

Model formulations for the machine scheduling problem with limited waiting time constraints

Jen-Shiang Chen*

Department of Industrial Engineering and Management

Far East College

49 Junghua Road, Shinshr Shiang

Tainan 744

Taiwan

R.O.C.

Jin-Shan Yang

Department of Management and Information Technology

Southern Taiwan University of Technology

Taiwan

R.O.C.

Abstract

This study considers the machine scheduling problem with limited waiting time constraints. We examine the machine environment of the open-shop, job-shop, flow-shop, and permutation flow-shop, and uses makespan as a measure performance. Eight mixed binary integer programming models are developed to optimally solve these problems.

Keywords : Scheduling, waiting time, integer programming, open-shop, job-shop, flow-shop.

1. Introduction

Most studies assume infinite waiting time between any two consecutive operations of each job [9]. There are many industries where the limited time constraint applies. For example, in a wafer fabrication process,

*E-mails: jschenc@ms25.hinet.net, jschen@cc.fec.edu.tw

Journal of Information & Optimization Sciences

Vol. 27 (2006), No. 1, pp. 225–240

© Taru Publications

0252-2667/06 \$2.00 + 0.25

the waiting time after the operations in furnace tubes is limited in order to prevent the absorption of the particulates in air. Except wafer fabrication, the proposed problem also exists in some practical applications such as food production [4], chemical production, and steel production [12].

This study relaxes the no-wait constraint [2, 6, 11, 13, 15] that job must be processed continuously without waiting time between consecutive machines. We consider a scheduling model which is similar to, but somewhat more flexible than the no-wait model. A review of the literature reveals that only two papers have addressed the limited waiting time constraint problem. Yang and Chern [16] considered a two-machine flow-shop sequencing problem with limited waiting time constraints. The objective was to minimize the makespan. This problem was showed to be NP-hard and a branch and bound algorithm was presented to solve the problem. Su [12] considered a hybrid two-stage flow-shop with a batch processor in stage 1 and a single processor in stage 2. The objective was to minimize the makespan. A heuristic algorithm and a mixed integer program were proposed.

In this study, we consider the open-shop, job-shop, flow-shop, and permutation flow-shop scheduling problems with limited waiting time constraints. The following terms are used to define the corresponding scheduling problems:

Open-shop (OS). There are m machines. Each job has to be processed again on each one of the m machines. However, some of these processing times may be zero. There are no restrictions with regard to the routing of each job through the machine environment. The scheduler is allowed to determine a route for each job, and different jobs may have different routes.

Job-shop (JS). In a job shop with m machines, each job has its own predetermined route to follow.

Flow-shop (FS). There are m machines in series. Each job has to be processed on each one of the m machines. All jobs have to follow the same route (i.e., they have to be processed first on machine 1, then machine 2, etc).

Permutation flow-shop (PFS). In flow-shop, after completion on one machine, a job joins the queue at the next machine. Usually, all queues are assumed to operate under the *First In First Out (FIFO)* discipline-

that is, a job cannot 'pass' another while waiting in a queue. If the *FIFO* discipline is in effect, the flow-shop is referred to as a *permutation* flow-shop.

This study assumes that each job has a limited waiting time constraint between any two consecutive operations. Each machine can process only one operation at a time and different operations of the same job cannot be processed simultaneously. Each job visits each machine at most once. Jobs are not preemptive. Buffer storage between any two machines is unlimited. All jobs are available simultaneously at time zero.

The purpose of this study is to concentrate on mixed binary integer programming (BIP) formulations of scheduling operations in OS, JS, FS, and PFS with limited waiting time constraints. This study presents two mixed BIP formulations for solving the OS with limited waiting time constraints problems, two mixed BIP formulations for solving the JS with limited waiting time constraints problems, two mixed BIP formulations for solving the FS with limited waiting time constraints problems, and two mixed BIP formulations for solving the PFS with limited waiting time constraints problems.

The remainder of the paper is organized as follows. Section 2 provides preliminary information. Section 3 addresses OS with limited waiting time constraints. Section 4 addresses JS with limited waiting time constraints. Section 5 addresses FS with limited waiting time constraints. Section 6 addresses PFS with limited waiting time constraints. Section 7 briefly draws conclusions.

2. Preliminaries

Mathematical programming-based scheduling research has received more and more attention from researchers as computer capacity increases and more efficient integer programming (IP) software becomes available [8]. Mathematical programming formulation is a natural way to attack machine scheduling problems [10]. Why are IP models studied at all? Morton and Pentico [7] addressed it with the following two reasons. First, there will always be certain special structure cases that are solvable. If we understand the general approaches, we may recognize them. Second, there are often various partial relaxations of the equations that can be solved and may be useful. Blazewicz et al. [1] surveyed the development

of mathematical programming formulations in scheduling. Most IP problems in scheduling involve mixed BIP, in which some variables are binary and some are continuous. This study employs the mixed BIP technique to describe the formations for the machine environment of the open-shop, job-shop, flow-shop, and permutation flow-shop.

To describe the problem, we introduce the following notations:

Symbol definition

J_i = job number i ;

M_k = machine number k ;

O_{ij} = operation number j of J_i (only used for the job-shop problems);

Problem parameters

M = a very large positive number;

n = number of jobs for processing at time zero;

m = number of machines in the shop;

L_i = the upper limited waiting time of any two consecutive operations of J_i ;

p_{ik} = the processing time of J_i on M_k ;

r_{ik} = 1 if J_i requires M_k ; 0 otherwise (only used for the open-shop problems);

r_{ijk} = 1 if O_{ij} requires M_k ; 0 otherwise (only used for the job-shop problems);

N_i = number of operations of J_i , that is, $N_i = \sum_{k=1}^m r_{ik}$ for the open-shop problems and $N_i = \sum_{j=1}^m \sum_{k=1}^m r_{ijk}$ for the job-shop problems (only used for the open-shop and job-shop problems);

E_k = number of operations that might ever be processed on M_k , i.e.,
 $E_k = \sum_{i=1}^n \sum_{j=1}^{N_i} r_{ijk}$ (only used for the job-shop problems);

Decision variables

C_{\max} = maximum completion time or makespan; $C_{\max} = \max_{i=1}^n C_i$;

g_{ir} = the starting time of the r th operation of J_i (only used for the open-shop problems);

h_{kq} = the starting time of the operation in the sequence position q on M_k (only used for the job-shop, flow-shop, and permutation flow-shop problems);

- $w_{irk} = 1$ if the r th operation of J_i is scheduled on M_k ; 0 otherwise (only used for the open-shop problems);
- $x_{ikq} = 1$ if J_i is scheduled in the q th position for processing on M_k ; 0 otherwise (only used for the job-shop and flow-shop problems);
- $s_{ik} =$ the earliest start time of J_i on M_k (only used for the job-shop, flow-shop, and permutation flow-shop problems);
- $z_{ii'k} = 1$ if J_i precedes $J_{i'}$ on M_k (not necessarily immediately); 0 otherwise (only used for the open-shop, job-shop, and flow-shop problems);
- $z_{ii'}$ = 1 if J_i precedes $J_{i'}$ (not necessarily immediately); 0 otherwise (only used for the permutation flow-shop problems);
- $y_{iq} = 1$ if J_i is assigned to sequence position q ; 0 otherwise (only used for the permutation flow-shop problems).

3. Open-shop with limited waiting time constraints

In open-shops, there are no restrictions with regard to the routing of each job through the machine environment. We must to determine a route for each job, and different jobs may have different routes. This section presents two mixed BIP formulations, namely OS-1 and OS-2, for solving open-shop scheduling problems with limited waiting time constraints.

3.1 *The OS-1 model*

The OS-1 model bases on the precedence relationship between operations of the same job on consecutive machines in the processing requirements. The binary w_{irk} that the model uses is restricted, and specifies the order in which the r th operation of J_i is processed on M_k . The following model employs the concept of one-operation-one-position to describe the open-shop scheduling problems with limited waiting time constraints.

Minimize C_{\max} (1)

Subject to $w_{irk} \leq Mr_{ik} \quad i = 1, 2, \dots, n; r = 1, 2, \dots, N_i; k = 1, 2, \dots, m$ (2)

$$\sum_{r=1}^{N_i} w_{irk} = r_{ik} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (3)$$

$$g_{ir} \leq Mw_{irk} \quad i = 1, 2, \dots, n; r = 1, 2, \dots, N_i; k = 1, 2, \dots, m \quad (4)$$

$$g_{ir} + \sum_{k=1}^m p_{ik}w_{irk} \leq g_{i,r+1} \quad i = 1, 2, \dots, n; r = 1, 2, \dots, N_i - 1 \quad (5)$$

$$g_{i,r+1} - \left(g_{ir} + \sum_{k=1}^m p_{ik} w_{irk} \right) \leq L_i \quad i = 1, 2, \dots, n; \\ r = 1, 2, \dots, N_i - 1 \quad (6)$$

$$g_{i,N_i} + \sum_{k=1}^m p_{ik} w_{i,N_i,k} \leq C_{\max} \quad i = 1, 2, \dots, n \quad (7)$$

$$C_{\max} \geq 0; g_{ir} \geq 0 \quad i = 1, 2, \dots, n; r = 1, 2, \dots, N_i; \\ w_{irk} \text{ is binary } i = 1, 2, \dots, n; r = 1, 2, \dots, N_i; k = 1, 2, \dots, m. \quad (8)$$

In the above model, constraint sets (2) and (3) describe the feasible value of w_{irk} . Constraint set (4) enforces $g_{ir} = 0$ when $w_{irk} = 0$. Constraint set (5) guarantees that the starting time of $O_{i,r+1}$ is later than its finish time of O_{ir} . Furthermore, constraint set (6) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (7) gives the definition of C_{\max} which is to be minimized in the objective function (1). Finally, constraint set (8) specifies the non-negativity of C_{\max} and g_{ir} , and sets up the binary restrictions for w_{irk} .

3.2 The OS-2 model

This model uses binary variable $z_{ii'k}$ to express the 'either-or' relationship for the non-interference restrictions on M_k . The following model applies the concept of non-interference to solve open-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \quad (9)$$

$$\text{Subject to } g_{ir} + p_{ik} \leq g_{i,r+1} + M(1 - r_{ik}) \quad i = 1, 2, \dots, n; \\ r = 1, 2, \dots, N_i - 1; k = 1, 2, \dots, m \quad (10)$$

$$g_{ir} + p_{ik} \leq g_{i'r'} + M(2 - r_{ik} - r_{i'k}) + M(1 - z_{ii'k}) \\ 1 \leq i < i' \leq n; r = 1, 2, \dots, N_i; r' = 1, 2, \dots, N_{i'}; \\ k = 1, 2, \dots, m \quad (11)$$

$$g_{i'r'} + p_{i'k} \leq g_{ir} + M(2 - r_{ik} - r_{i'k}) + Mz_{ii'k} \\ 1 \leq i < i' \leq n; r = 1, 2, \dots, N_i; r' = 1, 2, \dots, N_{i'}; \\ k = 1, 2, \dots, m \quad (12)$$

$$g_{i,r+1} - (g_{ir} + p_{ik}) \leq L_i + M(1 - r_{ik}) \quad i = 1, 2, \dots, n; \\ r = 1, 2, \dots, N_i - 1; k = 1, 2, \dots, m \quad (13)$$

$$\begin{aligned}
 g_{i,N_i} + p_{ik} &\leq C_{\max} + M(1 - r_{ik}) & i = 1, 2, \dots, n; \\
 k &= 1, 2, \dots, m; & (14)
 \end{aligned}$$

$$\begin{aligned}
 C_{\max} \geq 0; g_{ir} \geq 0 & \quad i = 1, 2, \dots, n; r = 1, 2, \dots, N_i; \\
 z_{ii'k} \text{ is binary} & \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m. & (15)
 \end{aligned}$$

Constraint set (10) guarantees that the starting time of $O_{i,r+1}$ is later than its finish time of O_{ir} . Constraint sets (11) and (12) together enforce the requirement that only one operation may be processed on a machine at any time. That is, either $g_{ir} + p_{ik} \leq g_{i'r'} + M(2 - r_{ik} - r_{i'k})$ or $g_{i'r'} + p_{i'k} \leq g_{ir} + M(2 - r_{ik} - r_{i'k})$. If $r_{ik} = r_{i'k} = 1$, then either $g_{ir} + p_{ik} \leq g_{i'r'}$ or $g_{i'r'} + p_{i'k} \leq g_{ir}$. Meanwhile, constraint set (13) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (14) gives the definition of C_{\max} which is to be minimized in the objective function (9). Finally, constraint set (15) specifies the non-negativity of C_{\max} and g_{ir} , and sets up the binary restrictions for $z_{ii'k}$.

3.3 Comparison of OS-1 and OS-2

French [3] stated that the speed with which IP problems can be solved depends upon the number of variables and constraints in the problem, and that the dominant factor is the number of binary variables. Wilson [14], and Liao and You [5] showed that, if two formulations have the same number of binary variables, then the number of constraints is the next most influential element. Accordingly, Table 1 summarizes the sizes of these two models. The OS-2 model has the same number of continuous variables as that of the OS-1 model, but has $(mnN_i - (1/2)mn^2 + (1/2)mn)$ fewer binary variables and $\left(2m \sum_{i=1}^{n-1} \sum_{i'=i+1}^n N_i N_{i'} - 2mn - 2nN_i + n\right)$ more constraints continuous variables. Thus, the OS-2 model is theoretically better than the OS-1 model.

Table 1
Comparison of OS-1 and OS-2

Model	No. of binary variables	No. of constraints	No. of continuous variables
OS-1	mnN_i	$2mnN_i + mn + 2nN_i - n + 1$	$nN_i + 1$
OS-2	$(1/2)mn^2 - (1/2)mn$	$2m \sum_{i=1}^{n-1} \sum_{i'=i+1}^n N_i N_{i'} + 2mnN_i - mn + 1$	$nN_i + 1$

4. Job-shop with limited waiting time constraints

In a job-shop, each job has its own processing order and this may bear no relation to the processing order of any other jobs. This section presents two mixed BIP formulations, namely JS-1 and JS-2, for solving job-shop scheduling problems with limited waiting time constraints.

4.1 The JS-1 model

The JS-1 model bases on the precedence relationship between jobs on consecutive machines in the processing requirements. The binary x_{ikq} that the model uses is restricted, and specifies the order in which jobs are processed on the machine. The following model employs the concept of one-job-one-position to describe the job-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \quad (16)$$

$$\text{Subject to } \sum_{q=1}^{E_k} x_{ikq} = \sum_{j=1}^{N_i} r_{ijk} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (17)$$

$$\sum_{i=1}^n x_{ikq} = 1 \quad k = 1, 2, \dots, m; q = 1, 2, \dots, E_k \quad (18)$$

$$h_{kq} + \sum_{i=1}^n p_{ik} x_{ikq} \leq h_{k,q+1} \quad k = 1, 2, \dots, m; q = 1, 2, \dots, E_k - 1 \quad (19)$$

$$\begin{aligned} & \sum_{k=1}^m r_{ijk} h_{kq} + \sum_{k=1}^m r_{ijk} p_{ik} \\ & \leq \sum_{k=1}^m r_{i,j+1,k} h_{kq'} + M \left(2 - \sum_{k=1}^m r_{ijk} x_{ikq} - \sum_{k=1}^m r_{i,j+1,k} x_{ikq'} \right) \\ & \quad i = 1, 2, \dots, n; j = 1, 2, \dots, N_i - 1; q, q' = 1, 2, \dots, E_k \quad (20) \end{aligned}$$

$$\begin{aligned} & \sum_{k=1}^m r_{i,j+1,k} h_{kq'} - \left(\sum_{k=1}^m r_{ijk} h_{kq} + \sum_{k=1}^m r_{ijk} p_{ik} \right) \\ & \leq L_i + M \left(2 - \sum_{k=1}^m r_{ijk} x_{ikq} - \sum_{k=1}^m r_{i,j+1,k} x_{ikq'} \right) \\ & \quad i = 1, 2, \dots, n; j = 1, 2, \dots, N_i - 1; q, q' = 1, 2, \dots, E_k \quad (21) \end{aligned}$$

$$h_{k,E_k} + \sum_{i=1}^n p_{ik} x_{i,k,E_k} \leq C_{\max} \quad k = 1, 2, \dots, m \quad (22)$$

$$C_{\max} \geq 0 \quad h_{kq} \geq 0, \quad k = 1, 2, \dots, m; q = 1, 2, \dots, E_k$$

$$x_{ikq} = 0 \text{ or } 1 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m; q = 1, 2, \dots, E_k. \quad (23)$$

Constraint set (17) describes the feasible value of x_{ikq} . Constraint set (18) ensures that each of J_i must be placed in a unique position of M_k . Constraint set (19) enforces the technological requirements of each job. Furthermore, constraint set (20) restricts that the starting time of $O_{i,j+1}$ to be no earlier than its finish time of O_{ij} . Constraint set (21) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (22) gives the definition of C_{\max} which is to be minimized in the objective function (16). Finally, constraint set (23) specifies the non-negativity of C_{\max} and h_{kq} , and sets up the binary restrictions for x_{ikq} .

4.2 The JS-2 model

The JS-2 model utilizes integer binary variables $z_{ii'k}$ to express the 'either-or' relationship for the non-interference restrictions for individual machine. The following model applies the concept of non-interference to solve job-shop scheduling problems with limited waiting time constraints.

Minimize C_{\max} (24)

Subject to $\sum_{k=1}^m r_{ijk}(s_{ik} + p_{ik}) \leq \sum_{k=1}^m r_{i,j+1,k}s_{ik} \quad i = 1, 2, \dots, n;$

$$j = 1, 2, \dots, N_i - 1 \quad (25)$$

$$s_{i'k} + p_{i'k} \leq s_{ik} + M(1 - z_{ii'k}) \quad 1 \leq i < i' \leq n;$$

$$k = 1, 2, \dots, m \quad (26)$$

$$s_{ik} + p_{ik} \leq s_{i'k} + Mz_{ii'k} \quad 1 \leq i < i' \leq n;$$

$$k = 1, 2, \dots, m \quad (27)$$

$$\sum_{k=1}^m r_{i,j+1,k}s_{ik} - \sum_{k=1}^m r_{ijk}(s_{ik} + p_{ik}) \leq L_i \quad i = 1, 2, \dots, n;$$

$$j = 1, 2, \dots, N_i - 1 \quad (28)$$

$$\sum_{k=1}^m r_{i,N_i,k}(s_{ik} + p_{ik}) \leq C_{\max} \quad i = 1, 2, \dots, n \quad (29)$$

$$C_{\max} \geq 0, \quad s_{ik} \geq 0 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m$$

$$z_{ii'k} = 0 \text{ or } 1 \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m. \quad (30)$$

Constraint set (25) restricts that the starting time of $O_{i,j+1}$ to be no earlier than its finish time of O_{ij} . Constraint sets (26) and (27) enforce the requirement that only one job may be processed on M_k at any time. If both J_i and $J_{i'}$ are processed on M_k then either $s_{i'k} + p_{i'k} \leq s_{ik}$ or $s_{ik} + p_{ik} \leq s_{i'k}$. Constraint sets (26) and (27) together guarantee that one of the constraints must hold when the other is eliminated since $z_{ii'k}$ is a binary variable and M is a large enough positive number. Furthermore, constraint set (28) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (29) defines C_{\max} to be minimized in the objective function (24). Finally, constraint set (30) specifies the non-negativity of C_{\max} and s_{ik} and the binary restrictions of $z_{ii'k}$.

4.3 Comparison of JS-1 and JS-2

The JS-1 model has $(mnE_k - (1/2)mn^2 + (1/2)mn)$ more binary variables than that of the JS-2 model, has $(2nN_i(E_k)^2 - 2n(E_k)^2 + 2mE_k + 2mn - mn^2 - 2nN_i + n)$ more constraints and $(mE_k - mn)$ more continuous variables. Thus, the JS-2 model is theoretically better than the JS-1 model.

Table 2
Comparison of JS-1 and JS-2

Model	No. of binary variables	No. of constraints	No. of continuous variables
JS-1	mnE_k	$2nN_i(E_k)^2 - 2n(E_k)^2 + 2mE_k + mn + 1$	$mE_k + 1$
JS-2	$(1/2)mn^2 - (1/2)mn$	$mn^2 - mn + 2nN_i - n + 1$	$mn + 1$

5. Flow-shop with limited waiting time constraints

All jobs have to follow the same route in a flow-shop. This section presents two mixed BIP formulations, namely FS-1 and FS-2, for solving flow-shop scheduling problems with limited waiting time constraints.

5.1 The FS-1 model

The FS-1 model bases on the precedence relationship between jobs on consecutive machines in the processing requirements. The binary x_{ikq} that the model uses is restricted, and specifies the order in which jobs are

processed on the machine. The following model employs the concept of one-job-one-position to describe the flow-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \quad (31)$$

$$\text{Subject to } \sum_{q=1}^n x_{ikq} = 1 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (32)$$

$$\sum_{i=1}^n x_{ikq} = 1 \quad k = 1, 2, \dots, m; q = 1, 2, \dots, n \quad (33)$$

$$h_{kq} + p_{ik}x_{ikq} \leq h_{k,q+1} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m; \\ q = 1, 2, \dots, n-1 \quad (34)$$

$$h_{kq} + p_{ik}x_{ikq} \geq h_{k+1,q'} + M(2 - x_{ikq} - x_{i,k+1,q'}) \\ i = 1, 2, \dots, n; k = 1, 2, \dots, m-1; q, q' = 1, 2, \dots, n \quad (35)$$

$$h_{k+1,q'} - (h_{kq} + p_{ik}x_{ikq}) \leq L_i + M(2 - x_{ikq} - x_{i,k+1,q'}) \\ i = 1, 2, \dots, n; k = 1, 2, \dots, m-1; q, q' = 1, 2, \dots, n \quad (36)$$

$$C_{\max} = h_{mn} + \sum_{i=1}^n p_{im}x_{imn} \quad (37)$$

$$C_{\max} \geq 0, h_{kq} \geq 0 \quad k = 1, 2, \dots, m; q = 1, 2, \dots, n; \\ x_{ikq} = 0 \text{ or } 1 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m; q = 1, 2, \dots, n. \quad (38)$$

Constraint set (32) ensures that each operation of J_i is uniquely placed on M_k . Constraint set (33) satisfies the requirement that each position of M_k has a unique operation. Constraint sets (34) and (35) enforce the technological requirements of each job. Meanwhile, constraint set (36) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (37) gives the definition of C_{\max} which is to be minimized in the objective function (31). Finally, constraint set (38) specifies the non-negativity of C_{\max} and h_{kq} , and sets up the binary restrictions for x_{ikq} .

5.2 The FS-2 model

The FS-2 model utilizes integer binary variables $z_{ii'k}$ to express the 'either-or' relationship for the non-interference restrictions for individual machine. The following model applies the concept of non-interference to solve flow-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \tag{39}$$

$$\text{Subject to } s_{ik} + p_{ik} \leq s_{i,k+1} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m - 1 \tag{40}$$

$$s_{i'k} + p_{i'k} \leq s_{ik} + M(1 - z_{ii'k}) \quad 1 \leq i < i' \leq n; \\ k = 1, 2, \dots, m \tag{41}$$

$$s_{ik} + p_{ik} \leq s_{i'k} + Mz_{ii'k} \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m \tag{42}$$

$$s_{i,k+1} - (s_{ik} + p_{ik}) \leq L_i \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m - 1 \tag{43}$$

$$s_{im} + p_{im} \leq C_{\max} \quad i = 1, 2, \dots, n \tag{44}$$

$$C_{\max} \geq 0, s_{ik} \geq 0 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \\ z_{ii'k} = 0 \text{ or } 1 \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m. \tag{45}$$

Constraint set (40) enforces the technological requirements of each job. Constraint sets (41) and (42) enforce the requirement that only one job may be processed on M_k at any time. If both J_i and $J_{i'}$ are processed on M_k then either $s_{i'k} + p_{i'k} \leq s_{ik}$ or $s_{ik} + p_{ik} \leq s_{i'k}$. Constraint sets (41) and (42) together guarantee that one of the constraints must hold when the other is eliminated since $z_{ii'k}$ is a binary variable and M is a large enough positive number. Furthermore, constraint set (43) ensures that the waiting time of any two consecutive operations of J_i cannot greater than its upper limited waiting time L_i . Constraint set (44) defines C_{\max} to be minimized in the objective function (39). Finally, constraint set (45) specifies the non-negativity of C_{\max} and s_{ik} and the binary restrictions of $z_{ii'k}$.

5.3 Comparison of FS-1 and FS-2

The FS-1 model has the same number of continuous variables as the FS-2 model, but it has $((1/2)mn^2 + (1/2)mn)$ more binary variables and $(2mn^3 - 2n^3 + n + 1)$ more constraints. Thus, the FS-2 model is theoretically better than the FS-1 model.

Table 3
Comparison of FS-1 and FS-2

Model	No. of binary variables	No. of constraints	No. of continuous variables
FS-1	mn^2	$2mn^3 - 2n^3 + mn^2 + mn + 2$	$mn + 1$
FS-2	$(1/2)mn^2 - (1/2)mn$	$mn^2 + mn - n + 1$	$mn + 1$

6. Permutation flow-shop with limited waiting time constraints

In a permutation flow-shop, a job cannot 'pass' another while waiting in a queue. This section presents two mixed BIP formulations, namely PFS-1 and PFS-2, for solving permutation flow-shop scheduling problems with limited waiting time constraints.

6.1 The PFS-1 model

The FS-1 model bases on the precedence relationship between jobs on consecutive machines in the processing requirements. The binary y_{iq} that the model uses is restricted, and specifies the order in which jobs are processed. The following model employs the concept of one-job-one-position to describe the permutation flow-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \quad (46)$$

$$\text{Subject to } \sum_{q=1}^n y_{iq} = 1 \quad i = 1, 2, \dots, n \quad (47)$$

$$\sum_{i=1}^n y_{iq} = 1 \quad q = 1, 2, \dots, n \quad (48)$$

$$h_{11} = 0 \quad (49)$$

$$h_{1q} + \sum_{i=1}^n p_{i1} y_{iq} = h_{1,q+1} \quad q = 1, 2, \dots, n-1 \quad (50)$$

$$h_{k1} + \sum_{i=1}^n p_{ik} y_{i1} = h_{k+1,1} \quad k = 1, 2, \dots, m-1 \quad (51)$$

$$h_{kq} + \sum_{i=1}^n p_{ik} y_{iq} \leq h_{k+1,q} \quad k = 1, 2, \dots, m-1; q = 2, 3, \dots, n; \quad (52)$$

$$h_{kq} + \sum_{i=1}^n p_{ik} y_{iq} \leq h_{k,q+1} \quad k = 2, 3, \dots, m; q = 1, 2, \dots, n-1 \quad (53)$$

$$h_{k+1,q} - (h_{kq} + p_{ik}) \leq L_i + M(1 - y_{iq}) \quad i = 1, 2, \dots, n; \\ k = 1, 2, \dots, m-1; q = 1, 2, \dots, n \quad (54)$$

$$C_{\max} = h_{nm} + \sum_{i=1}^n p_{im} y_{in} \quad (55)$$

$$C_{\max} \geq 0, h_{kq} \geq 0 \quad k = 1, 2, \dots, m; q = 1, 2, \dots, n; \\ y_{iq} = 0 \text{ or } 1 \quad i = 1, 2, \dots, n; q = 1, 2, \dots, n. \quad (56)$$

Constraint sets (47) and (48) are the classical assignment problem, with (47) insuring that each job is assigned to just one sequence position, while (48) insures that each sequence position is filled with only one job. Constraint sets (49), (50), and (51) insure there is no idle time on M_1 , and that J_1 processes on all m machines without delay. Constraint set (52) insures that the start of each job on M_{k+1} is no earlier than its finish on M_k . Furthermore, constraint set (53) insures that the job in position $q+1$ in the sequence does not start on M_k until the job in position q in the sequence has completed its processing on that machine. Constraint set (54) ensures that the waiting time of any two consecutive operations of J_i cannot be greater than its upper limited waiting time L_i . Constraint set (55) defines C_{\max} to be the finish time of the last job processed on M_m . Finally, constraint set (56) specifies the non-negativity of C_{\max} and h_{kq} , and sets up the binary restrictions for y_{iq} .

6.2 The PFS-2 model

The PFS-2 model utilizes integer binary variables $z_{ii'}$ to express the 'either-or' relationship for the non-interference restrictions. The following model applies the concept of non-interference to solve permutation flow-shop scheduling problems with limited waiting time constraints.

$$\text{Minimize } C_{\max} \quad (57)$$

$$\text{Subject to } s_{ik} + p_{ik} \leq s_{i,k+1} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m-1 \quad (58)$$

$$s_{i'k} + p_{i'k} \leq s_{ik} + M(1 - z_{ii'}) \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m \quad (59)$$

$$s_{ik} + p_{ik} \leq s_{i'k} + Mz_{ii'} \quad 1 \leq i < i' \leq n; k = 1, 2, \dots, m \quad (60)$$

$$s_{i,k+1} - (s_{ik} + p_{ik}) \leq L_i \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m-1 \quad (61)$$

$$s_{im} + p_{im} \leq C_{\max} \quad i = 1, 2, \dots, n \quad (62)$$

$$C_{\max} \geq 0, s_{ik} \geq 0 \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m$$

$$z_{ii'} = 0 \text{ or } 1 \quad 1 \leq i < i' \leq n \quad (63)$$

Constraint set (58) insures that each job's start time on M_{k-1} is no earlier than that job's start time on M_k plus that job's processing time on M_k . Constraint sets (59) and (60) are the paired disjunctive constraints which insure that J_i either precede $J_{i'}$ or follows $J_{i'}$ in the sequence, but not both. If both J_i and $J_{i'}$ are processed on M_k then either $s_{i'k} + p_{i'k} \leq s_{ik}$ or $s_{ik} + p_{ik} \leq s_{i'k}$. Constraint sets (59) and (60) together guarantee that one of the constraints must hold when the other is eliminated since $z_{ii'}$ is a binary variable and M is a large enough positive number. Meanwhile,

constraint set (61) ensures that the waiting time of any two consecutive operations of J_i cannot be greater than its upper limited waiting time L_i . Constraint set (62) equates makespan to the maximum completion time of all jobs on the last machine. Finally, constraint set (63) specifies the non-negativity of C_{\max} and s_{ik} and the binary restrictions of z_{ij} .

6.3 Comparison of PFS-1 and PFS-2

The PFS-2 model has the same number of continuous variables as that of the PFS-1 model, but has $((1/2)n^2 + (1/2)n)$ more binary variables and $(n^2 - mn + m - 2n - 2)$ fewer constraints. Thus, the PFS-2 model is theoretically better than the PFS-1 model.

Table 4
Comparison of PFS-1 and PFS-2

Model	No. of binary variables	No. of constraints	No. of continuous variables
PFS-1	n^2	$mn^2 - n^2 + 2mn - m + n + 3$	$mn + 1$
PFS-2	$(1/2)n^2 - (1/2)n$	$mn^2 + mn - n + 1$	$mn + 1$

7. Conclusions

This study considers the open-shop, job-shop, flow-shop, and permutation flow-shop scheduling problems with limited waiting time constraints. Eight mixed BIP models are proposed for solving these scheduling problems. The models of OS-1 and OS-2 are for the open-shop with limited waiting time constraints. The models of JS-1 and JS-2 are for the job-shop with limited waiting time constraints. The models of FS-1 and FS-2 are for the flow-shop with limited waiting time constraints, while the models PFS-1 and PFS-2 are for the permutation flow-shop with limited waiting time constraints. This study employs ILOG OPL and CPLEX to verify the accuracy of the above eight proposed mixed BIP models. Theoretically speaking, OS-2 is better than OS-1, JS-2 is better than JS-1, FS-2 is better than FS-1, and PFS-2 is better than PFS-1.

References

- [1] J. Blazewicz, M. Dror and J. Weglarz, Mathematical programming formulations for machine scheduling: a survey, *European Journal of Operational Research*, Vol. 51 (1991), pp. 283–300.

- [2] J. L. Bouquarcd, J. C. Billaut, M. A. Kubzin and V. A. Strusevich, Two-machine flow shop scheduling problems with no-wait jobs, *Operations Research Letters*, Vol. 33 (2005), pp. 255–262.
- [3] S. French, *Sequencing and Scheduling: An introduction to the Mathematics of the Job-Shop*, Ellis Horwood, Chichester, U.K., 1982.
- [4] A. Hodson, P. Muhlemann and D. H. R. Price, A microcomputer based solution to a practical scheduling problem, *Journal of the Operational Research Society*, Vol. 36 (1985), pp. 903–914.
- [5] C. L. Liao and C. T. You, An improved formulation for the job-shop scheduling problem, *Journal of the Operational Research Society*, Vol. 43 (1992), pp. 1047–1954.
- [6] C. Y. Liaw, C. Y. Cheng and M. Chen, Scheduling two-machine no-wait open shops to minimize makespan, *Computers and Operations Research*, Vol. 32 (2005), pp. 901–917.
- [7] T. E. Morton and D. W. Pentico, *Heuristic Scheduling Systems*, Wiley, New York, 1993.
- [8] C. H. Pan, A study of integer programming formulations for scheduling problems, *International Journal of Systems Science*, Vol. 28 (1997), pp. 33–41.
- [9] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall, New Jersey, 2002.
- [10] A. H. G. Rinnooy Kan, *Machine Scheduling Problems: Classification, Complexity and Computations*, Martinus Nijhoff, The Hague, Holland, 1976.
- [11] C. J. Schuster and J. M. Framinan, Approximative procedures for no-wait job shop scheduling, *Operations Research Letters*, Vol. 31 (2003), pp. 308–318.
- [12] L. H. Su, A hybrid two-stage flowshop with limited waiting time constraints, *Computers & Industrial Engineering*, Vol. 44 (2003), pp. 409–424.
- [13] J. B. Wang and Z. Q. Xia, No-wait or no-idle permutation flowshop scheduling with dominating machines, *Journal of Applied Mathematics and Computing*, Vol. 17 (2005), pp. 419–432.
- [14] J. M. Wilson, Alternative formulations of a flow-shop scheduling problem, *Journal of the Operational Research Society*, Vol. 40 (1989), pp. 395–399.
- [15] G. J. Woeginger, Inapproximability results for no-wait job shop scheduling, *Operations Research Letters*, Vol. 32 (2004), pp. 320–325.
- [16] D. L. Yang and M. S. Chern, Two-machine flowshop sequencing problem with limited waiting time constraints, *Computers & Industrial Engineering*, Vol. 28 (1995), pp. 63–70.

Received July, 2005