ELSEVIER

Int. J. Production Economics 70 (2001) 185–197

international journal of
production
economics

www.elsevier.com/locate/dsw

Minimizing WIP inventory in reliable production lines

H.T. Papadopoulos[a,*], M.I. Vidalis[b]

[a]Department of Business Administration, University of the Aegean, GR-821 00 Chios, Chios Island, Greece
[b]Department of Mechanical and Industrial Engineering, University of Thessaly, GR-383 34 Volos, Greece

Received 20 December 1997; accepted 11 May 2000

Abstract

This work deals with the well-known buffer allocation problem in short reliable production lines. The objective is to find the optimal buffer allocation (OBA) that minimizes the average work-in-process (WIP) inventory, subject to a minimum required throughput. The study leads to some insights concerning the evolution of the throughput and WIP as a function of the ordered buffer allocations. In the case of balanced lines, it is observed that the so-called self-similarity phenomenon prevails. It is also pointed out that the choice of the throughput level is critical for finding the OBA that minimizes the average WIP. Also a heuristic algorithm is proposed to find the OBA, which reduces the search space by over 50% compared to enumeration.

Keywords: Stochastic modelling; Production lines; Optimization; Buffer allocation; WIP inventory; Heuristic algorithms; Self-similarity

1. Introduction and literature review

Over the years a large amount of research has been devoted to the analysis of production lines. Much of this research has been concerned with the design of such systems in cases where there is considerable inherent variability in the processing times at the various stations. These are common and complicated situations when human operators/assemblers are involved.

An asynchronous production line is one in which each workstation can pass parts on when its processing is complete as long as there is available buffer space in the next station. Fig. 1 depicts a K-station production line with $K-1$ intermediate buffers, denoted by $B_2, \ldots, B_K$.

This type of line is subject to manufacturing blocking (or blocking after service) and starving. Material is not 'pulled' by demand. Instead, models are operated in a 'push' mode, i.e., it is assumed that there is always a part available when needed at the first workstation and space is always available for the last workstation to dispose a finished part. By this it is assumed that the first station is never starved and the last station is never blocked.

One of the key questions that designers face in a serial production line is the buffer allocation problem, that is, how much buffer storage to allow in a production line and how to distribute it across the line. This is an important question because buffers have a great impact on the efficiency of a production line. They compensate for the blocking and the starving of the stations. Unfortunately buffer storage is expensive, due to both its direct cost and the increase of the work-in-process (WIP) inventories. Besides, many times in practice, the

* Corresponding author. Tel.: +30-271-35101; fax: +30-271-35099.
E-mail address: hpap@aegean.gr (H.T. Papadopoulos).

0925-5273/01/$ - see front matter
PII: S0925-5273(00)00056-6

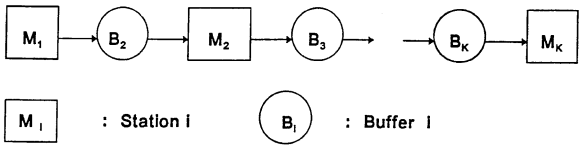

Fig. 1. A K-station production line with $K-1$ intermediate buffers.

limitation on the buffer space is imposed by the shop floor. The WIP includes the items at station 1 through station N including the one item being processed or held at station 1. However, it does not include any raw material held in front of station 1 as materials are assumed to be always available for process at station 1.

Previous work: There is much effort devoted to the modelling of production lines. For a systematic classification of papers dealing with stochastic modeling of manufacturing systems in general, the interested reader is addressed to a review paper by Papadopoulos and Heavey [1] and books such as [2–5].

A major classification of models can be done in two model classes: evaluative and generative (optimization) models. The former is concerned with the evaluation of systems' performance measures (see, for example, [6,7]), whereas models of the latter class try to optimize these measures by determining the optimal values of the decision variables involved. The approach of this paper falls into the second class since it produces buffer allocations subject to minimizing WIP, while achieving a minimum required throughput.

The majority of research papers on the OBA problem in production lines deal with the maximization of throughput (see, e.g., [8–11]). The paper by Conway et al. [8] reviews and extends much of the previous literature. Seong et al. [12] developed two heuristic algorithms for solving the buffer allocation problem in unreliable production lines.

This paper deals with the optimal allocation of a given total buffer space in short reliable production lines (i.e., we do not consider the case where the machines fail). The objective is to find the optimal allocation that minimizes the average work-in-process (WIP) inventory, provided the throughput exceeds a given level. This objective is different from the throughput maximization and it is realistic too. In practice it is usually required to find that the buffer allocation meets a specific order requirement minimizing simultaneously the average WIP inventory.

The contribution of the present work is that it presents some insights into the optimal buffer allocation (OBA) problem with the above objective. In the case of balanced lines it is observed by considering the throughput and WIP as a function of the ordered buffer allocations that the so-called self-similarity phenomenon prevails. It is also pointed out that the choice of the throughput level is critical for finding the OBA that minimizes the average WIP. Also, for the balanced case, a heuristic algorithm is proposed to find the OBA, which reduces the search space by approximately 50% compared to enumeration.

The paper is organized as follows. Section 2 describes the problem and the methodology that has been followed to solve it. Section 3 analyses balanced reliable production lines in order to determine the optimal buffer allocation that minimizes the average WIP. Section 3 also gives the heuristic algorithm developed to solve the buffer allocation problem for balanced lines. Section 4 concludes the paper and gives some directions for future research.

2. The OBA problem and methodology of our investigation

Define $B=(B_2,B_3,\ldots,B_K)$ the 'buffer vector', i.e., a vector with elements the buffer capacities of the $K-1$ buffers. $\mu=(\mu_1,\mu_2,\ldots,\mu_K)$ the mean service rates vector with elements the mean service rates of the K stations of the line.

In mathematical terms our problem (P) could be stated as follows:

(P) Find $B=(B_2,B_3,\ldots,B_K)$ to

$$\min \mathrm{WIP}(B)$$

subject to:

$$X_K(B)\geqslant X_0,$$

$$\sum_{i=2}^{K} B_i = N,$$

$B_i \geqslant 0$,

B_i integer $(i = 2, 3, \ldots, K)$,

where N is a fixed nonnegative integer, denoting the total buffer space available in the system. X_K denotes the throughput of the K-station line. This is a function of the moments of the service time distribution and the buffer capacities, B_i. The higher moments of service time distributions have a minor effect on the throughput of a production line. This only holds for variances smaller than one, as in the case of balanced reliable lines with Erlang-k ($k \geqslant 2$) service time distributions examined in this work. For this, only the first two moments of the service time distributions are taken into account.

$\max X_K$ denotes the maximum throughput that is attained when the certain buffer slots N are placed optimally within the intermediate buffers of the K-station line. This maximum throughput is determined numerically for the given values of N and K.

X_0 is a value that corresponds to a throughput level. In our analysis we considered three such levels, the following: $X_{0,1} = 0.90 \times \max X_K$, $X_{0,2} = 0.95 \times \max X_K$ and $X_{0,3} = 0.98 \times \max X_K$. These levels have enabled us to extract some inferences for the solution of the OBA problem.

The service times at each station of our model are allowed to follow either the exponential or the k-stage Erlang distribution, denoted by E_k, $k \geqslant 2$.

Definition of the equivalence classes of the buffer allocations: To study the evolution of throughput and work-in-process (WIP) the vectors of all feasible buffer allocations were classified into groups, depending on their values.

For a certain production line with K stations and a certain amount of buffer slots, N, that are to be allocated among the $K-1$ intermediate buffers, we denote by $\mathscr{B}$ the set of all these possible allocations. This set is

$$\mathscr{B} = \{B_1, B_2, \ldots, B_L\}, \tag{1}$$

where

$$L = \binom{N+K-2}{K-2} = \frac{(N+1)(N+2)\cdots(N+K-2)}{(K-2)!}. \tag{2}$$

The B_is, $1 \leqslant i \leqslant L$, are vectors with $K-1$ elements which are nonnegative integer numbers of the form

$$B_i = \{B_{i2}, B_{i3}, \ldots, B_{iK}\},$$

where, B_{ij}, $2 \leqslant j \leqslant K$, expresses the capacity (in buffer slots) of the jth buffer.

Set $\mathscr{B}$ is split into $N+1$ *equivalence classes* which are characterized as *classes of first generation*, the following:

[0], [1], [2], …, [N].

A class of first generation, say [I], $0 \leqslant I \leqslant N$, consists of all the allocations B_i, with $B_{i2} = I$. For example, class [2] consists of all the allocations B_i which have the first element $B_{i2} = 2$ (the first intermediate buffer has two slots).

All classes of first generation [I], $0 \leqslant I \leqslant N$, are divided into $N+1-I$ classes which are defined as *classes of second generation*, e.g., class [0] is divided into $N+1$ classes of second generation, class [1] is divided into N classes of second generation, …, class [N] is divided into 1 class of second generation.

Each class of second generation, e.g., class [I, J], $0 \leqslant I \leqslant N$, $0 \leqslant J \leqslant N+1-I$, consists of all the allocations with the first two elements equal to I and J, respectively, i.e., $B_{i2} = I$ and $B_{i3} = J$.

Each class of second generation is divided into $N+1-(I+J)$ classes, which are defined as *classes of third generation*. The same procedure is applied for the genesis of all the remaining classes up to the $(K-2)$-*generation class*. Each element of the $(K-2)$-*generation class* also specifies the buffer contents of the last buffer since the sum of the buffer places always equals N.

Definition of subsequent classes: Let $[m_1, m_2, \ldots, m_r]$ and $[n_1, n_2, \ldots, n_r]$ be two classes of the same generation r, $1 \leqslant r \leqslant K-3$. Then we say that $[n_1, n_2, \ldots, n_r]$ is *subsequent to* $[m_1, m_2, \ldots, m_r]$ if $m_i = n_i$, for $i = 1, 2, \ldots, r-1$, and $m_r + 1 = n_r$. This precisely defines the lexicographic ordering that produces the self-similarity property discussed later. By this definition class [0, 2] of the second generation is subsequent to class [0, 1] and class [2] of the first generation is subsequent to class [1].

Definition of central classes of equivalence: Given a certain amount N of total buffer slots, we define as central classes of equivalence the following: (i)

when N is an *odd* number, the two classes: $[(N-1)/2]$ and $[(N+1)/2]$ and (ii) when N is an *even* number, the class: $[N/2]$, where $[x]$ denotes the maximum integer less than or equal to x.

Methodology of our investigation: To solve the buffer allocation problem (P) we followed the steps:

- (S1) We modeled the queueing process of the production line as a finite state, continuous time Markov chain, due to the assumption of the Erlang-k ($k \geqslant 1$) distribution for the processing times;
- (S2) To calculate the throughput of the production lines we solved the sparse system of the steady-state probabilities of the resulting Markov chain by applying an algorithm developed by Heavey, Papadopoulos and Browne [13]. The algorithm gives the exact numerical solution for short K-station lines with finite intermediate buffers and phase-type service times. The number of states and the number of feasible allocations of the N buffer slots among the $K-1$ intermediate buffer locations increases greatly with N and K. The latter is given by

$$\binom{N+K-2}{K-2} = \frac{(N+1)(N+2)\cdots(N+K-2)}{(K-2)!}. \tag{3}$$

  For large systems the efficient decomposition method by Dallery and Gershwin may be employed to calculate the throughput of the systems (for a good description of this approximation method the interested reader is addressed to Dallery and Frein [14]). From these two algorithms both the throughput and the (average) WIP are calculated.
- (S3) At the balanced case we applied a new heuristic algorithm and reduced significantly the search space to find the optimal buffer allocation (OBA) that minimizes the average WIP (see Section 3).

The added value of the present work is that we have been able to obtain some insights into the OBA problem for minimizing the average WIP inventory in reliable production lines. This has been achieved by considering three throughput levels. We pointed out that the choice of the throughput level, X_0, is critical for finding the OBA that minimizes the average WIP. Finally the search algorithm itself constitutes one of the contributions of this work. This algorithm together with the observations we have made and some numerical results supporting these findings are given in Section 3.

3. The findings of our study and the heuristic algorithm

We have analysed systematically and thoroughly a number of reliable balanced production lines consisting of $K = 3$, 4, 5 and 6 stations, with the processing times following the exponential and the Erlangian distributions and with the number N of buffer slots varying as in Table 1. Dealing with larger production lines with greater number of buffer slots is rather problematic with the Markovian state model. However, this sample of production lines was enough to establish some rules of thumb for the optimal buffer allocation (OBA) problem.

3.1. The findings of our study

From the study of the systems above we have been able to extract the following observations.

Observation 1 (*Evolution of throughput as a function of the ordered buffer allocations*): In general, the throughput of a production line, for a certain number of stations K and a given number of total buffer slots N, follows the shape of an inverse bowl as a function of the lexicographic buffer ordering outlined above. This inverse bowl in turn is observed

Table 1
Balanced systems analysed in this study

K	Expon. Range of N	E_2 Range of N	E_3 Range of N	E_4 Range of N
3	1–20	1–20	1–20	1–20
4	1–30	1–20	1–20	1–16
5	1–20	1–17	1–12	1–9
6	1–17	1–4	—	—

to roughly split into N internal inverse bowls, corresponding with the classes of the first generation, which in turn, are split into $N, N-1, N-2, \ldots, 1$, internal bowls, corresponding with the classes of the second generation, and so on. This self-repeating structure is observed to occur $K-3$ times. The fact that at each generation class we find the same characteristics as at the previous generation is called the *self-similarity* property. We borrow this terminology from the general definition that a certain picture is termed self-similar when it can be split into a usually infinite number of pictures that are similar to each other and to the initial one.

Fig. 2 depicts graphically the evolution of throughput as a function of the ordered buffer allocations, for a balanced reliable line with $K=5$ stations and $N=5$ total buffer slots, that have to be allocated among the 4 intermediate buffer locations. The same characteristic occurs again and again but in a decreasing order. This is due to the fact that each equivalence class has more elements than its successor class of the same generation. Note that the reversibility property of finite production lines (see [15]) which states that under a reverse flow through the line the throughput remains equal is valid only for the throughput of the lines and not fot the WIP. For example, in Fig. 2 one may see that $X(0-0-0-5) = X(5-0-0-0) = 0.51$ and $X(0-0-1-4) = X(4-1-0-0) = 0.54$. However, from Fig. 3, it may be seen that $\mathrm{WIP}(0-0-0-5) = 3$ whereas, $\mathrm{WIP}(5-0-0-0) = 7.8$ and $\mathrm{WIP}(0-0-1-4) = 3.4$ whereas $\mathrm{WIP}(4-1-0-0) = 7.6$. This means that the resersibility property cannot be exploited in reducing the search space when we try to evaluate the average WIP.

Observation 2 (*Evolution of WIP as a function of the ordered buffer allocations*): The average work-in-process (WIP), for a certain value, N, of total buffer slots, is an increasing function of the ordered buffer

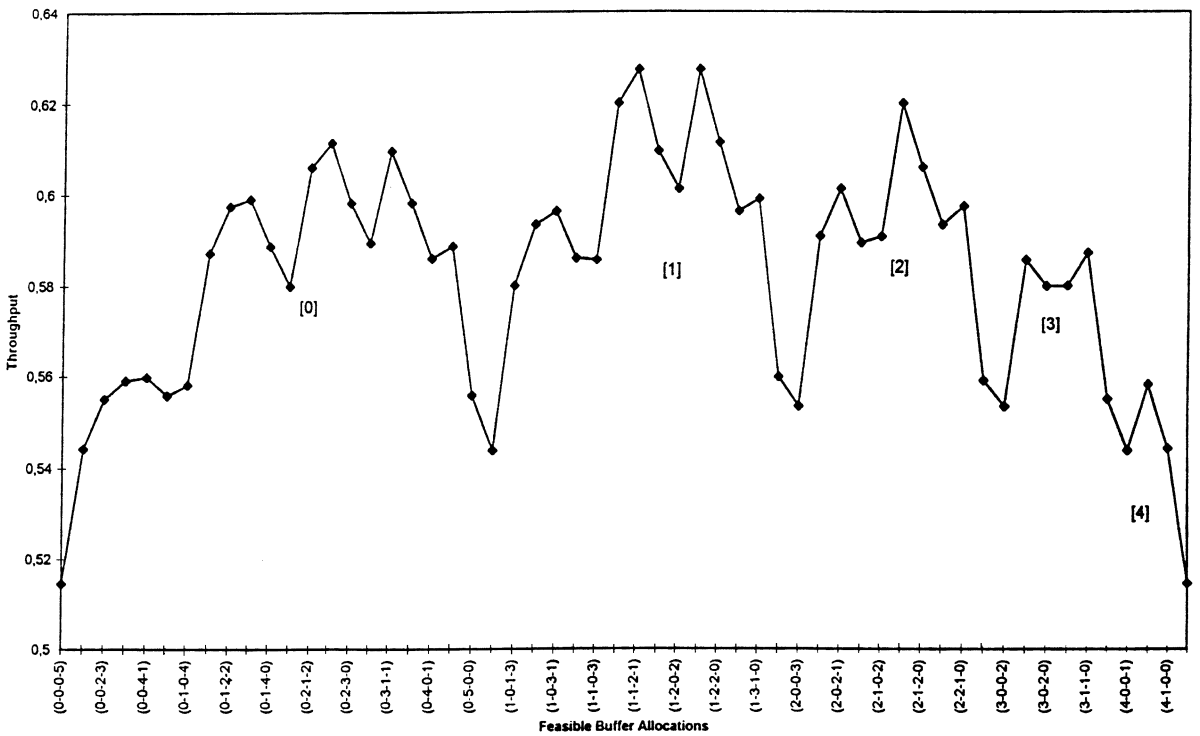

Fig. 2. Evolution of throughput as a function of the ordered buffer allocations, for $K=5$ and $N=5$: The self-similarity phenomenon.

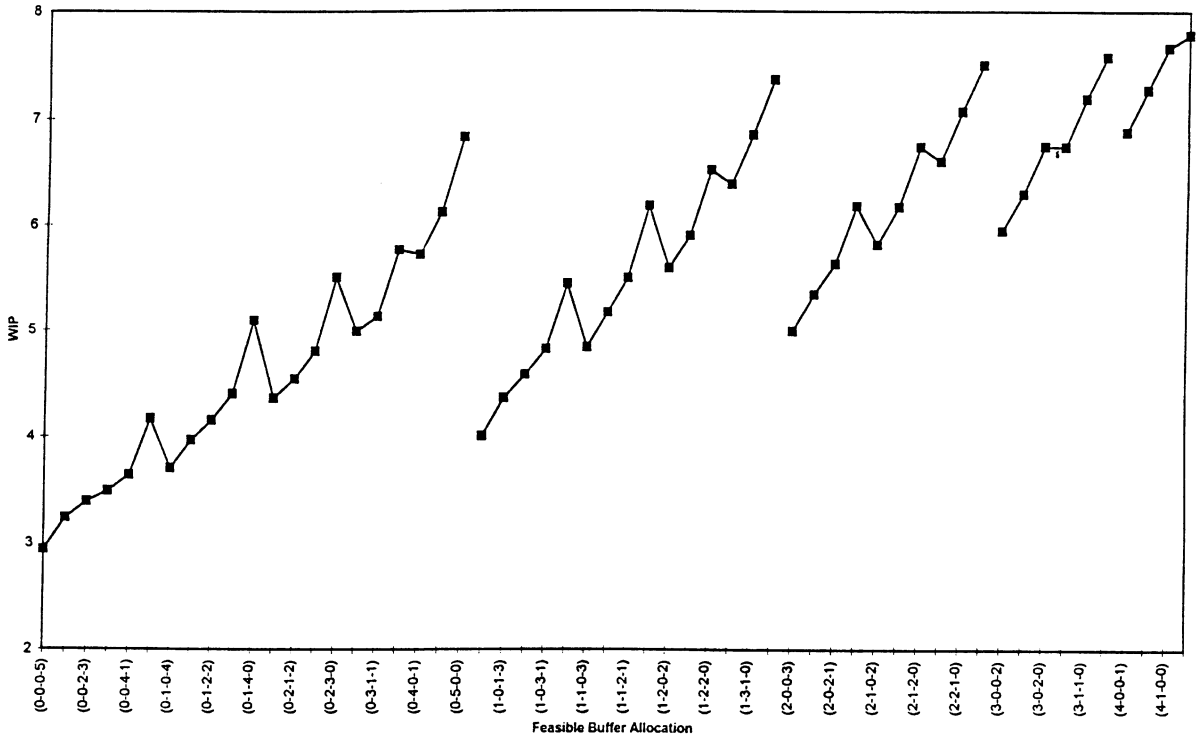

Fig. 3. Evolution of WIP as a function of the ordered buffer allocations, for $K = 5$ and $N = 5$: The self-similarity phenomenon.

allocations that belong to the classes of the $(K-3)$th generation. As we pass from one class of the $(K-3)$th generation to the next class, the average WIP decreases. For example, for $K = 5$ and $N = 5$, the allocation $(0-0-5-0)$ (which is the last of class [0, 0]) gives WIP = 4.1666, whereas the allocation $(0-1-0-4)$ (which is the first of the next class [0, 1]) gives WIP = 3.7041. Again, as in the case of the throughput (see Observation 1), it is observed that the self-similarity phenomenon appears in the evolution of the average WIP as a function of the ordered buffer allocations. More specifically we observe that its graph consists of N curves, each one of which is divided into $N, N-1, N-2, \ldots, 1$ curves, respectively, which are similar to each other and to the initial one. Graphically, this phenomenon is represented in Fig. 3. One may notice the discontinuities at the first allocation of each class, where the curve of WIP starts from lower values than those corresponding to the last allocations of the previous class.

Remark. The throughput and WIP are affected by the allocation schemes of the total buffer slots, N, into the $K-1$ intermediate buffer locations. Observing the evolution of these two measures of performance in terms of the feasible buffer allocations, we have been able to extract two observations:

1. Comparing the minimum and maximum values of the throughput, $X_K(N)$, and the average work-in-process, WIP(N), both as a function of N, one may see that the effect of the buffer allocation on the throughput is much less than that on the average WIP. For example, for $K = 4$ and $N = 7$, $\max X_4(7)/\min X_4(7) = 0.7183/0.5640 = 1.27$, whereas, the respective ratio $\max \mathrm{WIP}(7)/\min \mathrm{WIP}(7) = 8.9387/2.4169 =$

3.7. In Fig. 4 the evolution of $X_4(7)$ and WIP(7) is given as a function of the ordered buffer allocations. The major effect of the buffer allocation on the average WIP is shown by the curve that begins from the 2nd cycle and ends at the 8th cycle.

2. The effect of the buffer allocation on both the throughput and the WIP becomes more apparent when the exponential distribution is replaced by the Erlangian distribution with $k = 2, 3, \ldots$ phases of service (see Fig. 5).

Observation 3: For any number of stations, K, and any number of buffer slots, N, the OBA that minimizes the average WIP and satisfies both the restriction, $\sum_{i=2}^{K} B_i = N$, and the constraint, $X_K(B) \geqslant X_0$, has the following characteristic: The buffer slots are allocated primarily to the last buffer, to a minor extent to the one but last buffer, to a still minor extent to the second but last buffer, etc. This phenomenon ($B_K \geqslant B_{K-1} \geqslant B_{K-2} \geqslant \cdots$) is often called the *monotone increasing property*. As the throughput level increases, this allocation becomes more pronounced (see Table 2). One realization of the cases given in Table 2, for $K = 5$ and $N = 13$ is depicted graphically in Fig. 6.

From the results of Table 2 we see that the *monotone increasing property* of the buffer allocation (that is, $B_j \leqslant B_k$, for $j \leqslant k$) is not always valid. From a thorough investigation of many systems we came to the following observation. When the monotone increasing property is violated, for a

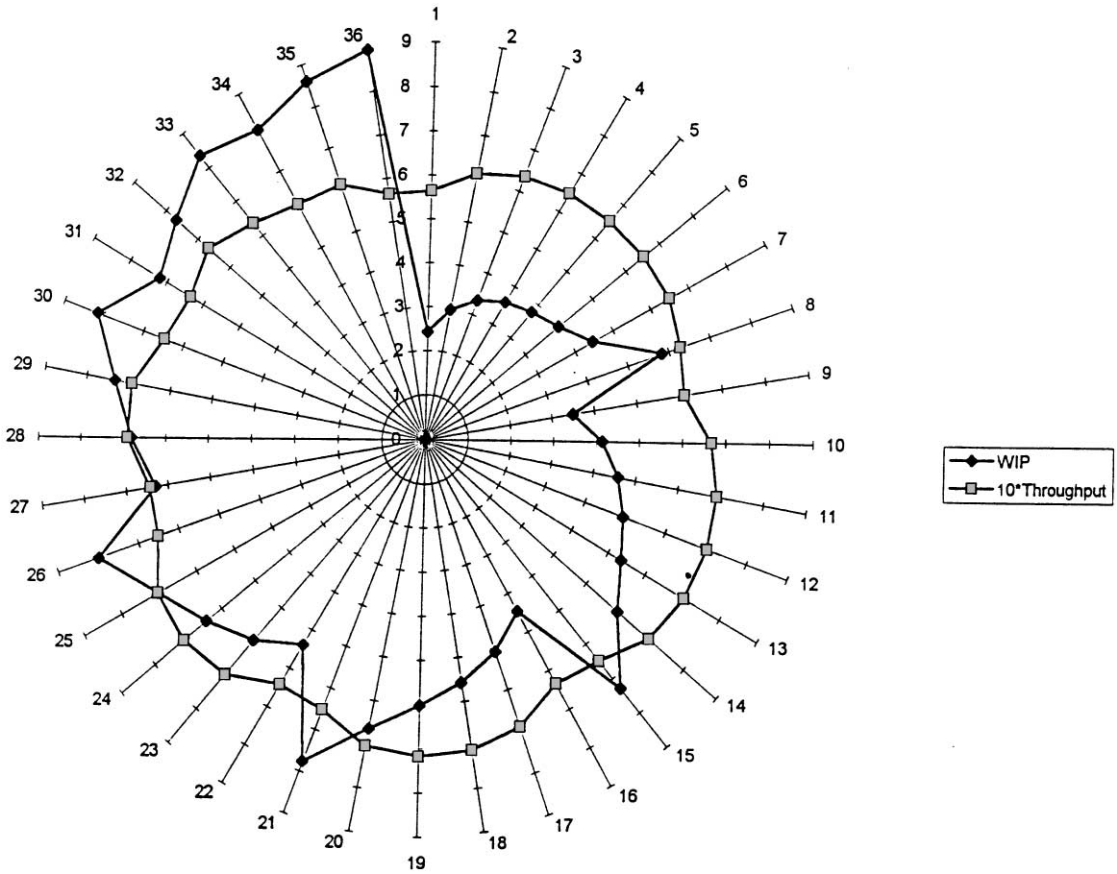

Fig. 4. Evolution of throughput and WIP as a function of the ordered buffer allocations, for $K = 4$ and $N = 7$.

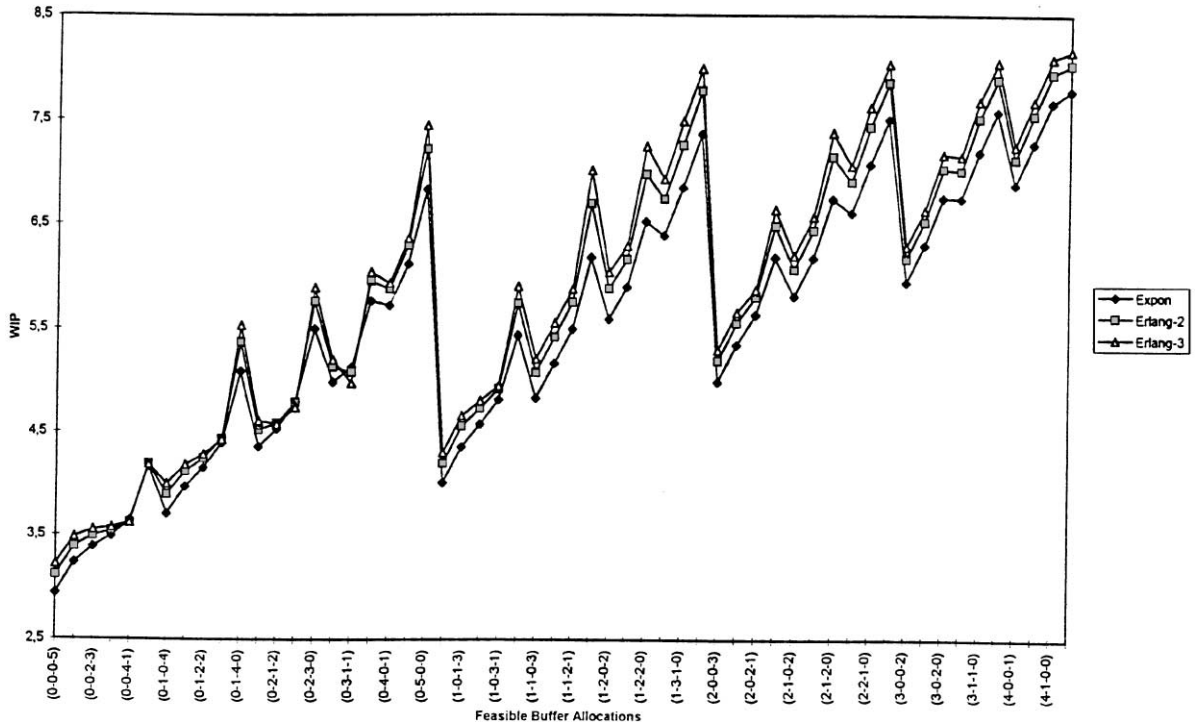

Fig. 5. The effect of service time distribution on the WIP, for $K = 5$ and $N = 5$.

Table 2
Some results supporting Observation 3

N	$(\mathrm{OBA})_{\min \mathrm{WIP}}$	X_5	$X_{0,1}$	$(\mathrm{OBA})_{\min \mathrm{WIP}}$	X_5	$X_{0,2}$	$(\mathrm{OBA})_{\min \mathrm{WIP}}$	X_5	$X_{0,3}$
11	$(0-3-5-3)$	0.6470	0.6463	$(1-2-3-5)$	0.6846	0.6822	$(1-4-3-3)$	0.7049	0.7037
12	$(1-1-3-7)$	0.6589	0.6569	$(1-2-6-3)$	0.6935	0.6934	$(2-2-4-4)$	0.7153	0.7153
13	$(1-1-5-6)$	0.6665	0.6660	$(1-3-4-5)$	0.7065	0.7030	$(2-3-3-5)$	0.7268	0.7252
14	$(1-2-2-9)$	0.6765	0.6752	$(1-4-4-5)$	0.7170	0.7127	$(2-3-5-4)$	0.7376	0.7352
15	$(1-2-3-9)$	0.6865	0.6817	$(1-4-5-5)$	0.7206	0.7196	$(2-4-4-5)$	0.7463	0.7423
16	$(1-2-4-9)$	0.6921	0.6893	$(1-5-5-5)$	0.7277	0.7276	$(2-4-5-5)$	0.7518	0.7506
17	$(1-2-6-8)$	0.6972	0.6962	$(2-3-4-8)$	0.7386	0.7349	$(2-5-5-5)$	0.7599	0.7581
18	$(1-3-4-10)$	0.7089	0.7033	$(2-3-5-8)$	0.7437	0.7423	$(2-6-5-5)$	0.7659	0.7658
19	$(1-3-4-11)$	0.7089	0.7085	$(2-3-7-7)$	0.7487	0.7478	$(3-4-5-7)$	0.7732	0.7715

certain value N of the buffer slots:

(i) the throughput level is just achieved by the buffer allocation that minimizes the average work-in-process (this is denoted by $(\mathrm{OBA})_{\min \mathrm{WIP}}$);

(ii) the $(\mathrm{OBA})_{\min \mathrm{WIP}}$ is a buffer allocation belonging to the middle of a certain class of allocations and

(iii) for $N + 1$ the monotone increasing property holds again, but now the $(\mathrm{OBA})_{\min \mathrm{WIP}}$ belongs to a subsequent equivalence class of the same or next generation.

For example, from Table 2, for $N = 11$, $K = 5$ and $\mu_1 = \cdots = \mu_K = 1$, $\max X_5$ is numerically calculated, by enumeration, to be equal to 0.7181 and

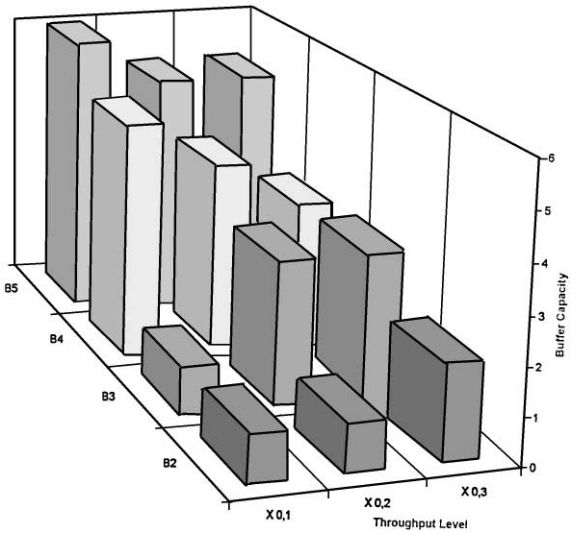

Fig. 6. Graphical representation of the OBA for the case $K = 5$ and $N = 13$ of Table 2.

is attained for the buffer allocation $(2-3-3-3)$ or its symmetric buffer allocation $(3-3-3-2)$. Then $X_{0,1} = 0.90 \times \max X_5 = (0.90) \times (0.7181) = 0.6463$, and the OBA that minimizes the average WIP is $(\mathrm{OBA})_{\min \mathrm{WIP}(B)} = (0-3-5-3)$. This belongs to the class [0] of the first generation and it gives a throughput greater than 0.6463. For $N = 12$ slots the monotone increasing property holds again by giving $(\mathrm{OBA})_{\min \mathrm{WIP}(B)} = (1-1-3-7)$ which belongs to class [1] of the first generation.

Observation 4 (*The optimal buffer allocation for the limiting cases:* (*a*) $X_0 = \min X_K$ *and* (*b*) $X_0 = \max X_K$)

(a) For any number of stations, K, and any number of buffer slots, N, the OBA that minimizes the average WIP (i.e., the $(\mathrm{OBA})_{\min \mathrm{WIP}(B)}$) and satisfies the restriction, $\sum_{i=2}^{K} B_i = N$, is the configuration

$(\mathrm{OBA})_{\min \mathrm{WIP}(B)} = (0, 0, \ldots, N)$.

The result conjectures that there is a tendency for the buffer slots to be located at the last buffer, B_K (see Table 3 for a few results supporting this finding). Notice that the constraint, $X_K(B) \geqslant X_0$, has been relaxed in this case (in other words, $X_0 = \min X_K$).

Table 3
Some results supporting Observation 4

K	N	$(\mathrm{OBA})_{\min \mathrm{WIP}(B)}$
3	1	$(0-1)$
3	2	$(0-2)$
3	3	$(0-3)$
	⋮	⋮
3	1	$(0-1)$
3	10	$(0-10)$
4	1	$(0-0-1)$
4	2	$(0-0-2)$
4	3	$(0-0-3)$
	⋮	⋮
4	10	$(0-0-10)$

The observation is supported by the following remark: Since the first station is never starved and the last station is never blocked, the last buffer is emptied more readily. As we move down the line the average WIP in each buffer decreases. So, if we allocate all N slots to the last buffer, there is no essential effect on the average WIP of this buffer. The stationary probabilities of $1, 2, \ldots, N$ slots being at the last buffer (B_K) remains extremely small and the average WIP of the whole system is the minimum. As the first station is always busy, buffer B_2 is always filled in, contributing to the increase of WIP. For this, to minimize WIP, one has to allocate more buffer slots to the downstream buffers.

(b) For any number of stations, K, and any number of buffer slots, N, the OBA that minimizes the average WIP and satisfies the two restrictions, $\sum_{i=2}^{K} B_i = N$ and $X_K(B) \geqslant X_0$, presents the following form: As the throughput rate increases to its maximum level, $\max X_K$, then the OBA which minimizes the average WIP tends to be the same as that maximizing the throughput of the production line. This observation has been extracted from numerous cases. As an example take the case of a line with $K = 3$ stations and $N = 8$ buffer slots. For the throughput levels $X_{0,1}$, $X_{0,2}$, $X_{0,3}$ and $X_{0,4}$ that correspond to 90%, 95%, 98% and 99.99%, respectively, of the maximum throughput, the OBAs are, respectively, (1–7), (2–6), (3–5) and (4–4), whereas

the OBA which maximizes the throughput of this line is the uniform allocation (4–4).

3.2. The algorithm

Exploiting the findings of our study and especially those of Observation 3, for the balanced lines, we reduced the search space by approximately 50%. A heuristic algorithm was developed for this particular problem.

The algorithm uses as input data the number of stations, K, of the line, the mean service rates, μ_i, $i = 1, 2, \ldots, K$, the total amount of buffer slots, N, that have to be allocated among the $K-1$ intermediate buffer locations, $B_2, \ldots, B_K$ and the throughput level X_0 that must be exceeded (three such levels were considered: $X_{0,1}$, $X_{0,2}$, $X_{0,3}$).

The algorithm searches for the values, $M_2, M_3, \ldots, M_{K-2}$, which correspond to the values of classes of 1st, 2nd, …, $(K-3)$th generation, up to which the average throughput increases and, at these particular values, it attains its maximum value. The steps of this search algorithm may be summarized as follows:

Step 1 (*Initialization phase*): At this phase the upper limits of the buffer capacities are determined.

Step 1.1: Put $B_2 = B_3 = \cdots = B_{K-2} = 0$ and search for the maximum value $M = M_{K-1}$ that buffer B_{K-1} can take such that for any $j = 0, 1, \ldots, N$:

$$X_K(0-0-\cdots-0-M-(N-M)) \geqslant X_K(0-0-\cdots-0-j-(N-j)). \quad (4)$$

In other words, being at the class (of buffer allocations) of the $(K-3)$th generation, try to determine the buffer allocation that maximizes throughput.

Step 1.2: Put $B_2 = \cdots = B_{K-3} = 0$ and search for the maximum value $M = M_{K-2}$ that buffer B_{K-2} can take (it has been empirically derived that $M_{K-2} \leqslant M_{K-1} < N$) such that for any $j = 0, 1, \ldots, M_{K-1}$:

$$\max X_K \text{ of class } [0-0-\cdots-0-M] \geqslant \max X_K \text{ of class } [0-0-\cdots-0-j]. \quad (5)$$

In other words, being at the class of $(K-4)$th generation, try to determine that sub-class of the $(K-3)$th generation which maximizes throughput.

Step 1.3: Find the upper values of the remaining buffers, i.e., the values $M_{K-3}, M_{K-4}, \ldots, M_3, M_2$ ($[x]$ again denotes the maximum integer less than or equal to x):

$$B_2 = 1, \ldots, M_2\left(= \left[\frac{N}{K-1}\right]\right),$$

$$B_3 = 0, \ldots, M_3(= M_4 - 1),$$

$$\vdots = \vdots$$

$$B_{K-4} = 0, \ldots, M_{K-4}(= M_{K-3} - 1),$$

$$B_{K-3} = 0, \ldots, M_{K-3}(= M_{K-2} - 1),$$

$$B_{K-2} = 0, \ldots, M_{K-2} \text{ determined by Step 1.2},$$

$$B_{K-1} = 0, \ldots, M_{K-1} \text{ determined by Step 1.1},$$

$$B_K = N - \sum_{i=2}^{K-1} B_i.$$

Step 2 (*Search phase*): At this phase the algorithm searches for the optimal buffer allocation which minimizes the average WIP and gives a throughput that exceeds the given level X_0 in the reduced search space (as given by the values, $M_{K-1}, \ldots, M_2$, determined in Steps 1.1–1.3).

Example. In a line with $K = 5$ stations and $N = 5$ buffer slots the number of all possible buffer allocations is 56. Take as $X_0 = X_{0,2} = 0.95 \times 0.6275 = 0.5961$ and apply the steps of the proposed heuristic algorithm. (*Note*: In Tables 4a and 4b the average WIP is written only when the throughput of the respective buffer allocation exceeds the given level, $X_{0,\,2} = 0.5961$).

From Step 1.1 the upper value of buffer B_{K-1}, $M_{K-1} = M_4$, is found equal to 4, as the buffer allocation $(0-0-4-1)$ gives the maximum throughput, 0.5597, after 6 buffer allocations of class [0, 0] (see Table 4a).

From Step 1.2 the algorithm finds $M_{K-2} = M_3 = 2$ since class [0–2] gives the maximum throughput $X_5 = 0.6114$. The number of searches are 12: 5 in class [0, 1], 4 in class [0, 2] and 3 in class [0, 3] (see Table 4a).

Finally, from Step 1.3, $M_2 = [5/(5-1)] = 1$ is obtained. This is the end of the initialization phase.

At the search space (Step 2), the values of the buffers are determined: $B_2 = 1$, $B_3 = 0, \ldots, 2$, $B_4 = 0, \ldots, 4$, $B_5 = N - \sum_{i=2}^{4} B_i$.

Table 4a
Steps 1.1 and 1.2: Searching in classes [0, 0], [0, 1], [0, 2] and [0,3]

Trial no.	Equiv. class	Buffer allocation	X_5	WIP
1		(0 − 0 − 0 − 5)	0.5146	
2		(0 − 0 − 1 − 4)	0.5441	
3	[0–0]	(0 − 0 − 2 − 3)	0.5550	
4		(0 − 0 − 3 − 2)	0.5590	
5		(0 − 0 − 4 − 1)	0.5597	
6		(0 − 0 − 5 − 0)	0.5557	
7		(0 − 1 − 0 − 4)	0.5580	
8		(0 − 1 − 1 − 3)	0.5872	
9	[0–1]	(0 − 1 − 2 − 2)	0.5974	4.1518
10		(0 − 1 − 3 − 1)	0.5990	4.3964
11		(0 − 1 − 4 − 0)	0.5887	
12		(0 − 2 − 0 − 3)	0.5800	
13		(0 − 2 − 1 − 2)	0.6061	4.5340
14	[0–2]	(0 − 2 − 2 − 1)	0.6114	4.7960
15		(0 − 2 − 3 − 0)	0.5982	5.5007
16		(0 − 3 − 0 − 2)	0.5895	
17	[0–3]	(0 − 3 − 1 − 1)	0.6096	5.1276
18		(0 − 3 − 2 − 0)	0.5982	5.7633

Remark. Since class [0] was already checked in Steps 1.1 and 1.2, B_2 takes only the value 1. The number of searches in Step 2 are 12, the following: 5 in class [1, 0], 4 in class [1, 1] and 3 in class [1, 2], as given in Table 4b.

Again from all these iterations we see that none of them gives WIP less than 4.1518, the minimum found so far. Thus, for this particular example, the buffer allocation that minimizes the average WIP is (0 − 1 − 2 − 2) and the corresponding minimum WIP is 4.1518 for the selected throughput level $X_0 = X_{0,2} = 0.5961$, that has to be exceeded. The total number of searches from all steps of the algorithm is 30 as compared with the 56 allocations from enumeration. This means that the algorithm leads to a 46% reduction in the number of searches to find the optimal buffer allocation. Table 5 tabulates more numerical results for four-, six- and seven-station lines with the last column of the table showing the number of searches needed by the algorithm to attain the OBA which minimizes the average WIP. The latter compared against the total number of buffer allocations gives a percentage

Table 4b
Step 2: Searching in classes [1–0], [1–1] and [1–2]

Trial no.	Equiv. class	Buffer allocation	X_5	WIP
19		(1 − 0 − 0 − 4)	0.5438	
20		(1 − 0 − 1 − 3)	0.5801	
21	[1–0]	(1 − 0 − 2 − 2)	0.5935	
22		(1 − 0 − 3 − 1)	0.5963	4.8100
23		(1 − 0 − 4 − 0)	0.5860	
24		(1 − 1 − 0 − 3)	0.5857	
25		(1 − 1 − 1 − 2)	0.6202	5.1638
26	[1–1]	(1 − 1 − 2 − 1)	0.6275	5.4941
27		(1 − 1 − 3 − 0)	0.6096	6.1794
28		(1 − 2 − 0 − 2)	0.6012	5.5889
29	[1–2]	(1 − 2 − 1 − 1)	0.6275	5.8978
30		(1 − 2 − 2 − 0)	0.6114	6.5231

reduction (PR) in the search space that is well over 50%. PR is defined by

$$\mathrm{PR} = 1 - \frac{\text{number of searches}}{\text{total number of buffer allocations}}.$$

In all cases examined in our study, the proposed algorithm provided the optimal solution.

4. Conclusions, contribution of the study and further research

In this work we studied the optimal buffer allocation in short reliable production lines. We examined the average work-in-process (WIP) in conjunction with the throughput of the system. More specifically we studied the evolution of the average WIP and throughput for all the ordered buffer allocations of a certain total number of buffer slots, N, among the $K - 1$ intermediate buffer locations. The vectors of the buffer allocations were classified systematically into equivalence classes, something that facilitated a lot in the analysis of the evolution of the average WIP and throughput as a function of these allocations.

We observed that in the case of reliable balanced production lines, the average WIP is an increasing function of the ordered buffer allocations. In this

Table 5
Some results for the reduction in the search space

K	N	$\max X_K$	Throughput level	$(\mathrm{OBA})_{\min \mathrm{WIP}(B)}$	PR
4	18	0.8280	0.90×0.8280	$(1-9-8)$	$1-\frac{91}{190}=52\%$
6	10	0.6669	0.95×0.6669	$(1-1-2-2-4)$	$1-\frac{423}{1{,}001}=58\%$
7	8	0.6090	0.90×0.6090	$(0-1-1-2-2-2)$	$1-\frac{461}{1{,}287}=64\%$
7	6	0.5805	0.95×0.5805	$(0-1-1-1-1-2)$	$1-\frac{210}{462}=55\%$

case the graphical representation of the throughput forms the shape of an inverse bowl. Within this bowl smaller inverse bowls are formed that are further sub-divided into even smaller inverse bowls and so on (this has been named as the *self-similarity phenomenon*). This analysis of the initial inverse bowl into smaller bowls is extended up to $K-3$ times. It was observed that this phenomenon appears also to the evolution of the average WIP as a function of the ordered buffer allocations of the balanced lines.

The main objective of this study was the determination of the OBA that minimizes the average WIP, given that there are N buffer slots that have to be allocated, under the condition that the throughput must exceed a given level, X_0. We considered three such levels: $X_{0,1}=0.90\times\max X_K$, $X_{0,2}=0.95\times\max X_K$ and $X_{0,3}=0.98\times\max X_K$. Two different tendencies were observed, concerning the OBA. In the first one the average WIP is minimized when the buffer slots are located at the last buffers of the line. In the second the maximum throughput in the balanced lines is achieved when the buffer slots are equally allocated among the buffers and the remaining slots are usually located at the central buffers. This second case prevails when the throughput level X_0 tends to the maximum throughput of the system for a certain N.

The contribution of the present work is that we have been able to obtain some insights into the OBA problem for minimizing the average WIP inventory in reliable production lines. This has been achieved by considering three throughput levels. We pointed out that the choice of the throughput level, X_0, is critical for finding the OBA that minimizes the average WIP. Finally the search algorithm itself constitutes one of the contributions of this work.

As an area for further research we propose the investigation of the optimal buffer allocation in large unreliable production lines. The objective again would be the minimization of the average WIP satisfying on the other hand the condition that the throughput would exceed a given level. This case is much more complicated than the present one as more parameters are involved and one would have to investigate the effect of the service time distribution and the availability of the individual unreliable workstations on the optimal buffer allocation.

Acknowledgements

The authors wish to sincerely thank the two anonymous referees for their valuable comments. These helped indeed improve the appearance of the paper and to clarify some points.

References

[1] H.T. Papadopoulos, C. Heavey, Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines, European Journal of Operational Research 92 (1996) 1–27.
[2] S.B. Gershwin, Manufacturing Systems Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[3] J.A. Buzacott, J.G. Shanthikumar, Stochastic Models of Manufacturing Systems, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[4] H.T. Papadopoulos, C. Heavey, J. Browne, Queueing Theory in Manufacturing Systems Analysis and Design, Chapman & Hall, London, 1993.
[5] T. Altiok, Performance Analysis of Manufacturing Systems, Springer, New York, 1997.
[6] S. Yeralan, B. Tan, Analysis of multistation production systems with limited buffer capacity. Part I: The subsystem model, Mathematical and Computer Modelling 25 (7) (1997) 109–122.
[7] S. Yeralan, B. Tan, Analysis of multistation production systems with limited buffer capacity. Part II: The decomposition method, Mathematical and Computer Modelling 25 (11) (1997) 109–123.
[8] R. Conway, W. Maxwell, J.O. McClain, L.J. Thomas, The role of work in process inventory in serial production lines, Operations Research 36 (2) (1988) 229–241.
[9] F.S. Hillier, K.C. So, The effect of machine breakdowns and interstage storage on the performance of production line systems, International Journal of Production Research 29 (10) (1991) 2043–2055.
[10] F.S. Hillier, K.C. So, The effect of the coefficient of variation of operation times on the allocation of storage space in production line systems, IIE Transactions 23 (2) (1991) 198–206.
[11] F.S. Hillier, K.C. So, R.W. Boling, Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times, Management Science 39 (1) (1993) 126–133.
[12] D. Seong, S.Y. Chang, Y. Hong, Heuristic algorithms for buffer allocation in a production line with unreliable machines, International Journal of Production Research 33 (7) (1995) 1989–2005.
[13] C. Heavey, H.T. Papadopoulos, J. Browne, The throughput rate of multistation unreliable production lines, European Journal of Operational Research 68 (1993) 69–89.
[14] Y. Dallery, Y. Frein, On decomposition methods for tandem queueing networks with blocking, Operations Research 41 (2) (1993) 386–399.
[15] E.J. Muth, The reversibility property of production lines, Management Science 25 (2) (1979) 152–158.