Scientia Iranica, Vol. 10, No. 4, pp 401–409

SCIENTIA
IRANICA

Buffer Allocation Problem, a Heuristic Approach

S.K. Chaharsooghi* and N. Nahavandi[1]

Because of the fundamental role of production flow lines in modern industry, considerable attention has paid to how improve their efficiency. A particular performance measure indicating efficiency is throughput. Throughput is the number of completed units of work per time period in steady state. In general, the throughput of a production line increases, or at least remains constant, as the buffer capacities increase. A larger throughput is usually the consequence of a larger buffer configuration, which, in turn, yields a larger inventory accumulation. One of the main problems with designing a production flow line is to determine the appropriate amount of buffers between stations in order to achieve the goals. This is known as a Buffer Allocation Problem (BAP). The BAP is a nonlinear, multi-objective one in integer variables. The purpose of this paper is to present a heuristic algorithm to find the optimal allocation of buffers that maximizes throughput. The main idea is to find the amount of buffers under which the system operates, as a system with infinite buffer capacity, so the stations will be independent of each other and throughput becomes maximum. Numerical results are used to demonstrate the accuracy of the proposed algorithm. The algorithm finds the allocation where it's throughput is maximum, or only slightly less than maximum, but where it's total buffer capacity is considerably less than optimal allocation.

INTRODUCTION

Buffers have a significant impact on the performance measures of a production line. The particular performance measures of a production line are WIP and throughput that are representative of the effectiveness and efficiency of operations.

Higher effectiveness implies lower WIP levels and higher throughput means higher efficiency in flow lines. Therefore, efficiency and effectiveness contradict each other in flow lines. When the capacity of a buffer is increased, the throughput increases at a decreasing rate and becomes asymptotic to the case with an infinite-capacity buffer. So, a larger throughput is usually the consequence of a larger buffer configuration, which, in turn, yields a larger inventory accumulation. Thus, the operation of a flow line having a higher throughput, obtained by placing buffers between the stations, may not be quite as effective due to its inventory buildup.

Thus, the main goals in designing the flow line are to maximize efficiency and effectiveness. The problem is to determine the appropriate amount of buffers between stations in order to achieve these goals. So, the Buffer Allocation Problem (BAP) is a combinatorial optimization problem and the problem can, therefore, be written as nonlinear and multi-objective with integer decision variables that are the buffer capacities.

This paper is organized as follows: In the following section, mathematical programming and generative models that are used to solve BAP are reviewed. In the next section, the proposed heuristic algorithm is explained and is validated by numerical examples. Finally, a summary and a conclusion are described.

LITERATURE REVIEW

Several previous studies exist in the literature that focus on the problem of determining the inventory buffer requirement. The buffer allocation problem is, essentially, a combinatorial optimization problem. The decision variables are the buffer capacities, denoted by x_i for the ith buffer. Since buffers include the positions on the machines, x_i will be at least one and only takes integer values. Lutz illustrated the combinatorial nature of the buffer allocation problem [1]. He showed the complexity of the problem by identifying the possible buffer allocation combinations; assuming that

*. *Corresponding Author, Department of Industrial Engineering, Tarbiat Modarres University, P.O. Box 14115-111, Tehran, I.R. Iran.*

1. *Department of Industrial Engineering, Tarbiat Modarres University, P.O. Box 14115-111, Tehran, I.R. Iran.*

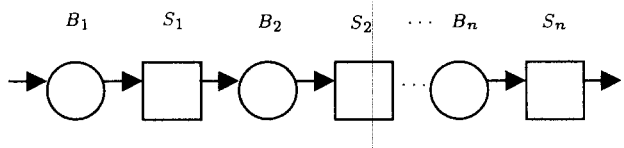

Figure 1. Flow line.

in a flow line there are $n+1$ stations and, so, there are n buffers between the stations and the total storage of u units is selected as described in Figure 1, where S_i represents station i and B_i is buffer i.

The number of different ways of allocating u storage space to n buffers can be determined using $\binom{u+(n-1)}{u}$. Each of these buffer arrangements is called a buffer profile, since each represents a unique combination of storage allocation and each will potentially result in a different output level of the line. The total number of buffer profiles for a line with up to u total storage space and n buffers, is the sum of the previous formula over the range of 0 to u, which yields this formula $\binom{u+n}{u}$. The line with 3 buffers and up to 3 total storage spaces, has 20 buffer profiles. If the problem is expanded to a five buffer line, evaluating up to 25 or more storage levels would require examining over 142,000 different buffer profiles [2]. In the following section, mathematical and generative models for solving BAP are reviewed.

Mathematical Programming Models

The buffer allocation problem is a stochastic, nonlinear programming problem with an integer decision vector. This problem is formulated as a nonlinear multiple-objective programming problem where the decision variables are the integers. Not only the BAP is a difficult NP-hard combinatorial optimization problem, but also it is made all the more difficult by the fact that the objective function is not obtainable in closed form to interrelate the integer decision variables, $\bar{x}$, and the performance measures such as θ throughput, $\bar{L}$ work-in-process, $\sum_i x_i$ total buffers allocated and other system performance measures, such as $\bar{\rho}$ system utilization for any but the most trivial situations.

Different types of mathematical programming formulation are written in accordance with the objectives. General formulation is as follows [3]:

$$\text{Extremize } Z = \{f_1(\bar{x}), f_2(\bar{x}), \cdots, f_p(\bar{x})\},$$

$$\text{s.t.} \quad g_i(\bar{x}) \leq 0,$$

where:

$f_1(\bar{x})$ = average system throughput (θ),
$f_2(\bar{x})$ = average work-in-process ($\bar{L}$),
$f_3(\bar{x})$ = total buffers allocated ($\sum_i x_i$),
$f_p(\bar{x})$ = average system utilization ($\bar{\rho}$),

The other form of problem can be formulated as follows:

$$\max \theta = f_1(\bar{x}),$$

$$\text{s.t.} \quad \sum_i x_i = u,$$

In this form, the problem is to determine how to distribute a finite sum of buffer space, u, in a flow line with $n+1$ stations. Recently, Harris and Powell developed an algorithm that efficiently found the optimal allocation for this problem [4]. This formulation does not consider the undesirable side of increasing buffer capacities, that is, holding high WIP levels. The other form is to consider both throughput and WIP in an objective function [5]:

$$P1\text{: maximize } Z = (R-V)\theta - H\bar{L}$$

$$\text{s.t.} \quad x_i \geq 1, \quad i = 1, 2, 3, \cdots, n,$$

where R is average revenue per unit, V is the average variable production cost and H is the average holding cost per unit.

Another way is to minimize the total buffer space for a given desired and feasible throughput, θ'. That is:

$$P2\text{: } \min \sum_i x_i$$

$$\text{s.t.} \quad \theta > \theta', \quad x_i \geq 1, \quad i = 1, 2, 3, \cdots, n.$$

Smith and Daskalaki modified formulation P1 and added another penalty term to the objective function to ensure that the buffers allocated were further penalized [6]. Thus, the objective becomes:

$$P3\text{: maximize } Z = (R-V)\theta - H\bar{L} - C\sum_i x_i.$$

For further description, refer to [3].

Generative Models

The difficulty in all the mentioned optimization problems is that there are no explicit expressions for average system throughput, θ, and average work-in-process, $\bar{L}$, in terms of x_i. These are usually obtained from the steady-state distribution of the jobs in the buffers. Thus, due to a lack of differentiable functions that are usually assumed in classical optimization problems, the approach used to solve the above optimization problems may be an ad-hoc one.

The general procedure is used by most researchers to find the optimal buffer levels. In this procedure, an evaluative model is used to obtain system performance measures, i.e. the mean buffer levels, which are then

used by the generative model in its search for the optimal value for a given objective function. Usually, either a queuing network model, a simulation model or an approximation model is used in evaluative models. Simulation models, when used in optimization scheme like this, have the disadvantage that only parameter estimations can be obtained. But simulation has the obvious advantage that arbitrary system configurations and the transient phase can be modeled.

In general, four generative (optimization) methods are used in the literature to search for optimal buffer sizes. In Table 1, a number of buffer optimization models for flow line are given. This table is based on an original table taken from [7] with additional references added.

Increasing Buffer Size by One at a Time

The simplest method is to change the buffer size manually, in increments of one. Another name for this method is the enumeration method [7]. It is obviously feasible for a very small system and in cases of designing balanced production lines with equal buffer sizes between work centers or production lines with a central buffer storage for in-process inventories [8].

Factorial Design of Experiment

The factorial design of an experiment is one of the simultaneous search methods with multiple variables. If one is unable to determine a specific mathematical form for the response function, such as throughput rate over buffer size, it may be useful to design a set of possible experiments in order to obtain information about the direction of an optimal solution. As the length of a production line increases, it is more difficult to process statistical insights into relations of system performance with buffer size, unless a large number of function evaluations are made. Thus, although this method can provide a brief insight into the possible location of the optimal solutions within the study domain, it may fail to provide a global optimum buffer size.

Table 1. Research pertaining to buffer allocation problem in flow line.

Authors [7]	Methodology		Objective	Number of Stations
	Evaluative Model	Generative Solution Model		
Altiok and Stidham (1983)	Coxian queuing network	Search method of Hooke and Jeeves	Maximize average profit	K
Anderson and Moodie (1969)	Simulation	Analytical method inventory model	Minimize inventory cost	K
Chow (1987)	Simulation	Dynamic programming	Maximize throughput	K
Hillier and So (1991)	Queuing network	Experimental design and Heuristic method	Maximize line utility	Four, five
Ho, Eyler and Chien (1979)	Simulation	Perturbation analysis	Maximize throughput	K
Jensen et al. (1991)	Mathematical analytical model	Dynamic programming	Maximize throughput	K
Kraemer and Love (1970)	Markov Chain	Simple mathematic analysis	Maximize net profit	Two
Masso and Smith (1974)	Simulation	Approximation	Maximize line utility	Three
Hillier, So and Boling (1993)	Mathematical analytical model	Enumeration approach	Maximize throughput	K
Park (1993)	Heuristic	Heuristic	Min. total buffer Max. desired thr.	
Powell and Pyke (1996)	Simulation		Maximize throughput	
Hurely (1996)	Heuristic	Heuristic	Maximize throughput	
Singh and Smith (1997)	Mathematical analytical model	Search method	Improve performance measures	K
Lutz, Davis and Sun (1998)	Simulation	Tabu search	Max. throughput Min. inventory	K
Harris and Powell (1999)	Simulation	Simplex search	Max. throughput	K

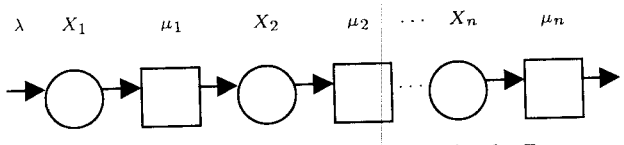

Figure 2. Flow line with n machines and n buffers.

Analytical Method

A number of analytical models have been developed, which consist of dynamic programming or the Markov process. In dynamic programming, an M-state buffer design problem is decomposed into a $(m-1)$ two stage problem and a recursive relationship is formulated. It must be noted that an optimal decision for the remaining stages must be independent of the decision adopted in the previous stages. In the Markov process, it must be assumed that the distribution function of the service times is exponential.

These methods address the combinatorial nature of the buffer allocation problem, but are not capable of modeling large or complex manufacturing lines.

Search Methods

Probably, frequently used generative models are those utilizing search methods. Search methods exist that can be used with queuing network models or stochastic simulation models. There are various non-linear optimization methods that can be applied to solve the buffer design optimization problem. They include the Gradient method, the Hooke and Jeeves method and the Tabu search. For further reading, refer to [2,5,8,9].

PROPOSED HEURISTIC ALGORITHM

In this section, a heuristic algorithm is proposed to find the particular allocation that maximizes throughput, which is defined as the number of completed units of work per time period in the steady state.

Consider a production flow line with n machines in a series. There is a buffer before each machine, as shown in Figure 2.

Where λ is the mean arrival rate to the system, μ_i is the mean service rate of machine i and X_i is the buffer size at machine i. X_i is included in position on the machine i, so it will be at least the one and only integer value. In most cases in the literature, arrival and service distributions are assumed to be exponential. So, it is assumed that arrivals at station one are generated from an infinite population, according to a Poisson distribution with mean arrival rate, λ, and the service time distribution at each machine, i, is exponential with mean rate μ_i. It is assumed that each machine has one server.

Case 1

Consider the case where there is no buffer limit in any machine. It means that the capacity of the buffers is infinite. In this case, Saaty [10] proved that for all i, the output from machine i, or equivalently the input to machine i, is Poisson with mean rate λ and that each machine may be treated independently as $M/M_i/1/\infty$ [10]. This means that for machine i, the steady state probabilities, $P_{i(ki)}$, are given by:

$$P_{i(ki)} = (1-\rho_i)\rho_i^{ki} \quad i = 1, 2, \cdots, n,$$

where $\rho_i = \lambda/\mu_i < 1, ki$ is the number of parts in the buffer of machine i and $P_{i(ki)}$ is the probability that there are ki parts in buffer machine i. So, $P_{i(ki)}$ is the marginal probability distribution of finding ki parts in the ith machine. Jackson showed that the joint distribution for all stations is written as closed-form and is given by:

$$P(k1, k2, k3, \cdots, kn) = P_1(k1)P_2(k2)P_3(k3)\cdots P_n(kn),$$

where $P(k1, k2, k3, \cdots, kn)$ is the probability that there are $k1$ parts in buffer machine 1 and $k2$ parts in buffer machine 2 and so on.

This result means that the number of parts at each machine is independent of the queue length at the other machines. In this way, the expected WIP and throughput is given by:

$$E(\text{WIP}) = \sum_{\forall k1}\sum_{\forall k2}\cdots\sum_{\forall kn}(k1 + k2 + \cdots + kn) \times P(k1, k2, \cdots, kn) = \sum_{k1} k1P(k1) + \cdots + \sum_{kn} knP(kn),$$

$$\text{throughput} = (1 - P_n(0))\,\mu_n,$$

where $P_n(0)$ is the probability that the buffer of machine n is empty and so the machine is starved.

As stated previously, as the total amount of buffer capacity increases, the throughput increases, but usually at a decreasing rate. So, maximum throughput is attained at $X_i = \infty$ for all i and, because of the large buffer configuration, WIP also reaches the maximum.

Case 2

In reality, because of space limitation and cost consideration, there are target levels for buffers between machines. As in the literature of the queuing theory, one station with a finite buffer capacity, X, is analyzed as $M/M/1/X$ and P_k is:

$$P_k = \frac{(1-r)r^k}{1-r^{x+1}},$$

where P_k is the probability of finding k parts in the buffer and $r = \frac{\lambda}{\mu}$. In this system, an effective input rate

to the system is $\bar{\lambda} = \lambda(1 - P_x)$, which is the accepted portion of arrivals where P_x is the probability that the buffer is full. So, traffic intensity is $\rho = \frac{\bar{\lambda}}{\mu}$.

In this case, minimum throughput yields when $X = 1$, so there is one part in the system and, consequently, the system has minimum WIP.

When finite buffers are allowed between successive machines, then the analysis of the flow line will be difficult. This is because individual machines cannot be solved in isolation and are dependent. Consequently, the buffer limitations give rise to blocking that causes stoppage at work stations due to lack of space or excessive accumulation of in-process inventory in the downstream stages. Similarly, starvation may exist in the manufacturing systems, causing idleness in stations due to lack of jobs to process in the upstream stages. So, the blocking and starving of machines are the main causes of inefficiency in production lines and difficulty in analysis of these systems. Because of the finite buffer capacity, work stations are not independent of each other, so $P(k1, k2, \cdots, kn)$ cannot be written as closed-form and is determined by using Markov chains. The number of states in the Markov chain increases rapidly with the amount of workstation and buffer capacity. For example, a line with five stations and buffer capacity $X_i = 3$ for each station, gives rise to a Markov chain with 19,402 states. So, it is impractical to solve the system using the Markov chain.

Case 3

Consider a flow line with n machines so that X_1 is finite and $X_2, X_3, \cdots, X_n$ are infinite. Since the buffer capacity of the first machine is finite, the flow of material into the system is limited and under controlled. $\bar{\lambda} = \lambda(1 - P_{X_1})$ is the proportion of materials entered into the system. Since the buffers capacity $X_2, X_3, \cdots, X_n$ are infinite, machines $1, 2, \cdots, n$ are independent from each other and there are no blocking effects between stations.

The main idea of the proposed heuristic method is to find the number of buffers under which, the system operates as the system with infinite buffer capacity. Since the machines are independent of each other, the steady state probabilities are easily defined (as closed form) and throughput reaches the maximum value. On the other side, the flow of material into the system is limited and so inventory accumulation is limited and the WIP of this line is much less than case 1.

In the proposed heuristic method, the effectiveness of buffers is measured by their effects on system throughput, which is the average number of parts produced per time unit in the steady state.

For the first machine, there is finite buffer capacity and so it is analyzed as $M/M/1/X_1$. To find the best value for X_1, the probability that the buffer is full is calculated and it is put equal to β.

$$P_1(X_1) = \frac{(1-r)r^{X_1}}{(1-r^{X_1+1})} = \beta, \quad r = \frac{\lambda}{\mu_1}.$$

β is an arbitrary value and the manager defines it. In fact, X_1 is defined so that in $100\beta\%$ time, the buffer of the first machine is full.

For the other machines, one must define the number of buffers under which, the system operates as the system with infinite buffer capacity. Machine i is analyzed as $M/M/1/\infty$. λ_i is the mean arrival rate to machine i and μ_i is the mean service rate of machine i. It is required to find buffer capacity X_i, so that the system $M/M/1/X_i$ becomes equivalent to system $M/M/1/\infty$. It means that the performance measures of two systems are equal.

If one calculates the probability of more than X_i parts in the machine i $(i = 2, 3, \cdots, n)$, then, the formula for X_i can be written as:

$$\sum_{j=X_i+1}^{\infty} P_i(j) = \sum_{j=X_i+1}^{\infty} \rho_i^j(1-\rho_i)$$

$$= (1-\rho_i) \sum_{j=X_i+1}^{\infty} \rho_i^j = \rho_i^{X_i+1}.$$

Probability of more than X_i parts in the buffer of the machine $i = \alpha_i$. And so;

$$\rho_i^{X_i+1} = \alpha_i, \text{ and } X_i = \frac{\ln \alpha_i}{\ln \rho_i} - 1.$$

where $\rho_i = \frac{\lambda_i}{\mu_i}$ and α_i are acceptable error and the manager defines it.

Description of Proposed Heuristic Algorithm

Notations

λ	mean arrival rate outside the system
λ_i	mean arrival rate to machine i, $i = 2, \cdots, n$
μ_i	mean service rate of machine i, $i = 1, 2, \cdots, n$
$(\text{out})_i$	mean output rate of machine i, $i = 1, 2, \cdots, n$
X_i	finite buffer size of machine i, $i = 1, 2, \cdots, n$
ρ_i	traffic intensity of machine i, $i = 1, 2, \cdots, n$
$p_i(X_i)$	the probability that machine i is full, $i = 1, 2, \cdots, n$
β	the probability that the buffer of the first machine is full
α_i	probability of being more than X_i parts in the buffer of machine i, $i = 2, 3, \cdots, n$.

Assumptions

The assumptions made are summarized below:

1. Arrivals occur only through the first queue;
2. Queues are arranged in a series topology with buffers preceding each server;
3. Failures, scrap and rework are not considered. They are often considered in service times (μ_i);
4. Production blocking is BAS. So, the part which has completed its service at the ith station has to go directly to the $(i+1)$st station. If the queue of the $(i+1)$st station is full at the moment of the service completion of the part at the ith station, then, the part waits, keeping the server idle at the ith station until the service at the $(i+1)$st station is completed. This is a common blocking phenomenon in production systems;
5. It is assumed that arrival and service rate are drawn from an exponential distribution. So, the input process to each station is assumed to be Poisson (although it is not). Therefore, all of the queues are $M/M/1/X_i$;
6. At each station, the service discipline is First-Come-First-Served.

Algorithm

Step 1: Station $i = 1$

Consider the first station as $M/M/1/X_1$. Determine buffer size (X_1) with given β by this formula:

$$r = \frac{\lambda}{\mu_1}, \quad P_1(X_1) = \frac{(1-r)r^{X_1}}{(1-r^{X_1+1})} = \beta,$$

and calculate:

$$P_1(0) = \frac{1-r}{1-r^{X_1+1}}, \quad (\text{out})_1 = \mu_1(1-P_1(0)).$$

Step 2: Station $i = i+1$,

$$\lambda_i = (\text{out})_{i-1}, \quad \rho_i = \frac{\lambda_i}{\mu_i}, \quad X_i = \frac{\ln \alpha_i}{\ln \rho_i} - 1,$$

$$P_i(0) = \frac{1-\rho_i}{1-\rho_i^{X_i+1}}, \quad (\text{out})_i = \mu_i(1-P_i(0)).$$

Step 3

Continue Step 2 until the last station ($i = n$).

Numerical Examples

In this section, the performance of the presented algorithm is reported on a different type of line. The success of the proposed heuristic algorithm depends on how closely it can reach the maximum throughput. The objective of the proposed heuristic method is to identify buffer spaces with the minimum possible storage level needed to achieve the highest output level.

It is assumed that arrivals to the first machine are Poisson with mean rate λ and service times at the ith machine are exponentially distributed with mean rate μ_i. Two classes of problems, balanced and unbalanced lines, are examined.

Balanced Lines

Sets 1-4 are examples of balanced lines. In these sets, the mean service rates of the machines are equal. The buffer allocation problem, in balanced lines with predetermined total buffer capacity, is well understood [11]. Many researchers have studied BAP in balanced lines. Their results show that for CV's below 1.0, the optimal allocation is also balanced, while for CV's more than 1.0, the bowl phenomenon becomes optimal [12-14]. In the proposed algorithm, BAP is solved while total buffer capacity is not predetermined.

Four sets are as follows:

Set 1: $n = 3,\ \lambda = 0.5,$
$\mu_1 = \mu_2 = \mu_3 = 3,$

Set 2: $n = 7,\ \lambda = 0.5,$
$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = 3,$

Set 3: $n = 4,\ \lambda = 3,$
$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 6,$

Set 4: $n = 7,\ \lambda = 3,$
$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = 6.$

To demonstrate the steps of the algorithm, Set 1 is solved with the values of $\beta = 0.01$ and $\alpha_i = 0.001 (i = 2, 3, \cdots, n)$ as follows:

Step 1:

$$r = \frac{\lambda}{\mu_1} = \frac{0.5}{3} = \frac{1}{6},$$

$$P_1(X_1) = \frac{(1-r)r^{X_1}}{(1-r^{X_1+1})} = \frac{(1-\frac{1}{6})\frac{1}{6}^{X_1}}{1-\frac{1}{6}^{X_1+1}} = 0.01,$$

$$X_1 = 2.46 \text{ (rounding)} \Rightarrow X_1 = 3,$$

$$P_1(0) = \frac{1-\frac{1}{6}}{1-\frac{1}{6}^4} = 0.8339$$

$$\Rightarrow (\text{Out})_{(1)} = (1-0.8339)3 = 0.498,$$

Step 2:

$$i = 2, \quad \lambda_2 = \text{Out}_{(1)}, \quad \rho_2 = \frac{\lambda_2}{\mu_2} = \frac{0.498}{3},$$

$$X_2 = \frac{\ln \alpha_2}{\ln \rho_2} - 1 \Rightarrow X_2 = 3,$$

$$P_2(0) = \frac{1 - \frac{0.498}{3}}{1 - \frac{0.498^4}{3}} = 0.8346 \Rightarrow \text{Out}_{(2)}$$

$$= (1 - 0.8346)3 = 0.496,$$

$$i = 3, \quad \lambda_3 = \text{Out}_{(2)}, \quad \rho_3 = \frac{\lambda_3}{\mu_3} = \frac{0.496}{3},$$

$$X_3 = \frac{\ln \alpha_3}{\ln \rho_3} - 1 \Rightarrow X_3 = 3$$

$$\Rightarrow \text{buffer allocation} : (X_1, X_2, X_3) = (3, 3, 3).$$

The manager defines α and β. It is obvious that different values of α and β yield different allocations. For example, for Set 1, if $\beta = 0.001$ and $\alpha_i = 0.001$, then, the buffer allocation (4,3,3) is resulted. The following parameter values are used throughout all sets reported in this paper: $\beta = 0.01$ and $\alpha_i = 0.001(i = 2, 3, \cdots, n)$.

Other sets are solved in this way and are shown in Table 2.

Unbalanced Lines

Sets 5 to 12 are examples of unbalanced lines. In the unbalanced lines, one or more bottleneck operations exist whose average unit-processing time is longer than that of other operations in the line. In the literature, it is well documented that the maximum attainable output level of an unbalanced line is determined by the performance and functioning of the bottleneck operation [11]. Thus, the output level of an unbalanced line is maximized by avoiding or reducing the idle time of the bottleneck operation caused by starvation or blockage. Hence, the buffers surrounding the bottleneck operation are of specific importance and are likely to contain larger storage spaces than other buffers in the line.

Single Bottleneck

One begins with four sets of experiments, in which the mean service rate of a single machine is less than other machines. Sets 5 to 8 are examples of unbalanced lines with one bottleneck. it must be noted that the position of the bottleneck is different in all sets.

Set 5: $n = 3, \lambda = 0.5,$

$$\mu_1 = \mu_3 = 3, \ \mu_2 = 1,$$

Set 6: $n = 7, \lambda = 0.5,$

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_7 = 3, \ \mu_6 = 1,$$

Set 7: $n = 5, \lambda = 0.5,$

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 3, \ \mu_5 = 1,$$

Set 8: $n = 7, \lambda = 0.5,$

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = 3, \ \mu_7 = 1.$$

The results of the proposed heuristic algorithm are shown in Table 2. As expected, a single bottleneck station draws buffers toward it.

Multi-Bottleneck

One attention now turns to the flow lines having two or more bottlenecks. To illustrate the power of the heuristic algorithm, four sets of production line with different numbers of machines and service rates are

Table 2. Comparison of proposed heuristic method and other methods.

	Proposed Heuristic Alg.		Smith and Daskalaki Method		First Space Finite and Others Infinite	
	Buffer Allocation	Throughput	Buffer Allocation	Throughput	Buffer Allocation	Throughput
Set 1	(3, 3, 3)	0.5	(9, 9, 10)	0.5	(3, Inf, Inf)	0.5
Set 2	(3, 3, 3, 3, 3, 3, 3)	0.49	(9, 9, 9, 9, 9, 9, 10)	0.5	(3, Inf, Inf, Inf, Inf, Inf, Inf)	0.5
Set 3	(6, 9, 9, 9)	2.95	(18, 6, 25, 25)	3	(6, Inf, Inf, Inf)	2.96
Set 4	(6, 9, 9, 9, 9, 9, 9)	2.98	(18, 6, 6, 6, 25, 25, 25)	3	(6, Inf, Inf, Inf, Inf, Inf, Inf)	2.98
Set 5	(3, 9, 3)	0.5	(10, 25, 10)	0.5	(3, Inf, Inf)	0.5
Set 6	(3, 3, 3, 3, 3, 9, 3)	0.49	(10, 10, 10, 10, 10, 25, 10)	0.5	(3, Inf, Inf, Inf, Inf, Inf, Inf)	0.49
Set 7	(3, 3, 3, 3, 9)	0.5	(10, 10, 10, 10, 25)	0.5	(3, Inf, Inf, Inf, Inf)	0.5
Set 8	(3, 3, 3, 3, 3, 3, 9)	0.48	(10, 10, 10, 10, 10, 10, 25)	0.5	(3, Inf, Inf, Inf, Inf, Inf, Inf)	0.48
Set 9	(3, 4, 3, 3, 9)	0.5			(3, Inf, Inf, Inf, Inf)	0.5
Set 10	(3, 3, 3, 3, 9, 4)	0.49			(3, Inf, Inf, Inf, Inf, Inf)	0.49
Set 11	(3, 3, 3, 3, 3, 9, 3, 4)	0.49			(3, Inf, Inf, Inf, Inf, Inf, Inf, Inf)	0.49
Set 12	(6, 3, 4, 3, 3, 3)	0.49			(6, Inf, Inf, Inf, Inf, Inf)	0.49

solved:

Set 9: $n = 5,\ \lambda = 0.5,$

$$\mu_1 = \mu_3 = \mu_4 = 3,\ \mu_2 = 2,\ \mu_5 = 1,$$

Set 10: $n = 6,\ \lambda = 0.5,$

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 3,\ \mu_5 = 1,\ \mu_6 = 2,$$

Set 11: $n = 8,\ \lambda = 0.5,$

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_7 = 3, \mu_6 = 1, \mu_8 = 2,$$

Set 12: $n = 6,\ \lambda = 0.5,$

$$\mu_2 = \mu_4 = \mu_5 = 3,\ \mu_1 = 1,\ \mu_3 = 2,\ \mu_6 = 4.$$

Table 2 shows the results. It can be seen that the bottleneck with the lower service rate draws more buffers toward it, even in the presence of the second bottleneck [4].

Evaluating the Proposed Heuristic Algorithm

In order to evaluate the proposed formulae for X_i that is $X_i = \frac{\ln \alpha_i}{\ln \rho_i} - 1$, each set is simulated with a buffer profile as form $(X_1, \text{inf}, \text{inf}, \cdots, \text{inf})$, which means a finite buffer at the first machine and an infinite buffer capacity for other machines.

SLAM (Pritsker and Associates, Inc.,1988) is used to simulate the problems. The throughput of the line is evaluated with the simulated time set to 11,000 min. The first 1,000 min of each run were viewed as the transient period and, therefore, were not used in the analysis. The remaining 10,000 min are viewed as representing the simulated steady state of the line. The number of units processed at the last operation, during the remaining 10,000, is used to represent the throughput of the line.

The result of the simulation for buffer allocation $(X_1, \text{inf}, \text{inf}, \cdots, \text{inf})$ for all sets is shown in Table 2. As can be seen, the throughput of the buffer allocation using the heuristic algorithm is equal or, in some cases, close to the buffer profile as $(X_1, \text{inf}, \text{inf}, \cdots, \text{inf})$. So, it can be resulted that buffer profiles $(X_1, \text{inf}, \text{inf}, \cdots, \text{inf})$ and $(X_1, X_2, X_3, \cdots, X_n)$ are equivalent.

The proposed algorithm is also compared with the Smith and Daskalaki method [15]. Smith and Daskalaki combine the expansion method and Powell's method into an overall design methodology to solve the BAP. The objective of their method was to determine the optimal allocation of buffers so that the throughput is maximized, while minimizing the holding and buffer storage costs. The basic objective function is as follows:

$$\text{maximize } Z = R(P - V) - HL,$$

where:

$R =$ average throughput of the nth machine,
$P =$ average revenue/item,
$V =$ average variable production cost,
$H =$ average holding cost/item,
$L =$ average total number of units in the production line at steady state.

The parameter values that are used in their method are $P = \$30.00/\text{unit}, V = \$10.00/\text{unit}, H = \$0.5/\text{unit}$ [15].

The results of the Smith and Daskalaki method for Sets 1 to 8 are shown in Table 2. As can be seen, the optimal buffer allocation proposed by the presented heuristic algorithm needs a smaller buffer capacity.

It is observed in all sets that in the proposed heuristic algorithm, the buffer capacity in the first station is smaller than that in the Smith and Daskalaki method. As a result, input to the system is controlled and more limited. Since average WIP decreases as buffer capacity decreases, it is expected that the proposed heuristic algorithm will have less WIP and so more effectiveness.

It is possible to discuss the parameters used in the Smith and Daskalaki method. For example, it is expected that by increasing the H value, the buffer capacity for each machine decreases. But the point is, the throughput of the heuristic method is equal or, in some sets, close to the smith method but needs less levels of storage. It means that two methods have almost the same throughput while the proposed heuristic method suggests less buffer capacity.

SUMMARY AND CONCLUSIONS

The BAP in a finite production flow line is one of the most difficult problems in performance modeling and in the design of production flow lines. As Altiok [16] has stated, this is due, in part, to the combinatorial nature of the problems and, in part, to the lack of explicit differentiable equations for the measure of performances involved in the design problems.

In this paper, a heuristic algorithm to solve buffer allocation problems is proposed. The main idea is to find the amount of buffers under which, the system operates as the system with infinite buffer capacity, so the stations will be independent of each other and throughput becomes maximum.

Examples demonstrate that the proposed algorithm finds optimal and near optimal allocations in balanced and unbalanced lines with one and more bottlenecks. The algorithm finds the allocation where it's throughput is maximum or only slightly less than maximum but it's total buffer capacity is considerably less than optimal allocation. It must be noted that the algorithm finds allocation without predetermined total buffer capacity. Consequently, the proposed algorithm

finds the optimal, or near optimal, allocation with less WIP.

Future research could be directed towards finding the effects of line length, variability, bottleneck, service time distributions, service discipline etc. in the optimal buffer allocation in flow lines with n stations.

REFERENCES

1. Lutz, C.M. "Determination of buffer size and location in scheduling systems", Ph.D. Dissertation, Terry College of business, The University of Georgia at Athens, GA (1995).
2. Lutz, C.M., Davis, K.R. and Sun, M. "Determining buffer location and size in production lines using tabu search", *European Journal of Operational Research*, **106**, pp 301-316 (1998).
3. Singh, A. and MacGregor Smith, J. "Buffer allocation for an integer nonlinear network design problem", *Computers and Operations Research*, **24**(5), pp 453-472 (1997).
4. Harris, J.H. and Powell, S.G. "An algorithm for optimal buffer placement in reliable serial lines", *IIE Transactions*, **31** (1999).
5. Altiok, T. and Stidham, S. "The allocation of interstage buffer capacities in production lines", *IIE Transactions*, **15**(4), pp 292-299 (1983).
6. MacGregor Smith, J. and Daskalaki, S. "Buffer space allocation in automation assembly lines", *Operations Research*, **36**(2), pp 343-358 (1988).
7. Papadopoulos, H.T., Heavy, C. and Brown, J., *Queuing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall (1993).
8. Park, T. "A two-phase heuristic algorithm for determining buffer sizes of production lines", *International Journal of Production Research*, **31**(3), pp 613-631 (1993).
9. Ho, Y.C., Eyler, M.A. and Chien, T.T. "A gradient technique for general buffer storage design in a production line", *International Journal of Production Research*, **17**, pp 557-580 (1979).
10. Saaty, T., *Elements of Queuing Theory*, McGraw-Hill (1961).
11. Powell, S.G. and Pyke, D.F. "Allocation of buffers to serial production lines with bottlenecks", *IIE Transactions*, **28**, pp 18-29 (1996).
12. Hillier, F.S. and So, K.C. "The effect of machine breakdowns and interstage storage on the performance of production line systems", *International Journal of Production Research*, **29**(10), pp 2043-2055 (1991).
13. Hillier, F.S. and So, K.C. "The effect of the coefficient of operation times on the allocation of storage space in production line systems", *IIE Transactions*, **23**(2), pp 198-206 (1991).
14. Hillier, F.S., So, K.C. and Boling, R.W. "Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times", *Management Science*, **39**(1), pp 126-133 (1993).
15. MacGregor Smith, J. and Daskalaki, S. "Buffer space allocation in automated assembly lines", *Operations Research*, **36**(2), pp 343-357 (1988).
16. Altiok, T. "Performance analysis of manufacturing systems", *Springer Series in Operations Research* (1997).