

# Scheduling for an automated three-machine flowshop manufacturing system

*Jinliang Cheng and Hiroshi Kise*

*Department of Mechanical and System Engineering,*

*Kyoto Institute of Technology*

*Matsugasaki, Sakyoku, Kyoto 606, JAPAN,*

*Tel: +81-75-724-7353 Fax: +81-75-724-7300*

*E-mail: cheng@hon.apptec.co.jp kise@ipc.kit.ac.jp*

## Abstract

This paper considers a scheduling problem of minimizing the maximum completion time for an automated flowshop manufacturing system such as FMS which consists of 3 machining cells with sufficient buffers, an AGV (automated guided vehicle) and loading/unloading stations. For this problem we propose a heuristic algorithm based on a fuzzy approximation (called fuzzy scheduling), and a branch-and-bound algorithm with fuzzy inferences. Computational experiences show that the fuzzy scheduling can give optimal or near optimal solutions, and the branch-and-bound algorithm can efficiently give optimal solutions with up to 400 parts with high probability over 90%.

## Keywords

FMS, AGV, flowshop scheduling, branch-and-bound method, fuzzy inference

## 1 INTRODUCTON

Many flexible manufacturing systems (FMS's) have been widely used around the world. An FMS can be characterized as a set of flexible machine tools connected by a material handling system and which is controlled by both computers and human operators [1]. On the other hand, the efficient implementation of such an FMS presents a complex set of issues to be solved. Among them are scheduling problems which optimally determine when and on what machine the parts are processed and how these parts are transported in the system [9].

Kise [5] has considered a two-machine, one-AGV (or a robot) scheduling problem with sufficient WIP-buffer at each machine and shown that the problem is an *NP*-hard 3-machine flowshop scheduling problem (denoted 3FSP). Kise et al. [6] have proposed a branch-and-bound algorithm for the same prob-

lem, and demonstrated by numerical experiments that the algorithm can exactly solve problem instances with up to 200 parts with high possibility. Stern and Vitner [10] have dealded with a two-machine, one-robot scheduling problem with part-dependent transport times. They have shown the NP-hardness of the problem and suggested an approximation algorithm.

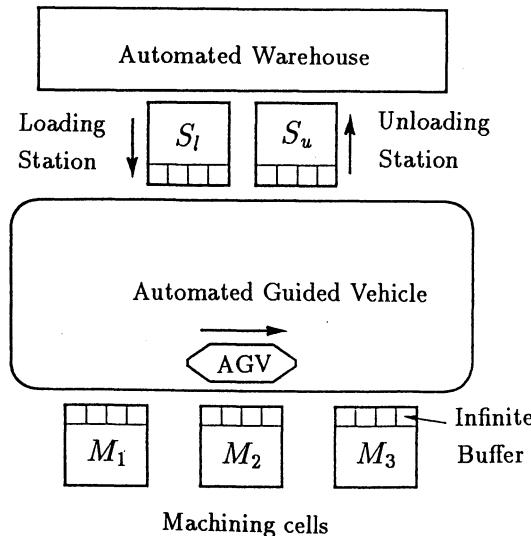
Sabuncuoglu and Hommertzheim [8] have considered an FMS scheduling problem by using a simulation model. They have analyzed the relative performances of machine and AGV scheduling rules against various due-date criteria. Ishii and Talavage [4] have proposed a mixed dispatching rule for each machine based on discrete event simulation in FMS scheduling. Their system includes two loading/unloading stations, four machines, three AGV's and six part types. This brief survey may suggest that scheduling problems arising in FMS's environment are hard to analytically be solved.

In this paper we deal with an automated flowshop manufacturing system such as FMS that consists of 3 machining cells with sufficient capacity of WIP-buffer, all of which are connected by an AGV (or a movable robot), and discuss an optimal scheduling problem that asks to minimize the makespan of  $n$  parts to be processed by the system. This paper aims to develop a branch-and-bound (BAB) algorithm that could efficiently solve large problem instances with high probability. Numerical experiments demonstrate that the BAB algorithm can solve large problem instances with up to 400 parts in reasonable time with quite high possibility.

## 2 MODEL DESCRIPTION

Figure 1 shows the physical layout of an FMS for study in this paper, which has a loading station  $S_L$ , an unloading station  $S_U$ , 3 (machining) cells (e.g., machining centers) with pallet storage carousel, an AGV or a movable robot and an automated warehouse. The cells, loading and unloading stations are arranged along a loop track on which the AGV can travel in unidirection. For this system the following assumptions are made:

- (1) A set of  $n$  parts,  $J = \{i|i = 1, 2, \dots, n\}$ , is available at time 0. (2) Each of 3 cells,  $M_l$  ( $l = 1, 2, 3$ ), can process at most one part at a time, and is never interrupted during processing.
- (3) Each part  $i \in J$  is processed in the order of  $M_1, M_2, M_3$  and the sequences of processing the parts at cells  $M_l$  ( $l = 1, 2, 3$ ) are the same.
- (4) Loading and unloading stations, and 3 cells have buffers for WIP(work-in-process). The capacity of each buffer is unlimited.
- (5) The AGV can carry at most one part at a time. The AGV has constant traveling speed and fixed pickup and drop rates.
- (6) A part is released from loading station  $S_L$  to the shop every time the AGV leaves  $S_L$ , then is carried to each cells,  $M_l$  ( $l = 1, 2, 3$ ) and leaves the shop at unloading station  $S_U$ . At each cell, the AGV stops to drop a part which should be processed on the cell and pick up a part which should be



**Figure 1** Shop configuration

processed on the next cell. The AGV leaves each cell  $M_l$  ( $l = 1, 2, 3$ ) without waiting there until  $M_l$  finishes a part if there is no part in the buffer.

(7) The processing times of part  $i$  on  $M_l$  ( $l = 1, 2, 3$ ), including set-up time, are known and represented by  $P_l(i)$  ( $i \in J$ ,  $l = 1, 2, 3$ ).

(8) The times required for the AGV to carry a part from  $S_L$  to  $M_1$ , from  $M_1$  to  $M_2$ , from  $M_2$  to  $M_3$ , from  $M_3$  to  $S_U$  and from  $S_U$  to  $S_L$ , including pickup and drop times, are known and represented by  $t_{01}, t_{12}, t_{23}, t_{34}, t_{40}$ , respectively. Such times are independent of the parts to be carried, thus, the time required for the AGV to travel a complete loop,  $t_v = t_{01} + t_{12} + t_{23} + t_{34} + t_{40}$ , is constant.

(9) An optimal sequence of processing the  $n$  parts to be found is one that minimizes the makespan, that is, the total elapse time between the time the first part is released from  $S_L$  and the time the last part is delivered to  $S_U$ .

Hereafter, this problem will be called the automated 3-machine flowshop scheduling problem (denoted A3FSP).

### 3 FORMULATION OF SCHEDULING PROBLEM

The finishing time of machining part  $i$  on  $M_l$  ( $l = 1, 2, 3$ ) is represented by  $F_l(i)$ , and the time of picking part  $i$  from  $M_l$  is represented by  $T_l(i)$ . Furthermore, the time part  $i$  leaves  $S_L$  and the time it arrives at  $S_U$  are represented by  $T_0(i)$  and  $F(i)$ , respectively. Then, the schedule of the  $k$ -th part  $j_k$  in any sequence,  $s = (j_1, j_2, \dots, j_n)$ , can be formulated as follows.

By assumption (6), it can be seen that

$$T_0(j_k) = (k-1)t_v, \quad k = 1, 2, \dots, n. \quad (1)$$

From assumption (1)~(8), the start of machining part  $j_k$  on  $M_l$  must be after the machining part  $j_{k-1}$  has finished and after  $j_k$  has been transferred to  $M_l$  by the AGV, so the finishing time of machining part  $j_k$  on  $M_l$  is given by

$$\begin{aligned} F_l(j_k) &= \max\{F_l(j_{k-1}), T_{l-1}(j_k) + t_{l-1,l}\} + P_l(j_k), \\ l &= 1, 2, 3; k = 1, 2, \dots, n, \end{aligned} \quad (2)$$

where  $F_l(j_0) = 0 (l = 1, 2, 3)$ .

Picking part  $j_k$  from  $M_l$  must be started at a time the AGV arrives there after the finishing of machining  $j_k$  on  $M_l$ , then

$$T_l(j_k) \geq t_{0l} + X_{lk}t_v \geq F_l(j_k), \quad l = 1, 2, 3; k = 1, \dots, n, \quad (3)$$

where  $X_{lk}$  is the minimal integer such that Eq.(3) holds, and  $t_{0l} = t_{01} + \dots + t_{l-1,l} (l = 2, 3, 4)$ . From Eq.(3)

$$X_{lk} \geq \lceil (F_l(j_k) - t_{0l})/t_v \rceil, \quad l = 1, 2, 3; k = 1, \dots, n, \quad (4)$$

where  $\lceil x \rceil$  is the minimal integer greater than  $x$ . By assumption (5), the AGV can pick  $j_k$  from  $M_l$  only after it picked part  $j_{k-1}$  from  $M_l$  and arrives at  $M_l$  again, i.e.,

$$T_l(j_k) \geq T_l(j_{k-1}) + t_v, \quad l = 1, 2, 3; k = 1, \dots, n, \quad (5)$$

where  $T_l(j_0) = 0 (l = 1, 2, 3)$ , then from Eqs.(3)~(5),

$$\begin{aligned} T_l(j_k) &= \max\{\lceil (F_l(j_k) - t_{0l})/t_v \rceil t_v + t_{0l}, T_l(j_{k-1}) + t_v\}, \\ l &= 1, 2, 3; k = 1, \dots, n. \end{aligned} \quad (6)$$

The time part  $j_k$  arrives at  $S_U$  is

$$F(j_k) = T_3(j_k) + t_{34}, \quad k = 1, \dots, n. \quad (7)$$

Thus the schedule can be computed by Eqs.(1), (2), (6) and (7), and the maximum completion time  $F_{max}(s)$  under the sequence  $s$  is

$$F_{max}(s) = F(j_n). \quad (8)$$

## 4 FUZZY SCHEDULING

### 4.1 An approximation of makespan

Problem A3FSP is NP-hard, even for the case of two-machine sysystem [5]. Thus, we need good heuristic for practical purposes. However, the above formulation that expresses makespan  $F_{max}$  by recursive equations (1), (2), (6), (7) and (8) make us somewhat difficult even to have an insight for a good heuristic. Thus we consider the following approximation of  $F_{max}(s)$ .

In Eq.(6) we relax assumption (5) to the one that the AGV can simultaneously carry parts existing in a buffer except  $S_L$ . Then the time when the AGV leaves cell  $M_l$  is given by

$$\hat{T}_l(j_k) = \lceil (F_l(j_k) - t_{0l})/t_v \rceil t_v + t_{0l}, l = 1, 2, 3, \quad (6)'$$

$\hat{T}_l(j_k)$  is a lower bound of  $T_l(j_k)$  in Eq.(6).

Let  $\hat{F}_l(j_k)$  be the finishing time of part  $j_k$  on cell  $M_l$  ( $l = 1, 2, 3$ ),  $\hat{F}(j_k)$  be the time when part  $j_k$  arrives at  $S_U$  and  $\hat{F}_{max}(s)$  be the makespan, all of which are computed by Eq.(6)' instead of Eq.(6). Then we have

$$\begin{aligned} \hat{F}_1(j_k) &= \max_{1 \leq p \leq k} \{(p-1)t_v + \sum_{h=p}^k P_1(j_h)\} + t_{01}, \\ \hat{F}_l(j_k) &\leq \max_{1 \leq p \leq k} \{\hat{F}_{l-1}(j_p) + \sum_{h=p}^k P_l(j_h)\} + t_v + t_{l-1,l}, \quad l = 2, 3, \\ \hat{T}_3(j_k) &\leq \hat{F}_3(j_k) + t_v, \quad k = 1, \dots, n. \end{aligned} \quad (9)$$

Thus, the following upper bound  $UB(s)$  of  $\hat{F}_{max}(s)$  can be obtained.

$$\hat{F}_{max}(s) \leq UB(s) = C(s) + 2t_v + t_{0,4} \quad (10)$$

where

$$C(s) = \max_{1 \leq o \leq p \leq q \leq n} \left\{ \sum_{h=1}^o t_v + \sum_{h=o}^p P_1(j_h) + \sum_{h=p}^q P_2(j_h) + \sum_{h=q}^n P_3(j_h) \right\} \quad (11)$$

Obviously minimizing  $UB(s)$  is equivalent to minimizing  $C(s)$ . The problem of minimizing  $C(s)$  is a 4FSP with processing time  $t_v, P_1(i), P_2(i), P_3(i)$  of part  $i$  and with no transportation time of the AGV. That is, the original A3FSP can be approximately reduced to a 4FSP.

## 4.2 Fuzzy scheduling method

The fuzzy scheduling method has been proposed to yield nearly optimal solutions for an  $m$ -machine FSP [2]. The basic idea of this heuristic is to use a membership function in the context of fuzzy inference for obtaining an approximate solution. The membership function represents a possibility that the dominance relation between parts holds even if its precondition does not hold. We describe it briefly below.

For the above reduced 4FSP of Eq.(11), let flow time of part  $j_k$  on the  $l$ -th machine ( $l = 2, 3, 4$ ) for a partial sequence of the first  $k$  parts,  $s_k = (j_1, \dots, j_k)$ , be defined by

$$FT_l(s_k) = FT_l(j_k) - FT_1(j_k), \quad (12)$$

where  $FT_l(j_k)$  is the finishing time of processing part  $j_k$  on the  $l$ -th machine

in sequence  $s_k$  for the 4FSP. Similarly flowtime  $FT_l(s_k, i)$  of part  $i$  in sequence  $(s_k, i)$  adding  $i$  after  $s_k$  and  $FT_l(s_k, i, j)$  of part  $j$  in sequence  $(s_k, i, j)$  adding  $j$  after  $(s_k, i)$  are defined.

**Theorem 1.** [2] Assume that two parts  $i$  and  $j$  are optimally processed immediately after partial sequence  $s_k$ , part  $i$  optimally precedes part  $j$  if

$$FT_l(s_k, i, j) \leq FT_l(s_k, j, i), \quad l = 2, 3, 4. \quad (13)$$

Eq.(13) rarely holds in realistic situations. But it is a sufficient condition for the optimal schedule. This suggests that if Eq.(13) approximately holds, then part  $i$  may precede part  $j$  in an optimal schedule with high possibility. We take advantage of this possibility for searching an optimal schedule, and represent it by a membership function in the fuzzy inference. That is,

$$D_l(s_k, i, j) = FT_l(s_k, i, j) - FT_l(s_k, j, i), \quad l = 2, 3, 4, \quad (14)$$

then the membership function that represents the degree that part  $i$  optimally precedes part  $j$  is given by

$$\mu_{s_k}(i, j) = 0.5 - \frac{D(s_k, i, j)}{2D_{max}(s_k)}, \quad (15)$$

where  $D(s_k, i, j) = \sum_{l=2}^4 \alpha_{l-1} D_l(s_k, i, j)$ ,  $D_{max}(s_k) = \max_{i,j} |D(s_k, i, j)|$  and  $\alpha_1, \alpha_2, \alpha_3$  ( $0 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$  and  $\sum_{l=1}^3 \alpha_l = 1$ ) are real numbers. Then, the degree of dominance of part  $i$  over the remaining parts under partial sequence  $s_k$  ( $k = 0, 1, \dots, n-1$ ) is given by

$$\mu_{s_k}^*(i) = \min_{j \in J_r} \mu_{s_k}(i, j), \quad (16)$$

then part  $i^*$  satisfying

$$\mu_{s_k}^*(i^*) = \max_{i \in J_r} \mu_{s_k}^*(i) \quad (17)$$

is identified as the part that immediately follows  $s_k$ , where  $J_r$  is the set of the remaining  $r (= n - k)$  parts.

The rule determining  $i^*$  by this way is referred to as fuzzy rule and the scheduling applying the fuzzy rule is referred to as fuzzy scheduling. We use a fuzzy schedule obtained as an approximate one of the original A3FSP, and its makespan (exactly computed by Eq.(6)) is used as an initial upper bound value of the BAB algorithm proposed next for the original A3FSP.

## 5 BAB ALGORITHM

It is assumed that the basic principle of BAB algorithm is well known (e.g., see [3]). Hence only the basic components of BAB algorithm are stated below.

## 5.1 Subproblem

Let the sequence of the first  $k$  parts fixed be  $s_k = (j_1, \dots, j_k)$ . The problem of determining an optimal sequence of the remaining  $r (= n - k)$  parts under the sequence  $s_k$  is called a subproblem of depth  $k$  and is represented by  $P(s_k)$ .

## 5.2 Lower bound

For a given sequence of the first  $k$  parts,  $s_k$  and an arbitrary sequence of the remaining  $r$  parts,  $\bar{s}_k = (j_{k+1}, \dots, j_n)$ , from Eqs.(1),(2) and (6) we have

$$\begin{aligned} T_1(j_n) &\geq \max\{F_1(j_k) + Y_{11}(\bar{s}_k), (k-1)t_v + Y_{01}(\bar{s}_k) + t_{01}\}, \\ T_2(j_n) &\geq \max\{F_2(j_k) + Y_{22}(\bar{s}_k), F_1(j_k) + Y_{12}(\bar{s}_k) + t_{12}, \\ &\quad (k-1)t_v + Y_{02}(\bar{s}_k) + t_{02}\}, \\ T_3(j_n) &\geq \max\{F_3(j_k) + Y_{33}(\bar{s}_k), F_2(j_k) + Y_{23}(\bar{s}_k) + t_{23}, \\ &\quad F_1(j_k) + Y_{13}(\bar{s}_k) + t_{13}, (k-1)t_v \\ &\quad + Y_{03}(\bar{s}_k) + t_{03}\}, \quad 0 \leq k \leq n-1, \end{aligned} \quad (18)$$

where

$$\begin{aligned} Y_{uv}(\bar{s}_k) = & \max_{k < q(u) \leq \dots \leq q(v-1) \leq n} \left[ \sum_{h=k+1}^{q(u)} P_u(j_h) + \dots + \right. \\ & \left. \sum_{h=q(v-1)}^n P_v(j_h) \right], \quad 0 \leq u \leq v \leq 3, \end{aligned} \quad (19)$$

and  $P_0(j_h) = t_v (h = 1, \dots, n)$ .

Now we consider the minimization of each  $Y_{uv}(\bar{s}_k)$  Eq.(19) that yields a lower bound of  $T_v(j_n)$  of Eq.(18), and leads to the one of the makespan  $F(j_n)$  of Eq.(8). The minimization of  $Y_{uv}(\bar{s}_k)$  is reduced to an FSP with processing time  $P_l(i)$  of part  $i$  on  $l$ -th machine. But it is NP-hard. Therefore, we consider the minimization of the following lower bound  $Y_{uv}^b(\bar{s}_k)$  of  $Y_{uv}(\bar{s}_k)$  instead of  $Y_{uv}(\bar{s}_k)$  itself, which is obtained by only considering the terms for  $k < q(u) = q(u+1) = \dots = q(v-1) \leq n$  and excluding the other terms of  $Y_{uv}(\bar{s}_k)$ , i.e.,

$$\begin{aligned} Y_{uv}(\bar{s}_k) &\geq Y_{uv}^b(\bar{s}_k) \\ &= \max_{k < q(v-1) \leq n} \left[ \sum_{h=k+1}^{q(v-1)} P_u(j_h) + \sum_{l=u+1}^{v-1} P_l(j_{q(v-1)}) \right. \\ &\quad \left. + \sum_{h=q(v-1)}^n P_v(j_h) \right] \end{aligned} \quad (20)$$

The minimization of  $Y_{uv}^b(\bar{s}_k)$  is reduced to a special 3FSP where the first machine and the third machine are separated by a non-bottleneck machine

with processing time  $\sum_{l=u+1}^{v-1} P_l(j)$  of part  $j$  [7]. Furthermore,

$$\begin{aligned} Y_{uv}^b(\bar{s}_k) &= \max_{k < q \leq n} \left[ \sum_{h=k+1}^q \left( \sum_{l=u}^{v-1} P_l(j_h) \right) + \sum_{h=q}^n \left( \sum_{l=u+1}^v P_l(j_h) \right) \right] - \sum_{l=u+1}^{v-1} Y_{ll}(\bar{s}_k) \\ &= Z_{uv}(\bar{s}_k) - \sum_{l=u+1}^{v-1} Y_{ll}(\bar{s}_k), \quad 1 \leq u < v \leq m \end{aligned} \quad (21)$$

and the problem of minimization  $Z_{uv}(\bar{s}_k)$  is reduced to a solvable 2-machine FSP with processing time  $a(i) = \sum_{l=u}^{v-1} P_l(i)$  on the first machine and  $b(i) = \sum_{l=u+1}^v P_l(i)$  on the second machine. Let  $Z_{uv}^*(\bar{s}_k)$  be its minimum value. Here we should note that processing of part  $j_{k+1}$  on machine  $M_u$  can not be started before  $F_u^*(j_k) = \max_{1 \leq l \leq u} [F_l(j_k) + \min A_{l,u-1}(\bar{s}_k) + t_{lu}]$ , and  $T_v(j_n)$  must be dependent of  $t_v$  and times the AGV goes around the loop track till  $T_v(j_n)$  (see Eq.(3)), then it can be easily seen by Eq.(18) through Eq.(21) that

$$g'_{uv}(s_k) = \lceil F_u^*(j_k) + Z_{uv}^*(\bar{s}_k) - \sum_{l=u+1}^{v-1} Y_{ll}(\bar{s}_k) + t_{uv} - t_{0v} \rceil t_v + t_{0v} \quad (22)$$

is a lower bound of  $T_v(j_n)$  for subproblem  $P(s_k)$ , where  $\min A_{u+1,u}(\bar{s}_k) = 0 (0 \leq u \leq 3)$ ,  $\min A_{uv}(\bar{s}_k) = \min_{k < h \leq n} [\sum_{l=u}^v A_l(j_h)] (1 \leq u \leq v \leq 3)$  and  $A_l(i) = \max \{P_l(i), t_v\}$ . Then,

$$g_{uv}(s_k) = \lceil (g'_{uv}(s_k) + \min A_{v+1,3}(\bar{s}_k) + t_{v3} - t_{03}) / t_v \rceil t_v + t_{0,m+1} \quad (23)$$

is a lower bound for subproblem  $P(s_k)$ .

We employ

$$g(s_k) = \max_{1 \leq u < v \leq 3} g_{uv}(s_k) \quad (24)$$

as a lower bound for  $P(s_k)$ . Lower bound  $g(s_k)$  for any subproblem  $P(s_k)$  except  $P(\emptyset)$ , i.e., the original problem can be computed in  $O(n)$  time and  $g(\emptyset)$ , the lower bound of the original problem in  $O(n \log n)$  time.

### 5.3 Fuzzy scheduling and fuzzy search

We use the fuzzy scheduling as mentioned before as the initial incumbent solution that takes a role of an upper bound of the optimal value. We adopt a depth-first search method that selects a subproblem with the smallest lower bound among the most recently generated ones, breaking ties by the fuzzy rule proposed in [2]. We call such search method the fuzzy search.

## 6 NUMERICAL EXPERIMENTS

In this section, performances of the BAB algorithm with fuzzy inferences proposed here are evaluated by the way of numerical experiments. For each  $n$  and  $t_v$ , 30 problem instances have been tested. The processing times of the parts ( $P_l(j)$ ) are generated for a uniform distribution over the integers from 1 to 99, inclusive. For weights  $\alpha_l$  ( $l = 1, 2, 3$ ) of membership function  $\mu_s(i, j)$  in Section 4.2, three types of function were applied to each problem instances. They are: (1) arithmetical progression weights,  $\alpha_l = l/6$ ; (2) equal weights,  $\alpha_l = 1/3$ ; (3) inverse arithmetical progression weights,  $\alpha_l = (4 - l)/6$  ( $l = 1, 2, 3$ ). Three schedules are obtained and the best one of them is adopted. In the fuzzy search of the BAB algorithm, only the arithmetical progression weights are used.

All programs were compiled in FORTRAN, and run on a DEC 3000 workstation (35MFLOPS). The running time (CPU time) of the BAB algorithm was limited within 5 min., and a problem instance that could not be solved within 5 min. has been identified as unsolved.

### 6.1 Performances of fuzzy scheduling

Table 1 and 2 show results obtained for test problems, where  $R_{FS}$  and  $R_{FC}$  represent average relative errors of the fuzzy scheduling and the FCFS (First Come First Served) scheduling, respectively. The comparisons with the FCFS scheduling in Tables 1 and 2 show the effect of the optimization by the fuzzy scheduling. Table 2 also shows the influence of the number of parts,  $n$  on the performance of the fuzzy scheduling. Table 3 shows how many problem instances can optimally solved by the fuzzy scheduling. We can conclude from these results that the fuzzy scheduling is superior.

**Table 1** Average relative errors(%) of fuzzy scheduling for  $n = 10$

$t_v/P_{mean}^*$	$R_{FS}$	$R_{FC}$
0.1	1.0	16.8
0.2	1.2	16.6
0.3	1.6	16.9
0.4	1.9	15.9
0.5	2.9	15.9
Average	1.7	16.4

\*:  $P_{mean} = 50$  is the mean processing time.

**Table 2** Average relative errors (%) of fuzzy scheduling for  $t_v = 15(t_v/P_{mean}) = 0.3$  for solved instances

$n$	10	20	30	40	50	100
$R_{FS}$	1.6	0.8	0.4	0.4	0.3	0.1
$R_{FC}$	16.7	12.1	10.8	8.3	7.9	5.4

**Table 3** Rate (%) of problem instances with  $n = 10$  for which the fuzzy scheduling gives optimal solutions

$t_v/P_{mean}$	0.1	0.2	0.3	0.4	0.5	Average
$A$	53	47	50	50	40	48

## 6.2 Evaluation of BAB algorithms

To examine the performance of the BAB algorithm with fuzzy inference, the following four kinds of BAB algorithms were implemented and compared:

(1)  $A$ : BAB algorithm with fuzzy inferences proposed here; (2)  $A_1$ : BAB algorithm  $A$  without fuzzy search; (3)  $A_2$ : BAB algorithm  $A$  without fuzzy scheduling; (4)  $A_3$ : BAB algorithm  $A_2$  without fuzzy search.

The rates of problem instances solved by these algorithms are shown in Table 4. It is evident that Algorithm  $A$  is superior to  $A_1$ ,  $A_2$  and  $A_3$ . The comparison with  $A_1$  as well  $A_3$  shows the effect of the fuzzy search on the performance of Algorithm  $A$ , and the comparison with  $A_2$  as well  $A_3$  shows the effect of the fuzzy scheduling on the performance of Algorithm  $A$ .

Table 5 show the influences of the number of parts on the solvability of Algorithm  $A$ . We can conclude that the BAB algorithm  $A$  can solve problem instances with 400 parts with high probability over 90%.

## 7 CONCLUSION

In this paper we have considered a scheduling problem of minimizing the makespan for an automated 3-machine flowshop such as FMS that consists of 3 machining cells with sufficient buffers, an AGV and loading/unloading stations, and shown that the problem can be approximately reduced to a 4-machine permutation flowshop scheduling problem. Based on this reduc-

**Table 4** Rate (%) of problem instances solved within 5 min. by BAB algorithms ( $t_v = 15(t_v/P_{mean} = 0.3)$ )

n	A	$A_1$	$A_2$	$A_3$
10-20	92	90	92	88
30-40	87	82	87	78
50-60	93	85	93	83
70-80	90	78	87	67
90-100	93	87	88	75
110-120	92	83	83	62
130-140	92	87	85	55
150-160	92	80	88	47
170-180	95	87	87	45
190-200	90	78	78	48
Average	92	84	87	65

**Table 5** Rate (%) of problem instances solved within 5 min. by Algorithm A for larger problems( $t_v = 15(t_v/P_{mean} = 0.3)$ )

$n$	A	$n$	A
210-220	90	310-320	83
230-240	88	330-340	100
250-260	92	350-360	92
270-280	92	370-380	97
290-300	90	390-400	83
Average		91	

tion we proposed a heuristic algorithm called the fuzzy scheduling, and a branch-and-bound algorithm with fuzzy inferences. Extensive numerical experiments show that the fuzzy scheduling can give optimal or near optimal solutions, and the branch-and-bound algorithm with fuzzy inferences can efficiently give optimal solutions to problem instances with up to 400 parts with high probability.

## 8 REFERENCES

- [1]Buzacott, J.A., and Yao, D.W. (1986) Flexible manufacturing systems: A review of models. *Management Science*, **32**, 890-905.
- [2]Cheng, J., Kise, H. and Matsumoto, H. (1997) A Branch-and-bound algorithm with fuzzy inference for a permutation flowshop scheduling problem. *European Journal of Operational Research*, **96**, 578-590.
- [3]Ibaraki, T. (1987) *Enumerative Approaches to Combinatorial Optimization Part I*, Annals of Operations Research (P.L. Hammer Ed.), **10**, J.C.Baltzer Science Publishers.
- [4]Ishii, N. and Talavage, J.J. (1994) A mixed dispatching rule approach in FMS scheduling. *The International Journal of Flexible Manufacturing Systems*, **6**, 69-87.
- [5]Kise, H. (1991) On an automated two-machine flowshop scheduling problem with infinite buffer. *Journal of the Operations Research Society of Japan*, **34**, 354-361.
- [6]Kise, H., K. Kohno, K., Shioyama, T and Kushiyama, T. (1992) Optimal scheduling for an automated two-machine manufacturing system. *Journal of Advanced Automation Technology*, **4**, 121-127.
- [7]Lageweg, B.J., Lenstra, J.K. and Rinnooy Kan, A.H.G. (1978) A general bounding scheme for the permutation flowshop problem. *Operations Research* **26**, 53-67.
- [8]Sabuncuoglu, I. and Hommertzheim, D.L. (1993) Experimental investigation of an FMS due-date scheduling problem: evaluation of machine and AGV scheduling rules. *The International Journal of Flexible Manufacturing Systems*, **5**, 301-323.
- [9]Stecke, K.E. (1985) Design, planning, scheduling, and control problems of flexible manufacturing systems. *Annals of Operations Research*, **3**, 3-12.
- [10]Stern, H.I. and Vitner, G. (1990) Scheduling parts in a combined production-transportation work cell. *Journal of the Operational Research Society*, **41**, 625-632.

## 9 BIOGRAPHY

Jinliang Cheng received M.E. from Wuhan University of Technology, China, and Dr. of Engineering from Kyoto Institute of Technology, Japan. He is now with Applied Technology Corporation, Japan. He is interested in production scheduling, modeling and analysis of manufacturing systems.

Hiroshi Kise is Professor at Department of Mechanical and System Engineering, Kyoto Institute of Technology. He received M.E. from Kyoto Institute of Technology and Dr. of Engineering from Kyoto University. His research interests include deterministic and stochastic scheduling, and production system modeling and analysis.