

Special Cases on Two Machine Flow Shop Scheduling with Weighted WIP Costs*

Jaehwan Yang**

School of Business Administration, University of Seoul,
13 Siripdae-gil, Dongdaemun-gu, Seoul. 130-743, Korea,

(Received: July 13, 2009 / Revised: September 25, 2009 / Accepted: September 25, 2009)

ABSTRACT

In this paper, we consider a relatively new two-machine flow shop scheduling problem where the unit time WIP cost increases as a job passes through various stages in the production process, and the objective is to minimize the total WIP (work-in-process) cost. Specifically, we study three special cases of the problem. First, we consider the problem where processing times on machine 1 are identical. Second, the problem with identical processing times on machine 2 is examined. The recognition version of the both problems is unary NP-complete (or NP-complete in strong sense). For each problem, we suggest two simple and intuitive heuristics and find the worst case bound on relative error. Third, we consider the problem where the processing time of a job on each machine is proportional to a base processing time. For this problem, we show that a known heuristic finds an optimal schedule.

Keywords: WIP Cost, Machine Flow Shop Scheduling, Heuristic Analysis

1. Introduction

In this paper we consider scheduling problems where the objective is associated with the *work in process* (WIP). We assume that the value of the product and the WIP cost increase as labor and material are added to a product. We call this new scheduling problem a two machine scheduling problem with weighted WIP cost. A major change from the classical flow shop scheduling problem is that the unit time WIP cost does not remain constant.

Minimizing WIP costs is an important criterion for many manufacturing facilities.

* This work was supported by the University of Seoul 2008 Research Fund.

** Corresponding author, E- mail: jyang@uos.ac.kr

Level of WIP inventory is often considered as one of the measures for production efficiency. While it is difficult to operate production lines without WIP inventory, most companies try to minimize WIP [16]. Some companies intentionally keep WIP inventory at work centers to improve utilization [18] or to hedge against due date penalties. However, for most companies, reducing unnecessary WIP inventory is a critical goal.

When the WIP cost per unit time of a job is constant throughout the manufacturing process, any scheduling problem where the objective is to minimize total completion time also minimizes the WIP inventory. However, this type of regular flow shop scheduling problems usually assume that WIP costs are equal and unchanged throughout the entire production process. In reality, the WIP costs increase while the production progresses in most industries or they may decrease in some industries such as refining industry.

A typical example of the problem can be found in any discrete manufacturing facility where there exists a sequential production process with multiple stages. For instance, in a consumer electronics, say television, manufacturing facility there usually exist three stages in the production process and they are automatic insertion (AI), manual insertion (MI), and final assembly (FA). During the AI stage, all small components are inserted on a printed circuit board (PCB) by an automatic insertion machine. Once the AI is finished, the PCB is moved to the next stage to be installed with medium size components manually by employees. Then, it finally moves to the next stage for the FA. Observe that as a WIP moves from one stage to a next stage, the value of the WIP increases. This is because the WIP at one stage contains additional valuable components and more labor hours compared to that at a previous stage. Since inventory holding cost of an item increases proportionally with the value of the item, the cost of a WIP at a particular stage should be different depending on the degree of completeness of the WIP.

This work considers a flow shop problem where there are two machines. Hence, this problem is a generalization of the two machine flow shop problem where the objective is to minimize the total completion time. The recognition version of the two machine flow shop problem is unary NP-complete (or NP-complete in strong sense) [8]. Several studies focus on developing efficient algorithms (see [1, 6, 7, 11, 13, 17, 19]). A heuristic procedure based on selecting jobs in shortest total processing time order is developed by Gonzalez and Sahni [9]. They show that the relative error of

their procedure is bounded above by 2.

The two machine scheduling problem with weighted WIP cost is first introduced by Yang and Posner [21]. They establish complexity of the problem and introduce several simple and intuitive heuristics. Yang [20] extends the problem further by introducing several variations of the problem and establishes their complexity. He also considers the case where WIP cost decreases as production progresses.

In this paper, we consider three special cases of the two machine scheduling problem with weighted WIP cost. First, we consider the problem where processing times on machine 1 are identical. Second, the problem with identical processing times on machine 2 is examined. The recognition version of the both problems is unary NP-complete [20]. Hence, using efficient heuristics to solve the problems is a suitable option. For each problem, we suggest two simple and intuitive heuristics which can be easily utilized in practice and find worst case bounds on relative error. Third, we also consider the problem where processing time of a job on each machine is proportional to a base processing time. For this problem, we show that a known heuristic, which is developed by Yang and Posner [21] to solve a general version of the problem, actually finds an optimal solution.

2. Notation and Preliminaries

The parameters of the problem are

N = set of jobs = $\{1, 2, \dots, n\}$

M = set of machines = $\{1, 2\}$

M_i = machine i for $i \in M$

p_{ij} = processing time of job j on machine i for $i \in M$ and $j \in N$.

The variables in our model are

σ_i = schedule of all jobs on machine i for $i \in M$

σ = schedule of all jobs = (σ_1, σ_2)

$C_{ij}(\sigma)$ = completion time of job j on machine i in schedule σ for $i \in M$ and $j \in N$

$S_{ij}(\sigma)$ = start time of job j on machine i in schedule σ for $i \in M$ and $j \in N$

$Q_{ij}(\sigma)$ = waiting time of job j before it starts processing on machine i in schedule σ for $i \in M$ and $j \in N$

$W_j(\sigma)$ = work in process cost for job j in schedule σ for $j \in N$

σ^* = an optimal schedule

z^* = value of optimal schedule σ^* .

When no confusion exists, we replace $C_{ij}(\sigma)$, $S_{ij}(\sigma)$, $Q_{ij}(\sigma)$ and $W_j(\sigma)$ with C_{ij} , S_{ij} , Q_{ij} and W_j , respectively. We let $[j]$ indicate the job in the j th position in schedule σ . For example, $p_{1[4]}$ is the processing time on M_1 of the fourth job in schedule σ .

The standard classification scheme for scheduling problems is $\alpha_1 | \alpha_2 | \alpha_3$, where α_1 describes the machine structure, α_2 gives the job characteristics and restrictions, and α_3 defines the objective [10]. We extend this scheme to provide for WIP costs by using W_j in the α_3 field. Following the standard scheduling classification schedule of Graham *et al.* [10], we refer to the problem of minimizing the WIP cost in two machine flow shop where processing times on M_1 are identical as $F2 | p_{1j} = p_1 | \sum W_j$.

A *schedule* defines a job order for each machine and a *permutation schedule* is a schedule in which every machine has the same job order. For $F2 \parallel \sum W_j$, the jobs are available at the start of the planning process. Also, no preemptions are allowed.

In a two machine flow shop system, there are four different types of WIP costs:

Type 1: Before being processed by the first machine (value of raw material).

Type 2: While being processed by M_1 (value of raw material + added components at M_1).

Type 3: Between being processed by M_1 and M_2 (value of raw material + added components at M_1 + labor and depreciation of M_1).

Type 4: While being processed by M_2 (value of raw material + added components at M_1 + labor and depreciation of M_1 + added components at M_2).

Let w_i be the weight (value) associated with WIP cost Type i , for $i = 1, 2, \dots, 4$ and we assume that $w_1 \leq w_2 \leq w_3 \leq w_4$. Completion time of job j is $C_{2j} = Q_{1j} + p_{1j} + Q_{2j} + p_{2j}$. It may be true that the actual weight varies depending on the job. But, in most flow lines the jobs seem to be similar due to the repetitive characteristic of the flow line. Consequently, a reasonable model is that the cost is proportional to the time spent. Thus, the WIP cost for job j is

$$W_j = w_1 Q_{1j} + w_2 p_{1j} + w_3 Q_{2j} + w_4 p_{2j}. \quad (1)$$

Note that $w_2 p_{1j}$ and $w_4 p_{2j}$ are fixed regardless of the job sequence. As a result, our goal is to minimize $\sum_{j=1}^n (w_1 Q_{1j} + w_3 Q_{2j})$.

The heuristic procedures we develop use the following well known rule to determine the order in which the jobs are processed [9].

SPT (Shortest Processing Time): When M_1 becomes available, an unscheduled job with the shortest total processing time is selected for processing.

Unless otherwise noted, we assume that the jobs are indexed in SPT order, ie. $p_{1j} + p_{2j} \leq p_{1,j+1} + p_{2,j+1}$ for $j = 1, 2, \dots, n-1$.

3. Problem Statement

In this section, we specifically introduce each of the special cases considered in the paper. Also, we describe how this paper is organized around these three special cases.

The first special case considered in the paper is the problem where processing times on machine 1 are identical, and the second special case is the problem with identical processing times on machine 2 is examined. More formally, the first case is the problem of minimizing the WIP cost in two machine flow shop where processing times on M_1 are identical, $F2 | p_{1j} = p_1 | \sum W_j$ and the second case is the problem of minimizing the WIP cost in two machine flow shop where processing times on M_2

are identical, $F2 | p_{1j} = p_{2j} | \sum W_j$. A typical example of these special cases can be found in a multi-stage manufacturing environment. In the previous example of consumer electronics manufacturing, two different models of TV sets, which belong to the same product lines and so produced in the same production line, may have an identical processing time at MI stage but they may have different processing times at FA stage and vice versa.

The recognition version of the both problems is unary NP-complete [20]. Hence, using efficient heuristics to solve the problems is a suitable option. In Sections 5 and 6, we introduce lower bounds for each of the problem. In Section 7 and 8, we suggest two simple and intuitive heuristics which can be easily utilized in practice for the two problems and find worst case bounds on relative error. For each heuristic, we find either a tight upper bound or an asymptotically attainable upper bound on relative error for the first two special cases.

The third special case we considered in the paper is the problem where processing time of a job on each machine is proportional to a base processing time. Formally, it is the problem of minimizing the WIP cost in two machine flow shop where $sp_{1j} = p_{2j}$ for all $j \in N$ and $s > 0$, $F2 \parallel \sum W_j$. An important practical situation where this occurs is when each operation consists of several identical tasks. For example, consider a two-step process where the first machine makes holes on a board and the second machine places a component in each hole. The processing time on each machine depends on the number of holes and the number of components that are put into the holes. Then, the processing time of a job on one machine is proportional to the processing time of the job on the other machine. Another example is when the processing time of a machine is proportional to a quantity associated with an order from a customer.

Early work on this problem appears in Ow [14]. Other works include Allahverdi [2], Choi *et al.* [4], Cheng and Shakhlevich [3], Hou and Hoogeveen [12], and Shakhlevich *et al.* [15].

In Section 9, we introduce another known heuristic for the third special case and finally in Section 10, we show that this heuristic finds an optimal schedule when the processing time of a job on one machine is proportional to processing time of the job on the other machine.

4. Basic Results

We begin by reviewing complexity of problems $F2|p_{1j} = p_1|\sum W_j$ and $F2|p_{2j} = p_2|\sum W_j$. The following theorem shows the complexity of the two special cases of problem $F2\|\sum W_j$.

Theorem 1 ([20]) *Problems $F2|p_{1j} = p_1|\sum W_j$ and $F2|p_{2j} = p_2|\sum W_j$ are unary NP-hard.*

We now discuss a couple of optimality properties. The first result shows that unlike most flow shop problems, the optimal solution to $F2\|\sum W_j$ may not be a permutation schedule.

Lemma 1 ([21]) *There exist instances where there is no optimal permutation schedule for problem $F2\|\sum W_j$.*

The next result shows that some optimal schedules for problem $F2\|\sum W_j$ requires inserted idle time on M_1 . As a result of Lemma 1, we may need to specify a schedule for each machine when necessary.

Lemma 2 ([21]) *There are instances of problem $F2\|\sum W_j$ where an optimal schedule requires inserted idle time on M_1 .*

As a result of Lemma 2, when we describe a schedule, we may need to specify the start time (or completion time) of each job as well as a job order when necessary. Having inserted idle time is an important distinction between problems $F2\|\sum C_j$ and $F2\|\sum W_j$. For problem $F2\|\sum C_j$, there always exists at least one optimal permutation schedule without any inserted idle time on M_1 [5].

5. Lower Bounds for $F2|p_{1j} = p_1|\sum W_j$

We establish two lower bounds on the value of a schedule for problem $F2|p_{1j} = p_1|\sum W_j$

$\sum W_j$. These lower bounds are used in the analysis of heuristics. Both bounds are based on the condition that there is no wait for machine 2. The first bound assumes that each job is processed as quickly as possible on M_1 .

Lemma 3

$$z^{L1} = \frac{w_1 n(n-1)p_1}{2} + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}.$$

Proof. From the definition of the WIP costs,

$$W_{[j]} \geq w_1(j-1)p_1 + w_2 p_1 + w_4 p_{2[j]}$$

for $j \in N$. Since jobs are ordered such that $p_1 + p_{2j} \leq p_1 + p_{2,j+1}$ for $j = 1, 2, \dots, n-1$ (or indexed in SPT order),

$$\sum_{j=1}^n W_j \geq \frac{w_1 n(n-1)p_1}{2} + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}. \quad \square$$

The next bound assumes that each job is processed as quickly as possible on M_2 .

Lemma 4

$$z^{L2} = w_1 \sum_{j=1}^{n-1} (n-j)p_{2j} + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}.$$

Proof. Because $w_1 \leq w_2 \leq w_3$, it is preferable to have a job wait before M_1 rather than M_2 . Thus for $j \in N$,

$$W_{[j]} \geq w_1(p_1 + \sum_{k=1}^{j-1} p_{2[k]} - p_1) + w_2 p_1 + w_4 p_{2[j]}.$$

Since the jobs are indexed in SPT order,

$$\sum_{j=1}^n W_j \geq w_1 \sum_{j=1}^{n-1} (n-j)p_{2j} + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}. \quad \square$$

We use z^{L1} and z^{L2} to bound z^* from below.

6. Lower Bounds for $F2 | p_{2j} = p_2 | \sum W_j$

In this section we establish another couple of lower bounds on the value of a schedule for problem $F2 | p_{2j} = p_2 | \sum W_j$. These lower bounds are used in the analysis of heuristics for problem $F2 | p_{2j} = p_2 | \sum W_j$. The analysis is similar to that in the previous section.

Lemma 5

$$z^{L3} = w_1 \sum_{j=1}^{n-1} (n-j)p_{1j} + w_2 \sum_{j=1}^n p_{1j} + w_4 np_2.$$

Proof. From the definition of the WIP costs,

$$W_{1j} \geq w_1 \sum_{k=1}^{j-1} p_{1[k]} + w_2 p_{1[j]} + w_4 p_2$$

for $j \in N$. Since the jobs are indexed in SPT order,

$$\sum_{j=1}^n W_j \geq w_1 \sum_{j=1}^{n-1} (n-j)p_{1j} + w_2 \sum_{j=1}^n p_{1j} + w_4 np_2. \quad \square$$

Lemma 6

$$z^{L4} = w_1 np_{11} + \frac{w_1 n(n-1)p_2}{2} + (w_2 - w_1) \sum_{j=1}^n p_{1j} + w_4 np_2.$$

Proof. Because $w_1 \leq w_2 \leq w_3$, it is preferable to have a job wait before M_1 rather than M_2 . Thus for $j \in N$,

$$W_{1j} \geq w_1 [p_{1[1]} + (j-1)p_2 - p_{1[j]}] + w_2 p_{1[j]} + w_4 p_2.$$

Since the jobs are indexed in SPT order,

$$\sum_{j=1}^n W_j \geq w_1 n p_{11} + \frac{w_1 n(n-1)p_2}{2} + (w_2 - w_1) \sum_{j=1}^n p_{1j} + w_4 n p_2. \quad \square$$

We use z^{L3} and z^{L4} to bound z^* from below.

7. Heuristic with No Inserted Idle Time

In this section, we introduce and analyze the worst case behavior of a heuristic. This heuristic is presented in Gonzalez and Sahni [9] to solve problem $F2 \parallel \sum C_j$. We apply the heuristic to problems $F2 \mid p_{1j} = p_1 \mid \sum W_j$ and $F2 \mid p_{2j} = p_2 \mid \sum W_j$. The heuristic processes jobs in SPT order and does not allow any inserted idle time. Consequently, it is a reasonable choice for managers who are interested in maximizing machine utilization.

Heuristic GS

0. Reindex jobs so that $p_{1j} + p_{2j} \leq p_{1,j+1} + p_{2,j+1}$ for $j = 1, 2, \dots, n-1$.

1. Schedule job 1 such that $C_{11} = p_{11}$ and $C_{21} = p_{11} + p_{21}$.

Schedule jobs 2, 3, \dots , n in index order on M_1 and M_2 such that $C_{1j} = C_{1,j-1} + p_{1j}$ and $C_{2j} = \max\{C_{2,j-1}, C_{1j}\} + p_{2j}$ for $j = 2, 3, \dots, n$.

2. Calculate W_j for $j = 1, 2, \dots, n$.

Output $\sum_{j=1}^n W_j$ and stop.

Heuristic GS runs in $O(n \log n)$ time. The resulting solution is a permutation schedule where there is no inserted idle time on M_1 . Let σ^{GS} be the schedule generated by Heuristic GS and z^{GS} be the cost of schedule σ^{GS} . We analyze Heuristic GS and find an upper bound on the relative error.

Example 1. Consider a two machine, three job problem where $w_1 = w_2 = 1$, $w_3 = w_4 = 1.5$, $p_{11} = p_{12} = 1$, $p_{13} = 5$, and $p_{21} = p_{22} = p_{23} = 3.5$. Heuristic GS generates the schedule $\sigma^{GS} = (1, 2, 3)$ where there exists no inserted idle time on either machine (see Figure 1). The cost is $z^{GS} = \sum_{j=1}^3 (w_1 Q_{1j} + w_2 p_{1j} + w_3 Q_{2j} + w_4 p_{2j}) = w_1(0+1+2) + w_2(1+1+5) + w_3(0+2.5+1) + w_4(3.5+3.5+3.5) = 31$. However, an optimal schedule $\sigma^* = (1, 2, 3)$ has inserted idle time of 1 on M_1 before job 2 starts. The cost of the optimal schedule is $z^* = w_1(0+2+3) + w_2(1+1+5) + w_3(0+1.5+0) + w_4(3.5+3.5+3.5) = 30$.

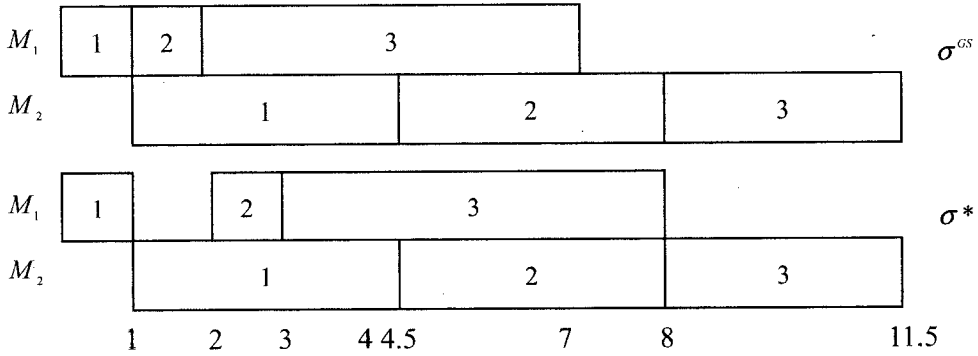


Figure 1. Example 1

7.1 Analysis for $F2 | p_{1j} = p_1 | \sum W_j$

We first analyze the worst case behavior of Heuristic GS for problem $F2 | p_{1j} = p_1 | \sum W_j$. The following theorem establishes a tight bound of Heuristic GS.

Theorem 2

$$z^{GS} / z^* \leq \frac{w_1 n(n-1)p_1 + 2w_3 \sum_{j=1}^{n-1} (n-j)p_{2j} + 2w_2 np_1 + 2w_4 \sum_{j=1}^n p_{2j}}{w_1 \max\{n(n-1)p_1, 2 \sum_{j=1}^{n-1} (n-j)p_{2j}\} + 2w_2 np_1 + 2w_4 \sum_{j=1}^n p_{2j}}$$

Further, this bound is tight.

Proof. From Lemmas 3 and 4,

$$z^* \geq \max\{z^{L1}, z^{L2}\}. \quad (2)$$

By the construction of GS,

$$W_j(\sigma^{GS}) \leq w_1(j-1)p_1 + w_2p_1 + w_3 \sum_{k=1}^{j-1} p_{2k} + w_4 p_{2j}$$

for $j \in N$. Therefore,

$$\sum_{j=1}^n W_j(\sigma^{GS}) \leq \frac{w_1 n(n-1)p_1}{2} + w_2 np_1 + w_3 \sum_{j=1}^{n-1} (n-j)p_{2j} + w_4 \sum_{j=1}^n p_{2j}. \quad (3)$$

From (2) and (3),

$$\begin{aligned} \frac{z^{GS}}{z^*} &\leq \frac{z^{GS}}{\max\{z^{L1}, z^{L2}\}} \\ &= \frac{w_1 n(n-1)p_1 + 2w_3 \sum_{j=1}^{n-1} (n-j)p_{2j} + 2w_2 np_1 + 2w_4 \sum_{j=1}^n p_{2j}}{2 \max\{z^{L1}, z^{L2}\}} \\ &= \frac{w_1 n(n-1)p_1 + 2w_3 \sum_{j=1}^{n-1} (n-j)p_{2j} + 2w_2 np_1 + 2w_4 \sum_{j=1}^n p_{2j}}{w_1 \max\{n(n-1)p_1, 2 \sum_{j=1}^{n-1} (n-j)p_{2j}\} + 2w_2 np_1 + 2w_4 \sum_{j=1}^n p_{2j}}. \end{aligned}$$

Now, we show that the bound is tight. Consider the instance where there are n jobs with processing times $p_{1j} = 0$ and $p_{2j} = 1$ for $j = 1, 2, \dots, n$. Since $p_{1j} + p_{2j} = p_{1k} + p_{2k}$ for all jobs $j, k \in N$, any job sequence can result from Heuristic GS. Suppose that permutation schedule $\sigma^{GS} = (1, 2, \dots, n)$ where there exists no inserted idle time. Then, the solution value is $z^{GS} = w_3(n^2 - n)/2 + w_4 n$. An optimal permutation schedule $\sigma^* = (1, 2, \dots, n)$ has the same sequence as σ^{GS} but schedule jobs on M_1

such that there is no wait time before a job on M_2 processes. Then, solution value $z^* = w_1(n^2 - n)/2 + w_4n$. Consequently, the relative error is $[w_3n(n-1) + 2w_4n]/[w_1n(n-1) + 2w_4n]$.

Observe that for this example, $\sum_{j=1}^{n-1} (n-j)p_{2j} = (n^2 - n)/2$. Hence, $\max\{n(n-1)p_1, 2\sum_{j=1}^{n-1} (n-j)p_{2j}\} = \max\{0, n^2 - n\} = n^2 - n$. Thus,

$$\begin{aligned} & \frac{w_1n(n-1)p_1 + 2w_3\sum_{j=1}^{n-1} (n-j)p_{2j} + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}{w_1\max\{n(n-1)p_1, 2\sum_{j=1}^{n-1} (n-j)p_{2j}\} + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}} \\ &= \frac{w_3n(n-1) + 2w_4n}{w_1n(n-1) + 2w_4n}. \end{aligned}$$

Therefore, the bound is tight. \square

Note that the bound has a minimum value of 1 when $w_1 = w_3$.

Remark 1 ([21]) For problem $F2 \parallel \sum W_j$, the best bound, which is asymptotically attainable, of Heuristic GS is 2 when $w_1 = w_3$.

7.2 Analysis for $F2 \mid p_{2j} = p_2 \mid \sum W_j$

We analyze the worst case behavior of Heuristic GS for problem $F2 \mid p_{2j} = p_2 \mid \sum W_j$. The procedure of the analysis and the result are similar to those for problem $F2 \mid p_{1j} = p_1 \mid \sum W_j$. The following theorem establishes a tight bound of Heuristic GS.

Theorem 3

$$z^{GS} / z^* \leq \frac{2w_1\sum_{j=1}^{n-1} (n-j)p_{1j} + w_3n(n-1)p_2 + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2}{2w_1\max\{\sum_{j=1}^{n-1} (n-j)p_{1j}, n(n-1)p_2 / 2 + np_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2}$$

Further, this bound is tight.

Proof. From Lemmas 5 and 6,

$$z^* \geq \max\{z^{L3}, z^{L4}\}. \quad (4)$$

By the construction of GS,

$$W_j(\sigma^{GS}) \leq w_1 \sum_{k=1}^{j-1} p_{1k} + w_2 p_{1j} + w_3(j-1)p_2 + w_4 p_2$$

for $j \in N$. Therefore,

$$\sum_{j=1}^n W_j(\sigma^{GS}) \leq w_1 \sum_{j=1}^{n-1} (n-j)p_{1j} + w_2 \sum_{j=1}^n p_{1j} + \frac{w_3 n(n-1)p_2}{2} + w_4 n p_2. \quad (5)$$

From (4) and (5),

$$\begin{aligned} \frac{z^{GS}}{z^*} &\leq \frac{z^{GS}}{\max\{z^{L3}, z^{L4}\}} \\ &= \frac{2w_1 \sum_{j=1}^{n-1} (n-j)p_{1j} + w_3 n(n-1)p_2 + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 n p_2}{2 \max\{z^{L3}, z^{L4}\}} \\ &= \frac{2w_1 \sum_{j=1}^{n-1} (n-j)p_{1j} + w_3 n(n-1)p_2 + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 n p_2}{2w_1 \max\{\sum_{j=1}^{n-1} (n-j)p_{1j}, n(n-1)p_2 / 2 + n p_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 n p_2}. \end{aligned}$$

Now, we show that the bound is tight. The instance we use is the same as the one in Theorem 2. Suppose that permutation schedule $\sigma^{GS} = (1, 2, \dots, n)$ where there exists no inserted idle time. Then, the solution value is $z^{GS} = w_3(n^2 - n)/2 + w_4 n$. An optimal permutation schedule $\sigma^* = (1, 2, \dots, n)$ has the same sequence as σ^{GS} but schedule jobs on M_1 such that there is no wait time before a job on M_2 processes. Then, solution value $z^* = w_1(n^2 - n)/2 + w_4 n$. Consequently, the relative error is $[w_3 n(n-1)$

$$+2w_4n]/[w_1n(n-1)+2w_4n].$$

Observe that for this example $\sum_{j=1}^{n-1}(n-j)p_{1j} = 0$. Hence,

$$2 \max\{\sum_{j=1}^{n-1}(n-j)p_{1j}, n(n-1)p_2/2 + np_{11} - \sum_{j=1}^n p_{1j}\} = 2 \max\{0, (n^2 - n)/2\} = n^2 - n. \text{ Thus,}$$

$$\begin{aligned} & \frac{2w_1 \sum_{j=1}^{n-1}(n-j)p_{1j} + w_3n(n-1)p_2 + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4np_2}{2w_1 \max\{\sum_{j=1}^{n-1}(n-j)p_{1j}, n(n-1)p_2/2 + np_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4np_2} \\ &= \frac{w_3n(n-1) + 2w_4n}{w_1n(n-1) + 2w_4n}. \end{aligned}$$

Therefore, the bound is tight. \square

8. Heuristic with No Wait Time

In this section, we introduce and analyze the worst case behavior of a heuristic that is a modification of Heuristic GS. This heuristic is first introduced by Yang and Posner [21]. Since $w_1 \leq w_3$, we try to improve the schedule by eliminating the waiting time before M_2 . Similar to Heuristic GS, Heuristic NW processes the jobs in SPT order. We now formally describe Heuristic NW.

Heuristic NW

0. Reindex jobs so that $p_{1j} + p_{2j} \leq p_{1,j+1} + p_{2,j+1}$ for $j = 1, 2, \dots, n-1$.

1. Schedule job 1 such that $C_{11} = p_{11}$ and $C_{21} = p_{11} + p_{21}$.

Schedule jobs 2, 3, \dots , n in index order on M_1 and M_2 such that

$$C_{1j} = \max\{C_{1,j-1} + p_{1j}, C_{2,j-1}\} \text{ and } C_{2j} = C_{1j} + p_{2j} \text{ for } j = 2, 3, \dots, n.$$

2. Calculate W_j for $j = 1, 2, \dots, n$.

Output $\sum_{j=1}^n W_j$ and stop.

Heuristic NW runs in $O(n \log n)$ time. The resulting solution is a permutation schedule where there is no wait time before a job is processed on M_2 . Let σ^{NW} be the schedule generated by NW and z^{NW} be the cost of schedule σ^{NW} . We analyze Heuristic NW and find an upper bound on the relative error.

Example 2. Consider the same instance from Example 1. Heuristic NW generates a schedule $\sigma^{NW} = (1, 2, 3)$ where there exists inserted idle time of 2 on M_1 before the start of job 2 (see Figure 2). Note that there exists no wait time on M_2 . The solution value $z^{NW} = \sum_{j=1}^3 (w_1 Q_{1j} + w_2 p_{1j} + w_3 Q_{2j} + w_4 p_{2j}) = w_1(0 + 3.5 + 4.5) + w_2(1 + 1 + 5) + w_3(0 + 0 + 0) + w_4(3.5 + 3.5 + 3.5) = 30.75$.

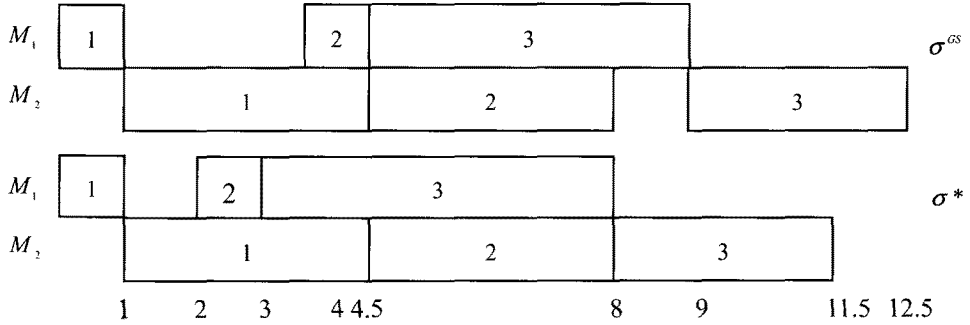


Figure 2. Example 2

8.1 Analysis for $F2 | p_{1j} = p_1 | \sum W_j$

We first analyze the worst case behavior of Heuristic NW for problem $F2 | p_{1j} = p_1 | \sum W_j$. The following theorem establishes a tight bound of Heuristic NW.

Theorem 4 *If there exist jobs q and r such that $p_{2q} < p_1 < p_{2r}$ for $q, r \in N$, then*

$$z^{NW} / z^* \leq \frac{w_1[(2n-i)(i-1)p_1 + 2\sum_{j=i}^{n-1}(n-j)p_{2j}] + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}{w_1 \max\{n(n-1)p_1, 2\sum_{j=1}^{n-1}(n-j)p_{2j}\} + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}$$

where p_{2i} is the smallest job to be greater than p_1 for $i \in N$. Otherwise, $z^{NW} / z^* = 1$. Further, this bound is asymptotically attainable.

Proof. From Lemmas 3 and 4,

$$z^* \geq \max\{z^{L1}, z^{L2}\}. \quad (6)$$

Since there is no waiting time before M_2 in σ^{NW} ,

$$\begin{aligned} W_j(\sigma^{NW}) &\leq w_1(p_1 + \sum_{k=1}^{j-1} \max\{p_{2k}, p_1\}) - w_1p_1 + w_2p_1 + w_4p_{2j} \\ &= w_1 \sum_{k=1}^{j-1} \max\{p_{2k}, p_1\} + w_2p_1 + w_4p_{2j} \end{aligned} \quad (7)$$

for $j \in N$. Now, we need to consider three different cases.

Case 1. $p_1 \geq \max_{j \in N}\{p_{2j}\}$.

Since $p_1 \geq \max_{j \in N}\{p_{2j}\}$, $\max_{k \in N}\{p_{2k}, p_1\} = p_1$. From (7),

$$W_j(\sigma^{NW}) \leq w_1(j-1)p_1 + w_2p_1 + w_4p_{2j}$$

for $j \in N$. Then,

$$\sum_{j=1}^n W_j(\sigma^{NW}) \leq \frac{w_1 n(n-1)p_1}{2} + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}.$$

Since $z^{L1} = w_1 n(n-1)p_1 / 2 + w_2 np_1 + w_4 \sum_{j=1}^n p_{2j}$, $z^{NW} / z^* = 1$.

Case 2. $p_1 \leq \min_{j \in N}\{p_{2j}\}$.

Similarly, we can show that $\sum_{j=1}^n W_j(\sigma^{NW}) = z^{L2}$. Hence, $z^{NW} / z^* = 1$.

Case 3. There exist q and r such that $p_{2q} < p_1 < p_{2r}$ for $q, r \in N$.

Suppose that p_{2i} is the smallest job to be greater than p_1 for $i \in N$. From (7), for $j = 1, 2, \dots, i$,

$$W_j(\sigma^{NW}) \leq w_1(j-1)p_1 + w_2p_1 + w_4p_{2j}$$

and for $j = i+1, i+2, \dots, n$,

$$W_j(\sigma^{NW}) \leq w_1[(i-1)p_1 + \sum_{k=i}^{j-1} p_{2k}] + w_2p_1 + w_4p_{2j}.$$

By adding up W_j 's, we have

$$\sum_{j=1}^n W_j(\sigma^{NW}) \leq w_1 \left[\frac{(2n-i)(i-1)p_1}{2} + \sum_{j=i}^{n-1} (n-j)p_{2j} \right] + w_2np_1 + w_4 \sum_{j=1}^n p_{2j}. \quad (8)$$

From (6) and (8),

$$\begin{aligned} \frac{z^{NW}}{z^*} &\leq \frac{z^{NW}}{\max\{z^{L1}, z^{L2}\}} \\ &= \frac{w_1[(2n-i)(i-1)p_1 + 2\sum_{j=i}^{n-1} (n-j)p_{2j}] + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}{2\max\{z^{L1}, z^{L2}\}} \\ &= \frac{w_1[(2n-i)(i-1)p_1 + 2\sum_{j=i}^{n-1} (n-j)p_{2j}] + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}{w_1 \max\{n(n-1)p_1, 2\sum_{j=1}^{n-1} (n-j)p_{2j}\} + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}. \end{aligned}$$

Next, we show that the bound is asymptotically attainable. Consider the instance where there are $2m$ jobs with processing times $p_{1j} = 1$ for $j = 1, 2, \dots, 2m$, $p_{2j} = 0$ for $j = 1, 2, \dots, m$, and $p_{2j} = 2$ for $j = m+1, m+2, \dots, 2m$. Since $p_{2j} = 0$ for $j = 1, 2, \dots, m$

and $p_{2j} = 2$ for $j = m+1, m+2, \dots, 2m$, jobs with $p_{2j} = 0$ must be processed first. Suppose that permutation schedule $\sigma^{NW} = (1, 2, \dots, 2m)$. Then, the solution value is $z^{NW} = w_1(2.5m^2 - 1.5m) + 2w_2m + 2w_4m$. An optimal schedule for the problem can be different with specific values of w_i 's and m . But for a problem with sufficiently a large value of m , an optimal permutation schedule $\sigma^* = (m+1, 1, m+2, 2, \dots, 2m, m)$ has solution value $z^* = w_1(2m^2 - m) + 2w_2m + w_3m + 2w_4m$. Consequently, the relative error z^{NW} / z^* goes to 1.25 as $m \rightarrow \infty$.

Observe that for this example $i = m+1$ and so $(2n-i)(i-1)p_1 = (3m-1)m = 3m^2 - m$. Also, $n(n-1)p_1 = 2m(2m-1) = 4m^2 - 2m$ and $\sum_{j=1}^{n-1} (n-j)p_{2j} = m^2 - m$. Hence, $\max\{n(n-1)p_1, 2\sum_{j=1}^{n-1} (n-j)p_{2j}\} = \max\{4m^2 - 2m, 2m^2 - 2m\} = 4m^2 - 2m$. Thus,

$$\begin{aligned} & \frac{w_1[(2n-i)(i-1)p_1 + 2\sum_{j=i}^{n-1} (n-j)p_{2j}] + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}}{w_1 \max\{n(n-1)p_1, 2\sum_{j=1}^{n-1} (n-j)p_{2j}\} + 2w_2np_1 + 2w_4\sum_{j=1}^n p_{2j}} \\ &= \frac{w_1(5m^2 - 3m) + 4w_2m + 4w_4m}{w_1(4m^2 - 2m) + 4w_2m + 4w_4m} \\ &\rightarrow 1.25 \end{aligned}$$

as $m \rightarrow \infty$. Therefore, the bound is asymptotically attainable. \square

Remark 2 ([21]) For problem $F2 \parallel \sum W_j$, the best asymptotically attainable bound of Heuristic NW is $2\beta / (\alpha + \beta) \geq 2$ where $\alpha = \min_{j \in N} \{p_{1j}, p_{2j}\}$ and $\beta = \max_{j \in N} \{p_{1j}, p_{2j}\}$.

8.2 Analysis for $F2 \mid p_{2j} = p_2 \mid \sum W_j$

In this subsection, we analyze the worst case behavior of Heuristic NW for problem $F2 \mid p_{2j} = p_2 \mid \sum W_j$. The following theorem establishes a tight bound of Heuristic NW.

Theorem 5 If there exist jobs q and r such that $p_{1q} < p_2 < p_{1r}$ for $q, r \in N$, then

$$z^{NW} / z^* \leq \frac{w_1[2np_{11} + 2\sum_{j=1}^{n-1}(n-j)p_{1j} + (2n-i+1)(i-2)p_2] + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2}{2w_1 \max\{\sum_{j=1}^{n-1}(n-j)p_{1j}, n(n-1)p_2/2 + np_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2}$$

where p_{1i} is the smallest job to be greater than p_2 for $i \in N$. Otherwise, $z^{NW} / z^* = 1$. Further, this bound is asymptotically attainable.

Proof. From Lemmas 5 and 6,

$$z^* \geq \max\{z^{L3}, z^{L4}\}. \quad (9)$$

Since there is no waiting time before M_2 in σ^{NW} ,

$$W_j(\sigma^{NW}) \leq w_1(p_{11} + \sum_{k=1}^{j-1} \max\{p_2, p_{1,k+1}\}) - w_1p_{1j} + w_2p_{1j} + w_4p_2 \quad (10)$$

for $j \in N$. Now, we need to consider three different cases.

Case 1. $p_2 \geq \max_{j \in N}\{p_{1j}\}$.

Since $p_2 \geq \max_{j \in N}\{p_{1j}\}$, $\max_{k \in N}\{p_2, p_{1k}\} = p_2$. From (10),

$$W_j(\sigma^{NW}) \leq w_1[p_{11} + (j-1)p_2] - w_1p_{1j} + w_2p_{1j} + w_4p_2$$

for $j \in N$. Then,

$$\sum_{j=1}^n W_j(\sigma^{NW}) \leq \frac{w_1[2p_{11} + n(n-1)p_2]}{2} + (w_2 - w_1)\sum_{j=1}^n p_{1j} + w_4np_2.$$

Since $z^{L4} = w_1n[2p_{11} + (n-1)p_2]/2 + (w_2 - w_1)\sum_{j=1}^n p_{1j} + w_4np_2$, $z^{NW} / z^* = 1$.

Case 2. $p_2 \leq \min_{j \in N}\{p_{1j}\}$.

Similarly, we can show that $\sum_{j=1}^n W_j(\sigma^{NW}) = z^{L3}$. Hence, $z^{NW} / z^* = 1$.

Case 3. There exist q and r such that $p_{1q} < p_2 < p_{1r}$ for $q, r \in N$.

Suppose that p_{1i} is the smallest job to be greater than p_1 for $i \in N$. From (7), for $j = 1, 2, \dots, i-1$,

$$W_j(\sigma^{NW}) \leq w_1[p_{11} + (j-1)p_2 - p_{1j}] + w_2 p_{1j} + w_4 p_2$$

and for $j = i, i+1, \dots, n$,

$$\begin{aligned} W_j(\sigma^{NW}) &\leq w_1[p_{11} + (i-2)p_2 + \sum_{k=i-1}^{j-1} p_{1,k+1} - p_{1j}] + w_2 p_{1j} + w_4 p_2 \\ &= w_1[p_{11} + (i-2)p_2 + \sum_{k=i}^{j-1} p_{1k}] + w_2 p_{1j} + w_4 p_2. \end{aligned}$$

By adding up W_j 's, we have

$$\sum_{j=1}^n W_j(\sigma^{NW}) \leq w_1[np_{11} + \sum_{j=i}^{n-1} (n-j)p_{1j} + (2n-i+1)(i-2)p_2 / 2] + w_2 \sum_{j=1}^n p_{1j} + w_4 np_2. \quad (11)$$

From (9) and (11),

$$\begin{aligned} \frac{z^{NW}}{z^*} &\leq \frac{z^{NW}}{\max\{z^{L3}, z^{L4}\}} \\ &= \frac{w_1[2np_{11} + 2\sum_{j=i}^{n-1} (n-j)p_{1j} + (2n-i+1)(i-2)p_2] + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 np_2}{2 \max\{z^{L3}, z^{L4}\}} \\ &= \frac{w_1[2np_{11} + 2\sum_{j=i}^{n-1} (n-j)p_{1j} + (2n-i+1)(i-2)p_2] + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 np_2}{2w_1 \max\{\sum_{j=1}^{n-1} (n-j)p_{1j}, n(n-1)p_2 / 2 + np_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2 \sum_{j=1}^n p_{1j} + 2w_4 np_2}. \end{aligned}$$

Next, we show that the bound is asymptotically attainable. Consider the instance where there are $2m$ jobs with processing times $p_{1j} = 0$ for $j = 1, 2, \dots, m$, $p_{1j} = 2$ for $j = m+1, m+2, \dots, 2m$, and $p_{2j} = 1$ for $j = 1, 2, \dots, 2m$. Since $p_{1j} = 0$ for $j = 1, 2, \dots, m$ and $p_{1j} = 2$ for $j = m+1, m+2, \dots, 2m$, jobs with $p_{1j} = 0$ must be processed first. Suppose that permutation schedule $\sigma^{NW} = (1, 2, \dots, 2m)$. Then, the solution value is $z^{NW} = 2.5w_1(m^2 - m) + 2w_2m + 2w_4m$. An optimal schedule for the problem can be different with specific values of w_i 's and m . But for a problem with sufficiently a large value of m , an optimal permutation schedule $\sigma^* = (1, m+1, 2, m+2, \dots, m-1, 2m)$ has solution value $z^* = 2w_1(m^2 - m) + 2w_2m + w_3(m-1) + 2w_4m$. Consequently, the relative error z^{NW} / z^* goes to 1.25 as $m \rightarrow \infty$.

Observe that for this example $i = m+1$. Hence, $\sum_{j=i}^{n-1} (n-j)p_{1j} = m^2 - m$ and $(2n-i+1)(i-2)p_2 = 3m(m-1) = 3m^2 - 3m$. Also, $\sum_{j=1}^{n-1} (n-j)p_{1j} = m^2 - m$ and $n(n-1)p_2 = 2m(2m-1) = 4m^2 - 2m$. Hence, $2 \max\{\sum_{j=1}^{n-1} (n-j)p_{1j}, n(n-1)p_2 / 2 + np_{11} - \sum_{j=1}^n p_{1j}\} = 2 \max\{m^2 - m, 2m^2 - 3m\} = 4m^2 - 6m$. Thus,

$$\begin{aligned} & \frac{w_1[2np_{11} + 2\sum_{j=i}^{n-1} (n-j)p_{1j} + (2n-i+1)(i-2)p_2] + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2}{2w_1 \max\{\sum_{j=1}^{n-1} (n-j)p_{1j}, n(n-1)p_2 / 2 + np_{11} - \sum_{j=1}^n p_{1j}\} + 2w_2\sum_{j=1}^n p_{1j} + 2w_4np_2} \\ &= \frac{w_1(5m^2 - 5m) + 4w_2m + 4w_4m}{w_1(4m^2 - 6m) + 4w_2m + 4w_4m} \\ &\rightarrow 1.25 \end{aligned}$$

as $m \rightarrow \infty$. Therefore, the bound is asymptotically attainable. \square

9. Heuristic with Minimum Waiting Cost

We introduce another heuristic which finds an optimal schedule for any fixed job sequence. The heuristic is first developed by Yang and Posner [21]. We use this

algorithm to optimally solve a special case of problem $F2 \parallel \sum W_j$ where $sp_{1j} = p_{2j}$ for all $j \in N$ and $s > 0$

The procedure begins with the schedule found by Heuristic GS. Since there exists no inserted idle time in Heuristic GS schedule, there may exist waiting time before M_2 . Starting with the last job, the procedure tries to eliminate those waiting times by delaying jobs as long as the solution value improves. The value of t provides the size of the block of jobs that are delayed only on M_1 . Now, we formally describe a procedure to solve $F2 \parallel \sum W_j$ when the job sequence σ is specified.

Algorithm FIXED

0. Input the job order.

Schedule the jobs using Heuristic GS. Set $C_{i[j]}$ to be the completion time in this schedule of job $[j]$ on machine i , for $i \in \{1, 2\}$ and $j \in N$.

1. Set $C_{1[n]} = C_{2[n]} - p_{2[n]}$.

Set $j = n$ and $t = 0$.

Go to Step 4.

2. Set $C_{1[j]} = \min\{C_{1[j+1]} - p_{1[j+1]}, C_{2[j]} - p_{2[j]}\}$.

If $C_{1[j]} + p_{2[j]} = C_{2[j]}$, then set $t = 0$ and go to Step 4.

Otherwise, set $t = t + 1$.

3. If $(n - j + 1) / t \geq w_3 / w_1$, then go to Step 4.

Otherwise, set $d = \min_{k=j, j+1, \dots, j+t-1} \{C_{2[k]} - p_{2[k]} - (C_{1[k+1]} - p_{1[k+1]}) \mid C_{2[k]} - p_{2[k]} - (C_{1[k+1]} - p_{1[k+1]}) > 0\}$ and $t' = \operatorname{argmin}_{k=j, j+1, \dots, j+t-1} \{C_{2[k]} - p_{2[k]} - (C_{1[k+1]} - p_{1[k+1]}) - d = 0\} - j$.

Set $C_{1[k]} = C_{1[k]} + d$ for $k = j, j+1, \dots, j+t-1$.

Set $C_{i[k]} = C_{i[k]} + d$ for $i = 1, 2$ and $k = j+t, j+t+1, \dots, n$.

Set $t = t'$.

4. If $j = 2$, then output schedule and stop.

Otherwise, $j = j - 1$.

5. If $n / (t + j) < w_3 / w_1$, then go to Step 2.

Otherwise, set $C_{1[k]} = \min\{C_{1[k+1]} - p_{1[k+1]}, C_{2[k]} - p_{2[k]}\}$ for $k = j, j-1, \dots, 2$.

Output schedule and stop.

The time requirement of Algorithm FIXED is $O(n \log n)$. By using FIXED, we now present the following heuristic.

Heuristic FB

0. Reindex jobs so that $p_{1j} + p_{2j} \leq p_{1,j+1} + p_{2,j+1}$ for $j = 1, 2, \dots, n-1$.

1. Call Algorithm FIXED.

2. Calculate W_j for $j = 1, 2, \dots, n$.

Output $\sum_{j=1}^n W_j$ and stop.

Since Algorithm FIXED requires $O(n \log n)$ time, the time requirement of FB is $O(n \log n)$ time. Let σ^{FB} be the schedule generated by Heuristic FB and z^{FB} be the cost of schedule σ^{FB} .

Example 3. Consider the same instance from Example 1. Heuristic FB generates a schedule $\sigma^{FB} = (1, 2, 3)$ where there exists inserted idle time of 1 on M_1 before job 2 starts. In this case, solution value $z^{FB} = \sum_{j=1}^3 (w_1 Q_{1j} + w_2 p_{1j} + w_3 Q_{2j} + w_4 p_{2j}) = w_1(0+2+3) + w_2(1+1+5) + w_3(0+1.5+0) + w_4(3.5+3.5+3.5) = 30$. For this instance, FB generates an optimal schedule.

10. Proportional Machines

In this section, we consider a special case of problem $F2 \parallel \sum W_j$. We examine the class of instances where $sp_{1j} = p_{2j}$ for all $j \in N$ and $s > 0$. For these instances, the processing time of a job on each machine is proportional to a base processing time.

In this section, we assume that the jobs are indexed so that $p_{11} \leq p_{12} \leq \dots \leq p_{1n}$. We establish the optimality of Heuristic FB for $F2 \parallel \sum W_j$ when $sp_{1j} = p_{2j}$ for all $j \in N$ and $s > 0$. Before we start the analysis we first review a known property of

FIXED.

Theorem 6 ([21]) *Suppose that a job sequence is given. Then, Algorithm FIXED generates an optimal set of start times.*

From Theorem 6, we need to establish only that an SPT job sequence is optimal for the problem. The next set of results establishes the optimality of FB.

Lemma 7 *Heuristic FB finds an optimal schedule if $sp_{1j} = p_{2j}$ for all $j \in N$ and $s \leq 1$.*

Proof. Since $0 < s \leq 1$, $p_{1j} \geq p_{2j}$. Also, since $sp_{1j} = p_{2j}$ for all $j \in N$, an SPT job processing order implies an SPT job processing order on M_1 . Thus, FB schedules jobs in SPT order. There is no inserted idle time on M_1 because $p_{1,j+1} \geq p_{2j}$ for $j \in \{1, 2, \dots, n-1\}$. Since jobs are sequenced in an SPT order and there does not exist any inserted idle time on M_1 , $\sum_{j=1}^n T_{1j}$ is minimized. Also, $T_{2j} = 0$ for all $j \in N$ because $p_{1j} \geq p_{2j}$ for all $j \in N$. Therefore, the result holds. \square

For the next three lemmas, we assume that there exists an optimal schedule $\bar{\sigma}$ where the job sequence is not SPT. We let l and t be the last pair of jobs in $\bar{\sigma}$ such that t immediately precedes l and $t > l$. In the next two lemmas, we show that inserting t after l gives a schedule σ' that is at least as good as $\bar{\sigma}$.

Lemma 8 *Heuristic FB finds an optimal schedule if $sp_{1j} = p_{2j}$ for all $j \in N$, $s > 1$, and $C_{1l}(\bar{\sigma}) = S_{1l}(\bar{\sigma})$.*

Proof. Since $p_{1l} < p_{2l} < p_{2t}$ and $C_{1l}(\bar{\sigma}) = S_{1l}(\bar{\sigma})$, we have that $C_{2t}(\bar{\sigma}) = S_{2t}(\bar{\sigma})$. In σ' , let $S_{1l}(\sigma') = S_{1l}(\bar{\sigma})$, $S_{2t}(\sigma') = S_{2t}(\bar{\sigma})$, $S_{1t}(\sigma') = C_{1l}(\sigma')$, and $S_{2t}(\sigma') = C_{2t}(\sigma')$. This schedule is possible because $p_{1l} \leq p_{1t}$ and $p_{1l} \leq p_{2t}$. Because $C_{1l}(\sigma') = C_{1l}(\bar{\sigma})$, and $C_{2t}(\sigma') = C_{2t}(\bar{\sigma})$, the jobs in $N \setminus \{l, t\}$ have the same completion times in both $\bar{\sigma}$ and σ' .

Since $p_{1l} < p_{1t}$, we have that $Q_{1l}(\bar{\sigma}) + Q_{1t}(\bar{\sigma}) > Q_{1l}(\sigma') + Q_{1t}(\sigma')$. Further, $Q_{2t}(\sigma') =$

$Q_{2l}(\bar{\sigma}) - (p_{2l} - p_{1l})$ and $Q_{2l}(\sigma') = Q_{2l}(\bar{\sigma}) + (p_{2l} - p_{1l})$. Since $p_{1l} < p_{1l}$, we have that $Q_{2l}(\sigma') + Q_{2l}(\sigma') < Q_{2l}(\bar{\sigma}) + Q_{2l}(\bar{\sigma})$. \square

When $s > 1$ and $C_{1l}(\bar{\sigma}) < S_{1l}(\bar{\sigma})$, there are two basic situations depending on whether $p_{1l} - p_{2l}$ is no larger than or greater than $Q_{2l}(\bar{\sigma})$. We consider the scenario where $p_{1l} - p_{2l} \leq Q_{2l}(\bar{\sigma})$ in Lemma 9, and the one where $p_{1l} - p_{2l} > Q_{2l}(\bar{\sigma})$ in Lemma 10. When $p_{1l} - p_{2l} \leq Q_{2l}(\bar{\sigma})$, the other jobs in the schedule are not delayed as a result of inserting job t after l . This situation is shown in Figure 3.

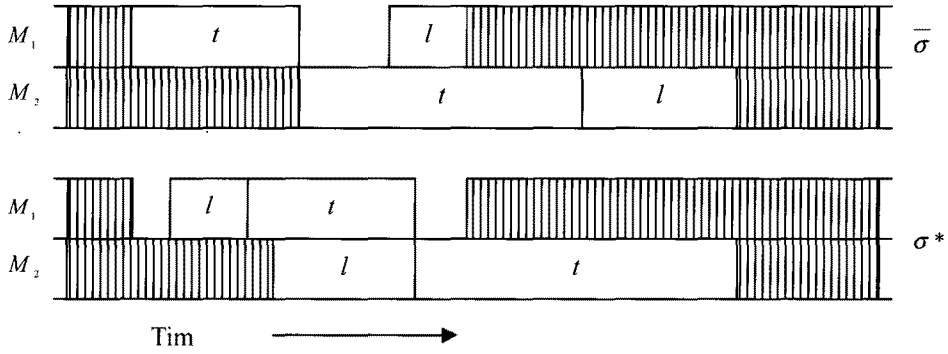


Figure 3. An example where $p_{1l} - p_{2l} \leq Q_{2l}(\bar{\sigma})$

Lemma 9 *Heuristic FB finds an optimal schedule if $sp_{1j} = p_{2j}$ for all $j \in N$, $s > 1$,*

$C_{1l}(\bar{\sigma}) < S_{1l}(\bar{\sigma})$, and $p_{1l} - p_{2l} \leq Q_{2l}(\bar{\sigma})$.

Proof. Because $C_{1l}(\bar{\sigma}) < S_{1l}(\bar{\sigma})$, $C_{1l}(\bar{\sigma}) = S_{2l}(\bar{\sigma})$. Otherwise, we can construct a better schedule by delaying job t on M_1 .

In σ' , let $S_{1l}(\sigma') = \min\{C_{1l}(\bar{\sigma}) - p_{1l}, C_{1l}(\bar{\sigma}), S_{2l}(\bar{\sigma}) + p_{2l} - p_{1l}\} - p_{1l}$, $S_{2l}(\sigma') = S_{2l}(\bar{\sigma})$, $S_{1t}(\sigma') = \min\{C_{1l}(\bar{\sigma}), S_{2l}(\bar{\sigma}) + p_{2l}\} - p_{1t}$, and $S_{2t}(\sigma') = S_{2l}(\bar{\sigma}) + p_{2l}$. Thus,

$$\begin{aligned} Q_{1l}(\sigma') + Q_{1t}(\sigma') &= S_{1l}(\sigma') + S_{1t}(\sigma') \\ &\leq C_{1l}(\bar{\sigma}) - p_{1l} + C_{1l}(\bar{\sigma}) - p_{1t} \end{aligned}$$

$$\begin{aligned}
 &= S_{1t}(\bar{\sigma}) + S_{1t}(\bar{\sigma}) \\
 &= Q_{1t}(\bar{\sigma}) + Q_{1t}(\bar{\sigma}).
 \end{aligned}$$

Consequently, the waiting cost before processing on M_1 is no larger in σ' than in $\bar{\sigma}$.

Because $Q_{1t}(\sigma') = S_{1t}(\sigma')$ and $Q_{1t}(\sigma') = S_{1t}(\sigma')$, we have that $Q_{1t}(\sigma') = \min\{Q_{1t}(\bar{\sigma}) + p_{1t} - p_{1t}', Q_{1t}(\bar{\sigma}) + p_{1t}', Q_{1t}(\bar{\sigma}) + p_{2t}\} - p_{1t}'$, and $Q_{2t}(\sigma') = \min\{Q_{2t}(\bar{\sigma}) + p_{1t}', Q_{2t}(\bar{\sigma}) + p_{1t} + p_{2t}\} - p_{1t}'$. Also,

$$Q_{2t}(\sigma') = \begin{cases} Q_{2t}(\bar{\sigma}) + p_{1t} - p_{2t}', & \text{if } Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) - p_{1t}' \\ 0, & \text{if } Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{1t}' - p_{1t}' \\ p_{1t} - p_{2t}', & \text{if } Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{2t}' - p_{1t}' \end{cases}$$

and

$$Q_{2t}(\sigma') = \begin{cases} Q_{2t}(\bar{\sigma}) + p_{2t} - p_{2t}', & \text{if } Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{1t}' - p_{1t}' \\ 0, & \text{if } Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{2t}'. \end{cases}$$

For each of these situations, it can be shown that $Q_{2t}(\sigma') + Q_{2t}(\sigma') \leq Q_{2t}(\bar{\sigma}) + Q_{2t}(\bar{\sigma})$.

Note that $C_{1t}(\sigma') = S_{1t}(\sigma') + p_{1t} = \min\{C_{1t}(\bar{\sigma}), S_{2t}(\bar{\sigma}) + p_{2t}\} \leq C_{1t}(\bar{\sigma})$. Also, $C_{2t}(\sigma') = S_{2t}(\bar{\sigma}) + p_{2t} + p_{2t}' = C_{2t}(\bar{\sigma})$. Consequently, the times that the jobs in $N \setminus \{l, t\}$ are processed in $\bar{\sigma}$ and σ' are identical. \square

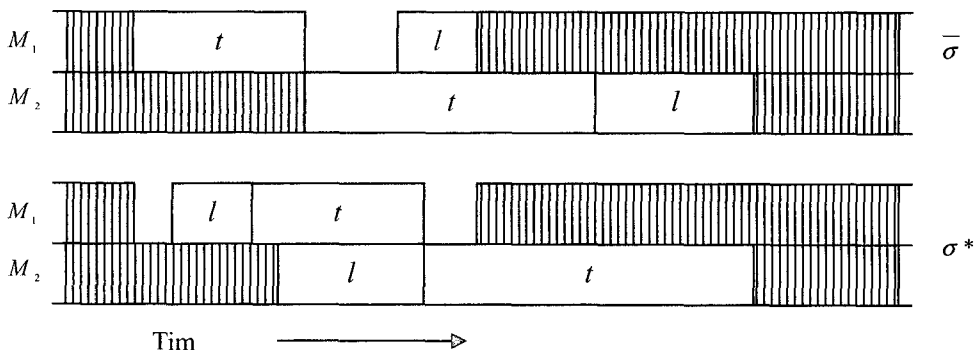


Figure 4. An example where $p_{1t} - p_{2t} > Q_{2t}(\bar{\sigma})$

When $p_{1t} - p_{2t} > Q_{2t}(\bar{\sigma})$, if t is inserted after l , then it is possible that jobs following job t are delayed. This could increase the solution value. This situation is shown in Figure 4. As a result, job t is delayed further in Lemma 10 to make sure that a new schedule is at least as good as $\bar{\sigma}$.

Lemma 10 *Heuristic FB finds an optimal schedule if $sp_{1j} = p_{2j}$ for all $j \in N$, $s > 1$, $C_{1t}(\bar{\sigma}) < S_{1t}(\bar{\sigma})$, $p_{1t} - p_{2t} > Q_{2t}(\bar{\sigma})$, and $t = l + 1$.*

Proof. Because $C_{1t}(\bar{\sigma}) < S_{1t}(\bar{\sigma})$, we have that $C_{1t}(\bar{\sigma}) = S_{2t}(\bar{\sigma})$. Otherwise, delaying job t on M_1 produces a better schedule. For $i, j \in N$, suppose jobs i and j are processed consecutively in σ .

Let σ' be the schedule where job t is inserted after l , and $S_{1t}(\sigma') = C_{1t}(\bar{\sigma}) - p_{1t} - Q_{2t}(\bar{\sigma})$, $S_{2t}(\sigma') = S_{2t}(\bar{\sigma})$, $S_{1t}(\sigma') = C_{1t}(\sigma')$, and $S_{2t}(\sigma') = C_{1t}(\sigma')$. Because $Q_{1t}(\sigma') = S_{1t}(\sigma')$ and $Q_{1t}(\sigma') = S_{1t}(\sigma')$, we have that $Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{1t} - p_{1t} - Q_{2t}(\bar{\sigma})$ and $Q_{1t}(\sigma') = Q_{1t}(\bar{\sigma}) + p_{1t} - Q_{2t}(\bar{\sigma})$. Also, $Q_{2t}(\sigma') = Q_{2t}(\bar{\sigma})$ and $Q_{2t}(\sigma') = 0$.

$$\begin{aligned} \text{Since } Q_{2t}(\bar{\sigma}) &= S_{2t}(\bar{\sigma}) - C_{1t}(\bar{\sigma}) = (Q_{1t}(\bar{\sigma}) + p_{1t} + p_{2t}) - (Q_{1t}(\bar{\sigma}) + p_{1t}), \\ Q_{1t}(\sigma') + Q_{1t}(\sigma') &= Q_{1t}(\bar{\sigma}) + p_{1t} - p_{1t} - Q_{2t}(\bar{\sigma}) + Q_{1t}(\bar{\sigma}) + p_{1t} - Q_{2t}(\bar{\sigma}) \\ &= Q_{1t}(\bar{\sigma}) + p_{1t} - Q_{2t}(\bar{\sigma}) - p_{2t} + Q_{1t}(\bar{\sigma}) \\ &< Q_{1t}(\bar{\sigma}) + Q_{1t}(\bar{\sigma}). \end{aligned}$$

Thus, the cost associated with jobs l and t decreases when t is inserted after l . Further,

$$\begin{aligned} C_{1t}(\sigma') &= Q_{1t}(\sigma') + p_{1t} \\ &= Q_{1t}(\bar{\sigma}) + 2p_{1t} - Q_{2t}(\bar{\sigma}) \\ &< S_{2t}(\bar{\sigma}) + p_{2t} - Q_{2t}(\bar{\sigma}) \\ &= C_{1t}(\bar{\sigma}), \end{aligned}$$

and

$$\begin{aligned}
 C_{2t}(\sigma') &= Q_{1t}(\sigma') + p_{1t} + p_{2t} \\
 &= Q_{1t}(\bar{\sigma}) + p_{1t} - Q_{2l}(\bar{\sigma}) + p_{1t} + p_{2t} \\
 &= (S_{1t}(\bar{\sigma}) + p_{1t} + p_{2t} + p_{2l}) - Q_{2l}(\bar{\sigma}) + p_{1t} - p_{2l} \\
 &= C_{2l}(\bar{\sigma}) - Q_{2l}(\bar{\sigma}) + p_{1t} - p_{2l}.
 \end{aligned}$$

As a result, job t completes $p_{1t} - Q_{2l}(\bar{\sigma}) - p_{2l}$ later in σ' than job l does in $\bar{\sigma}$.

Now, $p_{1t+1} \geq p_{1t} > p_{2l} + Q_{2l}(\bar{\sigma})$ implies that $C_{1l}(\bar{\sigma}) = S_{1t+1}(\bar{\sigma})$. Otherwise, delaying job l or expediting job $t+1$ produces a better schedule. As a result, the idle time on M_2 after job l is

$$p_{1t+1} - Q_{2l}(\bar{\sigma}) - p_{2l} \geq p_{1t} - Q_{2l}(\bar{\sigma}) - p_{2l}.$$

Hence, job t can be processed later on M_2 by $p_{1t+1} - Q_{2l}(\bar{\sigma}) - p_{2l}$ without delaying job $t+1$ or any other jobs. \square

Theorem 7 Heuristic FB finds an optimal schedule if $sp_{1t} = p_{2t}$ for all $t \in N$ and $s > 0$.

Proof. From Lemmas 7, 8, 9, and 10, the only situation that still has to be considered is when $s > 1$, $C_{1l}(\bar{\sigma}) < S_{1l}(\bar{\sigma})$, $p_{1t} - p_{2l} > Q_{2l}(\bar{\sigma})$, and $t \geq l+2$. Under these conditions, job t is inserted after job $t-1$.

When $Q_{2l}(\sigma') = 0$, depending on the situation, the proof is similar to the proofs

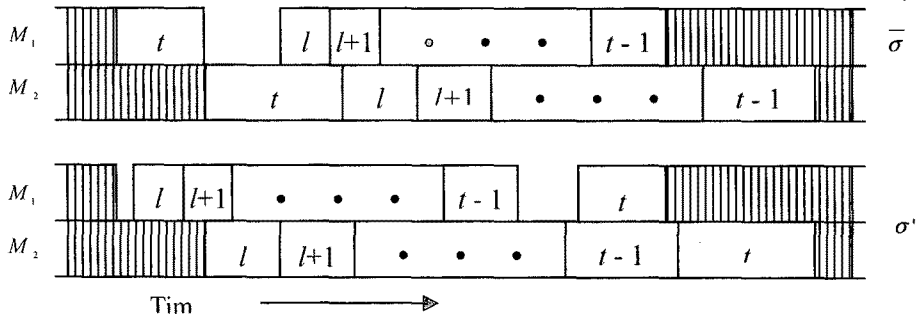


Figure 5. An example where job t is inserted after job $t-1$

of Lemmas 9 or 10. When $Q_{2t}(\sigma') > 0$, the proof is also similar to the proofs of Lemmas 9 or 10. However, some jobs in $\{l+1, l+2, \dots, t-1\}$ are delayed on M_1 to offset the increase of the solution because $Q_{2t}(\sigma')$ is positive (see Figure 5). \square

11. Summary and Further Research

We study the three special cases of the relatively new deterministic flow shop scheduling problem where different processing states have different WIP costs. Specifically, the first and the second cases are the problem where processing times on machine 1 are identical and the problem where processing times on machine 2 are identical, respectively. The recognition version of the both problems is known to be unary NP-complete.

For each problem, we suggest two simple and intuitive heuristics, GS and NW and find worst case bounds on relative error. For Heuristic GS, a tight upper bound on relative error is found for each of the problems. For Heuristic NW, we find an upper bound on relative error and the bound is asymptotically attainable for each of the problems.

The third special case is the problem where the processing time of a job on each machine is proportional to a base processing time. For this problem, we show that a known heuristic finds an optimal solution.

There are several important possible extension of this research. Design of a PTAS would be interesting. Also worth considering are more general cases of the problem such as including individual weights for each job.

References

- [1] Adiri, I. and N. Amit, "Openshop and flowshop scheduling to minimize sum of completion times," *Computers and Operations Research* 11 (1984), 275-284.
- [2] Allahverdi, A., "Two-machine proportionate flowshop scheduling with breakdowns to minimize maximum lateness," *Computers and Operations Research* 23 (1996), 909-916.

- [3] Cheng, T. C. E. and N. Shakhlevich, "Proportionate flow shop with controllable processing times," *Journal of Scheduling* 2 (1999), 253-265.
- [4] Choi, B.-C., S.-H. Yoon, and S.-J. Chung, "Minimizing the total weighted completion time in a two-machine proportionate flow shop with different machine speeds," *International Journal of Production Research* 44 (2006), 715-728.
- [5] Conway, R. W., Maxwell, W. L., and Miller, L. W., *Theory of scheduling*, Addison-Wesley, Reading, MA, 1967.
- [6] Croce, F. D., M. Ghirardi, and R. Tadei, "An improved branch-and-bound algorithm for the two machine total completion time flow shop problem," *European Journal of Operational Research* 139, (2002), 293-301.
- [7] Croce, F. D., V. Narayan, and R. Tadei, "The two-machine total completion time flow shop problem," *European Journal of Operational Research* 90 (1996), 227-237.
- [8] Garey, M. R., D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of Operations Research* 1 (1976), 117-129.
- [9] Gonzalez, T. and S. Sahni, "Flowshop and job shop schedules: Complexity and approximation," *Operations Research* 26 (1978), 36-52.
- [10] Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Annals of Discrete Mathematics* 5 (1979), 287-326.
- [11] Hoogeveen, J. A. and T. Kawaguchi, "Minimizing total completion time in a two-machine flowshop: analysis of special cases," *Mathematics of Operations Research* 24 (1999), 887-910.
- [12] Hou, S. X. and H. Hoogeveen, "The three-machine proportionate flow shop problem with unequal machine speeds," *Operations Research Letters* 31 (2003), 225-231.
- [13] Lee, W.-C. and C.-C. Wu, "Minimizing the total flow time and tardiness in a two-machine flow shop," *International Journal of Systems Science* 32 (2001), 365-373.
- [14] Ow, P. S., "Focused scheduling in proportionate flowshops," *Management Science* 31 (1985), 852-869.
- [15] Shakhlevich, N., H. Hoogeveen, and M. Pinedo, "Minimizing total weighted completion time in a proportionate flow shop," *Journal of Scheduling* 1 (1998), 157-168.
- [16] Sipper, D. and R. Shapira, "JIT Vs. WIP-a trade-off analysis," *International*

Journal of Production Research 27 (1989), 903-914.

- [17] van de Velde, S. L., "Minimizing the sum of the job completion times in the two-machine flow shop by Lagrangian relaxation," *Annals of Operations Research* 26 (1992), 257-268.
- [18] Vollman, T. E., W. L. Berry, and D. C. Whybark, *Manufacturing planning and control systems*, McGraw-Hill, New York, NY, 1997.
- [19] Wang, C., C. Chu, and J.-M. Proth, "Efficient heuristic and optimal approaches for $n/2/F/\sum C_i$ scheduling problems," *International Journal of Production Economics* 44 (1997), 225-237.
- [20] Yang, J., "Two machine flow shop scheduling problem with weighted WIP weight," *Computers and Operations Research* 36 (2009), 472-486.
- [21] Yang, J. and M. E. Posner, "Flow Shops with WIP and Value Added Costs," *Journal of Scheduling* (2009), in press.