THE STUDY OF OPTIMAL BUFFER ALLOCATION IN FMS USING GENETIC ALGORITHM AND SIMULATION

Youngkyun Lee
Kyungsup Kim
Joonho Park

Dept. of Computer Science & Industrial System Engineering
Yonsei University
134, Shinchon-dong, Sudaemoon-ku, Seoul, 120-749, Korea

ABSTRACT

In this paper, we present a new heuristic algorithm for buffer allocation in FMS (Flexible Manufacturing System). It is conducted by using a genetic algorithm and simulation. First, we model the system by using a simulation software, "Arena". Then, we apply a genetic algorithm to achieve an optimal solution. VBA blocks, which are kinds of add-in functions in Arena, are used to connect Arena with the genetic algorithm. The system being modeled has seven workstations, one loading/unloading station, and three AGVs (Automated Guided Vehicle). Also it contains three products, which each have their own machining order and processing times. We experimented with two kinds of buffer allocation problems with a proposed heuristic algorithm, and we will suggest a simple heuristic approach based on processing times and workloads to validate our proposed algorithm. The first experiment is to find a buffer profile to achieve the maximum throughput using a finite number of buffers. The second experiment is to find the minimum number of buffers to achieve the desired throughput. End of this paper, we compare the result of a proposed algorithm with the result of a simple buffer allocation heuristic based on processing times and workloads. We show that the proposed algorithm increase the throughput by 7.2%.

1 INTRODUCTION

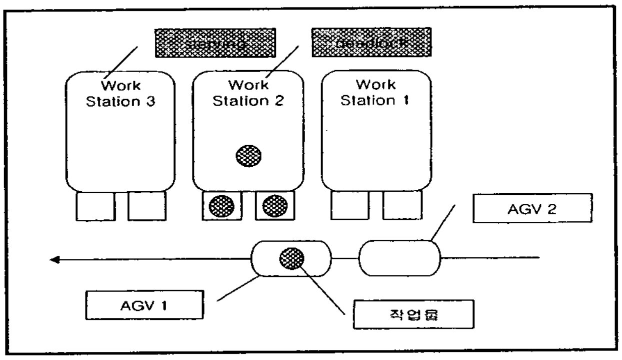
In modern industry, lot sizes have been reduced in production. Also, logistic time has been an important part of a whole processing time. Therefore, flexible manufacturing system (FMS) was proposed to decrease the logistic time in manufacturing system and to correspond to these environmental movements. The FMS consists of automated storage/retrieval system (AS/RS), automated guided vehicle (AGV), and numerical control machine (NC Machine). Until now, most of researches have been focused on JobShop scheduling, and AVG dispatching rule in the area of FMS. But proper use of buffers also can improve utilization of workstations and a throughput of a system, especially when the system contains bottleneck stations. However, considering cost and spaces buffers are finite resources. So it is important to allocate buffers properly for better performance of the system with a finite budget. Also, considering the desired throughput is fixed in designing a system, it is worthwhile to find the minimum total buffer capacity satisfying the desired throughput.

However, there has been little published research on buffer allocation in FMS because of its complexity, although it is of much importance. Most researches have been focusing on the simple production line system. In this study, we propose a new buffer allocation heuristic in FMS. We modeled the system by using a simulation software, Arena. Then, we apply a genetic algorithm to achieve an optimal solution. VBA blocks, which are kinds of add-in functions in Arena, are used to connect Arena with a genetic algorithm. The following steps conduct our research First, we generate a number of buffer profiles as a population in genetic algorithm. Next, we figure out the throughputs of each buffer profile by running simulation with a proposed model. Lastly, we apply a genetic algorithm with the throughputs obtained by simulation results of each buffer profile to get a better solution. We repeat step 2 and step 3. End of this paper, we compared a result of a proposed algorithm with a result of a simple buffer allocation heuristic based on processing times and workloads.

2 BUFFER CONCEPT AND LITERATURE REVIEW

In a manufacturing system, utilization of workstations may be decreased by machine failures, variable processing times, infeasible job assignments etc. However, these effects can be removed by proper use of buffers. But insufficient use of buffers will cause blocking, deadlock, and starving in FMS. These are shown in <Figure1>.

A blocking is a situation where an AGV waits to unload a job because of lack of input buffer spaces until the workstation will finish the precedence job. The AGV 1 is blocked in <Figure 1>. Suppose, if the input/output buffer of the workstation 1 are full of jobs, and the workstation 1 also has a job in it. Then the AGV 1 will be blocked until there is room in the output buffer space of workstation 1 but the AGV 2, which must do it, can't do it because the precedence AGV 1 is still blocked in <Figure 1>. This situation is called a deadlock. A deadlock is the worst case in a manufacturing system. Once a deadlock occurs, the system will be down. A starving is a situation where a workstation is idle because some related workstations are situated in the deadlock or the blocking. The workstation 3 is starving in <Figure 1>.



<Figure 1> Blocking, Deadlock, and Starving

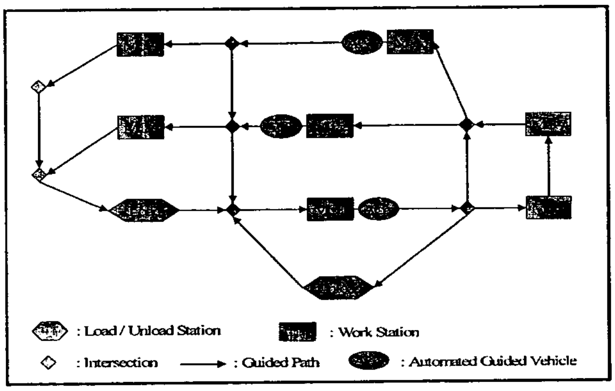
The studies about the buffer can be divided into three categories, by analytical approach. The three categories are: (1) mathematical approach, they have used mathematical approaches such as dynamic programming, markov chain and etc. (2) heuristic approach, they have used tabu search, and beam search. (3) analytical approach, they have analyzed the specific properties of an optimal buffer allocation.

In the area of a mathematical approach, most of researches have been focused on the simple production line system because of its own complexity in FMS. Jafri and Shanthikumar tried to find an optimal buffer allocation by using dynamic programming. They considered the production line as a stage with a machine, and located buffers between the stages. They suggested a simple heuristic to figure out a throughput of the system, and they compared their algorithm with a throughput of a greedy search. Also, Yamashita and Altok used a dynamic programming. In the area of a heuristic approach, Lutz used a tabu search and simulation. Park tried to find an optimal buffer allocation by using a dimension reduction and a beam search. Lastly, there have been analytical research, in which the inverted bowl has been found and the specific properties of an optimal buffer allocation have been analyzed. But these researches have important weakness, not being focused on FMS, but on the simple production line system, although it is of much importance.

3 ALGORITHM

3.1 System Configuration

The system being modeled is shown in <figure 2>. It has seven workstations, one loading/unloading station, and three AGVs. The AGVs are unidirectional. In this research, we make interarrival times to be very short and the capacity of the loading/unloading station to be infinite. Therefore, there will be no idle workstations due to lack of incoming jobs. So we can think a throughput of the system is effected by only the factors are in the system, such as AGV dispatching rule, buffer allocation, and Jobshop scheduling.



<Figure 2> The System

Until now, most research has dealt with single products. But it is necessary to deal with various products. In this paper, we experimented with four products. The processing orders and interarrival times of each product are shown in <Table 1>.

Product Type	Interarrival Time	Processing Order
A	EXPO(600)	1-2-1-2-3-4
B	EXPO(400)	4-5-6-7-5-7
C	EXPO(500)	1-2-4-2-4-6
D	EXPO(600)	4-5-3-6-7-5

<Table 1>

3.2 Problem Definition

We experimented with two kinds of buffer allocation problems with a proposed heuristic algorithm and will suggest a simple heuristic based on processing times and workloads to validate our proposed algorithm. When considering cost and space buffers are finite resources. So it is possible that the budget used for buffers could be fixed in the manufacturing system. Therefore, it is important to allocate buffers properly for the better performance of the system with the finite budget. Also, considering the desired throughput is fixed in designing a system, it is worthwhile to find the minimum total buffer capacity satisfying the desired throughput.

Our first experiment is to find a buffer allocation, which enables the system to achieve the maximum throughput. A throughput of the system is determined by many factors, such as processing times, AGV dispatching rules, buffer allocations and etc. If everything is fixed except buffer allocation then a throughput of the system is determined by only the buffer allocation, a number of input/output buffers. We express the input/output buffer strings as a vector, B, and the throughput as $T(B)$. The objective function and constraints are following.

$$Max\ T(B)$$

$$\begin{aligned} s.t\ \ & \sum_{i=1}^{N} b_i \le Total\ Buffer\ Capacity \\ & b_i \le Each\ Buffer\ Capacity\ \ i = 1,2,\dots,N \\ & B = (b_1, b_2, \dots b_N) \\ & N = Total\ Buffer\ Location \end{aligned}$$

The first constraint represents that the number of buffers used in the system that can't exceed the given total number of buffers. The second constraint means that each of input/output buffer has its own limited capacity.

Also, we conduct another experiment. We experiment to figure out the minimum total buffer capacities satisfying the desired throughput. Considering a desired throughput is fixed in designing a system, this experiment is meaningful. In this experiment, the objective function could be minimum number of buffers, which are used for the system. And another constraint is added to satisfy the desired throughput. Other things are unchanged when comparing previous experiment. The objective function and constraints are following.

$$Max \sum_{i=1}^{N} b_i$$

$$\begin{aligned} s.t\ \ & T(B) \ge Desired\ Throughput \\ & b_i \le Each\ Buffer\ Capacity\ \ i = 1,2,\dots,N \\ & B = (b_1, b_2, \dots b_N) \\ & N = Total\ Buffer\ Location \end{aligned}$$

Above problems have combinatorial properties. So it is hard to find an optimal buffer allocation, especially when the system is FMS. We can't reach an optimal solution with known algorithms so far. Therefore, we propose a new heuristic algorithm for buffer allocation in FMS. It is conducted by using a genetic algorithm and simulation.

3.3 Algorithm Steps

In genetic algorithms, one chromosome represents a string of input/output buffers. The system being modeled has seven workstations and each workstation has its own input/output buffer space. So we need 14 input/output buffer cells to allocate at each space. Therefore, one chromosome consists of 14 cells. For example, a chromosome, (0,1,2,1,1,0,0,2,2,1,0,0,2,1), means that there are zero input buffers and one output buffer in the first workstation. The following steps were used to conduct our research.

Step 1. Construct an Initial Population

In genetic algorithms, it's necessary to construct an initial population of n solutions. We make 10 solutions as an initial population. Some papers about the genetic algorithm have said that using a special algorithm to construct an initial population may make the algorithm go into the local optimal. Also, it takes much time to apply a special algorithm to construct an initial population. So we used a random generation method with some constraint to avoid getting infeasible solutions.

Step 2. Running Simulation

VBA blocks, which are kinds of add-in functions in Arena, allocate buffers at each input/output buffer space according to the given chromosomes. And then, we ran simulation with a given buffer allocation for all chromosomes in the population. This was repeated for each chromosome by ten times. End of replication, we take the mean throughputs of each chromosome.

Step 3. Reproduce the population to its original size by selections n solutions from the current population

There are so many selection methods, such as roulette wheel selection, ranking selection, and tournament selection In this study, the throughput used for selection is obtained by simulating during a day. So the difference of among throughputs is little distinguishable. Therefor, the tournament selection is more suitable to this research. The tournament selection procedure used in this research is following. First, we selected five chromosomes. Next, we chose the best one, which has the maximum throughput, among the five and made it a member of the next population. We repeated these steps, until we got the original sized population

Also, we conducted a mutation procedure. The mutation procedure made the solution escape from the local optimal.

	WS 1	WS 2	WS 3	WS 4	WS 5	WS 6	WS 7
Including Number	3	4	2	5	4	3	3
Processing Time	(133,173)	(152,181)	(165,192)	(126,160)	(82,100)	(106,133)	(130,150)
Index	5	6	7	3	1	2	4

<Table 2> Workstation Index

	Simple Heuristic		Proposed Heuristic		
Number	Buffer Profile	Throughput	Buffer Profile	Throughput	%
12	1,1,1,1,0,0,1,1,1,1,1,1,1,1	290	1,1,2,1,0,1,1,1,0,1,0,1,1,1	404	39.3
14	1,1,1,1,1,1,1,1,1,1,1,1,1,1	410	3,1,1,1,0,1,2,1,0,1,0,1,1,1	424	3.41
16	1,1,1,1,1,1,1,1,2,2,1,1,1,1	416	2,1,2,1,1,1,2,1,0,1,0,1,2,1	437	5.04
20	1,1,1,1,1,1,2,2,2,2,2,2,1,1	427	2,1,2,1,1,1,3,1,1,2,1,1,2,1	454	6.32
25	2,2,2,1,1,1,2,2,2,2,2,2,2,2	445	3,1,3,1,1,1,5,2,1,1,1,1,3,1	470	5.61
30	2,2,2,2,2,2,2,2,3,3,2,2,2,2	460	5,1,3,1,1,1,5,2,1,1,1,3,4,1	475	3.26

<Table 3> The Result

Step 4. Crossover and Mutation

The next step is a crossover and a mutation procedure. The next generation will have got good inherited characters by the crossover procedure. We applied a one-point crossover. The one-point crossover procedure follows. First, we generated a random number between 0 and 1 for each chromosome. If the number is less than the probability of a crossover, this buffer profile will be a candidate for the crossover. When we conduct the crossover procedure, the number of chromosomes must be an even number. So, if we get an odd number, we choose a chromosome randomly among the abandoned buffer chromosomes. Next, we conduct the crossover procedure. The example of one-point crossover procedure is following. Suppose we choose two chromosomes, buffer profiles, among the candidates. And a division point is decided between 7and 8 randomly. Then we will get two chromosomes, new buffer profiles, after conducting the crossover procedure.

Buffer Profile 1: (0,0,1,1,2,2,0 | 0,1,0,1,2,3)
Buffer Profile 2: (1,1,0,0,1,1,1 | 1,2,2,0,0,0)
New Buffer Profile 1: (0,0,1,1,2,2,0 | 1,2,2,0,0,0)
New Buffer Profile 2: (1,1,0,0,1,1,1 | 0,1,0,1,2,3)

If the probability of a mutation is high, the search may be a random search. On the other hand, if the probability is low, the search may go into the local optimal and will not be able to escape from the local optimal. In this study, the probability will be lower when the best solution is renewed, and the probability will be higher when there is no change in the best solution during the given number of iterations.

We generated three random numbers. The first one between 0 and 1 was used for comparing with the probability of a mutation. The second one also between 0 and 1 is used for determining whether the buffer will be added or will be subtracted. The last one between 1 and 3 is the number of buffers will be added or be subtracted.

Also, we added one special trick to get an optimal solution more fast. We make the algorithm add the buffers if utilization of the buffer is high, and subtract the buffers if utilization of the buffer is low.

4 DISCUSSION AND RESULTS

Our first experiment is to find the maximal throughput. In this study, the total buffer capacity is increased from 12 to 30. And the limit of input/output buffer spaces is fixed on 5. There have been no algorithms for buffer allocation in

FMS. So we propose a simple buffer allocation heuristic based on processing times and workloads to compare our proposed algorithm. The idea is that the workstations, which have many workloads and long processing time, will have high probability of being bottleneck stations. So these workstations need more buffers. We made the indexes by considering workloads and processing times. The indexes are shown <Table 2>. The buffers are allocated by the indexes. Higher index made the workstation have more buffers.

The result of first experiment is shown and is compared with a simple heuristic algorithm in <Table 3>. The throughput is increased by 10.5% when comparing with a simple heuristic algorithm. There is a special aspect in the result. When total buffer capacity is 12, the throughput increased by 39.3%. This is agreeable when considering pervious researches about buffers. Those researches said when total capacity of buffers is small, the system is affected more sensitively by adding or subtracting the buffers. In the result, it is proved that the buffer allocation can improve the throughput of the system with same total number of buffers, and it can be said that the proposed algorithm is agreeable when allocating buffers for improvement of the throughput in FMS.

In our second experiment, we fix the desired throughput to 360. And we found 8 buffers are enough to achieve the desired throughput. And the buffer profile is (0,1,1,0,0,1,0,1,0,1,0,1,1,1)

5 CONCLUSION

In this study, we figured out an optimal buffer allocation by using simulation and a genetic algorithm. Until now, it has been thought that finding an optimal buffer allocation is impossible in FMS. In this paper, we propose the method to find an optimal buffer allocation in FMS by just using simulation software, Arena, and a genetic algorithm.

Possible extension of our algorithm is to consider the buffer allocation with other factors, such as AGV dispatching rule, and the number of AGVs in the system. When the AGV dispatching rule or the number of AGVs is changed, the buffer allocation may be changed. Also, it is important to find an optimal buffer allocation with multi load AGV system.

REFERENCES

[1] H. Pierreval and L. Tautou (1997), Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions*, Vol 29, 181-189.

[2] Hideaki Yamashita and Tayfur Altiok (1998), Buffer capacity allocation for a desired throughput in production lines. *IIE Transactions*, Vol 30, 883-891.

[3] Mohsen A.Jafari and J. George Shanthikumar (1989), Determination of Optimal Buffer Storage Capacities and Optimal Allocation in Multistage Automatic Transfer Lines. *IIE Transactions*, Vol 21, 130-135.

[4] Christian M. Lutz, K.Roscoe Davis, Minghe Sun (1998), Determining buffer location and size in production lines using tabu search. *European Journal of Operational Research*, 301-316.

[5] G. A. Vorous and H. T. Papadopoulos (1998), Buffer Allocation In Unreliable Production Lines Using A Knowledge Based System. *Computers Ops Res*, Vol 25, 1055-1067.

[6] T. Park (1993), A two-phase heuristic algorithm for determining buffer sizes of production lines. *International Journal Of Production Research*, Vol 31, 613-631.

[7] Mark S. Hillier (2000), Characterizing the optimal allocation of storage space in production line systems with variable processing times. *IIE Transactions*, Vol 32, 1-8.

[8] Frederick S. Hillier and Kut C. So (1991), The Effect of the Coefficient of Variation of Operation Times on the Allocation of Storage Space in Production Line Systems. *IIE Transactions*, Vol 23, 198-206.

[9] 김여근, 윤복식, 이상복 (1997), 메타휴리스틱, *영지문화사*.

[10] KUT C. So (1997), Optimal buffer allocation strategy for minimizing work in process inventory in unpaced production lines. *IIE Transactions*. Vol 29. 81-88.

[11] Neil B, Marks (1987), Output analysis of an automated serial assembly line. *International Journal of Production Research*. Vol 25, No 8. 1197-1207.

[12] D.Seong, S.Y. Chang and Y.Hong (1995), Heuristic algorithms for buffer allocation in a production line with unreliable machines. *International Journal of Production Research*. Vol 33, No 7. 1989-2005.

AUTHOR BIOGRAPHIES

YOUNGKYUN LEE is a programmer in POSDATA. He received his BS in Industrial Engineering from the Hansung University, Seoul, South Korea, in 1999, and MS in Industrial System Engineering from the Yonsei University, Seoul, South Korea, in 2001. His research interests are in material handling systems such as AGV systems, and buffer allocation in FMS. His email address is <baram@ las.yonsei.ac.kr>.

KYUNGSUP LEE is an Associate Professor of Computer Science and Industrial System Engineering at Yonsei University in South Korea. He received his BS in Mechanical Engineering from the Yonsei University, Seoul, South Korea, in 1982, and MS in Industrial and Management Systems Engineering from the University of Nebraska-Lincoln in 1986, and Ph.D. in Industrial Engineering from North Carolina State University in 1993. His research interests are in material handling systems such as AGV systems. His email address is <kyungkim@yonsei.ac.kr>

JOONHO PARK is in the course of MS. He received his BS in Industrial Engineering from the Inha University, Inchon, South Korea, in 2000. His research interests are in supply chain management and discrete event simulation. His email address is <ie606@las.yonsei.ac.kr>