'90 KACC 1990. 10. 26~27

ON OPTIMAL CYCLIC SCHEDULING FOR A FLEXIBLE MANUFACTURING CELL

Hiroshi Kise, Shinji Nakamura and Yoshiyuki Karuno

Department of Mechanical and System Engineering
Kyoto Institute of Technology, Kyoto, Japan

ABSTRACT

This paper discusses an optimal cyclic scheduling problem for a FMC (Flexible Manufacturing Cell) modeled by a two-machine flowshop with two machining centers with APC's (Automated Pallet Changers), an AGV (Automated Guided Vehicle) and loading and unloading stations. Cyclic production in which similar patterns of production is repeated can significantly reduce the production lead-time and WIP (Work-In-Process) in such flexible, automated system. Thus we want to find an optimal cyclic schedule that minimizes the cycle time in each cycle. However, the existence of APC's as buffer storage for WIP makes the problem intractable (i.e., NP-complete). We propose an practical approximation algorithm that minimizes, instead of each cycle time, its upper bound. Performances of this algorithm are validated by the way of computer simulations.

1. INTRODUCTION

This paper considers a FMC (Flexible Manufacturing Cell) that consists of two machines such as machining centers with ATC (Automated Tool Changer), an AGV (Automated Guided Vehicle) and loading and unloading stations, all of which are controlled by computers. Each machine has an APC (Automated Pallet Changer) by which two jobs from and to that machine can simultaneously exchanged. The AGV can carry at most one job at a time. Loading and unloading stations that may constitute a part of an automated warehouse have enough capacity of storage for unfinished and finished jobs, respectively. Each job is picked up at the loading station, processed on two machines in the same order, and deposited in the unloading station.

Wassenhove et al [8] have surveyed over half the FMS's operating worldwide in 1983, and reported that there is a definite trend towards more integrated independent cells like the system discussed here. (Also see Yamazaki and Nagae [9] which shows a FMC quite similar to our model.) We consider our FMC as a part of a flexible production system that consists of manufacturing and assembling shops. The manufacturing shop consisting of disconnected FMC's are advantageous over large, complex FMS's, since the former can more easily be controlled and maintained from both hardware and software points of view. Thus even if many processing stages are required, the system can advantageously be disaggregated into disconnected FMC's (by subcontracting bottleneck processing, as discussed by Ravikumar and Vannelli [7]).

By cyclic production we mean here that similar patterns of production are repeated. Each cycle may have a different set of parts to be assembled into a product. This definition is more flexible than the conventional cyclic production in which the exact same pattern of production is repeated (e.g. see Graves et al [2] and Matsuo [5]). The automation of material handling as assumed here can drastically reduce set-up times for switching jobs. These reductions allow the system to process a variety of jobs without reducing the efficiency seriously. Then the cyclic production can significantly reduce the production lead-time and WIP(Work-In-process) in the entire production system as compared with commonly used batch production, if manufacturing and assembling are synchronized or adopt the JIT (Just-In-Time) system.

This paper discusses a scheduling problem of minimizing cycle time of each cycle in the above FMC. Kise et al [4] have shown that the problem can be solved in a polynomial time, if no WIP (i.e., no APC) is allowed in the FMC. However, the existence of APC's as buffer for WIP makes the problem intractable (i.e., NP-complete), even if transportation times of the AGV are neglected, and there is only one cycle (i.e., a classical makespan problem for a two-machine flowshop with finite buffer,[6]). Thus practical approximation algorithms should be developed. We propose an approximation algorithm which is based on the Gilmore and Gomory's algorithm [1] for a special traveling salesman problem, and minimizes, instead of each cycle time, its upper bound. Performances of this algorithm are validated by the way of computer simulations.

2. Description of the System

As schematically shown in Fig. 1, the system discussed here consists of two machines M_a and M_b, an AGV and a loading station S_l and an unloading stations S_u. Each machine have a single unit of APC by which a job to be transferred to the machine can be exchanged for a job awaiting transportation from that machine. It processes at most one job at a time. No pre-emption is allowed. The AGV sends at most one job at a time. Loading and unloading stations, S_l and S_u, have enough capacity of buffer storage for unfinished and finished jobs, respectively. Each job is processed on each machine exactly once in the same order. There are m cycles of the production to be processed by this system. The s-th cycle consists of n_s jobs for s=1,2,..,m.

Let $j^s(k)$ be the k-th job to be processed in the s-th cycle, the the system behaves as

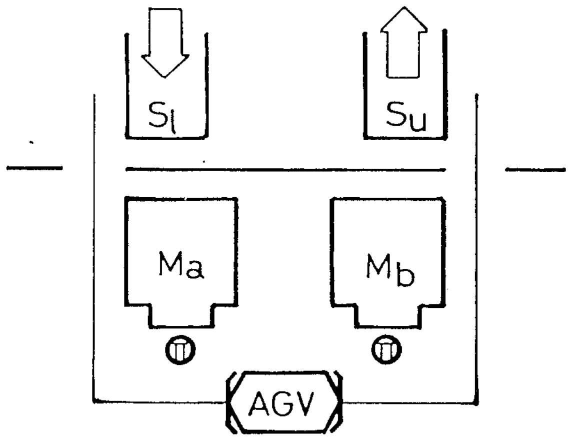


Fig.1 A Schematic of a FMC

follows, where

$$j^{s}(-k)=j^{s-1}(n_{s-1}-k),\ k=0,1,2,\ s=2,3,\ldots,m. \quad (1)$$

is assumed for notational convenience.

Step 1.

(1) The AGV picks up $j^1(1)$, the 1st job in the 1st cycle, at loading station S_l at time 0, and sends it to machine M_a.

(2) M_a starts processing $j^1(1)$, while the empty AGV goes back to S_l and sends the second job $j^1(2)$ to M_a where it is exchanged for $j^1(1)$ after it is finished on M_a.

(3) M_a starts processing $j^1(2)$, and the AGV sends $j^1(1)$ to machine M_b.

(4) M_b starts processing $j^1(1)$, while the empty AGV travels to S_l.
(Go to Step 2 after letting $k \leftarrow 2$ and $s \leftarrow 1$.)

Step 2.

(5) The AGV sends $j^s(k+1)$, the $(k+1)$-th job in the s-th cycle, to M_a from S_l, and exchanges it for job $j^s(k)$ after it is finished on M_a.

(6) M_a starts processing $j^s(k+1)$, while the AGV sends $j^s(k)$ to M_b where it is exchanged for $j^s(k-1)$ after it is finished on M_b.

(7) M_b starts processing $j^s(k)$, and the AGV sends $j^s(k-1)$ to unloading station S_u where the finished job is deposited.
(If $s = m$ and $k = n_s-1$, go to Step 4. Otherwise go to Step 3.)

Step 3.

(8) The empty AGV travels to S_l from S_u.
(Return to Step 2 after letting $k \leftarrow k+1$ if $k<n_s$, otherwise $k \leftarrow 0$ and $s \leftarrow s+1$.)

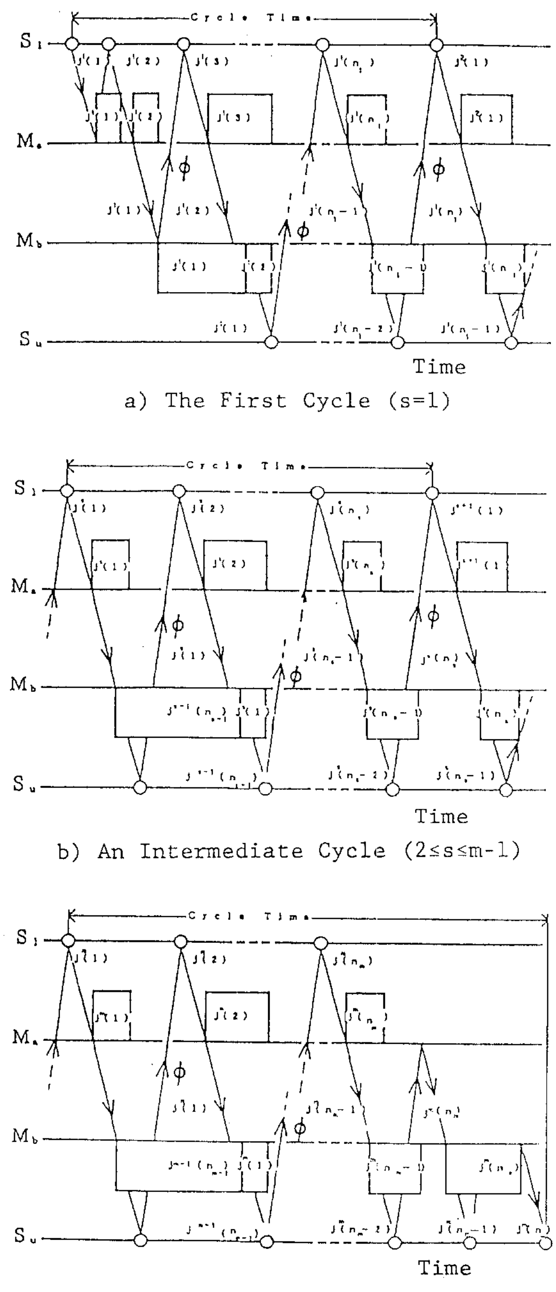
Step 4.

(9) The empty AGV travels to M_a from S_u, and sends $j^m(n_m)$, the last job in the last cycle to M_b after it is finished on M_b.

(10) M_b starts processing job $j^m(n_m)$, while the AGV sends $j^m(n_m-1)$ to S_u, and then travels to M_b.

(11) The AGV sends $j^m(n_m)$ to S_u after it is finished on M_b.
(Halt.) ∎

Fig. 2 illustrates the above behavior in the s-th cycle.



a) The First Cycle (s=1)

b) An Intermediate Cycle ($2 \le s \le m-1$)

c) The Last Cycle (s=m)

Fig.2 Gantt Charts of Cyclic Schedules

3. Formulation of Schedule

The following notations are used to formulate a schedule.

$p_a(j)$, $p_b(j)$: positive processing times of job j on machines M_a and M_b, respectively.
t_{1a}, t_{ab}, t_{bu}: nonnegative transportation times for the AGV to send a job from S_1 to M_a, from M_a to M_b and from M_b to S_u, respectively.
t_{a1}, t_{u1}, t_{ua}, t_{ub}: nonnegative times for the empty AGV travels from M_a to S_1, from S_u to S_1, from S_u to M_a and from S_u to M_b, respectively.
$t_{rnd} = t_{1a}+t_{ab}+t_{bu}+t_{u1}$: a turnaround time of the AGV.
$T_1(j)$: time instant when job j is loaded the AGV at S_1.
$T_a(j)$, $T_b(j)$: time instants when M_a and M_b start processing job j, respectively.
$T_u(j)$: time instant when job j is unloaded the AGV at S_u, i.e., the completion time of job j.

Let $\pi^s=[j^s(1),j^s(2),\dots,j^s(n_s)]$, $s=1,2,\dots,m$, be a sequence of n_s jobs to be processed in the s-th cycle. Then the schedule for π^s can be formulated as follows.

$$T^1[j^1(1)] = 0. \tag{2}$$

$$T_a[j^1(1)] = T_1[j^1(1)]+t_{1a}. \tag{3}$$

$$T_1[j^1(2)] = T_a[j^1(1)]+t_{a1}. \tag{4}$$

$$T_a[j^1(2)] = \max\{T_a(j^1(1)]+p_a[j^1(1)],\ T_1[j^1(2)]+t_{1a}\}. \tag{5}$$

$$T_b[j^1(1)] = T_a[j^1(2)]+t_{ab}. \tag{6}$$

$$\begin{aligned} T_1[j^1(3)] &= T_b(j^1(1)]+t_{b1},\\ T_1[j^1(k)] &= T_u[j^1(k-3)]+t_{u1},\ k=4,5,\dots,m, \text{ and}\\ T_1[j^s(k)] &= T_u[j^s(k-3)]+t_{u1},\\ &\quad k=1,2,\dots,n_s,\quad s=2,3,\dots,m. \end{aligned} \tag{7}$$

$$\begin{aligned} T_a[j^s(k+1)] &= \max\{T_a[j^s(k)]+p_a[j^s(k)],\\ &\qquad T_1[j^s(k+1)]+t_{1a}\},\\ &\quad k=0,1,\dots,n_s-1,\quad s=1,2,\dots,m. \end{aligned} \tag{8}$$

$$\begin{aligned} T_b[j^s(k)] &= \max\{T_a[j^s(k+1)]+t_{ab},\\ &\qquad T_b[j^s(k-1)]+p_b[j^s(k-1)]\},\\ &\quad k=1,2,\dots,n_s-1,\ s=1,2,\dots,m, \text{ and}\\ T_b[j^s(n_s)] &= \max\{T_a[j^{s+1}(1)]+t_{ab},\\ &\qquad T_b[j^s(n_s-1)]+p_b[j^s(n_s-1)]\},\\ &\quad s=1,2,\dots,m-1. \end{aligned} \tag{9}$$

$$\begin{aligned} T_u[j^s(k)] &= T_b[j^s(k+1)]+t_{bu},\\ &\quad k=1,2,\dots,n_s-1,\ s=1,2,\dots,m, \text{ and}\\ T_u[j^s(n_s)] &= T_b[j^{s+1}(1)]+t_{bu},\\ &\quad s=1,2,\dots,m-1. \end{aligned} \tag{10}$$

$$\begin{aligned} T_b[j^m(n_m)] &= \max\{T_u[j^m(n_m-2)]+t_{ua}+t_{ab},\\ &\qquad T_a[j^m(n_m)]+p_a[j^m(n_m)]+t_{ab},\\ &\qquad T_b[j^m(n_m-1)]+p_b[j^m(n_m-1)]\}. \end{aligned} \tag{11}$$

$$\begin{aligned} T_u[j^m(n_m)] &= T_b[j^m(n_m)]\\ &\quad +\max\{p_b[j^m(n_m)],\ t_{bu}+t_{ub}\} +t_{ub}. \end{aligned} \tag{12}$$

where (1) and

$$T_a[j^1(0)]=T_b[j^1(0)]=p_a[j^1(0)]=p_b[j^1(0)]=0, \tag{13}$$

are assumed for notational convenience.

For sequence $\pi=(\pi^1,\pi^2,\dots,\pi^m)$ with $\pi^s=(j(1),j(2),\dots,j(n_s))$ define each cycle time by

$$C_t(\pi^s) = T_1[j^{s+1}(1)]-T_1[j^s(1)],\ s=1,2,\dots,m-1,$$

$$\text{and } C_t(\pi^m) = T_u[j^m(n_m)]-T_1[j^m(1)]. \tag{14}$$

Let

$$A[j]=\max\{p_a[j],t_{rnd}\} \text{ and}$$

$$B[j]=\max\{p_b[j],t_{rnd}\}. \tag{15}$$

Then the following upper bound of each cycle time can be obtained.

Lemma 1. For sequence $\pi^s=[j^s(1),j^s(2),\dots,j^s(n_s)]$, $s=1,2,\dots,m$,

$$\begin{aligned} C_t(\pi^1) &\le \Sigma_{k=1}^{n_1-2} \max\{A[j^1(k+1)],\ B[j^1(k)]\}\\ &\quad +\max\{p_a[j^1(1)],t_{1a}+t_{a1}\} +T_{rnd}, \end{aligned} \tag{16}$$

$$\begin{aligned} C_t(\pi^s) &\le \Sigma_{k=0}^{n_s-1} \max\{A[j^s(k)],\ B[j^s(k-1)]\},\\ &\quad s=2,3,\dots,m-1, \text{ and} \end{aligned} \tag{17}$$

$$\begin{aligned} C_t(\pi^m) &\le \Sigma_{k=0}^{n_m-1} \max\{A[j^m(k)],\ B[j^m(k-1)]\}\\ &\quad +\max\{p_a[j^m(n_m)],\ t_{rnd},\ p_b[j^m(n_m-1)]\}\\ &\quad +\max\{p_b[j^m(n_m),t_{bu}+t_{ub}\}-t_{u1}, \end{aligned} \tag{18}$$

hold, where (1) is assumed.

Proof. In the following only the case of intermediate cycle, m>s>1, will be proven. (The remaining cases can be shown in a similar way.) It follows by (9), (8) and (7) that

$$\begin{aligned} T_b[j^s(k)] &= \max\{T_a[j^s(k+1)]+t_{ab},\\ &\qquad T_b[j^s(k-1)]+p_b[j^s(k-1)]\}\\ &= \max\{T_a[j^s(k)]+p_a[j^s(k)]+t_{ab},\\ &\qquad T_1[j^s(k+1)]+t_{1a}+t_{ab},\\ &\qquad T_b[j^s(k-1)]+p_b[j^s(k-1)]\}\\ &= \max\{T_a[j^s(k)]+p_a[j^s(k)]+t_{ab},\\ &\qquad T_b[j^s(k-1)]+t_{rnd},\\ &\qquad T_b[j^s(k-1)]+p_b[j^s(k-1)]\},\\ &\quad k=1,2,\dots,n_s-1. \end{aligned} \tag{19}$$

This means by (9) and (15) that

$T_b[j^s(k)] \leq T_b[j^s(k-1)]+\max\{A[j^s(k)], B[j^s(k-1)]\}, \; k=1,2,\ldots,n_s-1.$

Thus we have by (14), (7) and (10) that

$$C_t(\pi^s) = T_b[j^s(n_s-1)]-T_b[j^{s-1}(n_{s-1}-1)] \leq \Sigma_{k=0}^{n_s-1} \max\{A[j^s(k)], B[j^s(k-1)]\}. \;\blacksquare$$

4. An Approximation Algorithm

Our objective is to find a schedule that minimizes the cycle time in each cycle defined by (14). This problem is, however, NP-complete, even if $t_{rnd} = 0$, and there is only a single cycle, as shown by Papadimitriou and Kanellakis, [6]. we propose an approximation algorithm that minimizes the upper bound of each cycle time given in Lemma 1, instead of the cycle time itself. This algorithm is based on the Gilmore and Gomory's algorithm [1] which can solve the following traveling salesman problem (denoted TSP).

TSP: There are n cities $\{0,1,..,n-1\}$, each of which a traveling salesman has to visit exactly once. The cost of traveling from city i to city j is given by

$$c(i,j) = \max\{A(j), B(i)\}, \qquad (20)$$

where $A(k)$ and $B(k)$ are given for each city k. Thus the cost of tour $\pi=[j(0)-j(1)-\ldots-j(n-1)-j(0)]$ is given by

$$T_c(\pi) = \Sigma_{k=1}^{n-1} c[j(k-1),j(k)] + c[j(n-1),j(0)]$$
$$= \Sigma_{k=1}^{n-1} \max\{A[j(k)],B[j(k-1)]\} + \max\{A[j(0)],B[j(n-1)]\}. \qquad (21)$$

We ask an optimal tour that minimizes the tour cost (21).

Problem TSP is well known as a solvable case of the traveling salesman problem, since Gilmore and Gomory [1] have given an O(nlogn) time algorithm for it. The minimization of upper bound of each cycle time given by Lemma 1 can be reduced to an instance of problem TSP as shown in the following. We begin with an intermediate s-th cycle, i.e., $s=2,3,\ldots,m-1$. The 1st and the last cycles will be discussed after that.

Assume that we have already obtained an optimal sequence, $\pi^{s-1}=[j^{s-1}(1),j^{s-1}(2),\ldots,j^{s-1}(n_{s-1})]$ for the (s-1)-th cycle. Let $J^s=\{1,2,\ldots,n_s\}$ and $\pi^s=[j^s(1),j^s(2),\ldots,j^s(n_s)]$ be the set of n_s jobs and a sequence in the s-th cycle, respectively, then

$$UB^s = \max\{A[j^s(0)], B[j^s(-1)]\} + \Sigma_{k=1}^{n_s-1}\max\{A[j^s(k)],B[j^s(k-1)]\} \qquad (22)$$

is an upper bound of the minimum cycle time in the s-th cycle, as shown in Lemma 1. On the other hand, consider a TSP with set of n_s cities, $\{0,j^s(1),\ldots,j^s(n_s-1)\}$ and A[j] and B[j] given by (15) except that

$$A(0) = \max_{1\leq j\leq n_s} B(j), \text{ and}$$
$$B(0) = \max\{p_b[j^{s-1}(n_{s-1}), t_{rnd}\}. \qquad (23)$$

Then for tour $\pi^s=[j^s(0)-j^s(1)-..-j^1(n_s-1)-j^s(0)]$ with $j^s(0)=0$, the tour cost of (21) is given by

$$T_c(\pi^s) = \Sigma_{k=1}^{n_s-1} \max\{A[j^s(k)],B[j^s(k-1)]\}+A[0].$$

Thus by (22)

$$UB^s = T_c(\pi^s)-A[0]+\max\{A[j^s(0),B[j^s(-1)]\}.$$

This and the assumption on the (s-1)-th cycle already obtained mean that the minimum UB^s for job set, $\{j^s(1),j^s(2),\ldots,j^s(n_s-1)\}$, that is a closer upper bound of the minimum cycle time, can be obtained by solving the corresponding TSP. That is, if $\pi^s=[0-j^s(1)-..-j^s(n_s-1)-0]$ is an optimal tour, then sequence $\pi^s=[j^s(1),j^s(2),..,j^s(n_s-1)]$ minimizes UB^s for job set $\{j^s(1),..,j^s(n_s-1)\}$. Therefore, the minimum UB^s for the entire job set $\{1,2,..,n_s\}$, that is the closest upper bound of the minimum cycle time, can be obtained by solving n_s TSP's, and selecting the best among the n_s tours obtained, as shown in the following procedure.

Approximation Algorithm for the s-th Cycle :

Step 1.
(1) Let $J^s=\{1,2,..,n_s\}$ be the set of n_s jobs in the s-th cycle, and A[j] and B[j] for $j\in J^s$ are given by (15).
(2) Let A[0] and B[0] be given by (23).
(3) $T_c^* \leftarrow \infty$, and $k\leftarrow 0$.

Step 2. If $k < n_s$, then go to Step 3 after letting $k \leftarrow k+1$. Otherwise, let $\pi^s=[0-j^s(1)-..-j^s(n_s-1)-0]$ be a tour with $T_c[\pi^s] = T_c^*$, then $\pi^s =(j^s(1),j^s(2),\ldots,j^s(n_s-1),j^s(n_s))$ is a sequence with the minimum UB^s. Halt.

Step 3.
(4) Solve TSP with set of cities, $(J^s-\{k\})\cup\{0\}$, and let π^s be an optimal tour.
(5) If $T_c[\pi^s]<T_c^*$, then,

$j^s(n_s)\leftarrow k$, and $T_c^*\leftarrow T_c[\pi^s]$.

(6) Go to Step 2. $\blacksquare$

Table 1 shows a problem instance with three cycles, each consisting of 5 jobs, where p_a and p_b represent processing times on two machines. Fig.3 shows cyclic schedules obtained by applying the above algorithm to problem instance shown in Table 1, where $j^s(k)$, $s=1,2,3$, $k=1,2,..,5$, represent job k in the s-th cycle. Note that jobs in Table 1 are renumbered according to schedules obtained. It can be easily seen from Fig.3 (and Table 1) that each adjacent jobs in schedules obtained have a tendency to have

$$p_a[j^s(k)] \doteqdot p_b[j^s(k-1)].$$

Now we have the following theorem from the above discussion.

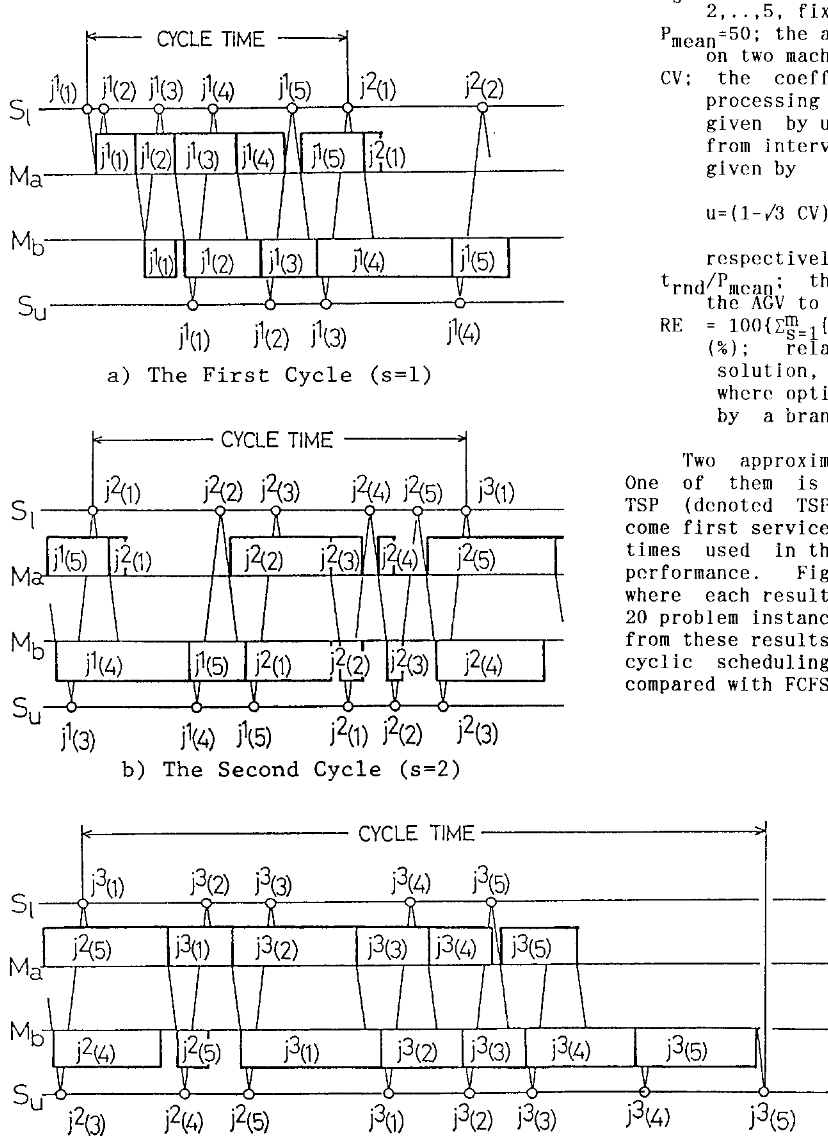
Theorem 1: The above algorithm gives a the minimum UB^s given by (22) in $O(n_s^2 \log n_s)$ time.

The above algorithm with modification of B[0]=0 in Step 1 minimizes UB^1, the upper bound

Table 1. Problem Instance

(m=3, n_s=5, t_{rnd}=6, P_{mean}=10, CV=0.5)

m \ n_s		1	2	3	4	5
1	Pa	5	5	8	6	8
	Pb	4	10	7	17	7
2	Pa	2	13	4	2	16
	Pb	11	3	2	14	4
3	Pa	8	16	9	8	10
	Pb	18	10	8	14	15



a) The First Cycle (s=1)

b) The Second Cycle (s=2)

c) The Last Cycle (s=3)

Fig.3 Cyclic Schedules Obtained by the Approximation Algorithm

for the first cycle given by (16). It also minimizes UB^m, the upper bound for the last (i.e., the m-th) cycle given by (18) by modifying

$$T_c[\pi^m] \leftarrow T_c[\pi^m] + \max\{p_b[j^m(n_m)],\ t_{bu}+t_{ub}\}.$$

as well as the modification of B[0]=0.

5. Computer Simulation

In order to validate the performance of the proposed approximation algorithm, some computer simulations were executed. That is, the following parameters were used:

- m=5; the number of cycles fixed to 5.
- n_s=10; the number of jobs in each cycle, s=1, 2,..,5, fixed to 10.
- P_{mean}=50; the average processing time of jobs on two machines fixed to 50.
- CV; the coefficient of variation of job processing times. $p_a(j)$ and $p_b(j)$ are given by uniform random integers taken from interval [u,v], where u and v are given by

  $u=(1-\sqrt{3}\ CV)P_{mean}$, and $v=(1+\sqrt{3}\ CV)P_{mean}$,

  respectively.
- t_{rnd}/P_{mean}; the ratio of turnaround time of the AGV to the mean processing time.
- RE $= 100\{\Sigma_{s=1}^{m}[C_t(\pi^s)-C_t(\pi^{s*})]\}/\Sigma_{s=1}^{m}C_t(\pi^{s*})$ (%); relative error of approximate solution, π^s to optimal solution, π^{s*}, where optimal solutions were calculated by a branch-and-bound algorithm [3].

Two approximation algorithms were compared; One of them is the proposed one based on TSP (denoted TSP), the other is the first come first service (denoted FCFS) which is some times used in the queuing analysis of system performance. Fig. 4 shows results obtained, where each result presents the mean value over 20 problem instances tested. It can be concluded from these results that the proposed approximate cyclic scheduling give quite good schedules as compared with FCFS scheduling.

6. Conclusion

An optimal cyclic scheduling problem for a FMC modeled by a two-machine flowshop with an AGV and APC's was discussed. Since this problem is NP-complete, an approximation algorithm based on Gilmore and Gomory's TSP algorithm was proposed. Computer simulations implemented showed that the approximation algorithm proposed gives schedules with mean relative errors within 2 (%) in wide ranges of The turnaround times of the AGV and the coefficient variation of job processing times.

Based on these results, We are now going on a study on an cyclic scheduling problem for a FMC with arbitrary number of APC's.

References

[1] Gilmore,P.C. and Gomory,R.E.,"Sequencing a One State-Variables Machine: A Solvable Case of the traveling salesman problem," ORSA, **12**(1964), 655-679.

[2] Graves,C.S., Meal,H.C., Stefek,D. and Zegmi, A.H.,"Scheduling Re-Entrant Flowshops," J. Opns. Mgmt., **3**(1983), 197-207.

[3] Karuno,Y., "A Study on Optimal Cyclic Scheduling for An Automated Two-Machine Manufacturing System with Finite Buffers," Graduation Thesis, Department of Mechanical and System Engineering, Kyoto Institute of Technology (1990), (in Japanese).

[4] Kise,H., Matsuo,H. and Sullivan,R.S., "Optimal Cyclic Scheduling for Automated Two-Machine Flowshop with No Buffer," Proc. 1990 Japan-U.S.A. Symp. on Flexible Automation III(1990), 1179-1185.

[5] Matsuo,H.,"Cyclic Sequencing Problems in the Two-Machine Permutation Flowshop: Complexity, Worst Case and Average Case Analysis," Working Paper, Department of Management, The University of Texas at Austin(1988).

[6] Papadimitriou,C.H. and Kanellakis,P.C., "Flowshop Scheduling with Limited Temporary Storages," ACM, **27**(1980), 553-549.

[7] Ravikumar,K. and Vannelli,A.,"Strategic Subcontracting for Efficient Disaggregated Manufacturing," I.J.P.R., **25**(1987), 1715-1728.

[8] Wassenhove,L.V.,"A Planning Framework for a Class of FMS," Opns. Res. Proc. 1988, 524-53, Springer-Verlag Berlin Heidelberg 1989.

[9] Yamazaki,T. and Nagae,A.,"Market Oriented FMS and CIM: A Megatrend for '90's Manufacturing Industries," Proc. 1990 Japan-U.S.A. Symp. on Flexible Automation," I(1990), 7-12.

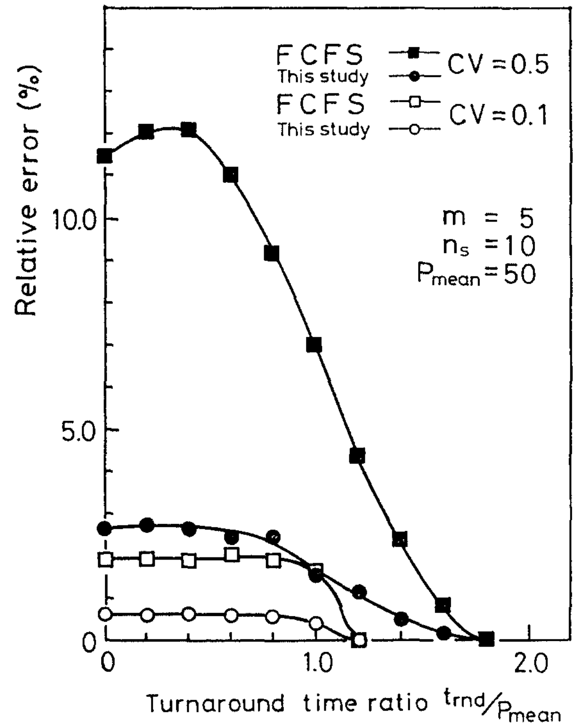


Fig.4 Performance of Approximation Algorithms