Journal of the Operations Research
Society of Japan
Vol. 40, No. 3, September 1997

OPTIMAL SCHEDULING FOR AN AUTOMATED m-MACHINE FLOWSHOP

Jin-Liang Cheng Hiroshi Kise Yoshiyuki Karuno
Kyoto Institute of Technology

(Received April 1, 1996; Revised November 18, 1996)

Abstract This paper considers a scheduling problem of minimizing the maximum completion time (i.e., the makespan) for an automated flowshop manufacturing system such as FMS which consists of m 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 in very short time, and the branch-and-bound algorithm can efficiently give optimal solutions to problem instances with three-machines and up to 400 parts with high probability over 90%.

1. Introduction

Many flexible manufacturing systems (FMS's) have been implemented 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 [2]. On the other hand, the efficient implementation of such FMS's presents a complex set of issues to be solved from both long- and short-term perspectives and a hierarchical framework for these intractable issues has been proposed [23]. 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.

Quite a few analytical studies on the optimal scheduling of n parts for automated two- or more machine flowshops have been reported. Kise et al. [14] have considered two-machine, one-AGV flowshop scheduling problems without Work-in-process (WIP) buffer, and given $O(n^3)$ time algorithms based on the well known Gilmore-Gomory algorithm for finding optimal schedules. Kise [15] has also considered a two-machine, one-AGV (or a robot) scheduling problem with sufficient WIP buffer at each machine and shown that the problem is a special case of the classical three machine flowshop scheduling problem (denoted 3-FSP), and NP-hard. Kise et al. [16] have proposed a branch-and-bound algorithm for the same problem, and demonstrated by numerical experiments that the algorithm can exactly solve problem instances with up to 200 parts with high possibility in reasonable running time. Panwalker [20] has studied a two-machine, one-robot flowshop scheduling problem without WIP buffer at the first machine, but sufficient WIP buffer at the second machine, and given a quadratic time optimal algorithm. Levner et al. [19] have considered a two-machine robotic cell in the same configuration as that in Panwalker's except the non-negligible loading/unloading operation times are allowed, and given an $O(n \log n)$ time optimal algorithm. Cheng and Kise [4] have given optimal scheduling algorithms for automated two-machine manufacturing systems with intermediate operations. Stern and Vitner [24] have dealed 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. Sethi et al. [22]

have studied a problem of sequencing parts and robot moves in a flow line manufacturing system without WIP buffer at each machine. They have developed a cycle time formulas and obtained necessary and sufficient conditions for various robot cycles to be optimal in two and three machine cells for producing a single part type. They have shown that the problem of scheduling different kinds of parts for a specific sequence of robot moves in a two-machine cell can be formulated as a solvable case of traveling salesman problem. Hall et al. [7] have also studied robot move and part sequencing problems for the same system as Sethi et al.'s. They have provided an efficient algorithm that simultaneously optimizes the robot move and part sequencing for multiple part type problems in a two machine cell. For a three machine cell producing multiple part types, they have proved that four out of the six potentially optimal robot move cycles for producing one unit allow efficient identification of the optimal part sequence.

Kats [12] has considered a problem of cyclic no-wait scheduling of identical parts on several sequential machines in a production line when the transportation of the parts between the machines is performed by a number of identical robots. He has found the minimal number of robots needed to meet a given schedule for all possible cycle lengths and given an $O(m^6)$ time optimal algorithm. Hitz [8] studied the input sequence problem of a dedicated flowshop with periodic demand. An optimal off-line scheme and a heuristic were developed.

There are a number of studies on the scheduling of parts and AGV's in general FMS enviroments using simulation. Kimemia and Gershwin [13] have used a simulation model to evaluate an off-line scheduling algorithm for the system similar to Hitz's except that routing flexibility was allowed. Their systems include four machines and two part types, and they have considered machine breakdown and in-system storage capacity. Above studies, however, have not considered the impact of the material handling system on the FMS scheduling problem. Sabuncuoglu and Hommertzheim [21] 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 [11] 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.

In this paper we deal with an automated flowshop manufacturing system such as FMS that consists of a loading station, m machining cells with unlimited WIP buffer, an unloading station and an AGV (or a moving robot) that sends at most one part at a time, and discuss an optimal scheduling problem that asks to minimize the makespan (i.e., the maximum completion time) of the n parts to be processed by the system. We can find many FMS's, FMC's or FTL's in real situations that can be modeled by this system (e.g., see [1]). The above review, however, shows that there have so far been few algorithms that can efficiently give exact optimal solutions to such systems with 3 or more cells, due to the barrier of the NP-hardness. This paper aims to develop a branch-and-bound (BAB) algorithm that could efficiently solve large problem instances with high probability. The configuration of this paper is as follows.

Section 2 describes the problem in detail. Section 3 formulates the problem exactly. Section 4 shows that the problem can approximately be reduced to an $(m+1)$-machine flowshop problem, and proposes a heuristic algorithm (called fuzzy scheduling) based on the $(m+1)$-machine flowshop problem. Section 5 proposes a branch-and-bound algorithm utilizing fuzzy inference for solving the problem exactly. Section 6 provides computational experiences.

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, m (machining) cells (e.g., machining centers) with pallet storage carousel, an AGV 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:

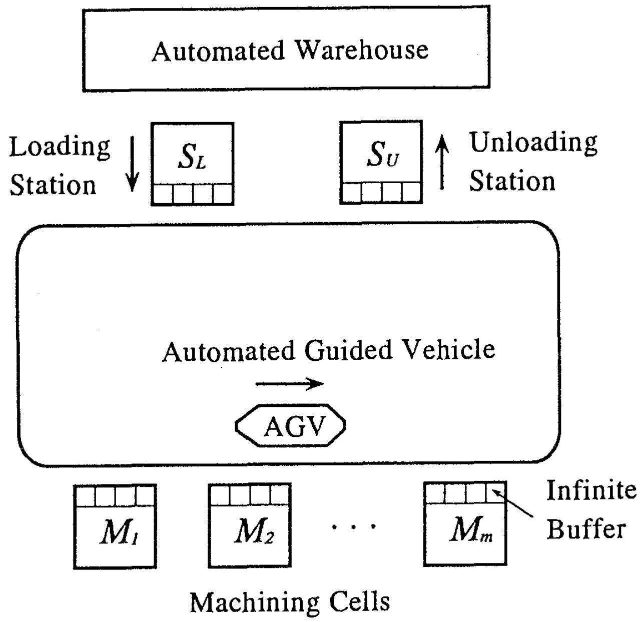

Figure 1: Shop configuration

(1) A set of n parts, $J = \{i | i = 1, 2, \cdots, n\}$, is available at time 0.
(2) Each of m cells, M_l ($l = 1, 2, \cdots, m$), 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 $M_1, M_2, \cdots, M_m$ and the sequences of processing the parts at cells M_l ($l = 1, 2, \cdots, m$) are the same.
(4) Loading and unloading stations, and m cells have buffers (pallet storage carousels) for pre- and/or post-process inventory. 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 ($l = 1, 2, \cdots, m$) 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 processed on the next cell. The AGV leaves each cell M_l ($l = 1, 2, \cdots, m$) without waiting there for M_l to finish a part if there is no part in the buffer.
(7) The processing time of part i on M_l, including set-up time, are known and represented by $P_l(i)$ ($i \in J,\ l = 1, 2, \cdots, m$).
(8) The times required for the AGV to carry a part from S_L to M_1, from M_1 to $M_2, \cdots$, from M_m to S_U and from S_U to S_L, including pickup and drop times, are known and represented by $t_{01}, t_{12}, \cdots, t_{m,m+1}, t_{m+1,0}$, respectively. Such times are independent of the parts to be carried, thus, the time required for the AGV to travel a complete loop, $t_v = t_{01} + t_{12} + \cdots + t_{m,m+1} + t_{m+1,0}$, 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 when the first part is released from S_L and the time when the last part is delivered to S_U.

The set-up time mentioned under assumption (7) may actually be dependent on the processing sequence. However, the variance in set-up times is insignificant relative to the processing times when the material handling is implemented by automated equipments such as automated pallet changers (APC's), automated tool changers (ATC's), and/or robot hands.

Hereafter, this problem will be called the automated flowshop scheduling problem (denoted AFSP).

3 Formulation of Scheduling Problem

The time when M_l finishes processing part i is represented by $F_l(i)$, and the time when AGV picks up part i from M_l by $T_l(i)$. Furthermore, the time when part i is released from S_L and the time when part i is delivered to 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, $s = (j_1, j_2, \cdots, j_n)$, can be formulated as follows.

By assumption (6), it can be seen that

$$T_0(j_k) = (k-1)t_v, \quad k = 1, 2, \cdots, n. \tag{1}$$

From assumptions (1)~(8), M_l can start processing part j_k only after it has finished the processing part j_{k-1} and part j_k has been transferred to M_l by the AGV, so the time $F_l(j_k)$ when M_l finishes processing part j_k is equivalent to the sum of the processing time $P_l(j_k)$ and the maximum between the time $F_l(j_{k-1})$ when M_l finishes processing part j_{k-1} and the arrival time $T_{l-1}(j_k) + t_{l-1,l}$ of part j_k at M_l, i.e.,

$$F_l(j_k) = \max\{F_l(j_{k-1}), T_{l-1}(j_k) + t_{l-1,l}\} + P_l(j_k), \quad l = 1, 2, \cdots, m; k = 1, 2, \cdots, n, \tag{2}$$

where $F_l(j_0) = 0$ $(l = 1, 2, \cdots, m)$.

Picking up part j_k from M_l must be after M_l finishes processing j_k, then

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

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

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

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

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

where $T_l(j_0) = 0 (l = 1, 2, \cdots, m)$, then from Eqs.(3)~(5) we have

$$T_l(j_k) = \max\{\lceil (F_l(j_k) - t_{0l})/t_v \rceil t_v + t_{0l}, T_l(j_{k-1}) + t_v\}, \quad l = 1, 2, \cdots, m; k = 1, 2, \cdots, n. \tag{6}$$

The time when part j_k arrives at S_U is

$$F(j_k) = T_m(j_k) + t_{m,m+1}, \quad k = 1, 2, \cdots, n. \tag{7}$$

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

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

Hereafter, the optimal sequence that minimizes $F_{max}(s)$ is represented by s^*.

4 Fuzzy Scheduling

Problem AFSP is NP-hard, even for the case of $m = 2$ [15]. Thus, we need good heuristic algorithm for practical purposes. However, the above formulation that expresses makespan F_{max} through recursive equations (1), (2), (6), (7) and (8) makes us somewhat difficult even to have an insight for a good heuristic.

In order to overcome this difficulty, we here take the following approach. We firstly show that the AFSP approximately reduces to a classical flowshop scheduling problem (denoted FSP) that has no AGV. This approximation tell us that the fuzzy scheduling method that has already been developed, and demonstrated to be a good heuristic for the FSP [5], could also be good for our AFSP. That is, the sequence for the FSP obtained by the fuzzy heuristic can also be used as a sequence for the AFSP to obtain a good schedule. Later, this schedule will, furthermore, be improved by a branch-and-bound algorithm.

4.1 An approximation of makespan

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

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

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

Let $\hat{F}_l(j_k)$ be the time when M_l finishes part j_k, $\hat{F}(j_k)$ be the time when part j_k arrives at unloading station 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 the following.

Lemma 1. For a given sequence $s = (j_1, j_2, \cdots, j_n)$,

$$\hat{F}_1(j_k) = \max_{1 \le p \le k} \{(p-1)t_v + \sum_{h=p}^{k} P_1(j_h)\} + t_{01}, \tag{9}$$

$$\hat{F}_l(j_k) \le \max_{1 \le p \le k} \{\hat{F}_{l-1}(j_p) + \sum_{h=p}^{k} P_l(j_h)\} + t_v + t_{l-1,l}, \quad l = 2, 3, \cdots, m, \tag{10}$$

$$\hat{T}_m(j_k) \le \hat{F}_m(j_k) + t_v, \qquad k = 1, 2, \cdots, n. \tag{11}$$

Proof. See appendix A.

Lemma 2. For a given sequence $s = (j_1, j_2, \cdots, j_n)$, the following upper bound $UB(s)$ of $\hat{F}_{max}(s)$ can be obtained.

$$\hat{F}_{max}(s) \le UB(s) = C(s) + (m-1)t_v + t_{0,m+1} \tag{12}$$

where

$$C(s) = \max_{1 \le q(0) \le \cdots \le q(m-1) \le n} \{\sum_{h=1}^{q(0)} t_v + \sum_{h=q(0)}^{q(1)} P_1(j_h) + \cdots + \sum_{h=q(m-1)}^{n} P_m(j_h)\}, \tag{13}$$

and $q(0), q(1), \cdots, q(m-1)$ are integers that satisfy $1 \le q(0) \le q(1) \le \cdots \le q(m-1) \le n$.

Proof. See appendix B.

Since t_v and $t_{0,m+1}$ are independent of sequence s, Lemma 2 means that minimizing $UB(s)$ is equivalent to minimizing $C(s)$. $C(s)$ has the same expression as the makespan of a classical flowshop scheduling problem that has no AGV, but $(m+1)$ machines with processing times $t_v, P_1(i), \cdots, P_m(i)$ for part i. That is, the original AFSP can be approximately reduced to the classical $(m+1)$-machine flowshop scheduling problem (denoted $(m+1)$-FSP).

4.2 Fuzzy scheduling method

The fuzzy scheduling method has been proposed to yield nearly optimal solutions for an m-machine FSP [5]. 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 $(m+1)$-FSP of Eq.(13), let flow time of part j_k on the l-th machine $(l = 2, \cdots, m+1)$ for a partial sequence of the first k parts, $s_k = (j_1, \cdots, j_k)$, be defined by

$$FT_l(s_k) = FT_l(j_k) - FT_1(j_k), \tag{14}$$

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 $(m+1)$-FSP. Similarly flowtimes $FT_l(s_k, i)$ 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. [5] Assume that two parts i and j are optimally processed immediately after partial sequence s_k, part i optimally precedes part j if

$$FT_l(s_k, i, j) \leq FT_l(s_k, j, i), \quad l = 2, 3, \cdots, m+1. \tag{15}$$

It is rare that precondition Eq.(15) simultaneously holds for all l's, but it is a sufficient condition for an optimal schedule, suggesting that if Eq.(15) 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 context of fuzzy inference [25]. That is, let

$$D_l(s_k, i, j) = FT_l(s_k, i, j) - FT_l(s_k, j, i), \quad l = 2, 3, \cdots, m+1, \tag{16}$$

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

$$\mu_{s_k}(i, j) = 0.5 - \frac{D(s_k, i, j)}{2D_{max}(s_k)}, \tag{17}$$

where $D(s_k, i, j) = \sum_{l=2}^{m+1} \alpha_{l-1} D_l(s_k, i, j)$, $D_{max}(s_k) = \max_{i,j} |D(s_k, i, j)|$ and $\alpha_1, \cdots, \alpha_m$ $(0 \leq \alpha_1, \cdots, \alpha_m \leq 1$ and $\sum_{l=1}^{m} \alpha_l = 1)$ are real numbers to be appropriately determined (see Section 6). Then, the degree of dominance of part i over the remaining parts under partial sequence s_k $(k = 0, 1, \cdots, n-1)$ is given by

$$\mu^*_{s_k}(i) = \min_{j \in J_r} \mu_{s_k}(i, j), \tag{18}$$

and part i^* satisfying

$$\mu^*_{s_k}(i^*) = \max_{i \in J_r} \mu^*_{s_k}(i) \tag{19}$$

is then 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 based on the fuzzy rule is referred to as fuzzy scheduling. We use a sequence obtained by applying the fuzzy scheduling to the FSP as that of the AFSP, and then compute its makespan (by Eq.(6)), that will be used as an initial upper bound value of the BAB algorithm proposed next for the original AFSP.

5 BAB Algorithm

It is assumed that the basic principle of BAB algorithm is well known (e.g., see [9, 10]). Hence only the basic components of BAB algorithm are stated below.

5.1 Subproblem

Let the sequence of the first k parts fixed be $s_k = (j_1, \cdots, j_k)$. 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

Lemma 3. For a given sequence of the first k parts, s_k (for $\forall k = 0, 1, \cdots, n-1$), and an arbitrary sequence of the remaining parts, $\overline{s}_k = (j_{k+1}, \cdots, j_n)$,

$$\begin{aligned} T_1(j_n) &\geq \max\{F_1(j_k) + Y_{11}(\overline{s}_k), (k-1)t_v + Y_{01}(\overline{s}_k) + t_{01}\}, \\ T_2(j_n) &\geq \max\{F_2(j_k) + Y_{22}(\overline{s}_k), F_1(j_k) + Y_{12}(\overline{s}_k) + t_{12}, (k-1)t_v + Y_{02}(\overline{s}_k) + t_{02}\}, \\ &\cdots\cdots, \\ T_m(j_n) &\geq \max\{F_m(j_k) + Y_{mm}(\overline{s}_k), F_{m-1}(j_k) + Y_{m-1,m}(\overline{s}_k) + t_{m-1,m}, \cdots, \\ &\qquad F_1(j_k) + Y_{1m}(\overline{s}_k) + t_{1m}, (k-1)t_v + Y_{0m}(\overline{s}_k) + t_{0m}\}, \end{aligned} \tag{20}$$

where

$$Y_{uv}(\overline{s}_k) = \max_{k<q(u)\leq\cdots\leq q(v-1)\leq n} [\sum_{h=k+1}^{q(u)} P_u(j_h) + \cdots + \sum_{h=q(v-1)}^{n} P_v(j_h)], \quad 0 \leq u \leq v \leq m, \tag{21}$$

and $P_0(j_h) = t_v (h = 1, \cdots, n)$, where $q(u), \cdots, q(v-1)$ are integers which satisfy $k < q(u) \leq \cdots \leq q(v-1) \leq n$.

Proof. See appendix C.

Now we consider the minimization of each $Y_{uv}(\overline{s}_k)$ of Lemma 3 that yields a lower bound of $T_v(j_n)$ of Eq.(20), and leads to the one of the makespan $F(j_n)$ of Eq.(8). The minimization of $Y_{uv}(\overline{s}_k)$ 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 $Y^b_{uv}(\overline{s}_k)$ of $Y_{uv}(\overline{s}_k)$ instead of $Y_{uv}(\overline{s}_k)$ itself, which is obtained by only considering the cases of $k < q(u) = q(u+1) = \cdots = q(v-1) \leq n$ in Eq.(21), i.e.,

$$\begin{aligned} Y_{uv}(\overline{s}_k) &\geq Y^b_{uv}(\overline{s}_k) \\ &= \max_{k<q(v-1)\leq n} [\sum_{h=k+1}^{q(v-1)} P_u(j_h) + \sum_{l=u+1}^{v-1} P_l(j_{q(v-1)}) + \sum_{h=q(v-1)}^{n} P_v(j_h)] \end{aligned} \tag{22}$$

The minimization of $Y^b_{uv}(\overline{s}_k)$ is reduced to a special 3-machine FSP 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 [18]. Furthermore,

$$\begin{aligned} Y^b_{uv}(\overline{s}_k) &= \max_{k<q\leq n} [\sum_{h=k+1}^{q} (\sum_{l=u}^{v-1} P_l(j_h)) + \sum_{h=q}^{n} (\sum_{l=u+1}^{v} P_l(j_h))] - \sum_{l=u+1}^{v-1} Y_{ll}(\overline{s}_k) \\ &= Z_{uv}(\overline{s}_k) - \sum_{l=u+1}^{v-1} Y_{ll}(\overline{s}_k), \quad 1 \leq u < v \leq m \end{aligned} \tag{23}$$

and the problem of minimizing $Z_{uv}(\overline{s}_k)$ over all $\overline{s}_k$ is reduced to a solvable 2-machine FSP with processing times $a(i) = \sum_{l=u}^{v-1} P_l(i)$ 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 the part j_{k+1} on machine M_u can not be started before

$$F^*_u(j_k) = \max_{1\le l\le u}[F_l(j_k) + \min_{k<h\le n}\sum_{o=l}^{u-1} A_o(j_h) + t_{lu}], \tag{24}$$

where $A_l(i) = \max\{P_l(i), t_v\}, 1 \le l \le m$. Also $T_v(j_n)$ must be dependent on t_v and the number of times the AGV goes around the loop track till $T_v(j_n)$ (see Eq.(3)), then it can be easily seen by Eq.(20) through Eq.(23) that

$$g^b_{uv}(s_k) = \lceil F^*_u(j_k) + Z^*_{uv}(\bar{s}_k) - \sum_{l=u+1}^{v-1} Y_{ll}(\bar{s}_k) + t_{uv} - t_{0v})/t_v\rceil t_v + t_{0v} \tag{25}$$

is a lower bound of $T_v(j_n)$ for subproblem $P(s_k)$. Then,

$$g_{uv}(s_k) = \lceil (g^b_{uv}(s_k) + \min_{k<h\le n}[\sum_{o=v+1}^{m} A_o(j_h)] + t_{vm} - t_{0m})/t_v\rceil t_v + t_{0,m+1} \tag{26}$$

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

We employ

$$g(s_k) = \max_{1\le u<v\le m} g_{uv}(s_k) \tag{27}$$

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 $O(m^2n)$ time and $g(\emptyset)$, the lower bound of the original problem in $O(m^2n\log n)$ time.

5.3 Fuzzy scheduling and fuzzy search

As mentioned before, we use the fuzzy scheduling for obtaining an initial incumbent solution for the BAB that plays a role of an upper bound of optimal value. We adopt a depth-first search for the BAB that selects a subproblem with the smallest lower bound among the most recently generated ones, breaking ties by the fuzzy rule described in Section 4.2. We call such search method fuzzy search.

6 Numerical Experiments

In this section, performances of the BAB algorithm with fuzzy inference proposed here are evaluated by the way of numerical experiments. For each n (the number of parts), m (the number of cells) and t_v (the time required for AGV to travel a complete loop), 30 problem instances have been tested. The processing times $P_l(j)$ of the part $j \in J$ on M_l $(l = 1, 2, \cdots, m)$ are given by uniformly distributed random integers from 1 to 100, inclusive. For weights α_l $(l = 1, \cdots, m)$ of membership function $\mu_s(i, j)$ in Section 4.2, three types of function were applied to each problem instance. They are: (1) arithmetical progression weights, $\alpha_l = 2l/m(m+1)$; (2) equal weights, $\alpha_l = 1/m$; (3) inverse arithmetical progression weights, $\alpha_l = 2(m-l+1)/m(m+1)$ $(l = 1, \cdots, m)$ (Note that $\sum_{l=1}^m \alpha_l = 1$ for all types). The best one of three kinds of schedule obtained is adopted as the initial solution in the BAB algorithm. In the fuzzy search of the BAB algorithm, only the arithmetical progression weights are used.

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

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

	m=3		m=4		m=5	
$t_v/(m+2)/P_{mean}$	R_{FS}	R_{FC}	R_{FS}	R_{FC}	R_{FS}	R_{FC}
0.02	1.0	16.8	1.9	19.0	4.1	17.3
0.04	1.2	16.6	2.3	18.9	4.4	16.2
0.06	1.6	16.9	2.7	18.4	4.2	16.0
0.08	1.9	15.9	3.1	17.8	4.8	14.5
0.10	2.9	15.9	3.5	17.8	6.3	14.7
Average	1.7	16.4	2.7	18.4	4.8	15.7

Table 2: Average relative errors (%) of fuzzy scheduling for $m = 3$ and $t_v = 15$ ($t_v/(m+2)/P_{mean} = 0.06$)

n	10	20	30	40	50	100
R_{FS}	1.6	0.8	0.4	0.4	0.3	0.1
	(<0.005)*	(0.01)	(0.02)	(0.03)	(0.06)	(0.54)
R_{FC}	16.7	12.1	10.8	8.3	7.9	5.4

*: Average CPU times in second.

6.1 Performance of fuzzy scheduling

The fuzzy scheduling presented above has been tested on a total of 630 problem instances with $n = 10$, $m = 3, 4, 5$, $t_v = (1 \sim 5)(m+2)$ (equivalently, $t_v/(m+2)/P_{mean} = 0.02 \sim 0.10$, where $P_{mean} = 50$ is mean processing time) and $n = 10, 20, 30, 40, 50, 100$, $m = 3$, $t_v = 15$ ($t_v/(m+2)/P_{mean} = 0.06$), all of which were exactly solved by the BAB algorithm.

Tables 1~3 show results obtained, where R_{FS} and R_{FC} represent average relative errors of the fuzzy scheduling and the FCFS (First come first served) scheduling, respectively. The comparison of the fuzzy scheduling with the FCFS scheduling in Tables 1 and 2 shows the effectiveness 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. As a result, solutions by the fuzzy scheduling are closer to optimal ones as n becomes larger. Table 3 shows how many problem instances can be optimally solved by the fuzzy scheduling.

We can conclude from these results that the fuzzy scheduling is superior to the FCFS scheduling especially for problem instances with $m = 3$.

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 for the purpose of comparison.

(1) A: BAB algorithm with fuzzy inference proposed here;

(2) A_1: BAB algorithm A without fuzzy search (i.e., the ordinary depth-first search method is adopted);

(3) A_2: BAB algorithm A without fuzzy scheduling (i.e., the initial incumbent value is set to ∞);

(4) A_3: BAB algorithm A_2 without fuzzy search (i.e., the one without fuzzy inference).

The rates of problem instances solved by these algorithms are shown in Table 4 for $m = 3$

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

$t_v/(m+2)/P_{mean}$	$m=3$	$m=4$	$m=5$
0.02	53	27	7
0.04	47	30	13
0.06	50	27	13
0.08	50	23	10
0.10	40	33	3
Average	48	28	9

and $t_v = 15$. It is evident that Algorithm A is superior to A_1, A_2 and A_3. The comparisons of A with A_1 as well A_3 show the effectiveness of the fuzzy search on the performance of Algorithm A, and the comparisons of A with A_2 as well A_3 show the effectiveness of the fuzzy scheduling on the performance of Algorithm A.

Tables 5 and 6 show the influences of the turnaround time of AGV, the number of parts and the number of cells on the solvability of Algorithm A. We can conclude from results in these tables that the BAB algorithm A can solve problem instances with 400 parts with high probability over 90% if the number of cells, m is restricted to 3.

7 Conclusion

In this paper we have considered a scheduling problem of minimizing the makespan for an automated m-machine flowshop such as FMS that consists of a few machining cells with sufficient buffers, an AGV and loading/unloading stations, and shown that the problem can be approximately reduced to a flowshop scheduling problem. Based on this reduction we proposed a heuristic algorithm called fuzzy scheduling, and a branch-and-bound algorithm with fuzzy inference. Extensive numerical experiments show that the fuzzy scheduling can give optimal or near optimal solutions in short time, and the branch-and-bound algorithm with fuzzy inference can efficiently give optimal solutions to problem instances with three-machines and up to 400 parts with high probability. These facts suggest that approximate and/or exact algorithms proposed here can successfully be applied to real manufacturing systems such as FMS's, FTL's and especially FMC's with a few machining centers.

Other possible topics for future research include the use of multiple AGV's in automated flowshops as a means of reducing the waiting time of parts and the idling time of machining cells. The scheduling of automated flowshops with limited buffer storage and the design of polynomial time approximation algorithms for such intractable problems are also important.

Acknowledgment

The authors would like to thank referees for their constructive comments and valuable suggestions for improving the earlier version of the paper. They would also like to thank Mr. Mikito Nakai who executed a part of numerical experiment in Section 6 as his undergraduate research. This study is partially supported by the Ministry of Education, Science and Culture of Japan under Science Research Grant-in-Aid.

Appendix A. Proof of Lemma 1

Proof. From Eqs.(1), (2) and (6)'

$$\hat{F}_1(j_k) = \max\{T_0(j_k) + t_{01}, \hat{F}_1(j_{k-1})\} + P_1(j_k),$$

Table 4: Rate (%) of problem instances solved within 5 minutes by BAB algorithms ($m = 3$, $t_v = 15$ $(t_v/(m+2)/P_{mean} = 0.06)$)

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: Average rates (%) of problem instances solved within 5 minutes by Algorithm A for $n = 10 \sim 200$

$t_v/(m+2)/P_{mean}$	$m = 3$	$m = 4$
0.02	86	61
0.04	88	75
0.06	92	77
0.08	85	69
0.10	74	56

Table 6: Rate (%) of problem instances solved within 5 minutes by Algorithm A for larger problems($m = 3$, $t_v = 15$ $(t_v/(m+2)/P_{mean} = 0.06)$)

n	A	n	A
210-220	90 (28)*	310-320	83 (55)
230-240	88 (30)	330-340	100 (66)
250-260	92 (33)	350-360	92 (72)
270-280	92 (53)	370-380	97 (81)
290-300	90 (37)	390-400	83 (91)
		Average	91

*:Average CPU times in second for solved instances.

$$= \max\{(k-1)t_v + P_1(j_k) + t_{01}, \hat{F}_1(j_{k-1}) + P_1(j_k)\}. \quad (28)$$

Since we have the same equation as Eq.(28) for k replaced by $(k-1)$, we have the following by substituting such equation into $\hat{F}_1(j_{k-1})$ of the right-hand side of Eq.(28),

$$\hat{F}_1(j_k) = \max\{(k-1)t_v + P_1(j_k) + t_{01}, (k-2)t_v + P_1(j_{k-1}) + P_1(j_k), \hat{F}_1(j_{k-2}) + P_1(j_{k-1}) + P_1(j_k)\}.$$

Repeating the above substitution for $k-2, \cdots, 1$, we have

$$\hat{F}_1(j_k) = \max_{1 \le p \le k}\{(p-1)t_v + \sum_{h=p}^{k} P_1(j_h)\} + t_{01}. \quad (29)$$

Replacing $\lceil x \rceil$ of Eq.(6)' by $(x+1)$, we have

$$\hat{T}_l(j_k) \le \hat{F}_l(j_k) + t_v, \quad l = 1, 2, \cdots, m. \quad (30)$$

Subsituting Eq.(30) into Eq.(2), we have

$$\hat{F}_l(j_k) \le \max\{\hat{F}_{l-1}(j_k) + P_l(j_k) + t_v + t_{l-1,l}, \hat{F}_l(j_{k-1}) + P_l(j_k)\}. \quad (31)$$

Since we have the same equation as Eq.(31) for k replaced by $(k-1)$, we have the following by substituting such equation into $\hat{F}_l(j_{k-1})$ of the right-hand side of Eq.(31),

$$\hat{F}_l(j_k) \le \max\{\hat{F}_{l-1}(j_k) + P_l(j_k) + t_v + t_{l-1,l}, \hat{F}_{l-1}(j_{k-1}) + P_l(j_{k-1}) + P_l(j_k) + t_v + t_{l-1,l}, \hat{F}_l(j_{k-2}) + P_l(j_{k-1}) + P_l(j_k). \quad (32)$$

Repeating the above substitution for $k-2, \cdots, 1$, we have

$$\hat{F}_l(j_k) \le \max_{1 \le p \le k}\{\hat{F}_{l-1}(j_p) + \sum_{h=p}^{k} P_l(j_h)\} + t_v + t_{l-1,l}, \quad l = 2, 3, \cdots, m. \quad (33)$$

Thus, from Eqs.(29), (33) and (30) we have Eqs.(9), (10) and (11).

Appendix B. Proof of Lemma 2

Proof. From Eqs.(10) and (11)

$$\begin{aligned}\hat{F}_{max}(s) &= \hat{T}_m(j_n) + t_{m,m+1} \\ &\le \hat{F}_m(j_n) + t_v + t_{m,m+1} \\ &\le \max_{1 \le q(m-1) \le n}\{\hat{F}_{m-1}(j_{q(m-1)}) + \sum_{h=q(m-1)}^{n} P_m(j_h)\} + 2t_v + t_{m-1,m+1}. \quad (34)\end{aligned}$$

Substituting Eqs.(9) and (10) into Eq.(34) for $l = m-1, \cdots, 1$ repeatedly, we have

$$\hat{F}_{max}(s) \le \max_{1 \le q(m-1) \le n}\{\max_{1 \le q(m-2) \le q(m-1)}\{\hat{F}_{m-2}(j_{q(m-2)}) + \sum_{h=q(m-2)}^{q(m-1)} P_{m-1}(j_h)\} + \sum_{h=q(m-1)}^{n} P_m(j_h)\} + 3t_v + t_{m-2,m+1}$$

$$= \max_{1\le q(m-2)\le q(m-1)\le n}\{\hat{F}_{m-2}(j_{q(m-2)}) + \sum_{h=q(m-2)}^{q(m-1)} P_{m-1}(j_h)$$

$$+ \sum_{h=q(m-1)}^{n} P_m(j_h)\} + 3t_v + t_{m-2,m+1}$$

$\cdots\cdots,$

$$\le \max_{1\le q(0)\le\cdots\le q(m-1)\le n}\{\sum_{h=1}^{q(0)} t_v + \sum_{h=q(0)}^{q(1)} P_1(j_h)$$

$$+\cdots+ \sum_{h=q(m-1)}^{n} P_m(j_h)\} + (m-1)t_v + t_{0,m+1}.$$

Thus, we have Eq.(12) and Eq.(13).

Appendix C: Proof of Lemma 3

Proof. The proof is done by induction on $l = 1, 2, \cdots, m$. Firstly, replacing $\lceil x \rceil$ by x and removing the term $T_l(j_{k-1}) + t_v$ in Eq.(6), we have the following,

$$\begin{aligned} T_l(j_k) &\ge F_l(j_k) \\ &= \max\{F_l(j_{k-1}), T_{l-1}(j_k) + t_{l-1,l}\} + P_l(j_k), \quad l = 1, \cdots, m; k = 1, \cdots, n. \end{aligned} \tag{35}$$

Now from Eq.(35) for $l = 1$ and $k = n-1$, and Eq.(1)

$$\begin{aligned} T_1(j_n) &\ge \max\{F_1(j_{n-1}), T_0(j_n) + t_{01}\} + P_1(j_n) \\ &= \max\{F_1(j_{n-1}), (n-1)t_v + t_{01}\} + P_1(j_n) \\ &= \max\{F_1(j_{n-1}) + P_1(j_n), (n-2)t_v + t_v + P_1(j_n) + t_{01}\}. \end{aligned} \tag{36}$$

From Eq.(21)

$$Y_{11}(\bar{s}_{n-1}) = P_1(j_n) \tag{37}$$

and

$$\begin{aligned} Y_{01}(\bar{s}_{n-1}) &= P_0(j_n) + P_1(j_n) \\ &= t_v + P_1(j_n). \end{aligned} \tag{38}$$

Thus, from Eqs.(36)$\sim$(38) we have

$$T_1(j_n) \ge \max\{F_1(j_{n-1}) + Y_{11}(\bar{s}_{n-1}), (n-2)t_v + Y_{01}(\bar{s}_{n-1}) + t_{01}\}. \tag{39}$$

In general, we have the following from Eq.(21)

$$Y_{11}(\bar{s}_k) = \sum_{h=k+1}^{n} P_1(j_h), \quad k = 0, 1, \cdots, n-1. \tag{40}$$

Eq.(40) can be rewritten as follows,

$$Y_{11}(\bar{s}_k) = P_1(j_{k+1}) + \sum_{h=k+2}^{n} P_1(j_h), \quad k = 0, 1, \cdots, n-1. \tag{41}$$

Since we have the same equation as Eq.(40) for k replaced by $(k+1)$, we have the following by replacing the right-hand side of Eq.(41) by such equation,

$$Y_{11}(\bar{s}_k) = P_1(j_{k+1}) + Y_{11}(\bar{s}_{k+1}), \quad k = 0, 1, \cdots, n-1. \tag{42}$$

Also from Eq.(21)

$$Y_{0,1}(\bar{s}_k) = \max_{k<q(0)\leq n}\{\sum_{h=k+1}^{q(0)} P_0(j_h) + \sum_{h=q(0)}^{n} P_1(j_h)\}$$
$$= \max_{k<q(0)\leq n}\{\sum_{h=k+1}^{q(0)} t_v + \sum_{h=q(0)}^{n} P_1(j_h)\}. \tag{43}$$

Eq.(43) can be rewritten as follows,

$$Y_{0,1}(\hat{s}_k) = \max\{t_v + P_1(j_{k+1}) + \sum_{h=k+2}^{n} P_1(j_n),$$
$$t_v + \max_{k+1<q(0)\leq n}[\sum_{h=k+2}^{q(0)} t_v + \sum_{h=q(0)}^{n} P_1(j_h)]\}. \tag{44}$$

Since we have the same equations as Eqs.(40) and (44) for k replaced by $(k+1)$, we have the following by replacing the right-hand side of Eq.(44) by such equations,

$$Y_{0,1}(\bar{s}_k) = \max\{t_v + P_1(j_{k+1}) + Y_{11}(\bar{s}_{k+1}), t_v + Y_{0,1}(\bar{s}_{k+1})\}. \tag{45}$$

Substituting Eq.(2) for $l=1$ and $k=n-1$ into $F_1(j_{n-1})$ of the right-hand side of Eq.(39), we have

$$\begin{aligned} T_1(j_n) \geq \max\{&F_1(j_{n-2}) + P_1(j_{n-1}) + Y_{11}(\bar{s}_{n-1}),\\ &T_0(j_{n-1}) + P_1(j_{n-1}) + Y_{11}(\bar{s}_{n-1} + t_{01}),\\ &(n-2)t_v + Y_{01}(\bar{s}_{n-1}) + t_{01}\}\\ = \max\{&F_1(j_{n-2}) + P_1(j_{n-1}) + Y_{11}(\bar{s}_{n-1}), (n-3)t_v\\ &+ \max[t_v + P_1(j_{n-1}) + Y_{11}(\bar{s}_{n-1}), t_v + Y_{01}(\bar{s}_{n-1})] + t_{01}\}. \end{aligned} \tag{46}$$

Replacing the right-hand side of Eq.(46) by Eqs.(42) and (45) for $k = n-2$, we have

$$T_1(j_n) \geq \max\{F_1(j_{n-2}) + Y_{11}(\bar{s}_{n-2}), (n-3)t_v + Y_{01}(\bar{s}_{n-2}) + t_{01}\}. \tag{47}$$

Repeating the above substitution and replacement for $n-3, n-4, \cdots, k$, we have

$$T_1(j_n) \geq \max\{F_1(j_k) + Y_{11}(\hat{s}_k), (k-1)t_v + Y_{01}(\hat{s}_k) + t_{01}\}. \tag{48}$$

Now assuming that Lemma3 (Eq.(20)) holds for $l \leq m-1$, i.e.,

$$\begin{aligned} T_l(j_n) \geq \max\{&F_l(j_k) + Y_{ll}(\bar{s}_k), F_{l-1}(j_k) + Y_{l-1,l}(\bar{s}_k) + t_{l-1,l},\\ &\cdots, F_1(j_k) + Y_{1l}(\bar{s}_k) + t_{1m}, (k-1)t_v + Y_{0l}(\bar{s}_k) + t_{0l}\}, \quad l = 2,3,\cdots,m-1. \end{aligned} \tag{49}$$

From Eq.(35) we have

$$T_m(j_n) \geq \max\{F_m(j_{n-1}), T_{m-1}(j_n) + t_{m-1,m}\} + P_m(j_n). \tag{50}$$

Substituting Eq.(49) for $l = m-1$ into $T_{m-1}(j_n)$ of the right-hand side of Eq.(50), we have

$$\begin{aligned} T_m(j_n) \geq \max\{&F_m(j_{n-1}) + P_m(j_n),\\ &F_{m-1}(j_k) + Y_{m-1,m-1}(\bar{s}_k) + P_m(j_n) + t_{m-1,m},\\ &F_{m-2}(j_k) + Y_{m-2,m-1}(\bar{s}_k) + P_m(j_n) + t_{m-2,m},\\ &\cdots, F_1(j_k) + Y_{1,m-1}(\bar{s}_k) + P_m(j_n) + t_{1m},\\ &(k-1)t_v + Y_{0,m-1}(\bar{s}_k) + P_m(j_n) + t_{0,m}\}. \end{aligned} \tag{51}$$

Substituting Eq.(2) for $l = m$ and $k = n-1$ into $F_m(j_{n-1})$ of the right-hand side of Eq.(51), we have

$$\begin{aligned}T_m(j_n) \geq \max\{&F_m(j_{n-2}) + P_m(j_{n-1}) + P_m(j_n),\\ &T_{m-1}(j_{n-1}) + P_m(j_{n-1}) + P_m(j_n) + t_{m-1,m},\\ &F_{m-1}(j_k) + Y_{m-1,m-1}(\bar{s}_k) + P_m(j_n) + t_{m-1,m},\\ &F_{m-2}(j_k) + Y_{m-2,m-1}(\bar{s}_k) + P_m(j_n) + t_{m-2,m},\\ &\cdots, F_1(j_k) + Y_{1,m-1}(\bar{s}_k) + P_m(j_n) + t_{1m},\\ &(k-1)t_v + Y_{0,m-1}(\bar{s}_k) + P_m(j_n) + t_{0,m}\}. \end{aligned} \tag{52}$$

As a matter of convenience, we introduce a new notation $Y^i_{uv}(\bar{s}_k)$ as follows,

$$\begin{aligned}Y^i_{uv}(\bar{s}_k) = &\max_{k<q(u)\leq\cdots\leq q(v-1)\leq i}\Big[\sum_{h=k+1}^{q(u)} P_u(j_h) + \cdots + \sum_{h=q(v-1)}^{i} P_v(j_h)\Big],\\ &0 \leq u \leq v \leq m; k < i. \end{aligned} \tag{53}$$

Since we have the same equation as Eq.(49) for n replaced by $i(> k)$, we have

$$\begin{aligned}T_{m-1}(j_i) \geq \max\{&F_{m-1}(j_k) + Y^i_{m-1,m-1}(\bar{s}_k), F_{m-2}(j_k) + Y^i_{m-2,m-1}(\bar{s}_k) + t_{m-2,m-1},\\ &\cdots, F_1(j_k) + Y^i_{1,m-1}(\bar{s}_k) + t_{1,m-1}, (k-1)t_v + Y^i_{0,m-1}(\bar{s}_k) + t_{0,m-1}\}. \end{aligned} \tag{54}$$

Substituting Eq.(54) for $i = n-1$ into $T_{m-1}(j_{n-1})$ of the right-hand side of Eq.(52), we have

$$\begin{aligned}T_m(j_n) \geq \max\{&F_m(j_{n-2}) + P_m(j_{n-1}) + P_m(j_n),\\ &F_{m-1}(j_k) + \max[Y^{n-1}_{m-1,m-1}(\bar{s}_k) + P_m(j_{n-1}) + P_m(j_n),\\ &Y_{m-1,m-1}(\bar{s}_k) + P_m(j_n)] + t_{m-1,m},\\ &F_{m-2}(j_k) + \max[Y^{n-1}_{m-2,m-1}(\bar{s}_k) + P_m(j_{n-1}) + P_m(j_n),\\ &Y_{m-2,m-1}(\bar{s}_k) + P_m(j_n)] + t_{m-2,m},\\ &\cdots, F_1(j_k) + \max[Y^{n-1}_{1,m-1}(\bar{s}_k) + P_m(j_{n-1}) + P_m(j_n),\\ &Y_{1,m-1}(\bar{s}_k) + P_m(j_n)] + t_{1m},\\ &(k-1)t_v + \max[Y^{n-1}_{0,m-1}(\bar{s}_k) + P_m(j_{n-1}) + P_m(j_n),\\ &Y_{0,m-1}(\bar{s}_k) + P_m(j_n)] + t_{0,m}\}. \end{aligned} \tag{55}$$

Repeating the above substitutions for $n-2, n-3, \cdots, k+1$, we have

$$\begin{aligned}T_m(j_n) \geq \max\{&F_m(j_k) + \sum_{k+1}^{n} P_m(j_h),\\ &F_{m-1}(j_k) + \max[Y^{k+1}_{m-1,m-1}(\bar{s}_k) + \sum_{k+1}^{n} P_m(j_h), \cdots, Y_{m-1,m-1}(\bar{s}_k) + P_m(j_n)] + t_{m-1,m},\\ &F_{m-2}(j_k) + \max[Y^{k+1}_{m-2,m-1}(\bar{s}_k) + \sum_{k+1}^{n} P_m(j_h), \cdots, Y_{m-2,m-1}(\bar{s}_k) + P_m(j_n)] + t_{m-2,m},\\ &\cdots,\\ &F_1(j_k) + \max[Y^{k+1}_{1,m-1}(\bar{s}_k) + \sum_{k+1}^{n} P_m(j_h), \cdots, Y_{1,m-1}(\bar{s}_k) + P_m(j_n)] + t_{1m},\\ &(k-1)t_v + \max[Y^{n-1}_{0,m-1}(\bar{s}_k) + \sum_{k+1}^{n} P_m(j_h), \cdots, Y_{0,m-1}(\bar{s}_k) + P_m(j_n)] + t_{0,m}\}. \end{aligned} \tag{56}$$

From Eq.(21) we have

$$\max[Y^{k+1}_{l,m-1}(\bar{s}_k) + \sum_{k+1}^{n} P_m(j_h), \cdots, Y_{l,m-1}(\bar{s}_k) + P_m(j_n)]$$

$$= \max[\max_{k<q(l)\leq\cdots\leq q(m-2)\leq k+1}\{\sum_{h=k+1}^{q(l)} P_l(j_h) + \cdots + \sum_{h=q(m-2)}^{k+1} P_{m-1}(j_h)\} + \sum_{k+1}^{n} P_m(j_h),$$

$$\cdots,$$

$$\max_{k<q(l)\leq\cdots\leq q(m-2)\leq n}\{\sum_{h=k+1}^{q(l)} P_l(j_h) + \cdots + \sum_{h=q(m-2)}^{n} P_{m-1}(j_h)\} + P_m(j_h)]$$

$$= \max_{k<q(l)\leq\cdots\leq q(m-2)\leq q(m-1)\leq n}\{\sum_{h=k+1}^{q(l)} P_l(j_h) + \cdots + \sum_{h=q(m-2)}^{q(m-1)} P_{m-1}(j_h) + \sum_{h=q(m-1)}^{n} P_m(j_h)\}$$

$$= Y_{lm}(\bar{s}_k), \quad l = 0, 1, \cdots, m-1. \tag{57}$$

Substituting Eq.(57) for $l = 0, 1, \cdots, m-1$ into the right-hand side of Eq.(56), we have

$$T_m(j_n) \geq \max\{F_m(j_k) + Y_{mm}(\bar{s}_k), F_{m-1}(j_k) + Y_{m-1,m}(\bar{s}_k) + t_{m-1,m}, \cdots, F_1(j_k) + Y_{1m}(\bar{s}_k) + t_{1m}, (k-1)t_v + Y_{0m}(\bar{s}_k) + t_{0m}\}. \tag{58}$$

Thus, Lemma 3 holds for $l = m$. This completes the proof of Lemma 3.

References

[1] Association of Mechanical Technology of Japan: *A Collection of FMS's in Japan*, Mashinisuto Publisher, 1982 (in Japanese).

[2] Buzacott, J.A., and Yao, D.W.: Flexible Manufacturing Systems: A Review of Models, *Management Science*, Vol.32 (1986) 890-905.

[3] Cheng, J., Kise, H., and Matsumoto, H.: A Branch-and-bound Algorithm with Fuzzy Inference for the 3-machine Flowshop Scheduling Problem, in: *Proceedings of 1994 Japan-U.S.A. Symposium on Flexible Automation–A Pacific Rim Conference–*, Kobe, Japan, July 11-18, 1994, 791-797.

[4] Cheng, J., and Kise, H.: Optimal Scheduling for Automated Two-machine Manufacturing Systems with Intermediate Operations, *Transactions of the Institute of Systems, Control and Information Engineers of Japan*, Vol.8 (1995) 65-71(in Japanese).

[5] Cheng, J., Kise, H., and Matsumoto, H.: A Branch-and-bound Algorithm with Fuzzy Inference for a Permutation Flowshop Scheduling Problem, *To appear in European Journal of Operational Research.*

[6] Garey, M.R., Johnson, D.S., and Sethi, R.: The Complexity of Flowshop and Jobshop Scheduling, *Mathematics of Operations Research* Vol.1 (1976) 117-129.

[7] Hall, N.G., Kamoun, H., and Sriskandarajah, C.: Scheduling in Robotic Cells: Classification, Two and Three Machine Cells, *To appear in Operations Research.*

[8] Hitz, K.: Scheduling of Flow Shops, *Report No. LIDS-R-1049.* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA(1979).

[9] Ibaraki, T.: *Enumerative Approaches to Combinatorial Optimization Part I,* Annals of Operations Research (P.L. Hammer Ed.), 10, J.C.Baltzer Science Publisher, 1987.

[10] Ibaraki, T.: *Enumerative Approaches to Combinatorial Optimization Part II,* Annals of Operations Research, 11, J.C.Baltzer Science Publisher, 1987.

[11] Ishii, N., and Talavage, J.J.: A Mixed Dispatching Rule Approach in FMS Scheduling, *The International Journal of Flexible Manufacturing Systems*, Vol.6 (1994) 69-87.

[12] Kats, V.B.: An Exact Optimal Cyclic Scheduling Algorithm for Multi-operator Service of a Production Line, *Automation and Remote Control*, Vol.43 (1982) 538-542.
[13] Kimemia, J., and Gershwin, S.: An Algorithm for the Computer Control of a Flexible Manufacturing System, *IIE Transactions*, Vol.15 (1983) 353-362.
[14] Kise, H., Shioyama, T., and Ibaraki, T.: Automated Two-machine Flowshop Scheduling: A Solvable Case, *IIE Transaction*, Vol.23 (1991) 10-16.
[15] Kise, H.: On an Automated Two-machine Flowshop Scheduling Problem with Infinite Buffer, *Journal of the Operations Research Society of Japan*, Vol.34 (1991) 354-361.
[16] Kise, H., Kohno, K., Shioyama, T., and Kushiyama, T.: Optimal Scheduling for an Automated Two-Machine Manufacturing System, *Journal of Japanese Society of Mechanical Engineers*, Vol.57 (1991), 1776-1782 (in Japanese), translated into English in *Journal of Advanced Automation Technology*, Vol.4 (1992) 121-127.
[17] Kise, H., Cheng, J., and Matsumoto, H.: A Branch-and-bound Technique Based on Fuzzy Inference (Application to the 3-Machine Flowshop Scheduling Problem), *T. IEE Japan*, Vol.114-c (1994) 470-475(in Japanese).
[18] Lageweg, B.J., Lenstra, J.K., and Rinnooy Kan, A.H.G.: A General Bounding Scheme for the Permutation Flowshop Problem", *Operations Research* Vol.26 (1978) 53-67.
[19] Levner, E., Kogan, K., and Levin, I.: Scheduling a Two-machine Robotic Cell: A Solvable Case, *Mathematics of Industrial Systems*, Vol.1 (1995)
[20] Panwalker, S.S.: Scheduling of a Two-machine Flowshop with Travel Time between Machines, *Journal of the Operational Research Society*, Vol. 42 (1991) 609-613
[21] Sabuncuoglu, I., and Hommertzheim, D.L.: Experimental Investigation of an FMS Due-date Scheduling Problem: Evaluation of Machine and AGV Scheduling Rules, *The International Journal of Flexible Manufacturing Systems*, Vol. 5 (1993) 301-323.
[22] Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., and Kubiak, W.: Sequencing of Part and Robot Moves in a Robotic Cell, *The International Journal of Flexible Manufacturing Systems*, Vol. 4 (1992) 331-358.
[23] Stecke, K.E.: Design, Planning, Scheduling, and Control Problems of Flexible Manufacturing Systems, *Annals of Operations Research*, Vol.3 (1985) 3-12.
[24] Stern, H.I., and Vitner, G.: Scheduling Parts in a Combined Production-transportation Work Cell, *Journal of the Operational Research Sociaty*, Vol. 41 (1990) 625-632.
[25] Terano, T., Asai, K., and Sugeno, M.: *Introduction to Fuzzy System*, Omu Publisher, 1987 (in Japanese).

Hiroshi Kise
Department of Mechanical and System Engineering,
Faculty of Engineering and Design,
Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto 606, Japan
E-mail: kise@ipc.kit.ac.jp