

# Virtual System Buffer Model in Flexible Manufacturing Systems with an AGV System

AGVS를 포함한 FMS에서의 가상 시스템버퍼 모델에 관한 연구

Kyung Sup Kim, Chong Ha Lee

김경섭\*, 이종하\*\*

## Abstract

This research is concerned with buffer management in a multi-cell FMS with an AGVS. Buffers in manufacturing systems are required to reduce blocking and starving caused by breakdowns, variability in process times, and diversity of part routing. Due to the high per unit buffer cost, which primarily consists of floor space and equipment costs, the total capacity of buffers in an FMS is very limited. Proper buffer management can provide a high system efficiency. This paper presents a buffer management model for a multi-cell FMS with an AGVS and a simulation study to compare the proposed model to a conventional buffer management model in a job shop FMS.

\* School of Electrical and Mechanical Engineering, Yonsei University

\*\* Dept. of Management Information Systems, Yuhan College

## 1. INTRODUCTION

An Flexible manufacturing system (FMS) consists of computer numerically controlled (CNC) machines organized in cells, a material handling system (MHS), and a computer control system for integrating the functions of CNC machines and the MHS. In a multi-cell FMS, a cell contains one or more CNC machines with local buffers, or a cell is specialized for quality inspection, buffering, or loading and unloading jobs into the system[11]. MHSs in a multi-cell FMS can be classified into a primary MHS and a secondary MHS. The primary MHS performs

of buffers in a multi-cell FMS with an AGVS is the subject of this paper.

An example of a multi-cell FMS is depicted in Figure 1. Buffers in a multi-cell FMS can be broadly classified into three types: machine-buffers, cell-buffers, and system-buffers. A machine-buffer is attached to a machine and is accessible only by parts that visit that machine. A cell-buffer is accessible by parts visiting that cell, while a system-buffer is accessible to any part in the system. Some multi-cell FMSs use all three buffer types while others use one or two buffer types.

In an automated job shop it is very plausible

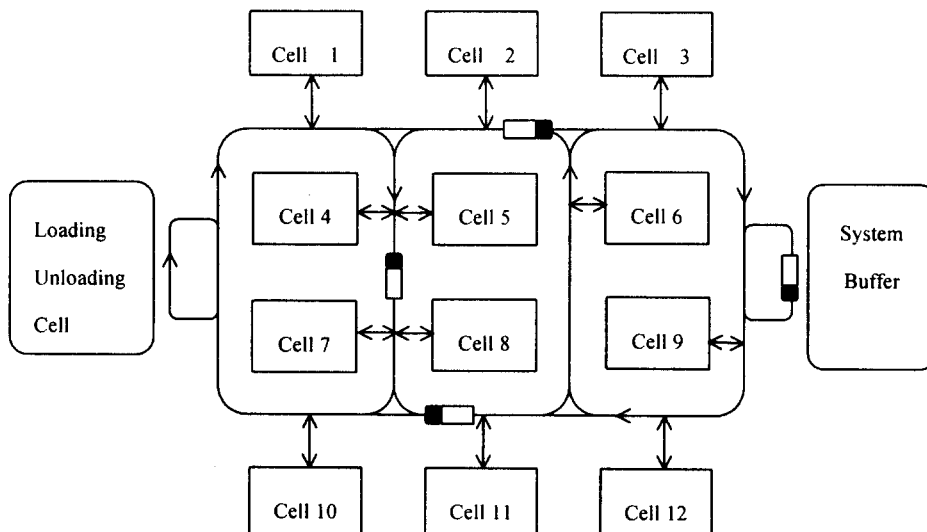


Figure 1. Layout of a multi-cell FMS with an AGVS

inter-cell movements of jobs, while the secondary MHS performs intra-cell movements of jobs. An automated guided vehicle system (AGVS) is an excellent choice for the primary MHS in a multi-cell FMS because of its flexibility in routing and ease of both guide path network modification and capacity expansion without production interruption. The management

that some parts require backtracking, that is, some parts must revisit a process that they have already visited. Furthermore, since parts have different processing steps, the flow of parts might not be unidirectional. These conflicts in the part's flow generally increase system congestion and affect the system throughput. A combination of backtracking, nonunidirectional

flow, and limited buffer space promote deadlock conditions in an FMS.

In automated manufacturing systems such as FMSs, deadlocks should be prevented or the system controller should be equipped with a deadlock resolution mechanism. Otherwise, deadlocks must be resolved by humans in real-time and consequently prohibit the operation of an FMS in an automated mode and affect the system productivity.

The total capacity of buffers in an FMS is very limited. This is due to the high per unit buffer cost that primarily consists of floor space and equipment costs. Therefore, in order to increase the system throughput and to reduce the deadlock possibility, a proper buffer management strategy must be considered before contemplating any increase in the buffer capacities. When devising a buffer management strategy for an FMS with AGVs, the unique characteristics of buffering in such a system must be considered. In this paper we discuss the problem of buffer management in an FMS with AGVs, and propose a buffer management strategy.

## 2. REVIEW OF LITERATURE

A large number of parts in the system cause excessive congestion and traffic on the shop floor and consequently increase the possibility of deadlock[13]. One approach to keep congestion and blocking low is to control the flow of unit loads into the system. Several researchers have been investigating the effects of the input control on the part flow in FMS environments [5,14,15, etc].

An usual input control for both conventional systems and FMSs attempts to balance the assigned workload on each machine and/or to keep the workload under a specified level[15].

Garetti et al.[5] carried out a simulation study for a static dedicated FMS by examining machine dispatching rules versus input control rules with utilization and mean flow time as the criteria. The results indicate that as the size of the system-buffer increases, the utilization value of the system rises, and the difference between the various input control rules diminishes. Sabuncuoglu and Hommertzheim[13] investigated the effects of the number of jobs allowed into an FMS with AGVs and examined different machine and AGV scheduling rules by comparing the mean flow time performance. The results suggested the use of a limit on the number of parts allowed in the system. Also, they pointed out that this limit is a function of scheduling rules and buffer capacities as well as the capacities of the machines and the AGVs. The results can be supported by the fact that if the number of parts is controlled to be less than a specific number, the system is free from deadlocks that are caused by part flow[7].

Buzacott and Shanthikumar[1] showed analytically that an FMS with only a system-buffer is superior to a system with only individual cell-buffers. However, it is not necessarily the case that an FMS with only a system-buffer achieves better throughput because the system-buffer can be occupied by parts waiting for a particular cell[20]. Kamoun and Kleinrock[6] proposed a scheme that controls the usage of a shared buffer (system-buffer) for a computer network node environment. In this scheme, a minimum number of spaces in the shared buffer are reserved for each node and also there is a limit on the maximum number of spaces that can be allocated to each node. Yamashita[20] proposed an approximation algorithm for the analysis of open queueing networks under this scheme. Sharing of

cell-buffers between cells can be found in a paper by Tang et al.[17]. Tang et al. used a multi-cell FMS that has only shared common cell-buffers to test the performance of various scheduling rules. In their simulation test, a part in a cell-buffer is sent to the next closest cell-buffer whenever a machine in the cell is blocked because the cell-buffer is full.

An alternative for the system with limited buffer capacity is changing the buffer configuration during the operations. Given the total capacity of buffers in each cell, it is possible, in some cells, to reallocate machine-buffers in real time for different part mixes to maximize the system throughput via a reprogramming of the automated material handling device (eg. robot) within each cell[18]. Unlike reallocation of buffers within a cell, reallocation of buffer capacities to cells for different part mixes is difficult without physically relocating buffers, and consequently the reallocation of system level might not be justified.

In an FMS with an AGVS, one of the important problems is the system deadlock caused by the limited buffer and the lack of mechanism to prevent it[4]. The deadlock problem in FMS environments is addressed by several authors. The approaches taken in the FMS research to solve deadlock problems include 1) Petri-net[2], 2) Graph theory[19], 3) Queueing network[7] and 4) others[9]. Most of the above approaches are implemented in real-time scheduling and control. Complexity and performance of each model is problem specific.

An increasing number of papers concerned with FMS scheduling and control continues to appear in the literature. However, research efforts to investigate the effect of different buffer management policies on FMS performance are very scarce and limited [16].

### 3. VIRTUAL SYSTEM BUFFER

At the machine and cell levels, buffers may be segregated into input and output buffers. The capacity of each segregated buffer is determined by the operational requirements of the production unit. Segregated buffers can be accessed sequentially or randomly. In a sequentially-accessible buffer, only the location at the ends may be accessed, whereas in a randomly-accessible buffer, any location in the buffer may be accessed. A sequentially-accessible buffer is generally less expensive to build and easier to control than a randomly-accessible buffer. The choice of priority rule for selecting a part from a sequentially-accessible buffer is limited to the first come first serve rule (FCFS) or the last come first serve rule (LCFS) which makes a sequentially-accessible buffer less flexible than a randomly-accessible buffer. However, a variety of priority rules can be used with a randomly-accessible buffer. Thus, by using a randomly-accessible buffer, the full flexibility potential of an FMS can be realized.

A non-segregated input and output buffer creates a common input/output (I/O) buffer for use by both incoming and outgoing parts. Preferably an I/O buffer should be randomly-accessible. Because managing a sequentially-accessible I/O buffer creates unavoidable machine starvation, and nullifies the advantages gained from nonsegregation of input and output buffers. Also, the primary advantage of a nonsegregated randomly-accessible I/O buffer over segregated buffers is reduction of machine and part blockings.

To reduce deadlocks as well as blocking, some parts should be sent to the system-buffer from cell-buffers. Also to reduce starving, some parts in the system-buffer should be sent to

corresponding cell-buffers. The decisions about the movements of parts between cell-buffers and system-buffers are when and which part should be sent to system-buffer from a cell-buffer and vice versa.

In many cases, the system controllers are designed with an assumption that the system-buffer is always available, though there is a limit on the number of parts that can use the buffer at a given moment. This assumption can lead a system to a deadlock. The deadlocks can be prevented by implementing a real-time deadlock resolution model. However, real-time deadlock resolution models require detailed system status information and may be too complex to be implemented in the system level. A simple solution of the deadlock problem caused by the capacity of the system-buffer is the addition of new system-buffers. However, space limitation and cost may prevent new system-buffers. Now we propose a system and cell-buffer management scheme that uses random accessibility of cell I/O buffers to increase the maximum number of parts allowed in the system without causing deadlocks.

Conventionally, only the system-buffer receives parts that are requested to leave their current cells while their ultimate destinations have no free spaces. However, if the system-buffer has no free spaces either, the parts have to wait at their current locations and will consequently block other parts. For the system with a small system-buffer, one alternative is to send those parts temporarily to other cell-buffers that have large available spaces and are close to the ultimate destinations of the parts. The most difficult obstacle to implement this idea is that to be efficient, cell-buffers should be randomly-accessible and capable of loading and unloading parts to AGVs,

and the system controller should have the information on the status of the each buffer. Indeed, it is difficult to find a system that uses this idea in conventional job shop environments. Technically, the concept of FMSs with AGVs can accommodate this scenario because in general FMSs with AGVs and common I/O cell-buffers do provide the randomly-accessible cell-buffers, such as a carousel, and also provide the integrated computer system. However, there has been no accepted control model to implement this idea with FMSs in combination with AGVs. Here, a virtual system-buffer is defined as the portion of a cell-buffer that is temporarily shared with other cells.

Decision variables in managing virtual system-buffers are: (1) what portion of each cell-buffer should be part of a virtual system-buffer, and (2) how and when virtual system-buffers will be used. The size of a cell's virtual system-buffer should be a function of the capacity and nature of the cell as well the total capacity of system-buffers in the system. For example, the size of a cell's virtual system-buffer when the cell is a bottleneck cell should be smaller than that when the cell is a nonbottleneck cell. In general, for each cell, the larger the portion of its cell-buffer is dedicated to the virtual system-buffer, the larger the increase in the system-buffer capacity. However, the possibility that the cell may be inefficient due to the lack of the buffer capacity dedicated to the cell also increases. To prevent such inefficiency, at least two places of cell-buffers in each cell should be reserved for the parts visiting the cell, one for the incoming parts and one for the outgoing parts. For example, suppose that a cell is located next to the bottleneck cells and all of its cell-buffers are shared with other cells. Then, there is a high possibility that its

cell-buffers become full with parts that need to be processed by the bottleneck cells. In this case, even though machines in the cell are all free, no part visiting the cell can be delivered to or removed from the cell unless a part heading to a bottleneck cell leaves the buffer. Therefore, it is reasonable to reserve at least one place for the incoming parts and one place for the outgoing parts. Consequently, the portion of a virtual system-buffer in a cell should be two places less than the total capacity of cell-buffers in the cell.

The management schemes that give different priority to the virtual system-buffer over the system-buffer are worthy to be considered. For example, suppose that an objective of the system design is to reduce the required system-buffer capacity. One alternative is to give a higher priority to the virtual system-buffer over the system-buffer. By doing so, less parts are going to the system-buffer, and consequently a smaller capacity of the system-buffer is required. However, since the capacity of the system-buffer is already known at the operational level, there is no particular reason to use the virtual system-buffer over the system-buffer. For another example, since it is more likely that parts in the system-buffer do not block other parts, giving higher priority to the parts in virtual-buffers over the parts in the system-buffer is more reasonable in terms of selecting the next incoming part.

After a part arrives at a system-buffer, the part does not leave the system-buffer until it is pulled by its destination machining cell. However, after a part arrives at a virtual system-buffer, it can be pushed to another virtual system-buffer or to a system-buffer as well pulled by its next machining cell. For example, when a cell-buffer is almost full, its cell

controller may send a part in its virtual system-buffer to other virtual system-buffer to make room for outgoing parts as well for incoming parts. Consequently, a part might take several trips before it goes to its next immediate cell and AGV demands may increase.

#### 4. PERFORMANCE TEST ON A VIRTUAL SYSTEM-BUFFER STRATEGY

It is intractable to test the performance of virtual system-buffer strategies for all possible combinations of parameters. Instead, a virtual system-buffer management model is proposed for minimizing the required size of the system-buffer without compromising throughput. To investigate the behavior of this model, the model is tested through a computer simulation on a hypothetical system. The results are then compared with that of a conventional model in which each cell has an input buffer and an output buffer.

##### 4.1 Virtual System-buffer Model

AGV Dispatching strategies fall into two categories: (1) Reservation Dispatching(RD) and (2) None Reservation Dispatching( NRD). The RD strategy is to dispatch idle AGVs only the parts that have reserved their destination, whereas an idle AGV is dispatched to a part that may or may not have reserved its destination at the time of the dispatching under the NRD strategy. The proposed model is developed for a multi-cell FMS where each cell has a randomly-accessible cell I/O buffer and the RD strategy is used as AGV dispatching rule.

Let the cells be numbered in an arbitrary

order, starting with Cell  $l$ . Let  $I/O_i$  denote the cell I/O buffer of the cell  $i$ . In addition, a transient part is defined as a part that resides in an I/O buffer of a cell but is neither an incoming nor outgoing part of the cell.

There are three types of parts that arrive at  $I/O_i$ : incoming parts, and transient parts from other cells, and outgoing parts from machines in the cell. Similarly, three types of parts depart from  $I/O_i$  to other places: incoming parts to machines in the cell  $i$ , outgoing parts and transient parts to their immediate machining cells or other virtual system-buffers or a system-buffer. Three sequential milestones are fundamental to part movements between buffers. They are: (1) arrival milestone which occurs when a part arrives at a cell I/O buffer, (2) reservation milestone which occurs when a part reserves its next destination place, and (3) departure milestone which occurs when a part departs from a cell I/O buffer. Between these three milestones, a part may have to wait until certain conditions are met such as the availability of its destination place and/or an AGV. The proposed model has three modules detailing action/events surrounding each milestone: (a) reservation request for a place in a cell I/O buffer, (b) part arrival at a cell I/O buffer, and (c) departure of a part from a cell I/O buffer. Each module contains actions that should be taken at each corresponding milestone.

The first module is the reservation request for a place in a cell I/O buffer. This module is invoked whenever a part tries to reserve a place at a cell I/O buffer. The request for the reserving a place in  $I/O_i$  is always granted if the buffer has more than two free places. If  $I/O_i$  has only two free places, an incoming part from another cell can reserve one of the places in  $I/O_i$ . If  $I/O_i$  does not have more than two free

places and there is at least one incoming part in  $I/O_i$ , then an outgoing part that is waiting for it at some places in Cell  $i$  can reserve one of the free places in  $I/O_i$ . If  $I/O_i$  has only one free place and there is no incoming part in  $I/O_i$ , then no part is granted the request. . The purpose of this strategy is to reduce the blockage of incoming parts. For instance, if outgoing parts occupy the total capacity of a cell I/O buffer, incoming parts cannot arrive even though all machines are idle. By holding outgoing parts at machines, such instances can be prevented.

Under this model, at least two places in  $I/O_i$  must be used only for Cell  $i$ . One of them is always reserved for the outgoing parts from resources within Cell  $i$ , whereas the other place can be used by an incoming part from other cells. It should be noted that the two places are not physically assigned and the locations of two places are continuously changing in  $I/O_i$ . This model aggressively uses cell I/O buffers as virtual system-buffers by only assigning two places for each cell. We will discuss the possible alternatives in the discussion section.

After a part arrives at a cell I/O buffer, the second module, part arrival at a cell I/O buffer, is invoked. In this module, two actions are taken: (1) switching a blocked outgoing part with an incoming part as mentioned above and (2) sending a part to a virtual system-buffer or a system-buffer if the cell I/O buffer is congested. When the part that has just arrived at a cell I/O buffer finds that there are less than two free places in the buffer, it pushes out a transient part in the buffer. If the transient part is not available, an outgoing part is selected by the FIFO rule and sent to other buffer.

When a cell I/O buffer has less than two available places, machines may be blocked. However, as soon as an incoming part arrives at

a cell I/O buffer, it pulls an outgoing part at a machine to the cell I/O buffer. That case is the only instance in which I/O<sub>i</sub> may become full. The pulled part changes its destination to a system-buffer instead of its ultimate destination cell. Further, the pulled part always forces another part in I/O<sub>i</sub> to the system-buffer. Therefore, the sum of parts that have the system-buffer as their next destination and the number of free places in each cell I/O buffer, is equal to or greater than two. Considering the assumption of infinite system-buffer capacity, if an I/O buffer has less than two free places, it is guaranteed that at least two places in the buffer will be freed later and an incoming part will reserve one of the places.

The third module, departure of a part from a cell I/O buffer, is invoked right after a part leaves from an I/O buffer. In this module, three actions are performed: (1) pulling parts in virtual or system-buffers, (2) finding an appropriate buffer if the departing part is a transient part, and (3) checking blocked parts for an available place by a part departure. A transient part reserves a place in the system-buffer first at the reservation stage. As soon as the part is loaded onto an AGV, it checks its destination machining cell. If the cell I/O buffer can receive an incoming part (more than one free place), it reserves one of the places at its next machine cell I/O buffer and cancels the reservation at the system-buffer. Otherwise, it is looking for a virtual system-buffer that is closest to its ultimate destination and has more than two free places. If thus virtual system-buffer can be found, the part is sent there. However, if there is no such virtual system-buffer, the part is sent to the system-buffer.

After a part enters a virtual or system-buffer, it cannot leave the buffer unless it is

pushed by another part such as mentioned in the first module or pulled by its ultimate next destination cell. The part flow from the virtual and system-buffers to the ultimate destination of parts is performed in the third module. Parts in virtual or system-buffers are pulled by Cell<sub>i</sub> whenever there are no incoming parts waiting for loading to Cell<sub>i</sub> at I/O<sub>i</sub>. The number of parts that can be transferred to I/O<sub>i</sub> from virtual system-buffers and the system-buffer is equal to a half of the free places in I/O<sub>i</sub> at the time of request.

#### 4.2 Conventional Buffer Management

The term conventional approach for buffer management, as used here, applies to the approach that physically divides each cell I/O buffer into one input buffer and one output buffer. This is one of the most conservative usages of a randomly-accessible cell I/O buffer. However, to reduce movements between system-buffer and cell-buffers, more conservative methods are used for both the part flow directions between system and cell-buffers. The part flow from the system-buffer to the cell input buffer selected is similar to the virtual system-buffer model where parts in the system-buffer are pulled by Cell<sub>i</sub> whenever there is no part waiting for loading at the input buffer of Cell<sub>i</sub>. The number of parts that can be transferred to the input buffer of Cell<sub>i</sub> from the system-buffer is equal to a half of the places in the input buffer of Cell<sub>i</sub> at the time of request. Also, a part in a cell output buffer is pushed to the system-buffer when the cell output buffer becomes full.



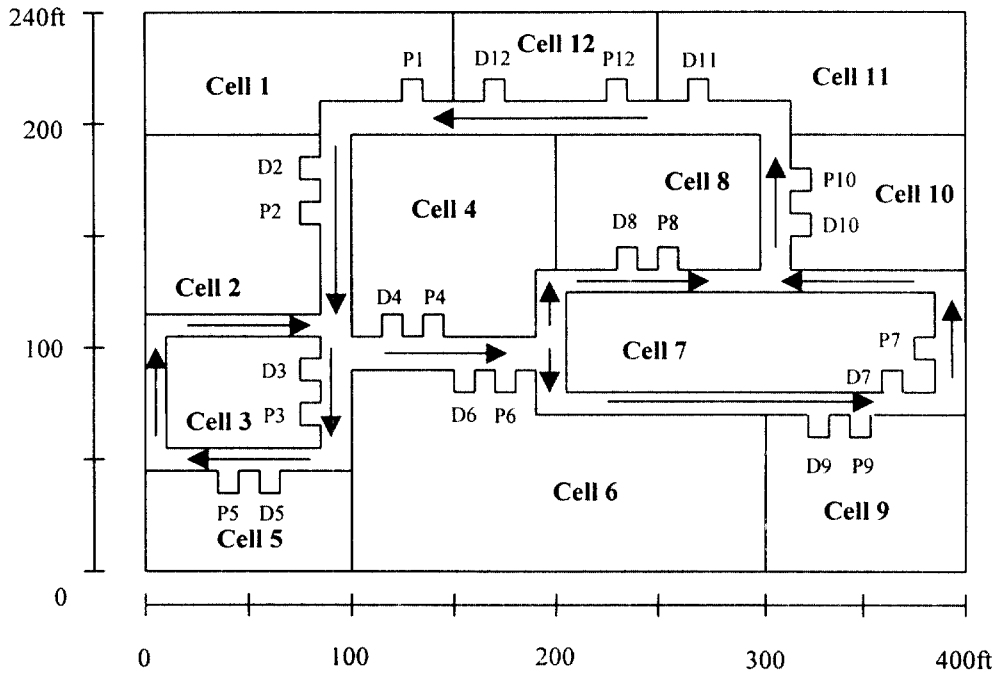


Figure 2. Job Shop Multi-Shop FMS and Guided Path Layout

Table 1. Capacities of Cells in the Job Shop FMS used for Simulation

Processing cell	1	2	3	4	5	6	7	8	9	10	11	12
Number of machines	5	3	1	7	1	5	4	3	2	2	$\infty$	-
Input buffer capacity	$\infty$	C	C	C	C	C	C	C	C	C	$\infty$	$\infty$
Output buffer capacity	C	C	C	C	C	C	C	C	C	C	-	$\infty$
I/O buffer capacity	$\infty$ / C*	2C	2C	2C	2C	2C	2C	2C	2C	2C	$\infty$ / -*	$\infty$

Where C=3, 4 or 5: Loading cell and unloading cell have segregated input output buffer allowed in the system(LWIP)

### 4.3 Experiment Design and Simulation System

A hypothetical job shop multi-cell FMS was selected for the experiment. The multi-cell FMS is depicted in Figure 2. The multi-cell FMS consists of 12 cells, including a loading department (cell 1), an unloading department (cell 11), and a system-buffer (cell 12). All jobs enter the system through the loading department and leave the system through the unloading department. A uni-directional AGV guided path network was designed for the system. Each processing cell consists of identical machines. Table 1 shows the capacities of each cell. The input buffers of the loading cell and the unloading cell are assumed to have infinite buffer capacities to allow unrestricted arrival and departure of parts.

Simulation was used to compare the performance of the virtual model against the conventional model for different experimental conditions. The models were tested under the following experimental conditions:

- \* different limits on the number of parts
- \* varying AGV fleet sizes;
- \* different buffer capacities

It is assumed that infinite raw parts at the input buffer of the loading cell are available. It is also assumed that the loading sequence of parts has already been decided. It may have been decided at planning level or according to the arrival time. Under this assumption, each part type completed is approximately proportional to each part type arrived so that it eliminates a possibility of using the alternative part selection loading policies. The same random number stream was used for all experiments to generate the loading sequence of

parts. Figure 3 represents the job mix used in this study. The figure includes five part families, their routings, and their mean processing times at each cell. All parts in a part family use the same routing sequence, but have different processing times. Processing time is randomly chosen from a uniform discrete function  $(0.5, 0.75, 1.0, 1.25, 1.5) \times (\text{mean processing time})$  at the cell using a different seed for each part family. The processing times include both the required setup time and the processing time at each processing cell.

Three limits on the number of parts allowed in the system (LWIP) are used. They are 44, 55, 66 that are  $4/3$ ,  $5/3$ , and  $6/3$  times of the number of machines in the system. A part is loaded into a loading machine in the loading cell if the number of parts in the system is less than the limit and there is an idle loading machine in the loading cell. The only other congestion control method used is that a part in the output buffer of the loading cell can only go to the next cell in the routing. Therefore, a part cannot go to the system-buffer before it is processed at least once in a machine other than in a loading machine.

The Shortest Processing Time (SPT) rule is selected as the machine scheduling rule. An idle machine selects a part in the cell input buffer that requires the shortest processing time in the machine (SPT). The Shortest Travel Distance (STD) rule is selected as the AGV dispatching rule for this study because the combination of SPT/STD is one of the best combinations of machine scheduling and AGV dispatching rules[14]. Also FCFS rule is used to select a part among several parts requesting AGVs at a buffer when an AGV arrives at a buffer to pick a part and find several parts waiting. The fleet size of AGVs studied is 7, 8, 9, 10, and 11

Cell	Part Family					Mean processing times at cell i per part:k	Normalized Work Load $\frac{K_i(\#of M/C_i) \cdot 100}{Max(K_i \#of M/C_i)}$
	1	2	3	4	5		
1	8.0	8.0	6.0	8.0	8.0	7.8	60.61%
2	6.0	6.0	4.5	6.0	6.0	5.85	75.39%
3				17.2		1.72	66.49%
4	22.8	9.2	7.6		26.4	17.34	95.77%
5				20.4		2.04	78.87%
6		12.4	23.4			6.06	46.86%
7	8.0	8.0	7.2	6.8		5.8	56.06%
8			14.0	26.0		7.76	100.00%
9	9.2				9.2	4.6	88.92%
10	4.0	4.0	3.0		4.0	3.5	67.65%
11	0.0	0.0	0.0	0.0	0.0		

Figure 3. Job Mix: Part Family Routings, Mean Processing Times and Probability Dist.

AGVs. The fleet size of 9 AGVs was obtained by the first analytical model in the paper by Egbelu[3] which assumes that the fraction of time an AGV travels empty is equal to the fraction of time it travels loaded. Therefore, the number of AGVs needed is equal to  $\{2 \cdot (\text{the required total time travel loaded/unit time}) + (\text{the number of load/unit time} \cdot \text{loading time} + \text{the number of unload/unit time} \cdot \text{unloading time})\}$ . The fleet sizes of AGVs selected cover 75% to 125% of the minimum required AGV fleet sizes.

Because of the dynamic behavior of the system and the simplicity, buffers are evenly distributed between cells and between input buffers and output buffers. For the conventional model, each cell has the same input buffer and output buffer sizes. Three different buffer sizes are studied; 3, 4, and 5 for all input buffers and output buffers in the system except the input buffer of the loading and unloading cells. Also, for the virtual model, corresponding buffer sizes for cell I/O buffers selected are 6, 8, and 10.

Consequently, the sum of an input buffer and an output buffer of each cell in the conventional model is identical to the capacity of a cell I/O buffer in the virtual model.

## 5. SIMULATION RESULTS

The simulation was developed using the SIMAN[12] general purpose simulation language and C programming language. Batch means method[8] was employed to obtain the steady state statistics for desired measure of performance. A pilot run was conducted to determine the batch size and the initial warm-up period. A warm-up run of 20 simulation hours and the size of 10 hours per batch were selected. A total of 220 simulation hours per each experiment was conducted to obtain 20 batches. The system was started with the empty and idle condition. It should be noted that the batch size of 10 simulation hours tends to be small to investigate a steady state behavior for a measure

of performance investigated for some extreme cases. However, because of the computation time required and the size of batch is sufficient for most other measures of performance and cases, a batch size of 10 hours was selected. Further discussions on this problem will be discussed in appropriate sections later.

### 5.1 Performance Measures

Four different measures were used to compare the performances of the virtual model against the conventional model for different experimental conditions. The measures of relative performance tested are: (1) Throughput, (2) Flow time, (3) AGV utilization and (4) Maximum number of parts in the system-buffer (MNPS) at a moment during entire simulation duration for each experiment. MNPS was selected to show the difference between models

on the required capacity of the system-buffer to have a deadlock free system. It should be noted that the observation value underestimates the actual required buffer size that guarantees the deadlock free system under the same system condition.

### 5.2 Relative Performance of the Buffer Management Models at Varying Buffer Capacities, AGV Fleet Sizes and LWIPs

The effect of different buffer capacities, AGV fleet sizes and LWIPs on the relative performance of the buffer management models was investigated in this study. The simulation results are summarized in Figure 4 through Figure 7.

Figure 4 illustrates the average throughputs of two models under different experimental

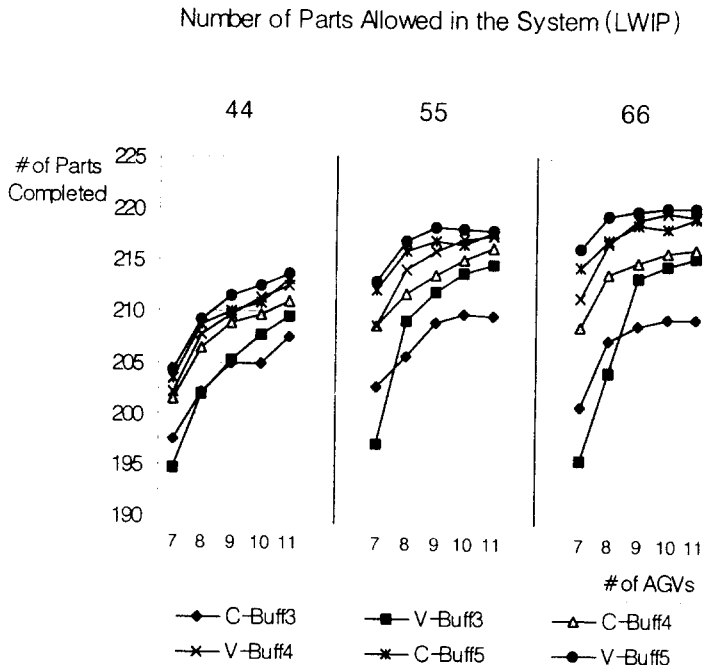


Figure 4. Throughput

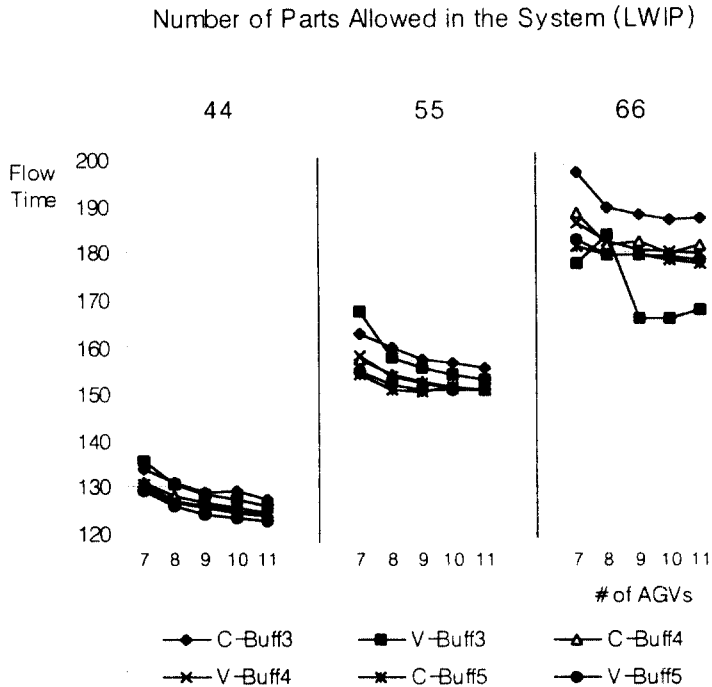


Figure 5. Average flow time

conditions. For each model, the throughputs monotonically increase as the number of AGVs and/or the buffer capacities increase over the entire spectrum of experiments. Also as the LWIPs increase, the throughputs increase except in the cases of small numbers of AGVs with small buffer capacities (buffer of size 3) for both models. This can be explained as follows: A relatively large number of parts in the system, as compared to the cell-buffer capacities, increases the possibility of blocking and the number of trips between the system-buffer and cell I/O buffers. Therefore, slow MHS (small AGV fleet size) cannot keep up with the demand, consequently causing the throughput to decrease. Between the two models, the virtual model provided better throughputs than the conventional model in all cases except in the

cases of small numbers of AGVs with a small buffer capacity (buffer of size 3). Also, as the LWIPs increase, the improvement of the virtual model over the conventional model also increases.

Figure 5 summarizes the simulation results of average flow times under the same experimental conditions. Figure 5 is almost the mirror image of Figure 4 (throughput). This phenomenon can be explained by well known Little's Law[10] which equates WIP levels to the throughput rate and flow time for steady-state systems. The results revealed that the WIP levels for all experiments are very close to the LWIP. In other words, when a part leaves the system, there is usually at least one idle loading machine so a new part can be loaded into the system immediately. Therefore, the increase in

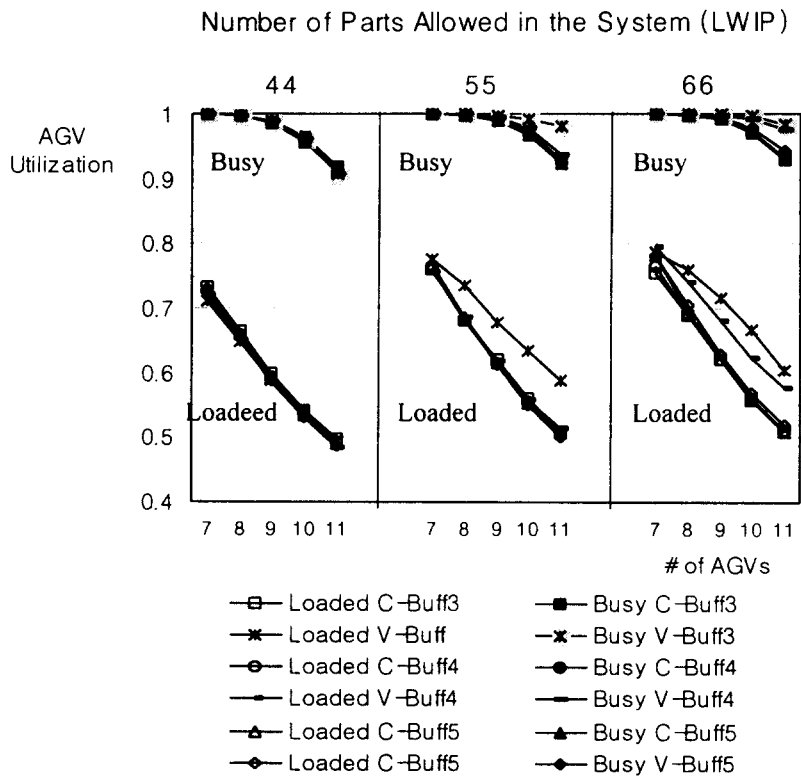


Figure 6. AGV utilization

throughput follows the decrease in flow time.

Also, although there is a monotonic decrease in the average flow time as the AGV fleet size and/or buffer capacities increase, the differences between flow time on the varying AGV fleet sizes and buffer capacities are insignificant compared to that of increasing LWIPs, except in the case of buffer size 3. It should be noted that the average flow times with a buffer size of 3 are underestimates of the average steady state flow time. The reasoning can be explained as following. Since there is a possibility that a part in the system-buffer may stay at the system-buffer for an extremely long period of time, perhaps staying there indefinitely, in order to calculate the steady state expectation of flow

time, a large warm-up and a large batch size are required. Furthermore, since the parts in the system-buffer have lower priority than the parts in the virtual system-buffers in the virtual model, the possibility of such cases is higher under the virtual model than that under the conventional model. In the case of buffer size 3 under the virtual model, it is observed that a part that entered the system-buffer at warm-up period still stayed there at the end of simulation. This is the only case in which the batch size and the warm-up run selected are not sufficient.

Figure 6 illustrates the average utilization of AGVs under different experimental conditions. When the LWIP was 44, there was no significant difference between the two buffer

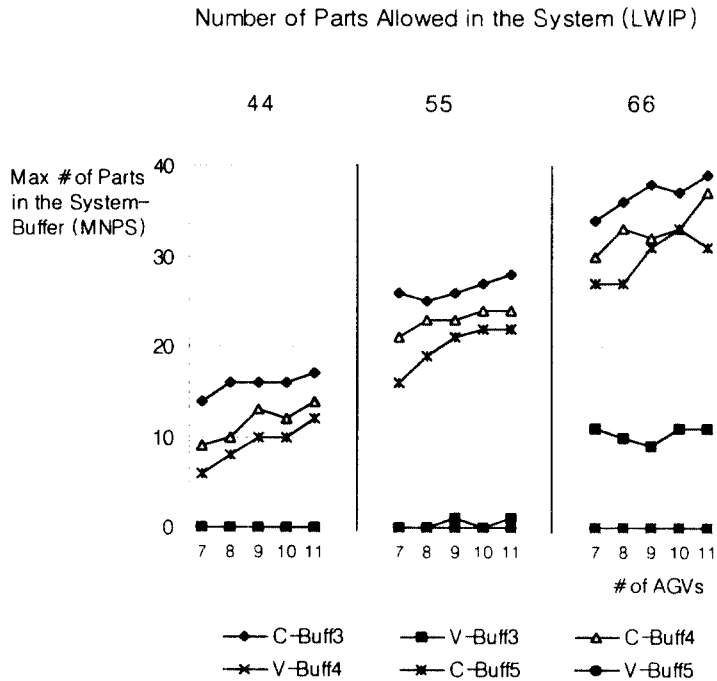


Figure 7. Maximum number of parts (MNPS) in the system-buffer

management models. However, when the LWIP was 55 or 66, the utilizations of AGVs under the virtual model tended to be higher than that under the conventional model, especially for small buffer sizes. This was somewhat expected. When buffer sizes are small relative to the LWIP selected, the cell I/O buffers are congested. Consequently, the possibility that a part in a virtual system-buffer has to be removed to another virtual system-buffer, in order to provide room for the parts that will be processed in the cell, is high. And it requires high AGV demand under the virtual model. However, because of conservative usage in addition to the infinite capacity of the system-buffer, the parts in the system-buffer do not push or block other parts. Therefore, the AGV utilizations under different LWIPs and

buffer sizes are close to each other under the conventional model.

Figure 7 clearly illustrates that the virtual model reduces the required capacity of the system-buffer. When the buffer size was 4 or 5, there were no parts that visited the system-buffer under the virtual model where as much as 38 parts at a moment were in the system-buffer under the conventional model.

## 6. CONCLUSION AND RECOMMENDATION

A buffer management model (the virtual model) for random job shop FMSs with AGVs was presented in this paper. The virtual model uses some portion of each local buffer as a system-buffer.

The simulation experiments conducted in this research clearly indicate that having a proper buffer management, suited for the system, could result in higher throughputs. The proposed buffer management model is superior over the conventional buffer management model in terms of throughputs and the required capacity of the system-buffer. In addition to the superiority of the virtual model over the conventional model, the following observations are obtained for general behaviors of multi-cell FMSs. With the increased LWIPs, throughputs, flow times, and MNPSs also increase. With the increased buffer capacity and/or AGV fleet sizes, in most cases throughputs and MNPSs increase while flow times and AGV utilizations decrease.

With more specific knowledge of the system environment at hand, the virtual model can be fine tuned accordingly. Some considerable alternatives that might improve the virtual model are: (1) Increasing the portion of each cell I/O buffer that can be used only by the parts for the cell; (2) Giving the same priority to the parts in the system-buffer with the parts in the virtual system-buffer; (3) Giving the same priority to the system-buffer with cell I/O buffers when a part requests a virtual system-buffer. Since the capacity of the system-buffer is already known at the operation level, there is no particular reason to use the system-buffer over virtual system-buffers; Finally, (4) since cell-buffers do function as system buffers, system-buffers are no longer needed unless it is cost effective. Instead, the capacity of cell-buffers may be increased.

## REFERENCES

- [1] Buzacott, J.A. and Shankthikumar, J.G., "Models for Understanding Flexible Manufacturing Systems," *AIIE Transactions*, Vol.12, No.4(1980), pp. 339-350.
- [2] D'souza, K.A. and Suresh K.K., "A Survey of Petri Net Applications in Modeling Controls for Automated Manufacturing Systems," *Computers in Industry*, Vol.24(1994), pp. 5-16.
- [3] Egbelu, P.J., "The Use of Non-simulation Approaches in Estimating Vehicle Requirements in an Automated Guided Vehicle Based Transport System," *Material Flow*, Vol.4 (1987), pp. 17-32.
- [4] Egbelu, P.J. and Tanchoco, J.M.A., "Characterization of Automatic Guided Vehicle Dispatching Rules," *Int. J. Prod Res.*, Vol.22, No.3(1984), pp. 359-374.
- [5] Garetti, M., Pozzetti, A. and Bareggi, A., "On-line Loading and Dispatching in Flexible Manufacturing Systems," *Int. J. Prod. Res.*, Vol.28, No.7(1990), pp.1271-1292.
- [6] Kamoun, F. and Kleinrock, L., "Analysis of Shared Finite Storage in a Computer Network Node Environment under General Traffic Conditions," *IEEE Transactions in Communications*, COM-28(1980), pp. 992-1003.
- [7] Kundu, S. and Akyildiz, I.F., "Deadlock Free Buffer Allocation in Closed Queueing Networks," *Queueing Systems*, Vol.4(1989), pp. 47-56.
- [8] Law, A.M. and Kelton, W.D., "Simulation Modeling and Analysis," McGrawhill, Second Edition, New York, 1991.
- [9] Leung, Y.T. and Sheen, G.J., "Resolving Deadlocks in Flexible Manufacturing Cells," *Journal of Manufacturing Systems*, Vol.12, No.4(1993), pp. 291-304.



- [10] Little, J.D.C., "A Proof for the Queueing Formula:  $L = w$ ," *Operations Research*, Vol.9, No.3(1961), pp. 383-387.
- [11] MacCarthy, B. L., and Jiyin Liu, "A New Classification Scheme for Flexible Manufacturing Systems," *Int. J. Prod. res.*, Vol. 31, No.2(1993), pp. 299-304
- [12] Pegden, C. D., Shannon, R.E. and Sadowski, R.P., "Introduction to Simulation Using SIMAN," MacGraw-Hill, New-York, 1990.
- [13] Sabuncuoclu, I. and Hommertzheim, D.L., "An Investigation of Machine and AGV Scheduling Rules in an FMS," in: K.E. Stecke and R. Suri (eds.), *Flexible Manufacturing Systems: Operations Research Models and Applications*, (Proc. 3rd ORSA/TIMS Conf.), Elsevier, Amsterdam, (1989), pp. 261-266.
- [14] Sabuncuoclu, I. and Hommertzheim, D.L., "Experimental Investigation of FMS Machine and AGV Scheduling Rules against the Mean Flow-time Criterion," *Int..J. Prod. Res.*, Vol.30, No.7(1992), pp. 1617-1635.
- [15] Stecke, K.E., "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems," *Management Science*, Vol.29, No.3 (1993), pp. 273-288.
- [16] Solot, P. and Vliet, M.V., "Analytical Models for FMS Design Optimization: A Survey," *The International Journal of Flexible Manufacturing Systems*, Vol.6(1994), pp. 209-233
- [17] Tang, L.L., Yih, Y. and Liu, C.Y., "A Study on Decision Rules of a Scheduling Model in an FMS," *Computers in Industry*, Vol.22 (1993), pp. 1-13.
- [18] Tansel, B.C. and Kiran, A.S., "Optimum Central Storage Location in Flexible Manufacturing Cells," *Journal of Manufacturing Systems*, Vol.7, No.2(1988), pp. 121-129.
- [19] Wysk, R.A., Yang, N.S. and Joshi, S., "Detection of Deadlocks in Flexible Manufacturing Cells," *IEEE Transactions on Robotics and Automation*, Vol.7, No.6 (1991), pp. 853-859.
- [20] Yamashita, H., "Approximate Analysis of Buffer Sharing Scheme in Arbitrary Configurations of Queueing Networks," *IIE Transactions*, Vol.27(1995) pp. 332-338.

## ● 저자소개 ●



## 김경섭

- 1982년 연세대학교 기계공학과 졸업 (학사)  
 1986년 University of Nebraska-Lincoln 산업공학과 졸업 (석사)  
 1993년 North Carolina State University 산업공학과 졸업 (박사)  
 1994년 삼성데이터시스템 선임연구원  
 1994년-현재 연세대학교 기계전자공학부 조교수  
 관심 분야: 물류시스템, 시뮬레이션 모델링 및 분석



## 이종하

- 1983년 고려대학교 산업공학과 졸업 (학사)  
 1987년 University of Nebraska-Lincoln 산업공학과 졸업 (석사)  
 1995년 University of Nebraska-Lincoln 산업공학과 졸업 (박사)  
 1996년-현재 유한대학 경영정보과 전임강사  
 관심 분야: FMS, 시뮬레이션, ERP