1	0	0	1	0.7	−0.2	−0.1	0.7	0	0	0.7	1	0	0	0.7	−0.2	−0.1
1	0	1	1	0.7	−0.2	−0.1	0.7	0	−0.1	0.6	1	0	0	0.7	−0.2	−0.1
1	1	0	1	0.7	−0.2	−0.1	0.7	−0.2	0	0.5	0	1	+0.1	0.8	−0.1	−0.1
1	1	1	0	0.8	−0.1	−0.1	0.8	−0.1	−0.1	0.6	1	−1	−0.1	0.7	−0.2	−0.2
1	0	0	1	0.7	−0.2	−0.2	0.7	0	0	0.7	1	0	0	0.7	−0.2	−0.2
1	0	1	1	0.7	−0.2	−0.2	0.7	0	−0.2	0.5	0	1	+0.1	0.8	−0.2	−0.1
1	1	0	1	0.8	−0.2	−0.1	0.8	−0.2	0	0.6	1	0	0	0.8	−0.2	−0.1
1	1	1	0	0.8	−0.2	−0.1	0.8	−0.2	−0.1	0.5	0	0	0	0.8	−0.2	−0.1
1	0	0	1	0.8	−0.2	−0.1	0.8	0	0	0.8	1	0	0	0.8	−0.2	−0.1
1	0	1	1	0.8	−0.2	−0.1	0.8	0	−0.1	0.7	1	0	0	0.8	−0.2	−0.1

```
threshold = 0.5
learning_rate = 0.1
weights = [0, 0, 0]
training_set = [((1, 0, 0), 1), ((1, 0, 1), 1), ((1, 1, 0), 1),
((1, 1, 1), 0)]

def sum_function(values):
    return sum(value * weights[index] for index, value in
enumerate(values))

while True:
    print '-' * 60
    error_count = 0
    for input_vector, desired_output in training_set:
        print weights
        result = 1 if sum_function(input_vector) > threshold
else 0
        error = desired_output - result
        if error != 0:
            error_count += 1
            for index, value in enumerate(input_vector):
                weights[index] += learning_rate * error * value
    if error_count == 0:
        break
```

Figure 3. Pseudo-code implemented in MATLAB 7.0.

Table 3. Number of part-types produced by different machines in different shops.

Decision variable	Quantity
X_{13}	5
X_{21}	5
X_{32}	2.5
X_{34}	2.5
X_{42}	5
X_{51}	2.5
X_{53}	2.5

Let us consider a single objective function model for cost, Min. Equation (7), subject to constraints (10), (11), (13), (15) and $X_{ij} \geq 0$, where $i = 1,2,\ldots, I$ and $j = 1,2,\ldots, J$. We solve this model with LINGO 9.0 (using the input values given earlier) and obtain: $X_{11} = 2$, $X_{14} = 3$, $X_{23} = 5$, $X_{32} = 1$, $X_{33} = 4$, $X_{43} = 5$, $X_{52} = 3$ and $X_{54} = 2$. However, these results are incomplete because they do not reflect: (1) the machines' reliability and the proposed loss function as reliability cost and (2) the interrelated cost-time Equation (14).

Let us consider a single objective function model for time, Min. Equation (8), subject to constraints (9), (12), (14), (15) and (16). We solve this model with LINGO 9.0 (using the input values given earlier) and obtain: $X_{12} = 5$, $X_{22} = 5$, $X_{33} = 5$, $X_{41} = 5$ and $X_{54} = 5$.

We consider the following objective functions with regards to cost minimisation (19) and time minimisation (20):

$$\text{Min} \quad \left(\sum_{j=1}^{J} \sum_{i=1}^{I} C_j(t) \cdot X_{ij} + \left(\left(\sum_{i=1}^{I} C_{t_i} + C_o \right) \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) + (C_d) \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right). \tag{19}$$

$$\text{Min} \quad \left(\left(\left(\sum_{j=1}^{J} t_{p_j} + t_m \right) \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) + \left(\sum_{j=1}^{J} \sum_{i=1}^{I} t_{w_{ij}} \cdot X_{ij} \right) \right). \tag{20}$$

Subject to:

$$\left(\left(N \sum_{j=1}^{J} t_{p_j} \right) \cdot \left(\sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) \right) + \left(\sum_{j=1}^{J} \sum_{i=1}^{I} t_{w_{ij}} \cdot X_{ij} \right) + \left(t_m \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) \leq \left(t_c \cdot \sum_{j=1}^{J} \sum_{i=1}^{I} X_{ij} \right) \tag{21}$$

$$C_d \cdot \sum_{j=1}^{J}\sum_{i=1}^{I} X_{ij} \leq \beta, \tag{22}$$

Equations (10), (13), (15), and $X_{ij} \geq 0$, where $i = 1,2,\ldots, I$ and $j = 1,2,\ldots, J$.

We solve the above bi-objective model with LINGO 9.0 (using the input values given earlier) and obtain: $X_{11} = 2$, $X_{13} = 3$, $X_{22} = 1$, $X_{24} = 4$, $X_{32} = 2$, $X_{33} = 3$, $X_{43} = 4$, $X_{44} = 1$, $X_{51} = 2$, $X_{53} = 1$ and $X_{54} = 2$. However, these results are incomplete because they do not reflect: (1) the machines' reliability and the proposed loss function as reliability cost and (2) the stochastic nature of the manufacturing system.

6. Extended results

The mathematical model gives us the $\theta_j s$ considering a confidence level as a reliability of the job-shop system. Now, we investigate the $\theta_j s$ for the shops separately by data collection. The aim is to analyse the $\theta_j s$ obtained from data collection which does not have any interactions with other shops, in comparison with the $\theta_j s$ gained from the mathematical model in the last section. We collect data for a specific working time $t = 24$ (min). Then for $t > 24$, our data are type I censored data. Assume $x_{1:n}, x_{2:n}, \ldots, x_{r:n}$ are the r censored data in a specific shop, then the estimated θ ($\hat{\theta}$) using maximum likelihood estimation (MLE) is as follows:

Assume that X follows an exponential distribution,

$$X \approx \mathrm{EXP}(\theta), \tag{23}$$

The cumulative distribution function is,

$$F(x) = 1 - e^{\frac{-x}{\theta}}, \tag{24}$$

The probability density function is,

$$f(x) = \frac{1}{\theta} e^{\frac{-x}{\theta}}, \tag{25}$$

Then for all $x_{i:n}$, the ordered statistic is,

$$g(x_{1:n}, x_{2:n}, \ldots, x_{r:n}) = \frac{n!}{(n-r)!}\left(1 - (1 - e^{\frac{-t_0}{\theta}})\right) \cdot \prod_{i=1}^{r} \frac{1}{\theta} e^{\frac{-x_{i:n}}{\theta}}, \tag{26}$$

The likelihood function is,

$$L(\theta) = \frac{n!}{(n-r)!} \cdot \theta^{-r} \cdot e^{-\frac{\sum_{i=1}^{r} x_{i:n} + (n-r)t_0}{\theta}}, \tag{27}$$

The logarithm of both sides of (27) gives,

$$\ln(L(\theta)) = \ln \frac{n!}{(n-r)!} - r \ln\theta - \frac{\sum_{i=1}^{r} x_{i:n} + (n-r)t_0}{\theta^2}, \tag{28}$$

The partial derivative of (28) with respect to θ is,

$$\frac{\partial \ln(L(\theta))}{\partial\theta} = -\frac{r}{\theta} + \frac{\sum_{i=1}^{r} x_{i:n} + (n-r)t_0}{\theta}, \tag{29}$$

Then if we set (31) equal to zero, we have:

$$\hat{\theta}_{\mathrm{MLE}} = \frac{\sum_{i=1}^{r} x_{i:n} + (n-r)t_0}{r}. \tag{30}$$

where t_0 is the end time of observation ($t = 24$) and r is the number of failures. While setting the manufacturing system with the exact value of θ_j obtained from the mathematical model is difficult due to mechanical specifications' changes during the manufacturing process, we propose a confidence interval for θ. Here, using $\frac{2r\hat{\theta}}{\theta} \approx \chi^2(2r)$ as a pivot ($2r$ is degree of freedom), we can set a confidence interval for θ. To set a confidence interval using $\frac{2r\hat{\theta}}{\theta} \approx \chi^2(2r)$, we obtain:

$$P\left[\chi^2_{1-\frac{\alpha}{2}}(2r) < \frac{2r\hat{\theta}}{\theta} < \chi^2_{\frac{\alpha}{2}}(2r)\right] = 1 - \alpha, \tag{31}$$

By inverting (31) and multiplying all fractions by $2r\hat{\theta}$ we obtain:

$$P\left[\frac{2r\hat{\theta}}{\chi^2_{\frac{\alpha}{2}}(2r)} < \theta < \frac{2r\hat{\theta}}{\chi^2_{1-\frac{\alpha}{2}}(2r)}\right] = 1 - \alpha, \tag{32}$$

where (34) is a two-sided $(1 - \alpha)$% confidence interval based on type I censored data. In our case, we collected data for a special working period. We were supposed to collect 30 observations but a failure occurred (type I censored data) in the 20th observation. Therefore, using (32) we can obtain the estimated θ.

$$\hat{\theta}_{\mathrm{MLE}} = \frac{235 + (30 - 20)10}{20} = 16.75. \tag{33}$$

Clearly the estimated θ is different from the one obtained from the mathematical model. Here, we configure the confidence interval for θ using the estimated value with a 0.95% confidence level (Equation (33)).

$P\left[\frac{2\times 20\times 16.75}{\chi^2_{0.025}(40)} < \theta < \frac{2\times 20\times 16.75}{\chi^2_{0.975}(40)}\right] = 1 - 0.05$; we obtain [11.3, 27.46] with a 95% confidence interval for θ.

The results show that the output produced in the integrated case was higher than the outputs produced by separate cost and time objective function considerations. In addition, the proposed model resulted in an increase in the availability of the manufacturing system through the improvements in the reliability of the machine types and the shops. These improvements help to maintain the system at a desirable level of reliability and to prevent sudden breakdowns.

7. Conclusions and future research directions

AGV systems complement the operation in manufacturing systems by providing integrated automated material handling that capitalises on the system's flexibility. Previous research considering AGVs systems has focused primarily on complex control strategies in MHSs. In this study, we focused on the time and cost measures in an optimisation model used to evaluate an MHS with AGVs. The automated manufacturing system considered in this study has the following physical characteristics: (1) the manufacturing system is a job shop and (2) single-load AGVs perform the material handling job in the shop.

We took into account the reliability of the manufacturing system because of the need for steadiness and stability in the system. Reliability was included in the model as a cost function. Furthermore, we considered bi-objective stochastic programming to optimise the time and cost objectives because of the uncertainties inherent in the optimisation parameters in real-world problems. Finally, we used perceptron neural networks to transform the bi-objective optimisation model into a single objective model.

The MHS proposed in this study could potentially be extended (or revised) to improve the effectiveness of a wide variety of decision tools in productivity improvement. For example, current trends in scheduling systems provide the production scheduler with powerful tools which can be used to optimise real-time workloads in various stages of production. The MHS could potentially be integrated within such a real-time scheduling system. The proposed method could also be potentially useful for general applications of business process improvement which strive to improve workflows within and between functional groups. These approaches often utilise systematic methods to process a large amount of imprecise and complex information to redesign critical business processes.

The limitations of this study and the future research directions are summarised as follows:

- Although we used optimisation in this study, alternatively, a simulation method or a heuristic algorithm could be used in lieu of the proposed model. Buzacott and Yao (1986) presented a comprehensive review of the analytical models developed for the design and control of automated and flexible manufacturing systems. They advocated analytical methods over the simulation models because analytical methods provide a better insight into the system performance. Ho et al. (2000, 490), Lee, Chew, and Manikam (2006, 1828), Kuo et al. (2007, 1002), and Crombecq, Laermans, and Dhaene (2011, 683) have also confirmed that although computer simulations are often the first choice for modelling systems and even for optimisation purposes, the simulation of complex systems with multiple input and output parameters can be both prohibitively expensive and time-consuming. On the other hand, heuristics seek good feasible solutions to optimisation problems where the complexity of the problem or the available time for its solution does not allow exact algorithms. Although the main measure of success for exact algorithms is time efficiency, we must often evaluate the quality of solutions when an exact optimum is not available (Gnanavel Babu et al. 2010).
- Although we did not experience this limitation in our study, the time complexity of the learning phase in some complex perceptron neural networks could be high depending on the heuristic used for calculating the weights

and the halting condition. In such complex perceptron neural networks, a large number of passes may be required. Each pass involves computation of the outcome for every training set point followed by modification of the weights.

- Although in this study we used the time and cost measures in an optimisation model, performance measures may be categorised on the basis of: time, cost, quality and flexibility measures. Consideration of quality and flexibility is a natural extension of the model proposed in this study.
- Although in this study we considered single-load AGVs, researchers (e.g. Ozden 1988; Lee, Tangjarukij, and Zhu 1996; Ho and Hsieh 2004) have already proven that multiple-load AGV systems have many advantages over single-load ones. Another natural extension of this research is using multiple-load AGVs in the optimisation model.

Acknowledgement

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References

Abdelmaguid, T. F., A. O. Nassef, B. A. Kamal, and M. F. Hassan. 2004. "A Hybrid GA/Heuristic Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles." *International Journal of Production Research* 42 (2): 267–281.

Ball, M. O., and F. L. Lin. 1993. "A Reliability Model Applied to Emergency Service Vehicle Location." *Operations Research* 41 (1): 18–36.

Beamon, B. M. 1995. *Reliability in Design of Fixed-path Material Handling Systems*. Cincinnati, OH: University of Cincinnati Working Paper.

Beamon, B. M. 1998. "Performance, Reliability, and Performability of Material Handling Systems." *International Journal of Production Research* 36 (2): 377–393.

Birge, J., and F. Louveaux. 1997. *Introduction to Stochastic Programming*. Berlin: Springer-Verlag.

Buzacott, J. A., and D. D. Yao. 1986. "Flexible Manufacturing Systems: A Review of Analytical Models." *Management Science* 32 (7): 890–905.

Carbone, R. 1974. "Public Facilities Location under Stochastic Demand." *INFOR* 12 (3): 261–270.

Charnes, A., and W. Cooper. 1959. "Chance-constraint Programming." *Management Science* 6 (1): 73–79.

Crombecq, K., E. Laermans, and T. Dhaene. 2011. "Efficient Space-filling and Non-collapsing Sequential Design Strategies for Simulation-based Modeling." *European Journal of Operational Research* 214 (3): 683–696.

Deroussi, L., M. Gourgand, and N. Tchernev. 2008. "A Simple Metaheuristic Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles." *International Journal of Production Research* 46 (8): 2143–2164.

Edwards, W. 1977. "How to Use Multiple Attribute Utility Measurement for Social Decision-making." *IEEE Transactions on Systems, Man, and Cybernetics* 7 (5): 326–340.

Farling, B. E., C. T. Mosier, and F. Mahmoodi. 2001. "Analysis of Automated Guided Vehicle Configurations in Flexible Manufacturing Systems." *International Journal of Production Research* 39 (18): 4239–4260.

Gnanavel Babu, A., J. Jerald, A. Noorul Haq, V. Muthu Luxmi, and T. P. Vigneswaralu. 2010. "Scheduling of Machines and Automated Guided Vehicles in FMS Using Differential Evolution." *International Journal of Production Research* 48 (16): 4683–4699.

Ho, Y.-C., and P. F. Hsieh. 2004. "A Machine-to-loop Assignment and Layout Design Methodology for Tandem AGV Systems with Multiple-load Vehicles." *International Journal of Production Research* 42 (4): 801–832.

Ho, Y.-C., C. G. Cassandras, C.-H. Chen, and L. Dai. 2000. "Ordinal Optimisation and Simulation." *Journal of the Operational Research Society* 51 (4): 490–500.

Jain, A., P. K. Jain, F. T. S. Chan, and S. Singh. 2013. "A Review on Manufacturing Flexibility." *International Journal of Production Research* 51 (19): 5946–5970.

Keeney, R. L., and H. Raiffa. 1976. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. New York: Wiley.

Khosravi, A., S. Nahavandi, D. Creighton, and A. F. Atiya. 2011. "Comprehensive Review of Neural Network-based Prediction Intervals and New Advances." *IEEE Transactions on Neural Networks* 22 (9): 1341–1356.

Kleindorfer, P. R., H. C. Kunreuther, and P. J. H. Schoemaker. 1993. *Decision Sciences*. New York: Cambridge University Press.

Kohonen, T. 1988. "An Introduction to Neural Computing." *Artificial Neural Networks* 1 (1): 3–16.

Kros, J. F., M. Lin, and M. L. Brown. 2006. "Effects of the Neural Network S-Sigmoid Function on KDD in the Presence of Imprecise Data." *Computers & Operations Research* 33 (11): 3136–3149.

Kuo, Y., T. Yang, B. A. Peters, and I. Chang. 2007. "Simulation Metamodel Development Using Uniform Design and Neural Networks for Automated Material Handling Systems in Semiconductor Wafer Fabrication." *Simulation Modelling Practice and Theory* 15 (8): 1002–1015.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 2nd ed. Singapore: McGraw-Hill.

Le-Anh, T., and M. B. M. De Koster. 2006. "A Review of Design and Control of Automated Guided Vehicle Systems." *European Journal of Operational Research* 171 (1): 1–23.

Lee, J., M. Tangjarukij, and Z. Zhu. 1996. "Load Selection of Automated Guided Vehicles in Flexible Manufacturing Systems." *International Journal of Production Research* 34 (12): 3383–3400.

Lee, L. H., E. P. Chew, and P. Manikam. 2006. "A General Framework on the Simulation-based Optimization under Fixed Computing Budget." *European Journal of Operational Research* 174 (3): 1828–1841.

Maniya, K. D., and M. G. Bhatt. 2011. "A Multi-attribute Selection of Automated Guided Vehicle Using the AHP/M-GRA Technique." *International Journal of Production Research* 49 (20): 6107–6124.

Maxwell, W. L., and J. A. Muckstadt. 1982. "Design of Automatic Guided Vehicle Systems." *IIE Transactions* 14 (2): 114–124.

McCulloch, W., and W. Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *The Bulletin of Mathematical Biophysics* 5: 115–133.

Miriyala, K., and N. Viswanadham. 1989. "Reliability Analysis of Flexible Manufacturing Systems." *The International Journal of Flexible Manufacturing Systems* 2 (2): 145–162.

Özcan, U. 2010. "Balancing Stochastic Two-sided Assembly Lines: A Chance-constrained, Piecewise-linear, Mixed Integer Program and a Simulated Annealing Algorithm." *European Journal of Operational Research* 205 (1): 81–97.

Ozden, M. 1988. "A Simulation Study of Multiple-load-carrying Automated Guided Vehicles in a Flexible Manufacturing System." *International Journal of Production Research* 26 (8): 1353–1366.

Pamosoaji, A. K., P. T. Cat, and K. S. Hong. 2013. "Sliding-mode and Proportional-derivative-type Motion Control with Radial Basis Function Neural Network Based Estimators for Wheeled Vehicles." *International Journal of Systems Science*. doi:10.1080/00207721.2013.772678.

Pöyhönen, M., and R. P. Hämäläinen. 2001. "On the Convergence of Multiattribute Weighting Methods." *European Journal of Operational Research* 129 (3): 569–585.

Rosenblatt, F. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review* 65 (6): 386–408.

Saaty, T. L. 1980. *The Analytic Hierarchy Process*. New York: McGraw-Hill.

Sarker, B. R., and S. S. Gurav. 2005. "Route Planning for Automated Guided Vehicles in a Manufacturing Facility." *International Journal of Production Research* 43 (21): 4659–4683.

Sayarshad, H. R., and R. Tavakkoli-Moghaddam. 2010. "Solving a Multi Periodic Stochastic Model of the Rail–Car Fleet Sizing by Two-Stage Optimization Formulation." *Applied Mathematical Modelling* 34 (5): 1164–1174.

Shiode, S., and Z. Drezner. 2003. "A Competitive Facility Location Problem on a Tree Network with Stochastic Weights." *European Journal of Operational Research* 149 (1): 47–52.

Smith, D. J. 1993. *Reliability, Maintainability and Risk*. Oxford: Butterworth Heinemann.

Stewart, W., and B. Golden. 1983. "Stochastic Vehicle Routing: A Comprehensive Approach." *European Journal of Operational Research* 14 (4): 371–385.

Ultsch, A., D. Korus, and T. O. Kleine. 1995. "Integration of Neural Networks and Knowledge-Based Systems in Medicine." In *Artificial Intelligence in Medicine*, edited by Pedro Barahona, Mario Stefanelli, and Jeremy Wyatt, 425–426. Berlin: Springer.

Von Winterfeldt, D., and W. Edwards. 1986. *Decision Analysis and Behavioral Research*. Cambridge: Cambridge University Press.