Int J Adv Manuf Technol (2012) 62:371–383
DOI 10.1007/s00170-011-3802-8

ORIGINAL ARTICLE

Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm

Maghsoud Amiri · Ali Mohtashami

Received: 6 July 2011 / Accepted: 21 November 2011 / Published online: 8 December 2011

Abstract This paper presents a multiobjective formulation of the buffer allocation problem in unreliable production lines. Majority of the solution methods for buffer allocation problems assume that the process times, time between failures, and repair times are deterministic or exponentially distributed. This paper relaxes these restrictions by proposing a simulation-based methodology which can consider general function distributions for all parameters of production lines. Factorial design has been used to build a meta-model for estimating production rate based on a detailed, discrete event simulation model. We use genetic algorithm combined to line search method to solve the multiobjective model and determining the optimal (or near optimal) size of each buffer storage.

Keywords Buffer storage · Queuing network · Simulation · Design of experiments · Genetic algorithm · Multiobjective decision making · Line search

1 Introduction

A production line consists of machines connected in series and separated by buffers (Fig. 1). Each part is required to be processed on each machine during a fixed amount of time called process time. A production line for which the process times are equal at all machines be called a homogeneous (or balanced) line. In a nonhomogeneous (or nonbalanced) line, machines may take different lengths of time performing operations on parts [1]. Also it is obvious that in each station, there might be a number of parallel machines that help production and manufacturing system (Fig. 2).

The main aim in such problems is to optimize the production rate, throughput, or the profit of the line. However material flow may be disrupted by machines failure [4] or by differences among the service time of the stations. The inclusion of buffers increases the average production rate of the line by limiting the propagation of distributions, but at the cost of additional capital investments, floor space of the line, and inventory. Therefore appropriate buffer storage size must be determined in order to reduce manufacturing cost while maintaining a desirable production rate [5].

Determining buffer sizes and their appropriate positions is still a challenging problem. Substantial researches have been conducted on production line evaluation and optimization [6]. Buzacott derives an analytic formula for the production rate of two-machine, one-buffer lines in a deterministic process time model [7]. Chow discusses two factors that cause difficulties in solving the buffer allocation problem (BAP) [8]. Firstly it has not been possible to derive an analytical relation between performance of the transfer line and the storage capacities distribution in the line. Secondly, buffer sizing is a combinatorial optimization problem. Searching for an optimal solution by experimenting with a real line or a simulation model is time consuming. The invention of decomposition methods with unreliable machines and finite buffers [9] and its corresponding DDX algorithm [10] enables the numerical evaluation of the production rate of lines having more than two machines.

More recently, Huang et al. consider a flow shop-type production system and use a dynamic programming approach to maximize its production rate or minimize its work in process under a certain buffer allocation strategy [11]. Diamantidis and Papadopoulos [12] present a dynamic programming algorithm for optimizing buffer allocation

M. Amiri · A. Mohtashami (✉)
Allameh Tabataba'i University,
Tehran, Iran
e-mail: mohtashami@st.atu.ac.ir

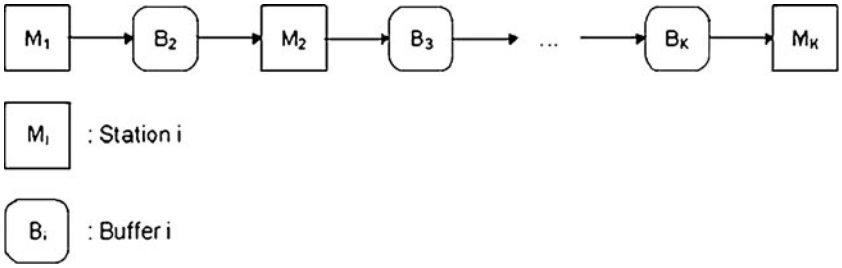

Fig. 1 A production line with K stations and $k-1$ buffer storages [2]

based on the aggregation method given by Lim et al. [13]. Smith and Cruz solve the BAP for general finite buffer queuing networks in which they minimize buffer space cost under the production rate constraint [14]. Manitz proposes a queuing model-based analysis of assembly lines with finite buffers and general service times, and he uses decomposition approach to analyze and allocate buffer storages [15]. Diamantidis and Papadopoulos analyze a two-workstation one-buffer flow line with parallel unreliable machines [16]. They assume that the process time, time between failures, and repair time of the parallel machines at each workstation are exponentially distributed. They employ a recursive algorithm that generates the transition matrix for any value of the intermediate buffer capacity, and through this, they encounter with the problem and solve it. Radhoui et al. develop a joint quality control and preventive maintenance policy for a randomly failing production system producing conforming and nonconforming units. The considered system consists of one machine designed to fulfill a constant demand. In order to palliate perturbations caused by the stopping of the machine to undergo preventive maintenance or an overhaul, a buffer stock is built up to ensure the continuous supply of the subsequent production line [17]. Sabuncuoglu and Gocgun develop a heuristic procedure to allocate buffers in serial production lines to analyze throughput [18]. Their procedure assumes that failure and repair times are exponentially distributed, and process times are deterministic. Vergara and Kim present a buffer placement method for serial production line in order to analyze throughput [19]. They assume that the process times are exponentially distributes, and the time between failures and the repair times are deterministic. Romulo et al. present a model to determine the optimal length of continuous production periods between maintenance actions and the optimal buffer inventory to satisfy demand during preventive maintenance or repair of a manufacturing facility [20]. Colledani and Tolio present an analytical method for evaluating the performance of production system, jointly considering in a unique framework quality and production logistic performance measures [21]. They assume that the stations process components by the same deterministic process time, scaled to time units. Abdul-Kader and Gharbi address the problem of capacity estimation of a multi-product production line composed of m unreliable workstations and $(m-1)$ intermediate buffers [22]. They propose a simulation-based experimental design methodology to improve the performance of the production line not to determine the exact amount of each intermediate buffer. In other words, their methodology determines just the effects of buffers with comparing to each other, on the cycle time of production line.

The aim of this paper is to propose a mathematical model (formulation) and also a methodology as a solution to maximize production rate (throughput of the line in a time range) and to minimize intermediate buffer storages under some constraints for unreliable production lines. Many of the previous works on this subject are limited because of considering some assumptions which could not be applied to all the problems in real cases. Process times and (or) time between failures are considered deterministic or exponentially distributed in many researches (e.g., [7, 11, 12, 15, 16, 18, 19, 21]). Also maintenance and repairing of machines and the nondeterministic nature of repair times are ignored as a fact in some researches.

According to previous researches, considering deterministic and (or) exponential distribution functions of parameters, some methods like queuing networks and decomposition can be used for buffer allocation, but in production systems, if we

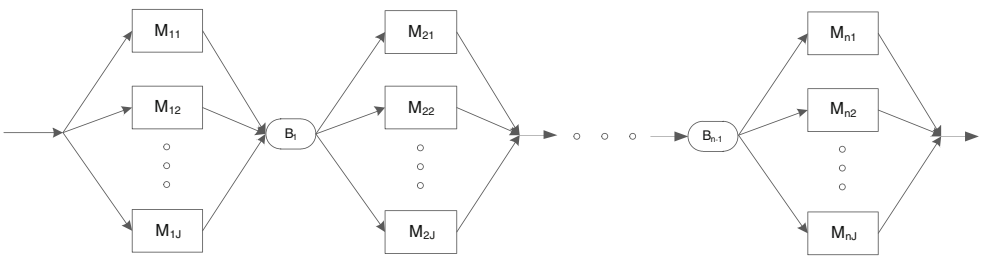

Fig. 2 Series–parallel production line [3]

assume some parameters as general distribution functions, these methods cannot be applicable. This paper relaxes these restrictions by proposing a simulation-based methodology which can considers general function distributions (e.g., normal, gamma, Weibull, uniform) for all parameters of production lines to solve the buffer allocation problem.

Recently, new approaches in the area of optimization research are presented to further improve the solution of optimization problems with complex nature. Over the past few years, the studies on evolutionary algorithms have shown that these methods can be efficiently used to eliminate most of the difficulties of classical methods. Evolutionary algorithms are widely used to solve engineering optimization problems with complex nature. Various research works are carried out to enhance the performance of evolutionary algorithms [23–27]. Yildiz proposes a new design optimization framework based on immune algorithm and Taguchi's method. He describes an innovative optimization approach that offers significant improvements in performance over existing methods to solve shape optimization problems [28]. Yildiz in another work proposes an effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry [29]. He proposes a novel optimization approach that is a new hybrid optimization approach based on the particle swarm optimization algorithm and receptor editing property of the immune system [30]. He introduces an improved harmony search algorithm to solve engineering optimization problems. To demonstrate the effectiveness and robustness of the proposed approach, he applies the approach to an engineering design and manufacturing optimization problem taken from the literature [31]. He introduces a new hybrid optimization approach by combining immune algorithm and simulated annealing algorithm. The main aim of his paper is to develop a novel optimization approach for the solution of engineering optimization problems [32]. He presents a new hybrid optimization approach based on immune algorithm and hill climbing local search algorithm. He claims that this research is the first application of immune algorithm to the optimization of machining parameters in the literature [33]. Yildiz et al. propose hybrid multiobjective shape design optimization using Taguchi's method and genetic algorithm. The objective is to contribute to the development of more efficient shape optimization approaches in an integrated optimal topology and shape optimization area with the help of genetic algorithms and robustness issues [34]. Yildiz and Saitou present a new method for synthesizing structural assemblies directly from the design specifications, without going through the two-step process. They present a new formulation in a continuum design domain, which enhances the ability to present complex structural geometry observed in real-world products [35]. This paper is organized as follows: Section 2 describes the basic assumptions, Section 3 presents the proposed methodology, in Section 4 numerical results are provided, and Section 5 concludes the paper.

2 Basic assumptions

The basic assumptions regarded in this research are listed below:

- The production system is considered as a queuing network;
- The entrances to the system may be from different nodes and time between entrances are generally distributed;
- The system consists of M stations, m_i, $i=1,\ldots, M$, and $M-1$ buffers b_i, $i=1,\ldots, M-1$, separating each consecutive pair of machines.
- Machine m_i is said to be starved during a time slot if b_{i-1} is empty; machine m_i is said to be blocked during a time slot if b_i is full and machine m_{i+1} is either under repair or blocked.
- Buffer b_i is characterized by its size, n_i, $i=1,\ldots, M-1$. One part is inserted into the buffer if upstream machine is up and neither starved nor blocked. One part is removed from the buffer if the downstream machine is up and neither starved nor blocked.
- Process time of the stations are generally distributed;
- The probability of machines failure is noted, and time between failures is generally distributed;
- Repairing time is considered, and it is generally distributed;

3 The proposed methodology

In this section we propose a methodology to design buffer storage capacity through following steps.

3.1 Simulation

This methodology employs simulation for a more realistic representation of the dynamic behavior of a system. Discrete event simulation is a very effective way of estimating almost any system performance given that the input data are accurate [36]. Computer simulation offers the advantage of tracing complex system processes (considering any distribution functions and complex relationship among the stations, elements, components, and sections), providing timely information on operating characteristics, and for these reasons, it has been adopted in this research.

3.2 Building meta-model using design of experiments

A simulation model is a representation of a real word system, whereas the term meta-model referred to herein is a

mathematical approximation of a simulation model [37, 38]. Meta-models are developed to obtain a better understanding of the nature of the true relationship between the input variables and the output variables of the system under study [39]. A number of mathematical functions have been used to develop meta-models [40]. The type of meta-models most commonly used in simulation studies have been polynomial regression models [38, 41–46]. In this study, two-level factorial design has been used to build a meta-model based on a detailed, discrete event simulation model. Factorial designs are widely used in experiments involving several factors where it is necessary to study the joint of the factors on a response [47]. The factorial design method is a statistical technique that evaluates at the same time all process (or any focus of study) variables in order to determine which ones really exert significant influence on the final response, which gives a better analysis of the response [48]. All variables are called factors, and different values chosen to study the factors are called levels [49, 50]. Important trends may be observed with these factorial designs, and the effect of each independent variable on the dependent one is estimated [51].

A complete replicate of a two-level factorial design requires $2\times2\times\ldots\times2=2^k$ observations and is called a 2^k factorial design (k denotes the number of factors). If we choose some designs with higher levels like a three-level factorial design (3^k), such a design requires $3\times3\times\ldots\times3=3^k$ observations. For example if we consider 20 buffer storages, the full 2^k design requires 2^{20}=1,048,576 observations and the full 3^k design requires 3^{20}=3,486,784,401 observations. Therefore with respect to 20 buffer storages, the 3^k design requires 3,485,735,825 observations more than 2^k (about 3,325 times more) that is too time consuming. These results indicate that the 2^k design is more efficient in this problem. But if the 2^k regression results are not satisfactory, other designs like 3^k design can be an alternative.

When a relatively large number of factors are evaluated, the total number of combinations may be too large. Furthermore, the high-order interactions (third, fourth, or superior) are small and may be mixed with the standard deviation of the effects. In this case, it is advisable and convenient to use a fractional factorial design (2^{k-p} case). The number of combinations is diminished, and the most important effects are determined but at the cost of decreasing the accuracy of the meta-model [49]. For example if we consider 20 buffer storages, the full 2^k design requires 2^{20}=1,048,576 observations, whereas considering a fractional factorial design like 2^{20-12} needs just 256 observations indeed at the cost of decreasing the accuracy of the meta-model. In the interpretation of the results generated by a complete or fractional factorial design, it is necessary to decide which calculated effects are significantly different from zero. The usual practice is using the concept of statistical significance (generally 95% of confidence). When analyzing the results of a factorial design, two statistic parameters are of relevance. The t statistics of a factor is obtained by the division of its effect by its error. This statistic parameter is dependent on the freedom degree, which is calculated by the subtraction of the number of calculated effects from the total number of experiments/trials available. The higher the t statistics, the higher is the significance of the corresponding factor. On the other hand, the p level, which represents the probability of error that is involved in accepting the effect as valid, is a decreasing index of the reliability of a result. The higher the p level, the less one can believe that the observed relation between factor and effect is reliable. The common practice is to consider 95% of confidence in a result, so that for an effect to be considered statistically significant, its p level must be less than 0.05. The two levels evaluated in a factorial design are coded by (+) and (−), representing the upper and lower levels, respectively [48]. In this paper we use design of experiments (DOE) and simulation to fit a meta-model to the production rate as response (y), considering the buffers as factors (X_i). In most factorial design problems, a low-order polynomial in some region of the independent variables is employed. If the response is well modeled by a linear function of the independent variables, then the approximating function is the first-order model (Eq. 1).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + e \tag{1}$$

If the statistics tests on the first-order meta-model are not acceptable or satisfactory, a second-order meta-model should be chosen consecutively (Eq. 2).

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum_{i<j} \sum \beta_{ij} x_i x_j + e \tag{2}$$

where β_0 denotes regression intercept, β_i denotes main or first-order effect of factor i, X_i denotes value of the factor i, β_{ii} denotes quadratic effect of factor i, β_{ij} denotes interaction between the factor i and j ($i\neq j$), e denotes fitting error of the regression model, and n denotes the number of factors.

Meta-model validation is determined based on the model's purpose [37, 52]. In this study, the validity of the meta-model with respect to the simulation model is determined by examining the model fit diagnostics. A lack-of-fit test is used to ascertain if the model adequately fits the data. A check of the distribution of the residuals leads to the determination of the validity of some of the model assumptions [53]. Comparing the meta-model output and simulation output is another way to evaluate the validity of the meta-model [41].

The methodology to build a regression meta-model (for production rate) we use the following:

Step 1: Define the problem. The problem to be solved and its boundaries need to be identified. Its limitations

and the factors to be considered should be defined (in this problem, the factors are buffer storages).

Step 2: Define the range of the factors (upper and lower limit of buffer storages).

Step 3: Specify the form of the meta-model.

Step 4: Develop the experimental design so that information on the response variable at a variety of input factors can be collected in an efficient way.

Step 5: Build a simulation model of production line and use the experimental design to generate the outputs (responses).

Step 6: Develop the regression meta-model. The regression meta-model to predict the dependent variable (production rate) can be developed using the output generated for the dependent variable (using simulation) and the input values obtained through the experimental design.

Step 7: If the meta-model is satisfactory, then stop; else, go to step 3.

3.3 Problem formulation

The first objective of the mathematical model is to maximize production rate:

$$\text{Max } Z_1 = f(x_i) \quad i = 1, 2, ..., n \tag{3}$$

Equation 3 is a regression meta-model, where X_i (decision variable) denotes the buffer size of station i, n denotes the number of buffer storages, and $f(X_i)$ estimates the production rate of the production line based on X_i. This regression meta-model is obtained using DOE and simulation (according to Section 3.2) with considering production rate as the response variable and X_i as the independent variables. This regression meta-model can be a first-order or higher order polynomial based on the validation tests (Section 3.2).

More buffer storages lead to higher production rate, yet at the cost of increasing, the holding cost of intermediate inventories. Therefore the second objective is to minimize the total buffer storages of the line.

$$\text{Min } Z_2 = \text{Total buffer size} = \sum_{i=1}^{n} X_i \tag{4}$$

In this model there are some constraints that are in relation with buffer size of each station. For environmental reasons (layout limitations), each buffer storage of the stations varies from a lower value to an upper value.

$$L_i \le X_i \le U_i \qquad \forall i = 1 \text{ to } n \tag{5}$$

$$X_i \ge 0 \text{ and integer} \qquad \forall i = 1 \text{ to } n \tag{6}$$

where U_i denotes the upper and L_i denotes the lower limit of buffer storages, X_i (decision variable) is the nonnegative and integer variable.

3.4 Solving the model

The mentioned problem is a multiobjective model (production rate objective and total buffer size objective) and should be solved to obtain the decision variables amount. A powerful method for multiobjective problem solving that generates different nondominated solutions based on the objective's weights is LP metric method. Equation 7 indicates the LP metric method for multiobjective problem solving [54, 55].

$$\min = \left[\sum_{j=1}^{m} W_j^p \left| \frac{f_j^* - f_j}{f_j^* - f_i^-} \right|^p \right]^{\frac{1}{p}} \tag{7}$$

where W_j denotes the weight of the objective j, m denotes the number of objectives, f_j^* denotes optimum value (ideal) of objective j, f_j^- denotes antioptimum value (anti-ideal) of objective j. Equation 7 combines the model objectives and finds the minimum of combined function. It seeks the optimum amount of decision variables in feasible region of the main formulated model as following:

$$x \in s \tag{8}$$

where "X" denotes the solution vector and "S" denotes the feasible region of the main formulated model.

The LP metric objective function is nonlinear for $p>1$ and also the first objective function (Eq. 3) may be nonlinear, thus genetic algorithm (GA) in this methodology is an appropriate optimization method and can be employed to find the near optimal solutions. Genetic algorithm is a powerful algorithmic approach that has been applied with great success to many difficult combinatorial problems. Genetic algorithm is a probabilistic search method stimulated by genetic evolution [56]. Genetic algorithms have proven to be very adaptable to a great variety of different optimization tasks [57]. The algorithms work with a population of possible solutions, which suffers evolution during the generations, an analogy borrowed from the Darwin's evolutionary theory. Each solution is coded as a collection (chromosome) of binary or real strings; each string represents a variable in the solution. The evolution is achieved by some genetic operators as reproduction, crossover, and mutation. The survival of the fittest is achieved by the assignment of a fitness function, usually defined as the objective function for the unconstrained optimization problem, or a combination of the objective function and a penalty function for constrained optimization [58, 59]. The set of solutions (i.e., the population) per iteration (generation) is fixed. At each iteration, pairs of individuals are selected

Table 1 GA parameters

Population size	Crossover rate	Mutation rate	Selection function	Crossover	Mutation
100	0.9	0.01	Tournament	Uniform	Uniform

randomly and are recombined into new solutions (crossover operator). A random change on the offspring generation is optionally applied (mutation operator). The newly created solutions are evaluated according to the fitness function [57]. Genetic algorithm can also borrow the idea from nature of coexistence of multiple niches in order to deal with multimodal optimization. A sharing concept (in an analogy to the sharing, in nature, of available resources, such as land and food) may be introduced artificially in GA population. This allows coexistence of multiple optimal solutions (both local and global). More details about the niching in GA may be found in [59].

GA often performs well in global search, but they are relatively slow and trapped in converging to local optimal [60]. On the other hand, the local improvement methods, such as gradient-based (line search, LS) procedures, can find the local optimum in a small region of the search space, but they are typically poor in a global search. Therefore, various strategies of hybridization have been suggested to improve performance of simple GAs [60, 61]. These hybridizations usually involve incorporating a neighborhood search heuristic as a local improver into a basic GA loop of recombination and selection. That is, local improver is applied to each newly generated offspring to move it to a local optimum one before inserting it into the enlarged population [62]. In this manner, GA is used to perform global exploration among a population (i.e., as a diversification tool), while the local improver is used to perform local exploitation around chromosomes (i.e., as an intensification tool). The general structure of the proposed hybrid genetic algorithm (HGA) for the problem is described as follows:

Step 1. Initialization: create an initial population of population size solutions using constructive heuristics and randomly generated solutions.

Step 2. Recombination: recombine the solutions in the current population using genetic operators to create new individuals.

Step 3. Improvement: apply the local improvement method (LS) to replace each offspring with a local optimum one, and insert the improved offspring into the enlarged population.

Step 4. Selection: select population size solutions from the chromosomes in the enlarged population to form the next generation, and determine the best solution in the new population.

Step 5. Iteration: repeat steps 2–4 until the termination condition is reached.

Setting GA parameters are mostly based on empirical observations with respect to the problem variability [63]. The GA parameters used in this study are retrieved from the available design literature [61, 64–67]. The GA parameters are set as listed in Table 1.

4 Numerical example

Consider a production line with 18 stations (A to R) and one operating machine for each station. Figure 3 demonstrates the mentioned production line. In this production line, circles indicate the stations and triangles indicate buffer storages. In this demonstration, raw materials enter to the production line from different nodes and pass among the stations to get process. Station M assembles the outputs of the station I and J; station O assembles the outputs of station M and N; station Q assembles the outputs of station P and L; and station R performs some complementary operations, and the output of the station R is the final product of the production line. We now want to determine the optimum

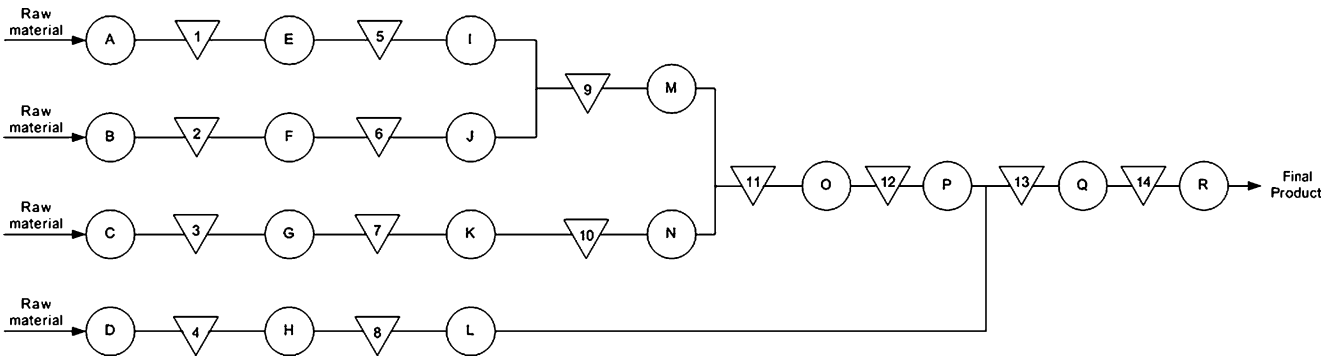

Fig. 3 A production line

Table 2 Process time, time between failures, and repair time of each station

Station	Process time	Time between failures	Repair time
A	Weibull (20, 10, 2)	Weibull (1,000, 70, 20)	Gamma (1,000, 250, 80)
B	Weibull (25, 11, 4)	Weibull (9,000, 200, 18)	Gamma (1,200, 250, 80)
C	Weibull (30, 11 4)	Weibull (9,500, 200, 18)	Gamma (200, 250, 80)
D	Weibull (20, 10, 6)	Weibull (9,000, 200 18)	Gamma (1,250, 250, 80)
E	Weibull (35, 4, 3)	Weibull (15,000, 130, 20)	Gamma (400, 300, 52)
F	Weibull (37, 9, 6)	Weibull (17,000, 240, 90)	Gamma (2,354, 300, 52)
G	Weibull (36, 10, 6)	Weibull (18,000, 240, 90)	Gamma (2,000, 325, 52)
H	Weibull (40, 9, 6)	Weibull (1,700, 240, 90)	Gamma (2,000, 300, 52)
I	Weibull (30, 11, 9)	Weibull (11,000, 350, 18)	Gamma (500, 300, 52)
J	Weibull (45, 16, 9)	Weibull (9,000, 180, 40)	Gamma (700, 300, 52)
K	Weibull (46, 12, 9)	Weibull (9,000, 180, 40)	Gamma (756, 300, 52)
L	Weibull (48, 11, 12)	Weibull (9,400, 140, 70)	Gamma (730, 300, 52)
M	Exponential (40)	Weibull (12,000, 200, 80)	Gamma (1,500, 300, 52)
N	Weibull (44, 17, 9)	Weibull (8,650, 180, 40)	Gamma (700, 300, 52)
O	Weibull (27, 12, 10)	Weibull (8,100, 300, 25)	Gamma (2,659, 140, 60)
P	Exponential (43)	Weibull (8,230, 250, 30)	Gamma (2,768, 160, 65)
Q	Weibull (26, 12, 10)	Weibull (7,700, 324, 25)	Gamma (2,546, 140, 60)
R	Weibull (33, 14, 10)	Weibull (7,900, 400, 25)	Gamma (2,434, 143, 60)

capacity of buffer storages of each station. Table 2 presents the process time, time between failures, and repair time of each station. Table 3 presents the lower limit and upper limit of the buffer storages (decision variables). According to the first step of the proposed methodology, the queuing network of Fig. 3 is simulated by MATLAB software product based on the data in Table 2.

At this step we should choose an appropriate design of experiments which can be two level full factorial design (2^k). However considering the large number of stations and buffer storages, the full factorial design (2^k) is too time consuming. As the number of factors in a 2^k factorial design increase, the number of runs required for a complete replicate of the design rapidly outgrows the resources of most experiments [47]. In spite of the better results of performing full factorial design or even response surface designs, we use two-level fractional factorial design (2^{k-p}) with low resolution to illustrate the performance of numerical example of the proposed methodology. Although it is still recommended if the 2^{k-p} design regression results are not satisfactory, researchers use full factorial or response surface designs for problem solving in real cases. The term "resolution" is used in DOE literature to express what can be estimated from a particular fractional design. For example a resolution V fractional factorial design would give main effects and two-way interactions unconfounded with each other. Consequently, a 99.8% reduction in data collection can be achieved by using a 2_v^{14-9} fractional design (14 factor), which represents only 32 combinations of factors (e.g., 320 simulations, if 10 replications are performed) in comparison with two-level full factorial design. These 32 combinations are generated in DOE++ software product.

Table 3 Lower limit and upper limit of buffer storages

Buffer	Lower limit	Upper limit
1	0	200
2	0	180
3	0	250
4	0	170
5	0	200
6	0	240
7	0	160
8	0	210
9	0	150
10	0	220
11	0	140
12	0	230
13	0	200
14	0	150

2_v^{14-9} fractional design generates 32 case, based on upper and lower limits of factors coded by +1 (upper limit) and −1 (lower limit). Each case of these combinations is simulated in MATLAB software product to analyze the results on the response (production rate). The simulation runs are performed for 5,000,000 time unit on a PC with Core 2 Duo CPU (2.00 GHz) and 1.99 GB of RAM and each run takes long about 5 to 6 min. For example

Table 4 Analysis of variance for response (considering the main effects)

Source of variation	Degrees of freedom	Sum of squares	Mean squares	*F* ratio	*p* value
Model	14	3.43E+07	2.45E+06	1.607	0.1752
Main effects	14	3.43E+07	2.45E+06	1.607	0.1752
Residual	17	2.59E+07	1.53E+06		
Lack of fit	17	2.59E+07	1.53E+06		
Total	31	6.03E+07			
R square	56.96%				

case "1, −1, −1, 1, 1, 1, −1, −1, 1, −1, −1, 1, 1, −1" that is one of the 32 cases in two-level fractional factorial design for 14 factor indicates a run with upper limit of X_1, X_4, X_5, X_6, X_9, X_{12}, X_{13} and lower limit of X_2, X_3, X_7, X_8, X_{10}, X_{11}, X_{14}. Simulation of this case leads to produce 430 final products.

In accordance with the methodology presented in Section 3.2, a regression meta-model considering only the main effects of the factors is developed. Analysis of variance for response is presented in Table 4.

Fisher's *F* test in Table 4 for regression does not demonstrate a high significance for the regression model and the obtained meta-model explains only 56.96% of the output (production rate) variability. Thus, regarding our objectives, the results of the used meta-model are not satisfactory. Consequently, a new regression meta-model is evaluated considering now both the main effects of the factors and their two-way interactions. Analysis of variance for response of the new meta-model is presented in Table 5. Fisher's *F* test in Table 5 for regression demonstrates a high significance for the regression model (F=17.7984 and P=0.0545). The larger the *F* value, the more important it is that the factor influenced the response variable. The meta-model obtained explains now more than 95% of the output variability (*R* square=99.61%). The normality test for residual has been performed using one-sample Kolmogorov–Smirnov test, and results are shown in Table 6. The amount of *p* value (0.361) approves the residual normality as another model fit diagnostics. Therefore we have considered that it is satisfactory regarding our objectives. Results of the regression analysis are presented in Table 7.

Considering Table 7 data, the regression meta-model is as follows:

$$\begin{aligned}\text{Production rate} = {} & 931.25 + 55.8125{*}X_1 + 24{*}X_2 + 313.6875{*}X_3 \\ & +87.6875{*}X_4 + 325.1875{*}X_5 + 329.25{*}X_6 \\ & +97.375{*}X_7 + 372.8125{*}X_8 + 464.375{*}X_9 \\ & +411.375{*}X_{10} + 202.1875{*}X_{11} + 300.0625{*}X_{12} \\ & +135{*}X_{13} + 258.25{*}X_{14} + 387.3125{*}X_1{*}X_2 \\ & +76.5{*}X_1{*}X_3 + 296.75{*}X_1{*}X_4 + 45.75{*}X_1{*}X_5 \\ & +125.4375{*}X_1{*}X_6 + 325.3125{*}X_1{*}X_7 \\ & +7.25{*}X_1{*}X_8 + 15.6875{*}X_1{*}X_9 \\ & +110.0625{*}X_1{*}X_{10} + 284.875{*}X_1{*}X_{11} \\ & +125.5{*}X_1{*}X_{12} + 278.3125{*}X_1{*}X_{13} \\ & +143.3125{*}X_1{*}X_{14} + 360.625{*}X_3{*}X_{12} \\ & +314.0625{*}X_3{*}X_{14}\end{aligned} \tag{9}$$

This meta-model (Eq. 9) estimates the production rate (first objective of the model), based on the different amounts of the buffer storages (x_1 to x_{14}) and the aim is to maximize this objective.

In order to test the validity of the meta-model, we use the approach suggested by Durieux and Pierreval [41]. Ten combinations of input values of the design factors are randomly selected. The simulation output is then compared to the predicted values for these combinations using Eq. 9. These results are given in Table 8. The absolute error is given by the following formula:

$$\textit{Absolute error} = \frac{100 * |\textit{meta model output} - \textit{simulation output}|}{\textit{Simulation output}} \tag{10}$$

The average absolute error between the simulation model and regression meta-model in the ten experimental runs is

Table 5 Analysis of variance for response (considering the main effects and the interaction between factors)

Source of variation	Degrees of freedom	Sum of squares	Mean squares	*F* ratio	*p* value
Model	29	6.01E+07	2.07E+06	17.7984	0.0545
Main effects	14	3.43E+07	2.45E+06	21.0816	0.0462
2-way interaction	15	2.57E+07	1.71E+06	14.7341	0.0653
Residual	2	2.33E+05	1.16E+05		
Lack of fit	2	2.33E+05	1.16E+05		
Total	31	6.03E+07			
R square	99.61%				

Table 6 One-sample Kolmogorov–Smirnov test

N		32
Normal parameters	Mean	0.0000
	Standard deviation	86.63995
Most extreme differences	Absolute	0.163
	Positive	0.163
	Negative	−0.163
Kolmogorov–Smirnov *Z*		0.924
p value		0.361

3.65%. Thus, we consider this meta-model to be accurate enough. The second objective of the model is to minimize the total buffer size (Eq. 11).

$$\text{Min} = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{12} + X_{13} + X_{14} \quad (11)$$

The combined objective function of production rate maximization and total buffer size minimization using LP metric is shown in Eq. 12.

$$\text{Min} = \left[W_1{}^p \left| \frac{-7205.0625 - (-\text{Eq.9})}{-6754.125} \right|^{\text{p}} + W_2^p \left| \frac{-14 - (\text{Eq.10})}{-28} \right|^p \right]^{\frac{1}{p}} \quad (12)$$

The optimum value (ideal) of Eq. 9 is $f_1{}^* = 7{,}205.0625$ (considering all decision variables equal to 1). 7,205.0625 is a negative number in Eq. 12 because of converting the production rate maximization to minimization function. The antioptimum value (anti-ideal) of Eq. 9 is $f_1{}^- = 450.9375$ (considering all decision variables equal to −1). Considering negative number as −450.9375 in Eq. 12 (because of converting the production rate maximization to minimization function), therefore, it can be concluded that $f_1{}^* - f_1{}^- =$

Table 7 Regression analysis considering the main effects and the interaction between factors

Term	Effect	Coefficient	Standard error	*T* value	*p* value
Intercept		931.25	60.2986	15.444	0.0042
X_1	111.625	55.8125	60.2986	0.9256	0.4524
X_2	48	24	60.2986	0.398	0.7291
X_3	627.375	313.6875	60.2986	5.2022	0.035
X_4	175.375	87.6875	60.2986	1.4542	0.2831
X_5	650.375	325.1875	60.2986	5.393	0.0327
X_6	658.5	329.25	60.2986	5.4603	0.0319
X_7	194.75	97.375	60.2986	1.6149	0.2477
X_8	745.625	372.8125	60.2986	6.1828	0.0252
X_9	928.75	464.375	60.2986	7.7013	0.0164
X_{10}	822.75	411.375	60.2986	6.8223	0.0208
X_{11}	404.375	202.1875	60.2986	3.3531	0.0786
X_{12}	600.125	300.0625	60.2986	4.9763	0.0381
X_{13}	270	135	60.2986	2.2389	0.1545
X_{14}	516.5	258.25	60.2986	4.2829	0.0504
X_1X_2	774.625	387.3125	60.2986	6.4232	0.0234
X_1X_3	153	76.5	60.2986	1.2687	0.3322
X_1X_4	593.5	296.75	60.2986	4.9213	0.0389
X_1X_5	91.5	45.75	60.2986	0.7587	0.5272
X_1X_6	250.875	125.4375	60.2986	2.0803	0.173
X_1X_7	650.625	325.3125	60.2986	5.395	0.0327
X_1X_8	14.5	7.25	60.2986	0.1202	0.9153
X_1X_9	31.375	15.6875	60.2986	0.2602	0.8191
X_1X_{10}	220.125	110.0625	60.2986	1.8253	0.2095
X_1X_{11}	569.75	284.875	60.2986	4.7244	0.042
X_1X_{12}	251	125.5	60.2986	2.0813	0.1729
X_1X_{13}	556.625	278.3125	60.2986	4.6156	0.0439
X_1X_{14}	286.625	143.3125	60.2986	2.3767	0.1406
X_3X_{12}	721.25	360.625	60.2986	5.9807	0.0268
X_3X_{14}	628.125	314.0625	60.2986	5.2085	0.0349

Table 8 Validation of the meta-model

Run	1	2	3	4	5	6	7	8	9	10	Average absolute error
X_1	200	0	200	200	0	0	0	3	200	200	
X_2	0	180	0	0	180	180	0	8	170	180	
X_3	0	0	250	0	250	0	250	243	250	250	
X_4	0	0	0	170	170	0	0	6	170	170	
X_5	0	0	0	0	0	200	200	200	198	200	
X_6	240	240	0	240	0	240	240	240	239	240	
X_7	160	160	160	0	0	160	0	4	159	160	
X_8	210	0	0	0	0	0	210	199	208	210	
X_9	0	150	150	150	150	150	150	143	150	150	
X_{10}	220	220	220	220	220	0	220	216	219	220	
X_{11}	140	0	0	140	140	140	0	0	140	140	
X_{12}	230	0	230	0	230	230	230	219	222	230	
X_{13}	0	200	200	0	0	0	0	0	200	200	
X_{14}	0	150	0	150	0	0	150	143	149	150	
Simulation	755	466	822	974	811	620	4,584	4,388	7,323	7,323	
Meta-model	780	491	796	948	836	594	4,701	4,588	7,057	7,205	
Absolute error (%)	3.311	5.365	3.163	2.669	3.083	4.194	2.552	4.558	3.632	1.611	3.414

(−7,205.0625)−(−450.9375)=−6,754.125. As the same, the optimum value (ideal) of Eq. 11 is $f_2^{*}=-14$, and the anti-optimum value (anti-ideal) of Eq. 11 is $f_2^{-}=14$, therefore it can be concluded that $f_2^{*}-f_2^{-}=(-14)-(14)=-28$. Buffer storages are bounded to an upper limit and lower limit as follows:

$$-1 \le X_i \le 1 \quad \forall i = 1 \text{ to } 14 \tag{13}$$

Considering Eq. 12 as objective function and constraints of the model, the problem has been solved by HGA with the line search method which has been coded in MATLAB software product given in the GA parameters in Table 1. GA can do global search in entire space, but there is no way for exploring the search space within the convergence area of generated GA. Therefore, it is sometimes impossible or insufficient for GA to locate an optimal solution in the optimization problems with complex search spaces and constraints. To overcome this weakness, hybridization of GA with conventional search techniques of line search is adopted in this section.

The results of problem solving for different objective's weights considering $p=1$ are not satisfactory regarding our objectives, therefore we solve the model again considering $p=2$. These results are shown in Table 9.

Each X_i ($i=1$ to 14) row is divided into two parts, the upper row indicates the coded values (CV) that is the output of the optimization model and the lower row indicates the noncoded values (NCV) that are the problem solutions. The CV and NCV can be converted to each other by Eq. 14 [47].

$$\mathrm{CV} = \frac{X_i - \frac{U_i+L_i}{2}}{\frac{U_i-L_i}{2}} \tag{14}$$

where U_i denotes the upper limit of X_i (noncoded value) and L_i denotes the lower limit of X_i. For example, considering X_3, the upper limit is equal to 250 and the lower limit is equal to 0 (Table 2), so for any noncoded value of X_3 between [0, 250], the coded value may be calculated by Eq. 14. For example:

$$X_3 = 0;\ \mathrm{CV} = \frac{0-\frac{250+0}{2}}{\frac{250-0}{2}} = -1$$
$$X_3 = 250;\ \mathrm{CV} = \frac{250-\frac{250+0}{2}}{\frac{250-0}{2}} = 1$$

Therefore the coded values (output of the model) have been converted to noncoded values by Eq. 14.

Each column in this table indicates a solution and decision maker (or decision makers) may choose one of them. For example, considering 60% for production rate objective's weight and 40% for total buffer size objective's weight, we have "X_1=199, X_2=150, X_3=250, X_4=46, X_5=15, X_6=239, X_7=155, X_8=181, X_9=149, X_{10}=215, X_{11}=140, X_{12}=229, X_{13}=198, X_{14}=149."

This solution results in production rate of 5,617 and total buffer size of 2,315. If the decision maker wants to increase the weight of minimizing buffer storages objective (e.g.,

Table 9 Results of problem solving (P=2)

W_1		0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
W_2		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
x_1	Coded	0.996	0.990	0.998	0.990	−0.960	−0.888	−0.999	−0.831	−0.842
	Noncoded	200	199	200	199	4	11	0	17	16
x_2	Coded	0.884	0.970	0.899	0.669	−0.993	−0.934	−0.980	−1.000	−0.502
	Noncoded	170	177	171	150	1	6	2	0	45
x_3	Coded	0.999	1.000	0.975	0.998	1.000	0.972	−0.664	−0.657	−0.796
	Noncoded	250	250	247	250	250	247	42	43	26
x_4	Coded	0.997	0.994	0.838	−0.456	−0.791	−0.949	−0.995	−0.864	−0.809
	Noncoded	170	169	156	46	18	4	0	12	16
x_5	Coded	0.982	0.987	0.040	−0.852	0.945	−0.765	−0.849	0.271	−0.681
	Noncoded	198	199	104	15	195	24	15	127	32
x_6	Coded	0.995	0.976	0.962	0.994	−0.950	−0.973	0.063	−0.860	−1.000
	Noncoded	239	237	235	239	6	3	128	17	0
x_7	Coded	0.992	0.997	0.863	0.944	−0.731	−0.999	−0.784	−0.840	−0.986
	Noncoded	159	160	149	155	21	0	17	13	1
x_8	Coded	0.985	0.756	0.963	0.729	0.991	0.382	0.902	−0.778	−0.711
	Noncoded	208	184	206	181	209	145	200	23	30
x_9	Coded	0.995	0.965	0.995	0.992	0.980	0.907	0.944	0.472	−0.582
	Noncoded	150	147	150	149	149	143	146	110	31
x_{10}	Coded	0.988	0.977	0.988	0.958	0.832	−0.406	−1.000	−0.906	−0.569
	Noncoded	219	217	219	215	202	65	0	10	47
x_{11}	Coded	0.999	0.987	0.988	0.995	−0.968	−0.932	−0.552	−0.946	−0.719
	Noncoded	140	139	139	140	2	5	31	4	20
x_{12}	Coded	0.934	0.996	0.994	0.987	0.962	0.976	−0.953	−0.944	−0.925
	Noncoded	222	229	229	229	226	227	5	6	9
x_{13}	Coded	1.000	0.908	0.841	0.980	−0.805	−0.925	−1.000	−0.288	−0.316
	Noncoded	200	191	184	198	19	7	0	71	68
x_{14}	Coded	0.984	0.989	0.995	0.986	0.847	0.996	−0.902	−0.808	−0.443
	Noncoded	149	149	150	149	139	150	7	14	42
Objective function of LP metric		0.10097	0.19827	0.29056	0.36958	0.35113	0.34039	0.27803	0.23369	0.16806
Objective function of throughput		7,057	6,974	6,538	5,617	3,965	2,880	2,021	976	213
Objective function of buffer size		2,674	2,647	2,539	2,315	1,441	1,037	593	467	383

W_1=50%; W_2=50%), the solutions are "X_1=4, X_2=1, X_3=250, X_4=18, X_5=195, X_6=6, X_7=21, X_8=209, X_9=149, X_{10}=202, X_{11}=2, X_{12}=226, X_{13}=19, X_{14}=139." This solution results in production rate of 3,965 and total buffer size of 1,441. As it can be seen, increasing the weight of total buffer size minimization objective, results in decrease of total buffer size but at the cost of decrease of production rate to.

5 Conclusion

This paper proposes a multiobjective formulation of the buffer allocation problem in unreliable production lines. Maximization of production rate and also minimization of total buffer size are two main objectives of this research. We present a methodology as a solution to overcome some limitations of the previous works. In many of the previous works, the solution methods have been proposed with assumptions of deterministic and (or) exponential distribution function of process times, time between failures, and repair times. However in practice these parameters may be nondeterministic and even generally distributed (e.g., normal, gamma, Weibull, uniform, etc.). Thus, this study employs a simulation-based approach to consider general distribution functions for all parameters of a production line like process times, time between failures, and repair times. The proposed

methodology employs 2^k fractional factorial design and simulation to build a meta-model to estimate the production rate of production line based on the buffer storages of the stations. Also it employs hybrid genetic algorithm with the line search method to solve the nonlinear problem and to determine some near optimal nondominated solutions for buffer allocation. The use of genetic algorithm for buffer allocation problem is not new, but the main contribution of this paper is to propose a methodology which is not confined to describing the parameters of production lines only by deterministic or exponential distribution functions and can overcome this limitation.

References

1. Nahas N, Ait-Kadi D, Nourelfath M (2006) A new approach for buffer allocation in unreliable production lines. Int J Prod Econ 103(2):873–881
2. Vidalis M, Papadopoulos C, Heavey C (2005) On the workload and 'phaseload' allocation problems of short reliable production lines with finite buffers. Comput Ind Eng 48(4):825–837
3. Nahas N, Nourelfath M, Ait-Kadi D (2008) Ant colonies for structure optimization in a failure prone series–parallel production system. J Qual Maint Eng 14(1):7–33
4. Shi C, Gershwin S (2009) An efficient buffer design algorithm for production line profit maximization. Int J Prod Econ 122(2):725–740
5. Massim Y, Yalaouib F, Amodeob L, Chateletc E, Zeblaha A (2010) Efficient combined immune-decomposition algorithm for optimal buffer allocation in production lines for throughput and profit maximization. Comput Oper Res 37(4):611–620
6. Dallery Y, Gershwin S (1992) Manufacturing flow line systems: a review of models and analytical results. Queuing Syst Theory Appl 12:3–94
7. Buzacott J (1967) Automatic transfer lines with buffer stocks. Int J Prod Res 5(3):183–200
8. Chow W (1987) Buffer capacity analysis for sequential production lines with variable processing times. Int J Prod Res 25(8):1183–1196
9. Gershwin S (1987) An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Oper Res 35(2):291–305
10. Dallery Y, David R, Xie X (1988) An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. IIE Trans 20(3):280–283
11. Huang M, Guang C, Pao L, Chou Y (2002) Buffer allocation in flow-shop-type production system with general arrival and service patterns. Comput Oper Res 29(2):103–121
12. Diamantidis A, Papadopoulos C (2004) A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines. Math Probl Eng 3:209–223
13. Lim J, Meerkov S, Top F (1990) Homogeneous, asymptotically reliable serial production line: theory and a case study. IEEE Trans Autom Control 35(5):524–534
14. Smith J, Cruz F (2005) The buffer allocation problem for general finite buffer queuing networks. IIE Trans 37(4):343–365
15. Manitz M (2008) Queuing model based analysis of assembly lines with finite buffers and general services times. Comput Oper Res 35 (8):2520–2536
16. Diamantidis C, Papadopoulos T (2009) Exact analysis of a two-workstation one-buffer flow line with parallel unreliable machines. Eur J Oper Res 197(2):572–580
17. Radhoui M, Rezg N, Chelbi A (2009) Integrated model of preventive maintenance, quality control and buffer sizing for unreliable and imperfect production systems. Int J Prod Res 47(2):389–402
18. Sabuncuoglu E, Gocgun Y (2006) Analysis of serial production lines: characterisation study and a new heuristic procedure for optimal buffer allocation. Int J Prod Res 44(13):2499–2523
19. Vergara H, Kim D (2009) A new method for the placement of buffers in serial production lines. Int J Prod Res 47(16):4437–4456
20. Romulo I, Pernando P, Valdes J (2004) Optimal buffer inventory and preventive maintenance for an imperfect production process. Int J Prod Res 42(5):959–974
21. Colledani M, Tolio T (2011) Integrated analysis of quality and production logistics performance in manufacturing lines. Int J Prod Res 49(2):485–518
22. Abdul-Kader W, Gharbi A (2002) Capacity estimation of a multi-product unreliable production line. Int J Prod Res 40(18):4815–4834
23. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. Proceeding of the fifth international conference on genetic algorithm, San Mateo, CA 416–423
24. Poidlena JR, Hendtlass T (1998) An accelerated genetic algorithm. Appl Intell 8:103–111
25. Deb K (1999) Evolutionary algorithms for multi-criterion optimization in engineering design. Kanpur Genetic Algorithm Laboratory KanGAL 1999–2001
26. Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203
27. Luh GC, Chueh CH, Liu WW (2003) MOIA: multi-objective immune algorithm. Eng Optim 35:143–164
28. Yildiz A (2009) A new design optimization framework based on immune algorithm and Taguchi's method. Comput Ind 60:613–620
29. Yildiz A (2009) An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry. J Mater Process Technol 209:2773–2780
30. Yildiz A (2009) A novel particle swarm optimization approach for product design and manufacturing. Int J Adv Manuf Technol 40:617–628
31. Yildiz A (2008) Hybrid Taguchi-Harmony search algorithm for solving engineering optimization problems. Int J Ind Eng Theory Appl Pract 15(3):286–293
32. Yildiz A (2009) Hybrid immune-simulated annealing algorithm for optimal design and manufacturing. Int J Mater Prod Technol 34 (3):217–226
33. Yildiz A (2009) A novel hybrid immune algorithm for global optimization in design and manufacturing. Robot Comput-Integr Manuf 25:261–270
34. Yildiz A, Ozturk N, Kaya N, Ozturk F (2007) Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm. Struct Multidiscip Optim 34(4):277–365
35. Yildiz A, Saitou K (2011) Topology synthesis of multicomponent structural assemblies in continuum domains. J Mech Des 133(1)
36. Lavoie P, Kenne J, Gharbi A (2009) Optimization of production control policies in failure-prone homogenous transfer lines. IEEE Trans 41:209–222
37. Kleijnen JPC, Sargent RG (2000) A methodology for fitting and validating meta-models in simulation. Eur J Oper Res 120:14–29
38. Kleijnen JPC (1987) Statistical tools for simulation practitioners. Marcel Dekker, New York
39. Noguera J, Watson E (2006) Response surface analysis of a multi-product batch processing facility using a simulation meta-model. Int J Prod Econ 102:333–343
40. Barton RR (1992) Meta-models for simulation input–output relations. In: Swain JJ, Goldsman D, Crain RC, Wilson JR (eds) Proceedings of the 1992 Winter Simulation Conference: 289–299

41. Durieux S, Pierreval H (2003) Regression meta-modeling for the design of automated manufacturing system composed of parallel machines sharing a material handling resource. Int J Prod Econ 1–10
42. Dengiz B, Akbay KS (2000) Computer simulation of a PCB production line: meta-modeling approach. Int J Prod Econ 63:195–205
43. Safizadeh MH (1990) Optimization in simulation: current issues and the future outlook. Nav Res Logist 37:807–825
44. Madu CN, Kuei CH (1994) Regression meta-modeling in computer simulation—the state of the art. Simul Pract Theory 2:27–41
45. Kleijnen JPC (1979) Regression meta-models for generalizing simulation results. IEEE SMC-9 2:93–96
46. Kleijnen JPC (1981) Regression analysis for simulation practitioners. J Oper Res Soc 32:35–43
47. Montgomery D (2001) Design and analysis of experiments. Wiley, New York
48. Costa C, Wolf M, Maria R, Rubens M (2005) Factorial design technique applied to genetic algorithm parameters in a batch cooling crystallization optimization. Comput Chem Eng 29(10):2229–2241
49. Barros N, Scarminio I, Bruns R (2001) Como fazer experimentos: pesquisa e desenvolvimento na ci^encia e naind'ustria. Editora da Unicamp Camp 20:83–184
50. Box G, Hunter W, Hunter J (1978) Statistic for experimenters—an introduction to design data analysis and model building. Wiley, New York
51. Rodrigues J, Toledo E, Maciel F (2002) A tuned approach of the predictive adaptative GPC controller applied to a fedbatch bioreactor using complete factorial design. Comput Chem Eng 26 (10):1493–1500
52. Kleijnen JPC (1995) Verification and validation of simulation models. Eur J Oper Res 82:145–162
53. Panis RP, Myers RH, Houck EC (1994) Combining regression diagnostics with simulation meta-models. Eur J Oper Res 73:85–94
54. Yu P (1973) A class of solutions for group decision problems. Manag Sci 19(8):936–946
55. Zeleny M (1982) Multiple criteria decision making. McGraw-Hill, New York
56. Holland J (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Michigan
57. Fuhner T, Jung T (2004) Use of genetic algorithms for the development and optimization of crystal growth processes. J Cryst Growth 266(1–3):229–238
58. Deb K (1998) Genetic algorithm in search and optimization: the technique and applications. In Proceeding of the International Workshop on Soft Computing and Intelligent Systems, Machine Intelligence Unit 58–87
59. Deb K (1999) An introduction to genetic algorithms. In Sadhana-Academy Proceedings in Engineering Sciences 24 Part 4–5: 293–315
60. Wang HF, Wu KY (2004) Hybrid genetic algorithm for optimization problems with permutation property. Comput Oper Res 31 (14):2453–2471
61. Gen M, Cheng R (1996) Optimal design of system reliability using interval programming and genetic algorithms. Comput Ind Eng 31 (1–2):237–240
62. Turabieh H, Sheta A, Vasant P (2007) Hybrid optimization genetic algorithm (HOGA) with interactive evolution to solve constraint optimization problems for production systems. Int J Comput Sci 1 (4):395–406
63. Elegbede C, Adjallah K (2003) Availability allocation to repairable systems with genetic algorithms: a multi-objective formulation. Reliab Eng Syst Saf 82(3):319–330
64. Gupta RK, Bhunia AK, Roy D (2009) A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliability of components. J Comput Appl Math 232:275–284
65. Kumar R, Izui K, Yoshimura M, Nishiwaki Sh (2009) Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization. Reliab Eng Syst Saf 94:891–904
66. Yokota T, Gen M, Li Y (1996) Genetic algorithm for non-linear mixed integer programming problems and its applications. Comput Ind Eng 30(4):905–917
67. Yang J, Hwang M, Sung T, Jin Y (2000) Application of genetic algorithm for reliability allocation in nuclear power plants. Reliab Eng Syst Saf 65(3):229–238