http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (2): 257-265, 2010
ISSN 1812-5638

Minimizing Makespan and Mean Flow Time in Two-Versatile-Machine Flow-Shop with Alternative Operations

[1]Ming-Cheng Lo, [2]Jen-Shiang Chen and [3]Yong-Fu Chang
[1]Department of Business Administration, Ching Yun University,
229, Chien-Hsin Road, Jung-Li, Taiwan 320, Republic of China
[2]Department of Business Administration,
[3]Department of Innovation and Entreneurship, Far East University,
49 Junghua Road, Shinshr Shiang, Tainan, Taiwan 744, Republic of China

Abstract: This study addresses the two-machine flow-shop scheduling problems in which both machines are versatile and thus alternative operations are possible. The performance measures are the mean flow time and makespan of jobs, respectively. The problem is formulated as two integer programming models and two heuristics are developed. Computational results are provided to demonstrate the efficiency of the models and the effectiveness of the heuristics. The integer programming technique is inefficient even for small problems. For the set of problems with known optimal solutions, the average percentage errors of heuristics H_1 and H_2 are within 10%, respectively. For the set of problems with unknown optimal solutions, the average percentage errors of heuristic H_1 for solving the mean flow time is 20.09%, while the average percentage errors of heuristic H_2 for solving the makespan is 20.84%.

Key words: Scheduling, two-versatile-machine flow-shop, alternative operations, integer programming, heuristic

INTRODUCTION

Automated Manufacturing Systems (AMS) may require a large capital investment, however, the key to success in implementation of AMS is the effective utilization of manufacturing resources (e.g., machines, tools, fixtures, pallets and feeders) through the application of efficient scheduling algorithms. In contrast to the conventional assumption that only one of each type of machine is available, some of the machines can perform alternative operations as well as their primary operations. Therefore, the assignment of factory resources to production tasks can exploit the versatility of AMS. Restated, rather than the traditional approach of separating process planning and production scheduling, an AMS can integrate these two functions by including alternative operations in operation routing during the scheduling stage. Such a system aims to simultaneously optimize operation allocation among machines and operation loading sequence to machines. Wilhelm and Shin (1985) concluded that the implementation of alternative operations could reduce flow time and the in-process inventory in a Flexible Manufacturing System (FMS). Srihari and Greene (1988) considered alternate routings strategies that could prevent bottlenecks, reduce in-process inventory, balance machine utilization and minimize flowtime.

Liao *et al*. (1995) presented two integer programming models for a permutation flow-shop where one or more processors had the flexibility to perform other operations besides their own and developed a heuristic to solve the problem with predetermined job sequence. Lee and Mirchandani (1988) studied a two-versatile-machine flow-shop problem and showed that the problem could be reduced to three versions of zero-setup, one-setup and two-setup problems. Cheng and Wang (1999) considered the NP-complete one-setup version of the problem studied by Lee and Mirchandani (1988) and derived a tight worst-case error bound for the heuristic presented by Lee and Mirchandani (1988) and proposed another heuristic with a tight worst-case error bound of 3/2. Pan and Chen (1997) showed that scheduling the two-machine flow-shop where either one or both machines are versatile to perform alternative operations was NP-complete and developed a branch-and-bound algorithm to solve the problem optimally. Cheng and Wang (1998) presented a general pseudo-polynomial dynamic programming scheme for the problem studied by Pan and Chen (1997) and showed that the solution scheme could be modified to solve the problems

Corresponding Author: Ming-Cheng Lo, Department of Business Administration, Ching Yun University, No. 229, Jiansing Road, Jhongli City, Taoyuan County, Taiwan 320, Republic of China

investigated by Lee and Mirchandani (1988). Chen and Pan (2005) also studied the two-versatile-machine flow-shop scheduling problem but they used the mean tardiness as the performance measure. Additionally, Low *et al*. (2008) addressed a two-stage hybrid flowshop scheduling problem with unrelated alternative machines.

Since, in practice, many machines have the potential to conduct more than one operation, production scheduling should take the machine versatility into consideration. The situation in which the processing time for a particular operation differs among machines is considered in this study. In semiconductor manufacturing, a certain operation can be processed on multiple machines with different processing times. This study investigates the scheduling problem in a two-machine flow-shop where both machines are versatile and alternative operations are allowed. The performance measures are the mean flow time and makespan of jobs, respectively. Mean flow time is important for minimizing work in process and lead time without impacting production capacity. A minimum makespan usually implies a high utilization of the machine(s). The problem is formulated as two integer programs and two heuristics are developed. Computational experiments are used to test the efficiency of the integer programming formulations and the effectiveness of the heuristics in generating near optimal solutions for minimizing the mean flow time and determining the schedule makespan.

PROBLEM DESCRIPTION

The two-versatile-machine flow-shop scheduling problem (2VFSP) under investigation is described as follows: There are n independent jobs. Each job has two operations V and W and the processing of operation V followed by operation W. The shop contains two machines, M1 and M2. Each machine can perform both operations of V and W. Primarily, M1 performs operation V and M2 performs operation W. An alternative operation refers to the situation where operation V of a job is not processed on M1, or operation W is not processed on M2. The processing times of an operation on different machines are different. Pan and Chen (1997) assumed that the processing time of an operation on its primary machine was always shorter than that on the alternative machine. According to this assumption, Pan and Chen (1997) developed several fathoming rules to solve the problem. In practice, the processing time of an operation on its primary machine is usually not shorter than that on the alternative machine. This study did not emphasize this assumption and instead considers the general case. The branch and bound algorithm provided by Pan and Chen (1997), but this algorithm could not to solve the general case. This general case is more complex and the extension to Pan and Chen (1997) is significant. The assumption that the processing time of an operation on its primary machine was always shorter than that on the alternative machine is the special case considered in this study. The performance measures are the mean flow time and makespan of jobs, respectively.

Theorem 1: Problem 2VFSP with mean flow time or makespan is NP-complete.

Proof: Clearly, the problem 2VFSP with mean flow time is NP-complete since the problem that minimizes the mean flow time for two-machine flow-shop without alternative operations is NP-complete (Pinedo, 2008). Additionally, Pan and Chen (1997) had shown that the problem 2VFSP with makespan was NP-complete.

As defined earlier, the versatile-machine flow-shop is actually a job-shop because the machines can be visited in any order. In traditional flow-shop problems, investigating two-machine is useful as it helps understand permutation schedules and gain other insights that apply to general multiple-machine problems. But this does not apply to a versatile-machine flow-shop. This study could similarly try to glean useful insight from two-machine problems that could become building blocks for solving larger problems.

The assumptions made for 2VFSP are summarized here. Machines are continuously available. The processing times of the operations of each job are known and fixed. The operations are not preemptable. All jobs are immediately ready for processing once production begins. Each machine can perform only one operation at a time. The setup time required for a machine to shift between operations is negligible and the transfer or transport time of a job between machines is also negligible.

INTEGER PROGRAMMING MODELS

Mathematical programming formulation is a natural way to solve machine-scheduling problems (Rinnooy-Kan, 1976). Most integer programming problems of scheduling involve mixed binary integer programming, that is, some variables are binary and the others are continuous. This section presents two integer programming models, including the 2VFSP_$\bar{F}$ model for the 2VFSP with minimal flow time as the criterion and the 2VFSP_C_{max} model for the 2VFSP with makespan as the criterion. The notations used in the mixed binary integer programming model are as follows:

M = A very large positive number
n = Number of jobs for processing at time zero
J_i = Job number i
M_k = Machine number k, k = 1, 2
O_{ij} = Operation number j of J_i, j = 1, 2
s_{ij} = The starting time of O_{ij}, j = 1, 2
p_{ijk} = The processing time of O_{ij} on M_k
X_{ij} = 1, if O_{ij} requires M_k, where j, k = , 2 and k ≠ j; 0, otherwise
F_i = The flow time of J_i, that is, the time that J_i spends in the workshop;
C_{max} = Makespan, that is, the maximum completion time of jobs
$Z^k_{iji'j'}$ = 1, if O_{ij} precedes $O_{i'j'}$ (not necessarily immediately) on M_k; 0, otherwise

The 2VFSP_$\bar{F}$ model: The 2VFSP_$\bar{F}$ model is first presented to solve a 2VFSP with mean flow time as the criterion. In 2VFSP, if O_{i1} precedes $O_{i'1}$ (not necessarily immediately) on M_1 ($1 \le i < i' \le n$), then $Z^1_{i1i'1} = 1$, while if O_{i1} and $O_{i'1}$ are not alternative operations (that is, $X_{i1} = X_{i'1} = 0$), then $s_{i'1}-s_{i1} \ge p_{i11}$ must hold. That is:

$$s_{i'1}-s_{i1} \ge p_{i11}-M(X_{i1}+X_{i'1})-M(1-Z^1_{i1i'1}) \tag{1}$$

If O_{i1} precedes $O_{i'2}$ (not necessarily immediately) on M_1 ($1 \le i < i' \le n$), then $Z^1_{i1i'2} = 1$, while if $O_{i'2}$ is an alternative operation but O_{i1} is not (that is, $X_{i1} = 0$ and $X_{i'2} = 1$), then $s_{i'2}-s_{i1} \ge p_{i11}$ must hold. That is:

$$s_{i'2}-s_{i1} \ge p_{i11}-M(X_{i1}+1-X_{i'2})-M(1-Z^1_{i1i'2}) \tag{2}$$

If O_{i2} precedes $O_{i'1}$ (not necessarily immediately) on M_1 ($1 \le i < i' \le n$), then $Z^1_{i2i'1} = 1$, while if O_{i2} is an alternative operation but $O_{i'1}$ is not (that is, $X_{i2} = 1$ and $X_{i'1} = 0$), then $s_{i'1}-s_{i2} \ge p_{i21}$ must hold. That is:

$$s_{i'1}-s_{i2} \ge p_{i21}-M(1-X_{i2}+X_{i'1})-M(1-Z^1_{i2i'1}) \tag{3}$$

If O_{i2} precedes $O_{i'2}$ (not necessarily immediately) on M_1 ($1 \le i < i' \le n$), then $Z^1_{i2i'2} = 1$, while if both O_{i2} and $O_{i'2}$ are alternative operations (that is, $X_{i2} = X_{i'2} = 1$), then $s_{i'2}-s_{i2} \ge p_{i21}$ must hold. That is:

$$s_{i'2}-s_{i2} \ge p_{i21}-M(2-X_{i2}-X_{i'2})-M(1-Z^1_{i2i'2}) \tag{4}$$

Similarly, if O_{ij} precedes $O_{i'j'}$ (not necessarily immediately) on M_2, for $1 \le i < i' \le n$; j, j′ = 1, 2; the four following constraints must hold.

$$s_{i'2}-s_{i2} \ge p_{i22}-M(X_{i2}+X_{i'2})-M(1-Z^2_{i2i'2}) \tag{5}$$

$$s_{i'1}-s_{i2} \ge p_{i22}-M(X_{i2}+1-X_{i'1})-M(1-Z^2_{i2i'1}) \tag{6}$$

$$s_{i'2}-s_{i1} \ge p_{i12}-M(1-X_{i1}+X_{i'2})-M(1-Z^2_{i1i'2}) \tag{7}$$

$$s_{i'1}-s_{i1} \ge p_{i12}-M(2-X_{i1}-X_{i'1})-M(1-Z^2_{i1i'1}) \tag{8}$$

If $O_{i'j'}$ precedes O_{ij} (not necessarily immediately) on M_1, for $1 \le i < i' \le n$; j, j′ = 1, 2; the four following constraints must hold.

$$s_{i1}-s_{i'1} \ge p_{i'11}-M(X_{i'1}+X_{i1})-M\,Z^1_{i1i'1} \tag{9}$$

$$s_{i2}-s_{i'1} \ge p_{i'11}-M(X_{i'1}+1-X_{i2})-M\,Z^1_{i2i'1} \tag{10}$$

$$s_{i1}-s_{i'2} \ge p_{i'21}-M(1-X_{i'2}+X_{i1})-M\,Z^1_{i1i'2} \tag{11}$$

$$s_{i2}-s_{i'2} \ge p_{i'21}-M(2-X_{i'2}-X_{i2})-M\,Z^1_{i2i'2} \tag{12}$$

If $O_{i'j'}$ precedes O_{ij} (not necessarily immediately) on M_2, for $1 \le i < i' \le n$; j, j′ = 1, 2; the four following constraints must hold.

$$s_{i2}-s_{i'2} \ge p_{i'22}-M(X_{i'2}+X_{i2})-M\,Z^2_{i2i'2} \tag{13}$$

$$s_{i1}-s_{i'2} \ge p_{i'22}-M(X_{i'2}+1-X_{i1})-M\,Z^2_{i1i'2} \tag{14}$$

$$s_{i2}-s_{i'1} \ge p_{i'12}-M(1-X_{i'1}+X_{i2})-M\,Z^2_{i2i'1} \tag{15}$$

$$s_{i1}-s_{i'1} \ge p_{i'12}-M(2-X_{i'1}-X_{i1})-M\,Z^2_{i1i'1} \tag{16}$$

The formulation for minimizing mean flow time with alternative operations in the two-machine flow-shop is as follows:

$$\text{Minimize } \frac{1}{n}\sum_{i=1}^{n} F_i \tag{17}$$

Subject to:

$$s_{i1}+p_{i11}(1-X_{i1})+p_{i12}X_{i1} \le s_{i2} \qquad i = 1, 2, ..., n \tag{18}$$

$$s_{i2}+p_{i22}(1-X_{i2})+p_{i21}X_{i2} = Fi \qquad i = 1, 2, ..., n \tag{19}$$

$$s_{i'j'}-s_{ij} \ge p_{ijk}-M(X_{ij}+X_{i'j'})-M(1-Z^k_{iji'j'})$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j = j' = k \tag{20}$$

$$s_{i'j'}-s_{ij} \ge p_{ijk}-M(X_{ij}+1-X_{i'j'})-M(1-Z^k_{iji'j'})$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j \ne j¢ \text{ and } j = k \tag{21}$$

$$s_{i'j'}-s_{ij} \ge p_{ijk}-M(1-X_{ij}+X_{i'j'})-M(1-Z^k_{iji'j'})$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j \ne j' \text{ and } j \ne k \tag{22}$$

$$s_{i'j'}-s_{ij} \ge p_{ijk}-M(2-X_{ij}-X_{i'j'})-M(1-Z^k_{iji'j'})$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j = j' \text{ and } j \ne k \quad (23)$$

$$s_{ij} - s_{i'j'} \ge p_{i'j'k} - M\,(X_{i'j'} + X_{ij}) - M\,Z^{k}_{iji'j'}$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j = j' = k \quad (24)$$

$$s_{ij} - s_{i'j'} \ge p_{i'j'k} - M\,(X_{i'j'} + 1 - X_{ij}) - M\,Z^{k}_{iji'j'}$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j \ne j' \text{ and } j \ne k \quad (25)$$

$$s_{ij} - s_{i'j'} \ge p_{i'j'k} - M\,(1 - X_{i'j'} + X_{ij}) - M\,Z^{k}_{iji'j'}$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j \ne j' \text{ and } j = k \quad (26)$$

$$s_{ij} - s_{i'j'} \ge p_{i'j'k} - M\,(2 - X_{i'j'} - X_{ij}) - M\,Z^{k}_{iji'j'}$$

$$1 \le i < i' \le n;\ j, j', k = 1, 2 \text{ and } j = j' \text{ and } j \ne k \quad (27)$$

$$F_i \ge 0,\ s_{ij} \ge 0,\ i = 1, 2, \ldots, n;\ j = 1, 2;$$

$$X_{ij} \text{ and } Z^{k}_{iji'j'} = 0 \text{ or } 1 \quad 1 \le i < i' \ge n;\ j, j', k = 1, 2 \quad (28)$$

Objective function (17) describes the minimization of the mean flow time, while constraint set (18) ensures the processing of O_{i2} can be started only after O_{i1} is finished. Constraint set (19) defines job flow time. Additionally, constraint set (20) is composed of constraints (1) and (5), (21) of (2) and (6), (22) of (3) and (7), (23) of (4) and (8), (24) of (9) and (13), (25) of (10) and (14), (26) of (11) and (15) and (27) of (12) and (16). The non-negativity and binary restrictions on F_i and s_{ij} and X_{ij} and $Z^{k}_{iji'j'}$, respectively, are specified in (28). The size of the above model is $4n^2$-2n binary variables, $8n^2$-6n constraints and 3n continuous variables.

The 2VFSP_C_{max} model: The 2VFSP_$\bar{F}$ model can be readily modified to a 2VFSP_C_{max} model by using makespan as the criterion. The 2VFSP_C_{max} model included constraint sets (20)-(27) of the 2VFSP_$\bar{F}$ model. The objective function (17) and constraint set (19) are changed to (29) and (30), respectively, as follow:

$$\text{Minimize } C_{max} \quad (29)$$

$$s_{i2} + p_{i22}\,(1 - X_{i2}) + p_{i21}\,X_{i2} \le C_{max} \quad i = 1, 2, \ldots, n \quad (30)$$

Additionally, the non-negativity and binary restrictions (28) are specified in (31).

$$C_{max} \ge 0,\ s_{ij} \ge 0,\ i = 1, 2, \ldots, n;\ j = 1, 2;$$

$$X_{ij} \text{ and } Z^{k}_{iji'j'} = 0 \text{ or } 1 \quad 1 \le i < i' \le n;\ j, j', k = 1, 2 \quad (31)$$

Consequently, the 2VFSP_C_{max} model, which minimizes the makespan of jobs for a 2VFSP, includes objective (29) and constraint sets (18), (20)-(27), (30) and (31). The size of the above model is $4n^2$-2n binary variables, $8n^2$-6n constraints and 2n+1 continuous variables.

HEURISTIC SCHEDULING ALGORITHM

Although, the integer programming approach can solve small problems, large problems are difficult to solve without considerable computation time; we therefore propose heuristics to solve larger problems.

The alternative operation type of a job refers to its processing route, namely, the machine on which the first and second operations are processed. Let t denote the alternative operation type of J_i. The value t = 0 represents that there is no alternative operation for J_i, or, the first operation is processed on M_1 and the second on M_2. For t = 1, the first operation of J_i is not an alternative one, while the second is; in other words, both operations are processed on M_1. Moreover, t = 2 denotes that the first operation of J_i is an alternative operation while the second is not, that is, both the first and second operations are processed on M_2. Lastly, let t = 3 designate the situation that both operations of J_i are alternative one, that is, the first operation is processed on M_2 and the second on M_1.

Rule 1: The processing route of J_i must obey one of the following four alternative operation types: (1) t = 0, that is, $X_{i1} = X_{i2} = 0$; (2) t = 1, that is, $X_{i1} = 0$ and $X_{i2} = 1$; (3) t = 2, that is, $X_{i1} = 1$ and $X_{i2} = 0$; (4) t = 3, that is, $X_{i1} = X_{i2} = 1$.

Minimizing the mean flow times: Heuristic H_1 is established for 2VFSP, with minimizing the mean flow times as the criterion. All jobs are first placed in the unassigned job set. Then we use the selecting rule to find the best job and its alternative operation type. This best job is removed from the unassigned job set when the best job is determined. The procedure is terminated until the unassigned job set is null.

Let TM_k (k = 1, 2) denote the current earliest finish time on M_k, $TM_k^{i,t}$ (t = 0, 1, 2, 3) denote the TM_k value when J_i is sequenced with alternative operation type t on the schedule and $F^{i,t} = \max\,\{TM_1^{i,t}, TM_2^{i,t}\}$. Let $J_{[q]}$ denote the job scheduled at the qth position in the processing sequence, $A = (J_{[1]}, J_{[2]}, \ldots, J_{[Q]})$ represent the subsequence of the Q jobs that have already been assigned.

Let $U = \{J_1, J_2, \ldots, J_n\} - \{(J_{[1]}, J_{[2]}, \ldots, J_{[Q]})\}$ be the set of unassigned (n - Q) jobs. The jobs in U and its alternative operation type are to be decided as follows. For each job J_i in U, calculate the $TM_k^{i,t}$ value and then use $TM_k^{i,t}$ to identify the best job J_{i^*} along with its t^* value. Next, sequence J_{i^*} at the (Q+1) position of A. Rule 2 states the computational process for $TM_k^{i,t}$ and rule 3 describes the procedure for selecting the best job J_{i^*} and its t^*.

Rule 2: Consider a job J_i in U. If its alternative operation type t = 0, then $TM_1^{i,0} = TM_1 + p_{i11}$, $TM_2^{i,0} = \max\{TM_1^{i,0}, TM_2\} + p_{i22}$ and $F^{i,0} = TM_2^{i,0}$. If t = 1, then $TM_1^{i,1} = TM_1 + p_{i11} + p_{i21}$, $TM_2^{i,1} = TM_2$ and $F^{i,1} = TM_1^{i,1}$. If t = 2, then $TM_1^{i,2} = TM_1$, $TM_2^{i,2} = TM_2 + p_{i12} + p_{i22}$ and $F^{i,2} = TM_2^{i,3}$. If t = 3, then $TM_2^{i,3} = TM_2 + p_{i12}$, $TM_1^{i,3} = \max\{TM_2^{i,3}, TM_1\} + p_{i21}$ and $F^{i,3} = TM_1^{i,3}$.

Rule 3: Determine the best job J_{i*} and its alternative operation type t^* as $\{(i, t)|\min\{F^{i,t}, J_i \text{ in U and } t = 0, 1, 2, 3\}\}$. If (i^*, t^*) is not unique, choose $(i^*, t^*) = \{(i, t)|\max\{TM_{dist}^{i,t}, J_i \text{ in U and } t = 0, 1, 2, 3\}\}$, where $TM_{dist}^{i,t} = |TM_1^{i,t} - TM_2^{i,t}|$. If a tie still exists, select one arbitrarily.

After determining the best alternative operation type t^* of J_{i*}, the flow time F_{i*} of J_{i*} is thus $F^{i*,t*}$. TM_k can be updated by setting $TM_k = TM_k^{i*,*t}$. The mean flow time $\bar{F}(H_1)$ can be calculated once the flow times of all jobs are determined.

The stepwise description of the heuristic H_1 is as follows:

- **Step 1:** ϕ denotes the null set. Set $TM_1 = TM_2 = 0$, A = ϕ and U = $\{J_1, J_2, \ldots, J_n\}$
- **Step 2:** Determine $TM_k^{i,t}$ and $F^{i,t}$ for each job in U according to rule 2
- **Step 3:** Identify the best job J_{i*} and its alternative operation type t^* using Rule 3
- **Step 4:** Set $TM_k = TM_k^{i*,t*}$ and $F_{i*} = F^{i*,t*}$
- **Step 5:** Delete job J_{i*} from U and add it to A
- **Step 6:** If U = ϕ, then stop. Otherwise, go to step 2

When completing heuristic H_1, the $\bar{F}(H_1)$ value can be obtained using $(1/n)\sum_{i=1}^{n} F_i$.

The complexity of heuristic H_1 can be investigated as follows: Step 1 performs the initialization procedure with time-complexity O (n). Step 2 then performs a selecting procedure to find the best job and its alternative operation type with time-complexity O (n^2). Step 3 identifies the best job and its alternative operation type with time-complexity O (n). Step 4 defines the current earliest finish time on each machine and job flow time with time-complexity O (n), while step 5 modifies the assigned and unassigned subsequence with time-complexity O (n). Finally, step 6 performs a comparison procedure with time-complexity O (n). Therefore, the time-complexity of heuristic H_1 is O (n^2).

Minimizing the makespan: Heuristic H_2 is established two heuristics for 2VFSP with minimizing the makespan as the criterion. All jobs are first placed in the unassigned job set. As defined in above, 2VFSP is actually an $n/2/G/C_{max}$ problem because the machines can be visited in any order. Jackson (1956) shown that the specific rules can optimize the makespan for the $n/2/G/C_{max}$ problem. The selecting rule is used to find the best job and its alternative operation type. This best job is removed from the unassigned job set when the best job is determined. The procedure is terminated until the unassigned job set is null.

Suppose the alternative operation types of the Q jobs are given and can be partitioned into four job types, as follows:

{A}: Set of jobs with t = 1, that is, jobs for processing on machine M_1 only

{B}: Set of jobs with t = 2, that is, jobs for processing on machine M_2 only

{AB}: Set of jobs with t = 0, that is, job for processing on both machines in the order M_1 then M_2

{BA}: Set of jobs with t = 3, that is, job for processing on both machines in the order M_2 then M_1

An optimal makespan then is obtained by the following theorem:

Theorem 2: Given the alternative operation type of jobs of the 2VFSP problem, an optimal makespan is determined by the following rule:

(1) On the machine M_1, arrange in the order {AB}, {A}, {BA}
(2) On the machine M_2, arrange in the order {BA}, {B}, {AB}
(3) All jobs in {AB} are scheduled according to Johnson, 1954, Johnson's rule, i.e., for J_i, $J_j \in$ {AB} and $i \neq j$, job J_i precedes job J_j on both machines if $\min\{p_{i11}, p_{j22}\} \leq \min\{p_{j11}, p_{i22}\}$
(4) All jobs in {BA} are scheduled according to Johnson (1954), Johnson's rule, i.e., for J_i, $J_j \in$ {BA} and $i \neq j$, job J_i precedes job J_j on both machines if $\min\{p_{i12}, p_{j21}\} \leq \min\{p_{j12}, p_{i21}\}$
(5) All other jobs in {A} and {B} are scheduled in arbitrary orders

Proof: If the alternative operation type of jobs is known, then the 2VFSP problem can be reduced to the $n/2/G/C_{max}$ problem. Jackson (1956) shown that the above rules can optimize the makespan for the $n/2/G/C_{max}$ problem.

Suppose there are Q jobs in set A which have already been assigned in the schedule and their alternative operation types are given. Then select one candidate job to enter set A from the unassigned (n-Q) jobs in set U. Four alternative operation types exist for this candidate job. Theorem 2 is applied to optimize the makespan for each alternative type of this candidate job. This procedure is repeated until all alternative types of all candidate jobs have been tested. The election job and that of the election alternative operation type can minimize the makespan among all candidate jobs as well as the makespan of alternative operation types. Let CM_k (k = 1, 2) denote the current earliest completion time on M_k and let $CM_k^{i,t}$ (t = 0, 1, 2, 3) denote the CM_k value when J_i is the candidate job with alternative operation type t on the schedule and $C^{i,t} = \max\{CM_1^{i,t}, CM_2^{i,t}\}$. $CM^{i,t}$ is used to identify the best job J_{i^*}, along with its t^* value. J_{i^*} and t^* thus are the election job and its alternative operation type, respectively. Theorem 2 states the computational process for $CM_k^{i,t}$ and Rule 4 describes the procedure for selecting the best job J_{i^*} and its t^*.

Rule 4: For the 2VFSP with minimizing the makespan as the criterion, determine the best job J_{i^*} and its alternative operation type t^* as $\{(i, t)|\min\{CM^{i,t}, J_i$ in the candidate jobs and $t = 0, 1, 2, 3\}\}$. If (i^*, t^*) is not unique, choose $(i^*, t^*) = \{(i, t)|\max\{CM_{dist}^{i,t}, J_i$ in the candidate jobs and $t = 0, 1, 2, 3\}\}$, where $CM_{dist}^{i,t} = |CM_1^{i,t} - CM_2^{i,t}|$. If a tie still exists, select one arbitrarily.

The stepwise description of the heuristic H_2 is as follows:

- **Step 1:** Set A = ϕ and U = $\{J_1, J_2, \ldots, J_n\}$
- **Step 2:** Select the combination of the candidate job J_i in U and its alternative operation type t. Determine $CM_k^{i,t}$ (k = 1, 2) and $C^{i,t}$ according to theorem 2
- **Step 3:** Identify the best job J_{i^*} and its alternative operation type t^* using Rule 4
- **Step 4:** Delete job J_{i^*} from U and add it to A
- **Step 5:** If U = ϕ, then stop. Otherwise, go to step 2

When completing the heuristic H_2, the $C_{max}(H_2)$ value can be obtained by C^{i^*,t^*}. The complexity of heuristic H_2 can be investigated as follows: Step 1 performs the initialization procedure with time-complexity O (n). Step 2 then performs a selecting procedure to find the best job and its alternative operation type with time-complexity O (n^2logn). Step 3 identifies the best job and its alternative operation type with time-complexity O (n), while step 4 modifies the assigned and unassigned subsequence with time-complexity O (n). Finally, step 5 performs a comparison procedure with time-complexity O (n). Therefore, the time-complexity of heuristic H_2 is O (n^2logn).

ILLUSTRATIVE EXAMPLE

Consider the five jobs, J_1 to J_5, with processing times listed in Table 1. First solve this problem by the proposed integer programming formulations. In the 2VFSP_$\bar{F}$ model, the optimal mean flow time obtained is 13.8 and the values of relevant variables are as follows: $s_{11} = 7$, $s_{12} = 15$, $s_{21} = 0$, $s_{22} = 2$, $s_{31} = 5$, $s_{32} = 9$, $s_{41} = 15$, $s_{42} = 18$, $s_{51} = 0$, $s_{52} = 8$; $X_{11} = 1$, $X_{12} = 1$, $X_{21} = 1$, $X_{22} = 0$, $X_{31} = 1$, $X_{32} = 1$, $X_{41} = 1$, $X_{42} = 1$, $X_{51} = 0$, $X_{52} = 0$; $F_1 = 17$, $F_2 = 5$, $F_3 = 15$, $F_4 = 23$ and $F_5 = 9$. In the 2VFSP_C_{max} model, the optimal makespan is 21 and the value of related variables are as

Table 1: Data of the illustrative example

Job	J_1	J_2	J_3	J_4	J_5
p_{i11}/p_{i12}	9/8	8/2	8/2	8/3	8/7
p_{i22}/p_{i21}	4/2	3/5	10/6	6/5	6/1

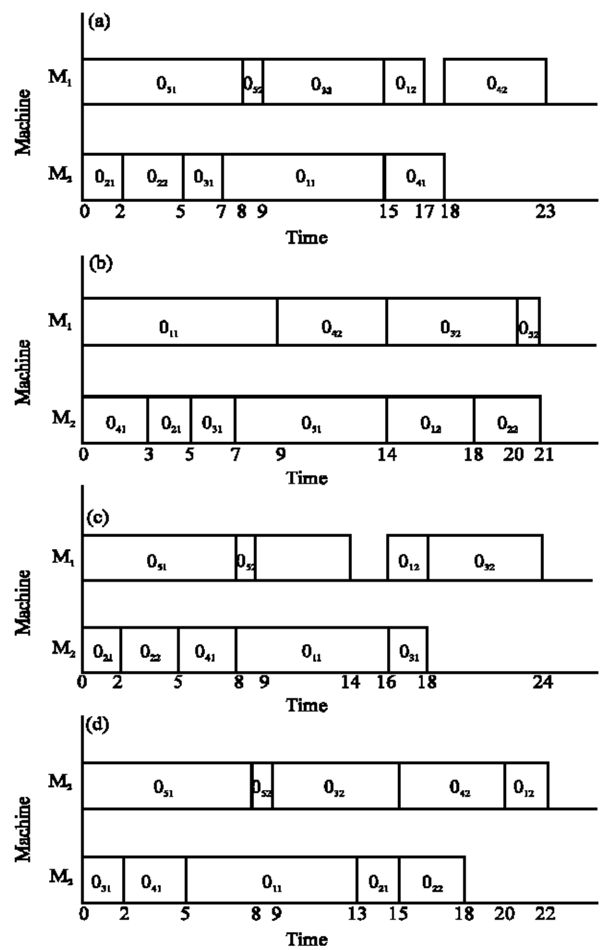

Fig. 1: Gantt chart of the schedule generated for the illustrative example. (a) Generated by the 2VFSP_$\bar{F}$ model, (b) generated by the 2VFSP_C_{max} model, (c) generated by heuristic H_1 and (d) generated by heuristic H_2

follows: $s_{11} = 0$, $s_{12} = 14$, $s_{21} = 3$, $s_{22} = 18$, $s_{31} = 5$, $s_{32} = 14$, $s_{41} = 0$, $s_{42} = 9$, $s_{51} = 7$, $s_{52} = 20$; $X_{11} = 0$, $X_{12} = 0$, $X_{21} = 1$, $X_{22} = 0$, $X_{31} = 1$, $X_{32} = 1$, $X_{41} = 1$, $X_{42} = 1$, $X_{51} = 1$, $X_{52} = 1$ and $C_{max} = 21$. The optimal schedules with mean flow time and makespan as criteria are shown in Fig. 1a and b, respectively.

To minimize the mean flow time problems, heuristic H_1 concurrently determines the sequence and the alternative operations type for each job. The best combination (i^*, t^*) follows the order (2, 2), (5, 1), (4, 3), (1, 3) and (3, 3). The job flow times found are: $F_1 = 18$, $F_2 = 5$, $F_3 = 24$, $F_4 = 14$ and $F_5 = 9$. Moreover, the mean flow time is 14 and Fig. 1c shows the schedule found by heuristic H_1.

For minimizing the makespan problems, the best combination (i^*, t^*) in heuristic H_2 follows the order (2, 2), (3, 3), (4, 3), (1, 3) and (5, 1). Additionally, the makespan is 22 and Fig. 1d shows the schedule found by heuristic H_2.

EXPERIMENTAL RESULTS

Computational experiments were conducted to test the effectiveness and efficiency of the mixed binary integer programming model and the proposed heuristic algorithm. The heuristic algorithm was coded in C. The mixed binary integer programming formulations were solved with ILOG CPLEX on an Intel P4/2.67 GHz with 512 M DRAM. The time limit is set to 86,400 sec for ILOG CPLEX. The processing times of jobs were generated randomly from a discrete uniform distribution with a range of 1 to 100.

The test problems were divided into two sets, one comprising of problems whose optimal solutions could be identified quickly by solving the integer programming formulations and the other containing problems whose optimal solutions were unknown. Three problem sizes, n = 10, 11 and 12, were tested with the 2VFSP_C_{max} model and n = 10 and 11, with the 2VFSP_C_{max} model in the first set and ten test problems were generated for each problem size. There were seven problem sizes tested in the second set, namely, n = 20, 50, 100, 200, 400, 800 and 1000, in which 100 problems were generated for each size. Thus, a total of 1450 (5×10 + 2×7×100) problems were randomly generated and tested.

Problems with known optimal solutions: Table 2 summarizes the computational results using known optimal mean flow time and makespan, respectively, for the first set of test problems. Notably, the efficiency of integer programming is reported based on the solution time (in seconds). Table 2 also shows the average percentage error. The percentage error is defined as:

Table 2: Computational results for problems with known optimal solutions

	Mean flow time		Makespan	
	Integer programming	Heuristic H_1	Integer programming	Heuristic H_2
Problem	Solution time (sec)	Error (%)	Solution time (sec)	Error (%)
10-1†	8.2415	5.0431	7.2514	6.8416
10-2	4.2134	9.4440	20.3210	9.4533
10-3	5.7854	7.1490	30.4502	10.1475
10-4	24.2178	0.0000	21.4509	8.1578
10-5	8.2147	9.5890	13.7450	0.0000
10-6	3.5141	6.9140	7.2370	9.1550
10-7	24.2314	9.8290	15.2045	11.0630
10-8	6.4857	0.0000	7.2350	11.8791
10-9	31.5612	0.0000	42.2103	0.0000
10-10	3.6524	8.8310	15.2450	7.4588
Avg.	12.0118	5.6799	18.0350	7.4156
11-1	84.2451	6.2717	11254.1551	6.7960
11-2	83.249	0.0000	5124.7851	12.1746
11-3	20.1250	9.1784	72.9681	13.4785
11-4	50.6587	4.1575	300.0851	0.0000
11-5	10.5874	2.0123	1058.3875	2.5145
11-6	95.0451	4.7246	1352.6540	5.6820
11-7	70.1254	9.1475	1293.2457	1.8567
11-8	23.1260	0.0000	708.8576	0.0000
11-9	251.0645	8.7183	3914.2580	3.8594
11-10	12.3542	8.6277	558.8964	7.2540
Avg.	70.0580	5.2838	2563.8293	5.3616
12-1	28975.8564	14.1930	N/A	N/A
12-2	7528.2517	3.3740	N/A	N/A
12-3	5852.6578	0.0000	N/A	N/A
12-4	8831.2561	11.9605	N/A	N/A
12-5	16387.5264	5.5891	N/A	N/A
12-6	8864.2578	12.6680	N/A	N/A
12-7	658.9511	0.0000	N/A	N/A
12-8	4645.6221	5.9510	N/A	N/A
12-9	17564.8524	7.0470	N/A	N/A
12-10	8925.2581	10.0141	N/A	N/A
Avg.	10823.4490	7.0797	N/A	N/A

†Specified by 10 jobs-1st problem. NA: Not available

$$\text{Percentage error} = \frac{S - S_0}{S_0} \times 100$$

where, S denotes the mean flow time (or makespan) obtained by the heuristics and S_0 represents the optimal mean flow time (or makespan) of the schedule. Table 2 yields the following observations.

- Integer programming technique is not efficient even for small problems. Experiments with the 2VFSP_$\bar{F}$ model demonstrate that this approach solved the 11-job problems with an average computation time of roughly 70 sec and took more than three hour for the 12-job problems. Moreover, results on the 2VFSP_C_{max} model reveal that it solved the 10-job problems with an average time of 18 sec and required approximately 42 min for the 11-job problems
- The average percentage error of heuristic H_1 or heuristic H_2 is below 10%

Problems with unknown optimal solutions

The mean flow time: To evaluate the solution quality for the mean flow time problem, the percentage error is defined as:

$$\text{Percentage error} = \frac{\bar{F}(H_1) - \bar{F}(LB)}{\bar{F}(LB)} \times 100$$

where, $\bar{F}$ (H_1) denotes the mean flow time obtained by heuristic H_1. $\bar{F}$ (LB) denotes the lower bound on the mean flow time and is calculated as follows:

Let r_i be the minimum completion time for J_i in the four alternative operation types defined by Rule 1. It is expressed as follows:

$$r_i = \min\{p_{i11} + p_{i22}, p_{i11} + p_{i21}, p_{i12} + p_{i22}, p_{i12} + p_{i21}\} \quad i = 1, 2, ..., n$$

Re-index r_i (i = 1, 2, ..., n) such that $r_{[1]} \le r_{[2]} \le \ldots \le r_{[n]}$, where $J_{[q]}$ is the job scheduled at the qth position in the processing sequence and $r_{[q]}$ is the minimum completion time for $J_{[q]}$. The term $p_{[q]jk}$ (q = 1, 2, ..., n; j = 1, 2; k = 1, 2) is the processing time of operation j of $J_{[q]}$ on M_k. Let $l^k_{[q]}$ (k = 1, 2; q = 1, 2, ..., n) be the total processing time of $J_{[q]}$ on M_k. If $r_{[q]} = p_{[q]11} + p_{[q]22}$, then $l^1_{[q]} = p_{[q]11}$ and $l^2_{[q]} = p_{[q]22}$; else if $r_{[q]} = p_{[q]11} + p_{[q]21}$, then $l^1_{[q]} = p_{[q]11} + p_{[q]21}$ and $l^2_{[q]} = 0$; else if $r_{[q]} = p_{[q]12} + p_{[q]22}$, then $l^1_{[q]} = 0$ and $l^2_{[q]} = p_{[q]12} + p_{[q]22}$; else, if $r_{[q]} = p_{[q]12} + p_{[q]21}$, then $l^1_{[q]} = p_{[q]21}$ and $l^2_{[q]} = p_{[q]12}$.

Let $TX^k_{[q]}$ (q = 1, 2, ..., n; k = 1, 2) be the current earliest finish time when $J_{[q]}$ scheduled on M_k It is expressed as follows:

$$TX^k_{[1]} = l^k_{[1]} \quad k = 1, 2$$

$$TX^k_{[q+1]} = TX^k_{[q]} + l^k_{[q]} \quad q = 1, 2, ..., n\text{-}1; k = 1, 2$$

Denote by $F_{[q]}$ (q = 1, 2, ..., n) the flow time of $J_{[q]}$. It is expressed as follows:

$$F_{[q]} = \max_{k=1,2} TX^k_{[q]} \quad q = 1, 2, ..., n$$

Therefore, the lower bound on the mean flow time is as follows:

$$\bar{F}(LB) = \frac{1}{n}\sum_{q=1}^{n} F_{[q]}$$

Since, $r_{[q]}$ is the minimum completion time for $J_{[q]}$. and $TX^k_{[q]}$ doesn't consider the precedent constraints (the idle time of each job is set to zero), $\bar{F}$ (LB) is a valid lower bound to the flow time problem. Table 3 shows the mean flow time and the average percentage error. The average percentage errors of heuristic H_1 for solving the mean flow time problem is 20.09%.

Table 3: Computational results for problems with unknown optimal solutions

	Mean flow time		Makespan	
	Heuristic H_1		Heuristic H_2	
n	(H_1)	Avg. % error	$C_{max}(H_2)$	Avg. % error
20	318.75	18.92	752.25	21.43
50	715.81	20.54	1882.58	21.06
100	1362.25	19.21	3593.28	20.97
200	2712.01	20.22	7268.12	20.05
400	5159.47	21.35	14685.84	21.68
800	10205.50	19.82	29350.45	19.42
1000	12412.12	20.58	36278.21	21.25

The makespan: To evaluate the solution quality for the makespan problem, the percentage error is defined as:

$$\text{Percentage error} = \frac{C_{max}(H_2) - C_{max}(LB)}{C_{max}(LB)} \times 100$$

where, $C_{max}(H_2)$ denotes the makespan obtained by heuristic H_2. $C_{max}(LB)$ denotes the lower bound on the makespan and is estimated as follows:

$$C_{max}(LB) = INT\left(0.5\sum_{i=1}^{n}\left(\min\{p_{i11}, p_{i21}\} + \min\{p_{i22}, p_{i12}\}\right)\right)$$

where, INT (x) denote the greatest integer less than or equal to x.

Table 3 shows the makespan and the average percentage error. The average percentage errors of heuristic H_2 for solving the makespan problem is 20.84%.

CONCLUSIONS

This study considers a two-machine flow-shop scheduling problem in which alternative operations are available to minimize the mean flow time and makespan of jobs, respectively. Two integer programming formulations are proposed and two heuristics are developed for solving large problems. The two integer programming formulations are the 2VFSP_$\bar{F}$ model for the mean flow times and the 2VFSP_C_{max} model for the makespans. Meanwhile, the two heuristics are heuristics H_1 for the mean flow times and heuristics H_2 for the makespans.

The computational results show that the integer programming technique is inefficient even for small problems. For the set of problems with known optimal solutions, the average percentage errors of heuristics H_1 and H_2 are within 10%, respectively. For the set of problems with unknown optimal solutions, the average percentage errors of heuristic H_1 for solving the mean flow time is 20.09%, while the average percentage errors of heuristic H_2 for solving the makespan is 20.84%.

Future research may be conducted to further improve the integer programming formulations. The use of meta-heuristic methods, such as taboo search, simulated annealing or genetic methods, is another method of solving 2VFSP.

REFERENCES

Chen, J.S. and J.C.H. Pan, 2005. Minimizing mean tardiness with alternative operations in two-machine flow-shop scheduling. Int. J. Syst. Sci., 36: 757-766.

Cheng, T.C.E. and G. Wang, 1998. A note on scheduling alternative operations in two-machine flowshops. J. Operat. Res. Soc., 49: 670-673.

Cheng, T.C.E. and G. Wang, 1999. A note on scheduling the two-machine flexible flowshop. IEEE Trans. Robotic Automat., 15: 187-190.

Jackson, J.R., 1956. An extension of Johnson's results on job lot scheduling. Naval Res. Logist. Q., 3: 201-203.

Johnson, S.M., 1954. Optimal two-and three-stage production schedules with setup times included. Naval Res. Logist. Q., 1: 61-68.

Lee, E.J. and P.B. Mirchandani, 1988. Concurrent routing, sequencing, and setups for a two-machine flexible manufacturing cell. IEEE J. Robotic Automat., 4: 256-264.

Liao, C., C. Sun and W. You, 1995. Flow-shop scheduling with flexible processors. Comput. Operat. Res., 22: 297-306.

Low, C., C.J. Hsu and C.T. Su, 2008. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. Comput. Operat. Res., 35: 845-853.

Pan, C.H. and J.S. Chen, 1997. Scheduling alternative operations in two-machine flow-shops. J. Operat. Res. Soc., 48: 533-540.

Pinedo, M., 2008. Scheduling: Theory, Algorithms and Systems. 3rd Edn., Springer, New York, ISBN: 0387789340.

Rinnooy-Kan, A.H.G., 1976. Machine Scheduling Problems: Classification, Complexity and Computations. 1st Edn., Martinus Nijhoff, The Hague, Holland.

Srihari, K. and T.J. Greene, 1988. Alternate routings in CAPP implementation in a FMS. Comput. Ind. Eng., 15: 41-50.

Wilhelm, W.E. and H. Shin, 1985. Effectiveness of alternate operations in a flexible manufacturing system. Int. J. Prod. Res., 23: 65-79.