On the Optimal Buffer Allocation of an FMS with Finite In-Process Buffers

Soo-Tae Kwon

Department of Information Systems,
School of Information Technology and Engineering, Jeonju University

Abstract. This paper considers a buffer allocation problem of flexible manufacturing system composed of several parallel workstations each with both limited input and output buffers, where machine blocking is allowed and two automated guided vehicles are used for input and output material handling. Some interesting properties are derived that are useful for characterizing optimal allocation of buffers for the given FMS model. By using the properties, a solution algorithm is exploited to solve the optimal buffer allocation problem, and a variety of differently-sized decision parameters are numerically tested to show the efficiency of the algorithm.

Keywords: FMS, Queueing Network, Throughput, Buffer.

1 Introduction

Flexible manufacturing systems (FMSs) have been introduced in an effort to increase productivity by reducing inventory and increasing the utilization of machining centers simultaneously. An FMS combines the existing technology of NC manufacturing, automated material handling, and computer hardware and software to create an integrated system for the automatic random processing of palletized parts across various workstations in the system.

The design of an FMS begins with a survey of the manufacturing requirements of the products produced in the firm with a view to identifying the range of parts which should be produced on the FMS. Then the basic design concepts must be established. In particular the function, capability and number of each type of workstation, the type of material handling system and the type of storage should be determined.

At the detailed design stage it will be necessary to determine such aspects as the required accuracy of machines, tool changing systems and the method of feeding and locating parts at machines. Then the number of transport devices, the number of pallets, the capacity of central and local storages must be determined, along with some general strategies for work transport to reduce delays due to interference.

One of key questions that the designer face in an FMS is the buffer allocation problem, i.e., how much buffer storage to allow and where to place it within the system. This is an important question because buffers can have a great impact on

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3982, pp. 767–776, 2006.

the efficiency of production. They compensate for the blocking and the starving of the workstations. Unfortunately, buffer storage is expensive both due to its direct cost and due to the increase of the work-in-process(WIP) inventories. Also, the requirement to limit the buffer storage can be a result of space limitations in the factory.

Much research has concentrated on queueing network model analyses to evaluate the performance of FMSs, and concerned with mathematical models to address the optimization problems of complex systems such as routing optimization, server allocation, workload allocation, buffer allocation on the basis of the performance model. Vinod and Solberg (1985) have presented a methodology to design the optimal system configuration of FMSs modeled as a closed queueing networks of multiserver queues. Buzacott and Yao (1986) have reviewed the work on modelling FMS with particular focus on analytical models. Shanthikumar and Yao (1989) have solved the optimal buffer allocation problem with increasing concave production profits and convex buffer space costs. Paradopoulos and Vidalis (1999) have investigated the optimal buffer allocation in short balanced production lines consisting of machines that are subject to breakdown. Enginarlar et al. (2002) have investigated the smallest level of buffering to ensure the desired production rate in serial lines with unreliable machines.

In the above-mentioned reference models, machines were assumed not to be blocked, that is, not to have any output capacity restriction. These days, the automated guided vehicle (AGV) is commonly used to increase potential flexibility. By the way, it may not be possible to carry immediately the finished parts from the machines which are subject to AGV's capacity restriction. The restriction can cause any operation blocking at the machines, so that it may be desirable to provide some storage space to reduce the impact of such blocking. In view of reducing work-in-process storage, it is also required to have some local buffers of proper size at each workstation. Sung and Kwon(1994) have investigated a queueing network model for an FMS composed of several parallel workstations each with both limited input and output buffers where two AGVs are used for input and output material handling, and Kwon (2005) has considered a workload allocation problem on the basis of the same model.

In this paper, a buffer allocation problem is considered to yield the highest throughput for the given FMS model (Sung and Kwon 1994). Some interesting properties are derived that are useful for characterizing optimal allocation of buffers, and some numerical results are presented.

2 The Performance Evaluation Model

The FMS model is identical to that in Sung and Kwon(1994). The network consists of a set of n workstations. Each workstation $i (i = 1, \cdots, n)$ has machines with both limited input and output buffers. The capacities of input and output buffers are limited up to IB_i and OB_i respectively, and the machines perform in an exponential service time distribution. All the workstations are linked to an automated storage and retrieval system (AS/RS) by AGVs which consist of

AGV(I) and AGV(O). The capacity of the AS/RS is unlimited, and external arrivals at the AS/RS follow a Poisson process with rate λ.

The FCFS (first come first served) discipline is adopted here for the services of AGVs and machines. AGV(I) delivers the input parts from the AS/RS to each input buffer of workstations, and AGV(O) carries the finished parts away from each output buffer of workstations to the AS/RS, with corresponding exponential service time distributions. Specifically, AGV(I) distributes all parts from the AS/RS to the workstations according to the routing probabilities $\gamma_i(\sum_{i=1}^{n} \gamma_i = 1)$ which can be interpreted as the proportion of part dispatching from the AS/RS to workstation i.

Moreover, any part (material) can be blocked on arrival (delivery) at an input buffer which is already full with earlier-arrived parts. Such a blocked part will be recirculated instead of occupying the AGV(I) and waiting in front of the workstation (block-and-recirculate mechanism). Any finished part can also be blocked on arrival at an output buffer which is already full with earlier-finished parts. Such a blocked part will occupy the machine to remain blocked until a part departure occurs from the output buffer. During such a blocking time, the machine cannot render service to any other part that might be waiting at its input buffer (block-and-hold mechanism).

Sung and Kwon(1994) have developed an iterative algorithm to approximate system performance measures such as system throughput and machine utilization. The approximation procedure decomposes the queueing network into individual queues with revised arrival and service processes. These individual queues are then analyzed in isolation. The individual queues are grouped into two classes. The first class consists of input buffer and machine, and the second one consists of output buffers and AGV(O). The first and second classes are called the first-level queue and the second-level queue, respectively.

The following notations are used throughout this paper ($i = 1, \cdots, n$):

λ	external arrival rate at AS/RS
λ_i	arrival rate at each input buffer i in the first-level queue
λ_i^*	arrival rate at each output buffer i in the second-level queue
μ	service rate of AGV
μ_i	service rate of machine i
$P(k_1, \cdots, k_n)$	probability that there are k_i units at each output buffer i in the second-level queue with infinite capacity.
$P(idle)$	probability that there is no unit in the second-level queue with infinite capacity.
$\prod(k_1, \cdots, k_n)$	probability that there are k_i units at each output buffer i in the second-level queue with finite capacity.
$\prod(idle)$	probability that there is no unit in the second-level queue with finite capacity.

The second-level queue is independently analyzed first to find the steady-state probability by using the theory of reversibility. The steady-state probability is derived as follows.

Lemma 1. (*refer to Sung and Kwon* 1994, *Theorem* 2)

The steady-state probability of the second-level queue is derived as

$$\prod(k_1, \cdots, k_n) = P(k_1, \cdots, k_n)/G$$
$$\prod(idle) = P(idle)/G \tag{1}$$

where,

$A = \{(k_1, \cdots, k_n) | 0 <= k_i <= OB_i, 1 <= i <= n\}$,
$G = \sum_{(k_1,\cdots,k_n)\in A} P(k_1, \cdots, k_n) + P(idle)$,
$P(k_1, \cdots, k_n) = (1-\rho)\cdot \rho^{(k_1+\cdots+k_n+1)} \cdot \frac{(k_1+\cdots+k_n)!}{k_1!\cdots k_n!} \cdot {q_1}^{k_1} \cdots {q_n}^{k_n}$,
$P(idle) = 1 - \rho$,
$\rho = \sum_{i=1}^{n} \lambda_i^*/\mu$,
$q_i = \lambda_i^* / \sum_{i=1}^{n} \lambda_i^*$.

It is followed by finding the clearance service time accommodating all the possible blocking delays that a part might undergo due to the phenomenon of blocking. The clearance time is derived from the steady-state probability of second-level queue. Then, the first-level queues are analyzed by this expected clearance time in the approach of the M/M/1/K queueing model.

3 The Buffer Allocation Problem

In FMS, a frequently encountered problem is concerned with how to allocate buffer space among several subsystems for maximizing the production rate (system throughput). In this section, the buffer allocation problem is considered to yield the highest throughput for the given performance evaluation model. The optimal buffer allocation problem can be stated as follows :

$$\begin{aligned} &Maximize \quad Z = TH(x_1, \cdots x_n, x_{n+1}, \cdots x_{2n}) \\ &s.t. \quad \sum_{i=1}^{2n} x_i \le S \end{aligned} \tag{2}$$

where

$TH(x_1, \cdots x_n, x_{n+1}, \cdots x_{2n})$ = the system throughput,

x_i = the number of buffers allocated to buffer i(input buffer : $1 \le i \le n$, output buffer : $n+1 \le i \le 2n$), that is$(IB_1, \cdots, IB_n, OB_1, \cdots, OB_n)$

S = the maximum total number of buffers to be allocated.

Despite of its practical importance, this buffer allocation problem has not been successfully studied in the literature. The major difficulty appears to be lack of known properties regarding the throughput of system as a function of its buffer capacity. Some interesting properties for the associated system throughput are now derived.

Property 1. In the first-level queue-alone subsystem, the throughput is a monotonically increasing concave function of its buffer size.

Proof:
Let $\rho_i(=\lambda_i/\mu_i)$ and IB_i denote the utilization and the buffer size of the first-level queue i, respectively. Then the throughput of the first-level queue i, $TH(\lambda_i, IB_i, \mu_i)$, can be derived as follows :

$$TH(\lambda_i, IB_i, \mu_i) = \mu_i \cdot (1 - \frac{1-\rho_i}{1-\rho_i^{IB_i+1}})$$

By the definition of $TH(IB_i)$,

$$\begin{aligned}
&TH(\lambda_i, IB_i+1, \mu_i) - TH(\lambda_i, IB_i, \mu_i) \\
&\quad = \mu_i \cdot (\frac{1-\rho_i}{1-\rho_i^{IB_i+1}} - \frac{1-\rho_i}{1-\rho_i^{IB_i+2}}) \\
&\quad = \mu_i \cdot (\frac{1}{1+\rho_i+\rho_i^2+\cdots+\rho_i^{IB_i}} - \frac{1}{1+\rho_i+\rho_i^2+\cdots+\rho_i^{IB_i+1}}) \\
&\quad > 0 \quad \text{for all} \quad \rho_i.
\end{aligned}$$

And,

$$\begin{aligned}
&2TH(\lambda_i, IB_i+1, \mu_i) - TH(\lambda_i, IB_i, \mu_i) - TH(\lambda_i, IB+2, \mu_i) \\
&\quad = 2 \cdot \mu_i \cdot (1 - \frac{1-\rho_i}{1-\rho_i^{IB_i+2}}) - \mu_i \cdot (1 - \frac{1-\rho_i}{1-\rho_i^{IB_i+1}}) - \mu_i \cdot (1 - \frac{1-\rho_i}{1-\rho^{IB_i+3}}) \\
&\quad = \mu_i \cdot (1-\rho_i)[\frac{1}{1-\rho_i^{IB_i+1}} + \frac{1}{1-\rho_i^{IB_i+3}} - \frac{2}{1-\rho_i^{IB_i+2}}] \\
&\quad = \frac{\mu_i \cdot (1-\rho_i)}{(1-\rho_i^{IB_i+1})(1-\rho_i^{IB_i+3})(1-\rho_i^{IB_i+2})} \cdot [\rho_i^{IB_i+3} + \rho_i^{IB_i+1} - 2\rho_i^{IB_i+2} \\
&\qquad\qquad + \rho_i^{2\cdot IB_i+5} + \rho_i^{2\cdot IB_i+3} - 2\rho_i^{2\cdot IB_i+4}] \\
&\quad = \frac{\mu_i \cdot (1-\rho_i)^3 \cdot (\rho_i^{IB_i+1} + \rho_i^{2\cdot IB_i+3})}{(1-\rho_i^{IB_i+1})(1-\rho_i^{IB_i+3})(1-\rho_i^{IB_i+2})} \\
&\quad > 0 \quad \text{for all} \quad \rho_i.
\end{aligned}$$

Thus, the throughput of the first-level queue is a monotonically increasing concave function of buffer size. This completes the proof.

Also, the throughput of the second-level queue is characterized as follows.

Property 2. In the second-level queue-alone subsystem, the throughput is a monotonically increasing concave function of its buffer size.

Proof:
Let $\rho(=\sum_{i=1}^{n} \frac{\lambda_i^*}{\mu})$ and $(OB_1, \cdots, OB_n)$ denote the utilization and the output buffer sizes of the second-level queue, respectively. Then the throughput of the second-level queue, $TH(\lambda_1^*, \cdots, \lambda_n^*, OB_1, \cdots, OB_n, \mu)$, can be derived as follows.

$$\begin{aligned}
TH(\lambda_1^*, \cdots, \lambda_n^*, OB_1, \cdots, OB_n, \mu) &= \mu \cdot (1 - \prod(idle)) \\
&= \mu \cdot (1 - \frac{1-\rho}{G(\lambda_1^*, \cdots, \lambda_n^*, OB_1, \cdots, OB_n, \mu)}) \\
&= \mu \cdot (1 - \frac{1}{\phi(\lambda_1^*, \cdots, \lambda_n^*, OB_1, \cdots, OB_n, \mu)})
\end{aligned}$$

where,

$$G(\lambda_1^*,\cdots,\lambda_n^*,OB_1,\cdots,OB_n,\mu) = (1-\rho)[1+\sum_{k_1=0}^{OB_1}\cdots\sum_{k_n=0}^{OB_n}\rho^{n+1}\frac{(k_1+\cdots+k_n)!}{k_1!\cdots k_n!}q_1^{k_1}\cdots q_n^{k_n}],$$

$$\phi(\lambda_1^*,\cdots,\lambda_n^*,OB_1,\cdots,OB_n,\mu) = 1+\sum_{k_1=0}^{OB_1}\cdots\sum_{k_n=0}^{OB_n}\rho^{n+1}\frac{(k_1+\cdots+k_n)!}{k_1!\cdots k_n!}q_1^{k_1}\cdots q_n^{k_n}$$

$$n = k_1+\cdots+k_n$$

And, let

$$\psi_1 = \sum_{k_1=0}^{OB_1}\cdots\sum_{k_i=OB_i+1}^{OB_i+1}\cdots\sum_{k_n=0}^{OB_n}\rho^{n+1}\frac{(k_1+\cdots+k_n)!}{k_1!\cdots k_n!}q_1^{k_1}\cdots q_n^{k_n}, \quad \text{and}$$

$$\psi_2 = \sum_{k_1=0}^{OB_1}\cdots\sum_{k_i=OB_i+2}^{OB_i+2}\cdots\sum_{k_n=0}^{OB_n}\rho^{n+1}\frac{(k_1+\cdots+k_n)!}{k_1!\cdots k_n!}q_1^{k_1}\cdots q_n^{k_n}$$

These lead to the relation $\psi_1 > \psi_2$, and it holds that

$$\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i+1,\cdots,OB_n,\mu) = \phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1$$

$$\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i+2,\cdots,OB_n,\mu) = \phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1+\psi_2$$

By the definition of $TH(\lambda_1^*,\cdots,\lambda_n^*,OB_1,\cdots,OB_n,\mu)$

$$\begin{aligned} &TH(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i+1,\cdots,OB_n,\mu) - TH(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu) \\ &= \mu\cdot[\frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)} - \frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1}] \\ &> 0 \quad \text{for all} \quad OB_i \end{aligned}$$

And, $2\cdot TH(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i+1,\cdots,OB_n,\mu)$

$$\begin{aligned} &- TH(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu) \\ &- TH(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i+2,\cdots,OB_n,\mu) \\ &= \mu\cdot[\frac{-2}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1} \\ &\quad + \frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)} \end{aligned}$$

$$+\frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1+\psi_2}]$$
$$=\mu\cdot[\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots OB_n,\mu)\cdot(\psi_1-\psi_2)+\psi_1^2+\psi_1\cdot\psi_2]$$
$$\cdot\frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1}$$
$$\cdot\frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)}$$
$$\cdot\frac{1}{\phi(\lambda_1^*,\cdots\lambda_n^*,OB_1,\cdots,OB_i,\cdots,OB_n,\mu)+\psi_1+\psi_2}$$
$$>0 \quad \text{for all} \quad OB_i$$

Thus, the throughput of the second-level queue is a monotonically increasing concave function of buffer size. This completes the proof.

Finally, the following result can be obtained by the comparison of the throughputs for buffer allocation scheme.

Property 3. In the second-level queue-alone subsystem, the balanced buffer allocation scheme maximizes the throughput.

Proof:
For simplification, the proof will be completed only for the case of $n = 2, S = 2$ and $\lambda_1 = \lambda_2$. Let TH^b and TH^{nb} be the throughput of the balanced allocation scheme case ($OB_1 = OB_2 = 1$) and the other one ($OB_1 = 2$), respectively.

$$TH^b - TH^{nb} = \mu(1-\frac{1-\rho}{G^b}) - \mu(1-\frac{1-\rho}{G^{nb}}) = \mu(1-\rho)\frac{G^b - G^{nb}}{G^b \cdot G^{nb}}$$

Since $q_1 = q_2$ and by the definition of G,

$$\begin{aligned} G^b - G^{nb} &= (1-\rho)[(1-\rho+\rho^2+2\rho^3 q_1 q_2) - (1+\rho+\rho^2 q_1+\rho^3 q_1^2)] \\ &= (1-\rho)[\rho^2(1-q_1)+\rho^3(2q_1q_2-q_1^2)] \\ &= (1-\rho)[\frac{1}{2}\rho^2+\frac{1}{2}\rho^3] \end{aligned}$$

Therefore, $TH^b - TH^{nb} \geq 0$. This complete the proof.

On the basis of properties, now consider the optimization problem in (2). Since the throughput is a monotonically increasing concave function of buffer capacities for both first-level and second-level queue as proved, the marginal allocation approach of Fox (1966) can be used to efficiently solve the optimal buffer allocation problem in (2). The idea of this approach is as follows.

Let $\triangle TH(x_1,\cdots,x_i,\cdots,x_{2n}) = TH(x_1,\cdots,x_i{+}1,\cdots,x_{2n}) - TH(x_1,\cdots,x_i,\cdots,x_{2n})$ for all i. Allocate the available buffer spaces to the buffer that would yield the largest increase in $\triangle F(x_i)$ one at a time. Continue this procedure until the available buffer spaces are exhausted.

The algorithm of the buffer allocation problem can be summarized as follows.

`Step 1.` Set $x_i = 0$, for all $i (= 1, \cdots, 2n)$.

`Step 2.` For all i, calculate $\triangle TH(x_1, \cdots, x_i, \cdots, x_{2n}) = TH(x_1, \cdots, x_i + 1, \cdots, x_{2n}) - TH(x_1, \cdots, x_i, \cdots, x_{2n})$ by using the performance evaluation model.

Table 1. The result of the parameter set 1

iteration	$x_1 = IB_1$	$x_2 = OB_2$	throughput	increment	allocation
0	0	0	.5262	-	
1	1	0	.6667	.1405	**
	0	1	.5846	.4941	
2	2	0	.7313	.0646	**
	1	1	.7311	.0644	
3	3	0	.7677	.0364	
	2	1	.7965	.0652	**
4	3	1	.83288	.03638	**
	2	2	.83287	.03637	
5	4	1	.8558	.0229	
	3	2	.8685	.0356	**

Table 2. The result of the parameter set 2

S	x_1 IB_1	x_2 IB_2	x_3 IB_3	x_4 IB_4	TH	$\triangle$	allocation	S	x_1 IB_1	x_2 IB_2	x_3 IB_3	x_4 IB_4	TH	$\triangle$	allocation
0	0	0	0	0	.6291	-									
1	1	0	0	0	.7043	.0752	**	6	3	1	1	1	.8867	.0142	
	0	1	0	0	.7043	.0752			2	2	1	1	.9003	.0278	**
	0	0	1	0	.6641	.035			2	1	2	1	.882	.0095	
	0	0	0	1	.6641	.035			2	1	1	2	.89	.0175	
2	2	0	0	0	.7378	.033		7	3	2	1	1	.9149	.0146	**
	1	1	0	0	.7758	.0715	**		2	3	1	1	.9149	.0146	
	1	0	1	0	.731	.0267			2	2	2	1	.912	.0117	
	1	0	0	1	.7438	.0395			2	2	1	2	.912	.0117	
3	2	1	0	0	.8087	.0329	**	8	4	2	1	1	.9232	.0083	
	1	2	0	0	.8087	.0329			3	3	1	1	.9293	.0144	**
	1	1	1	0	.807	.0312			3	2	2	1	.9217	.0068	
	1	1	0	1	.807	.0312			3	2	1	2	.9284	.0135	
4	3	1	0	0	.8268	.0181		9	4	3	1	1	.9376	.0083	
	2	2	0	0	.8406	.0319			3	4	1	1	.9376	.0083	
	2	1	1	0	.8312	.0225			3	3	2	1	.9379	.0086	**
	2	1	0	1	.8425	.0338	**		3	3	1	2	.9379	.0086	
5	3	1	0	1	.8629	.0204		10	4	3	2	1	.9427	.0048	
	2	2	0	1	.8663	.0238			3	4	2	1	.9479	.01	
	2	1	1	1	.8725	.03	**		3	3	3	1	.9394	.0015	
	2	1	0	2	.855	.0125			3	3	2	2	.9524	.0145	**

`Step 3.` Find k such that $\triangle TH(x_1, \cdots, x_k, \cdots x_{2n})$

$$= \max_{1 \le i \le 2n} \triangle TH(x_1, \cdots, x_i, \cdots x_{2n}).$$

Set $x_k = x_k + 1, S = S - 1$.
If $S > 0$, then go to step 2.

`Step 4.` Stop.

In order to illustrate the solution procedure, the buffer allocation problem on the basis of the given performance evaluation model is considered with parameter set 1 ($\lambda = 1, \mu_1 = \mu_2 = 2, \mu = 1, S = 5$), parameter set 2 ($\lambda = 1, r_1 = r_2 = 0.5, \mu_1 = \mu_2 = 2, \mu = 1, S = 10$) and parameter set 3 ($\lambda = 1, r_1 = r_2 = 0.5, \mu_1 = 1, \mu_2 = 2, \mu = 1, S = 10$), where S, λ, r_i, μ_i, and μ denote the maximum total number of buffers to be allocated, the arrival rate, the routing probability, the machine service rate, and the AGV service rate, respectively.

At first, a simple system with a single workstation is considered to illustrate the marginal allocation procedure, which is identical to the two stage transfer line. The results of the system with parameter set 1 are shown in Table 1. The table gives both the amount of increment and throughput results.

At second, the buffer allocation problem is considered to test the efficiency of the solution algorithm. For $S = 1, \cdots, 10$, the results of the system with parameter set 2 and 3 are shown in Table 2 and 3, respectively.

Table 3. The result of the parameter set 3

S	x_1 IB_1	x_2 IB_2	x_3 IB_3	x_4 IB_4	TH	△	allocation	S	x_1 IB_1	x_2 IB_2	x_3 IB_3	x_4 IB_4	TH	△	allocation
0	0	0	0	0	.5944	-									
1	1	0	0	0	.6662	.0718		6	3	2	1	0	.8547	.0162	
	0	1	0	0	.6729	.0785	**		2	3	1	0	.8592	.0207	
	0	0	1	0	.6219	.0275			2	2	2	0	.8502	.0117	
	0	0	0	1	.6297	.0353			2	2	1	1	.8692	.0307	**
2	1	1	0	0	.741	.0681	**	7	3	2	1	1	.8869	.0177	**
	0	2	0	0	.7074	.0345			2	3	1	1	.8835	.0143	
	0	1	1	0	.7049	.032			2	2	2	1	.8858	.0166	
	0	1	0	1	.7002	.0273			2	2	1	2	.8789	.0097	
3	2	1	0	0	.7731	.0321		8	4	2	1	1	.898	.0111	
	1	2	0	0	.7745	.0335	**		3	3	1	1	.901	.0141	
	1	1	1	0	.7728	.0318			3	2	2	1	.9014	.0145	**
	1	1	0	1	.771	.03			3	2	1	2	.8973	.0104	
4	2	2	0	0	.8061	.0316		9	4	2	2	1	.9105	.0091	
	1	3	0	0	.7931	.0186			3	3	2	1	.9177	.0163	**
	1	2	1	0	.8095	.035	**		3	2	3	1	.9068	.0054	
	1	2	0	1	.796	.0215			3	2	2	2	.9162	.0148	
5	2	2	1	0	.8385	.029	**	10	4	3	2	1	.9269	.0092	
	1	3	1	0	.8303	.0208			3	4	2	1	.9275	.0098	
	1	2	2	0	.8229	.0134			3	3	3	1	.924	.0063	
	1	2	1	1	.8376	.0281			3	3	2	2	.9281	.0104	**

The computational results present that the solution algorithm is very efficient. In case of $S = s$, the solution algorithm generated the optimal solution at the $s-th$ iteration. However, it is impossible in practice to allocate buffer spaces by conventional approach, since the number of allocating combination becomes explosively large as the number of S and workstation increase.

And, the results of Table 2 and 3 imply that the balanced buffer allocation scheme maximize the system throughput. That is, in order to maximize the system throughput, the buffer should be allocated depending on the routing probability and service rate of machine and AGV.

4 Conclusions

In this paper, a design aspect of a flexible manufacturing system composed of several parallel workstations each with both limited input and output buffers where two AGVs are used for input and output material handling is considered. The optimal design decision is made on the allocation of buffer spaces on the basis of the given performance evaluation model.

Some interesting properties are derived that are useful for characterizing optimal allocation of buffer spaces. The properties are then used to exploit a solution algorithm for allocating buffer spaces. A variety of differently-sized decision parameters are numerically tested to show the efficiency of the algorithm. The results present that the solution algorithm is very efficient.

Further research is to consider the cost factor more explicitly, and also to extend these concepts to general production systems.

References

1. Buzacott, J.A. and Yao, D.D.: Flexible Manufacturing Systems: A Review of Analytical Models. Management Science, **32**, No. 7, (1986) 890-905.
2. Enginarlar, E., Li, J., Meerkov, S.M., and Zhang, R.Q.: Buffer Capacity for Accommodating Machine Downtime in Serial Production Lines, International Journal of Production Research, **40**, No. 3, (2002) 601-624.
3. Fox, B.: Discrete Optimization via Marginal Analysis, Management Science, **13**, No. 3, (1966) 210-216.
4. Kwon, S.T.: On the Optimal Workloads Allocation of an FMS with Finite In-process Buffers, Lecture Notes in Computer Science, **3483**, (2005) 624-631.
5. Papadopoulos, H.T. and Vidalis, M.I.: Optimal Buffer Allocation in short-balanced unreliable production lines, Computers & Industrial Engineering, **37**, (1999) 691-710.
6. Shanthikumar, J. G. and Yao, D. D.: Optimal Buffer Allocation in a Multicell System, The International Journal of Flexible Manufacturing Systems, **1**, (1989) 347-356.
7. Sung, C.S. and Kwon, S.T.: Performance Modelling of an FMS with Finite Input and Output Buffers. International Journal of Production Economics, **37**, (1994) 161-175.
8. Vinod, B. and Solberg, J.J.: The Optimal Design of Flexible Manufacturing Systems, International Journal of Production Research, **23**, No. 6, (1985) 1141-1151.