International Journal of Management Science
Vol 13, No 2, November 2007

Customer Order Scheduling Problem on Parallel Machines with Identical Order Size*

Jaehwan Yang**
School of Business Administration, University of Seoul,
90 Jeonnong-dong, Dongdaemun-gu, Seoul, 130-743, Korea

(Received Dec. 2006; Revised May 2007; Accepted May 2007)

ABSTRACT

This paper considers a scheduling problem where a customer orders multiple products (jobs) from a production facility. The objective is to minimize the sum of the order (batch) completion times. While a machine can process only one job at a time, multiple machines can simultaneously process jobs in a batch. Although each job has a unique processing time, we consider the case where batch processing times are identical. This simplification allows us to develop heuristics with improved performance bounds. This problem was motivated by a real world problem encountered by foreign electronics manufacturers.

We first establish the complexity of the problem. For the two parallel machine case, we introduce two simple but intuitive heuristics, and find their worst case relative error bounds. One bound is tight and the other bound goes to 1 as the number of orders goes to infinity. However, neither heuristic is superior for all instances. We extend one of the heuristics to an arbitrary number of parallel machines. For a fixed number of parallel machines, we find a worst case bound which goes to 1 as the number of orders goes to infinity. Then, a tighter bound is found for the three parallel machine case. Finally, the heuristics are empirically evaluated.

Keywords: Scheduling, Heuristics, Complexity Analysis

1. Introduction

In the typical customer order scheduling problem, each order has a set of products (jobs), called a batch, which needs to be processed. Once all jobs in the batch are com-

* This work was supported by New Professor Research Fund of 2006 from University of Seoul.
** Corresponding author, E-mail: jyang@uos.ac.kr

pleted, the entire batch is dispatched. The composition of the jobs in the batch is prespecified. No setup times are assumed between different jobs or different batches. Also, the objective is associated with the completion time of the batches instead of the completion time of each job. The completion time of the batch is the latest completion time of any job in the batch. While a machine can process only one job at a time, several machines can simultaneously process jobs in a given batch.

This paper considers a variation of this customer order scheduling problem where the total processing time of each batch is identical. This variation is motivated by a real world problem of an electronics manufacturing company that produces personal computer monitors and owns several subsidiaries located in foreign countries. Due to high transportation costs, the manufacturing facility only ships full container units. Exporting in full container units is common practice in some of the consumer electronics industry. Consequently, subsidiaries wait until their purchase order is large enough to fill a container. For most of the products produced, the size of the product is approximately proportional to the time spent. Hence, producing products for any container usually takes about the same time. As a result, each order takes approximately the same amount of time.

No previous research considers this version of customer order scheduling problem where batch processing times are identical for all batches. However, several studies consider the more general customer order scheduling problems. The customer order scheduling problem is different from most of other batch scheduling problems because the objective is associated with the completion time of the batches instead of the completion time of each job. Julien and Magazine [12] study a single machine problem where the objective is to minimize the total batch completion time. A job-dependent setup time is assumed between two different types of jobs. They develop a dynamic programming algorithm that runs in polynomial time for the problem where there exist two types of jobs and the batch processing order is fixed. Coffman *et al.* [6] consider a similar problem where the batch processing order is not fixed. Baker [2] also considers a problem similar to Coffman *et al.* [6]. However, for one type of job, those jobs processed during the same production run are not available until the completion of the production run. This restriction is called batch availability. For batch availability, see Santos and Magazine [15]. Gupta *et al.* [11] consider the single machine problem where each order must have one job from each of several job classes. Also, there is a setup time whenever the job class changes. Gerodimos *et al.* [8] study single machine problems where each batch has one common job and one distinct job. Ding [7] and Liao [13] also examine similar problems.

Blocher and Chhajed [3] examine the customer order scheduling problem in the parallel machine environment where the objective is to minimize the sum of batch completion times. They show that the recognition version of the problem is unary NP-complete for the three parallel machine case and is at least binary NP-complete for the two parallel machine case. Also, they develop several heuristic methods and two lower bounds. Blocher *et al.* [4] extend the problem to a job shop. Yang and Posner [17] consider the same problem with two parallel machines, and introduce three simple heuristics and find tight worst case bounds on relative errors of 2, 9/7, and 6/5, respectively. Yang [16] establishes the complexity of different customer order scheduling problems. When the machine-job assignment is fixed, Roemer and Ahmadi [14] show that the recognition version of the problem is unary NP-complete for the two parallel machine case with the objective of minimizing the sum of batch completion times. Ahmadi *et al.* [1] develop three lower bounds and several heuristics for the problem with the objective of minimizing the sum of weighted batch completion times.

We first introduce some notation. Then, we establish the complexity of the problem and review a few properties of an optimal schedule. For the two parallel machine case, we introduce two simple but intuitive heuristics, and find their worst case bounds on relative error. One bound is tight and the other bound goes to 1 as the number of batches goes to infinity. We show that neither heuristic performs better for all instances. One of the heuristics for the two parallel machine case is extended to an arbitrary number of parallel machine case. For a fixed number of machines, we find a worst case bound which goes to 1 as the number of batches goes to infinity. The bound is tightened for the three parallel machine case. Finally, the heuristics are empirically evaluated.

2. Notation

The decision variables in our models are

σ_k = schedule of all jobs on machine k for $k \in M$ where M = set of machines = $\{1, 2, \cdots, m\}$ and m = number of machines

σ = schedule of all jobs $= (\sigma_1, \sigma_2, \cdots, \sigma_m)$.

Other notation that is used in this work include

n = number of jobs

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

b = number of batches

B = set of batches $= \{1, 2, \cdots, b\}$

n_i = number of jobs in batch i for $i \in B$

B_i = set of jobs in batch i for $i \in B = \{\sum_{j=1}^{i-1} n_j + 1, \sum_{j=1}^{i-1} n_j + 2, \cdots, \sum_{j=1}^{i} n_j\}$

$\beta_i(\sigma)$ = first job selected for processing in B_i for $i \in B$

p_j = processing time of job j for $j \in N$

P_i $= \sum_{j \in B_i} p_j$ = total processing time of batch $i \in B$

$C_i(\sigma_k)$ = completion time of batch i on machine k for $i \in B$ and $k \in M$

$C_i(\sigma)$ = completion time of batch i in schedule σ for $i \in B$ $= \max_{k \in M} C_i(\sigma_k)$

z^* = value of optimal schedule.

We represent $\beta_i(\sigma)$ as β_i and $C_i(\sigma)$ as C_i when there is no ambiguity. The standard classification scheme for scheduling problems (Graham *et al.* [10]) is $\alpha_1 | \alpha_2 | \alpha_3$, where α_1 describes the machine structure, α_2 gives the job characteristics or restrictive requirements, and α_3 defines the objective function to be minimized. In the first field, P means the parallel machine structure, and the number after P indicates a fixed number of machines rather than an arbitrary number. Also, $P_i = \overline{P}$ is placed in the α_2 to describe the job characteristic such that all batch processing times are identical. Finally, we extend this scheme to provide for batch completion times by using C_{B_i} in the α_3 field. This notation is used to eliminate the confusion between our problem and the classical scheduling problem. For example, $P2 | P_i = \overline{P} | \sum C_{B_i}$ is the problem where there exist two parallel machines, all batch processing times are identical, and the objective is to minimize the total batch completion time.

3. Complexity and Basic Results

The following theorem establishes the complexity of problem $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$.

Theorem 1 : *The recognition version of problem* $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$ *is at least binary NP-complete.*

Proof. Note that the recognition version of problem $P2 \parallel C_{\max}$ is binary NP-complete. By letting $b = 1$, problem $P2 \parallel C_{\max}$ reduces to $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. Hence, $P2 \parallel C_{\max}$ is a special case of $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. □

Similarly, the following theorem establishes the complexity of problem $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$.

Theorem 2 : *The recognition version of problem* $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$ *is unary NP-complete.*

Proof. Note that the recognition version of problem $P3 \parallel C_{\max}$ is unary NP-complete [5]. By letting $b = 1$, problem $P3 \parallel C_{\max}$ reduces to $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. Hence, $P3 \parallel C_{\max}$ is a special case of $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. □

Note that there is no restriction on delaying jobs. Hence, for problem $P \parallel \Sigma C_{B_i}$, there exists an optimal schedule without inserted idle time. As a result, we only consider those schedules where there is no inserted idle time.

Also, we say that batch $i \in B$ is *separated* if on some machine $k \in M$, jobs in batch i are not processed consecutively. Then, we have the following property for an optimal schedule.

Lemma 1 : (Blocher and Chhajed [3]) *For problem* $P \parallel \Sigma C_{B_i}$, *there exists an optimal schedule where no batch is separated.*

As a result of Lemma 1, we only consider those schedules where batches are not separated.

4. Two Parallel Machines

In this section, we introduce two heuristic procedures to find a schedule for problem $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. For each of the heuristics, a worst case bound on relative error is presented.

4.1 The Job LPT Heuristic

In order to find a schedule for problem $P2 \mid P_i = \bar{P} \mid \Sigma C_{B_i}$, we introduce a heuristic procedure that has the tight worst case bound on relative error of 7/6. This heuristic is first introduced by Blocher and Chhajed [3] for problem $P \parallel \Sigma C_{B_i}$. For problem $P2 \parallel \Sigma C_{B_i}$, Yang and Posner [17] show that the tight worst case bound for the heuristic is 6/5 (they call the heuristic H2). Hence, we establish that the heuristic has the tighter worst case bound on relative error with $P2 \mid P_i = \bar{P} \mid \Sigma C_{B_i}$ than with $P2 \parallel \Sigma C_{B_i}$.

To find a schedule, the heuristic tries to obtain the maximum benefit of the LPT (Longest Processing Time First) rule. For this rule, when a machine becomes available, an unscheduled job with the longest processing time is selected for processing and assigned to the first available machine. Based on the current partial schedule, this rule is used to decide both the job processing order and the machine-job assignment. Originally, the SB (Shortest Batch Processing Time First) rule is applied to determine the batch sequence [3]. However, since P_i is fixed for $i = 1, 2, \cdots, b$, the LS (List Scheduling) rule determines the order sequence. For this rule, an unscheduled job with the smallest index is scheduled first when a machine is available. We now formally describe the heuristic.

Heuristic BC

0. For $i = 1, 2, \cdots, b$, reindex the jobs so that $p_j \geq p_{j+1}$ if $j, j+1 \in B_i$.
   Set $i = j = 1$ and $F_1 = F_2 = 0$.
1. Select the first available machine $k = \arg\min\{F_1, F_2\}$. Choose job j in batch i and assign it to machine k. Set $F_k = F_k + p_j$ and $j = j + 1$.
   Repeat Step 1 until all jobs in batch i are scheduled.
2. Set $C_i = \max\{F_1, F_2\}$.
   If $i = b$, then output $\sum_{i=1}^{b} C_i$ and stop.
   Otherwise, set $i = i + 1$ and go to Step 1.

In Step 0, reindexing the jobs in each batch $i = 1, 2, \cdots, b$ requires $O(\sum_{i=1}^{b} n_i \log n_i)$ time. Since all other operations require $O(n)$ time, the time requirement of BC is

$O(n\log n)$.

In order to establish a worst case bound of heuristics, we define some notation used throughout the paper. This type of notation is first introduced in Yang and Posner [17]. For $k, i \in B$, suppose k is the last batch to complete before i in schedule σ. Let

$$\delta_i(\sigma) = \begin{cases} C_i(\sigma) - C_k(\sigma) & \text{if only one machine processes batch } i \\ |C_i(\sigma_1) - C_i(\sigma_2)| & \text{if both machines process batch } i. \end{cases}$$

For each $i \in B$, $\delta_i(\sigma)$ is the absolute difference between completion time of batch i on machine 1 and machine 2 in σ. If batch i is processed on only one machine, then $\delta_i(\sigma)$ is the difference between the completion time of batch i and the completion time of the last batch to complete before i (see Figure 1). If batch i is the first batch to complete, then we assume $C_k(\sigma) = 0$. We use $\delta_i(\sigma)$ to provide a description of the completion time of batch i. If $G \subseteq B$ is the set of batches that complete no later than batch i, then

$$C_i(\sigma) = \frac{\sum_{\ell \in G} P_\ell + \delta_i(\sigma)}{2}. \tag{1}$$

Notice that since P_ℓ is known for $\ell = 1, 2, \cdots, i$, $C_i(\sigma)$ only depends on the size of $\delta_i(\sigma)$. For $i \in B$, let $\bar{C}_i$ be the minimal makespan of just the jobs in batch i and let $\bar{\gamma}^i$ be the corresponding schedule. Similarly, let C_i^L be the makespan of the schedule generated by the LPT rule for batch i, and let γ^i be the corresponding schedule. Let $\bar{\delta}_i = 2\bar{C}_i - \bar{P}$ and $\delta_i^L = 2C_i^L - \bar{P}$ for $i \in B$. For $i \in B$, $\bar{\delta}_i$ is the difference between completion time of batch i on the two machines in $\bar{\gamma}^i$. Similarly, δ_i^L is the difference between completion time of batch i on the two machines in γ^i.

For notational convenience, let σ^{BC} be the schedule found by BC and z^{BC} be total completion time of this schedule. Also, let $\delta_i^{BC} = \delta_i(\sigma^{BC})$ for $i \in B$. Throughout this section, we assume that in σ^{BC}, batches complete in their index order and $p_j \geq p_{j+1}$ if $j, j+1 \in B_i$. To establish that the worst case bound of 7/6 for BC, we review

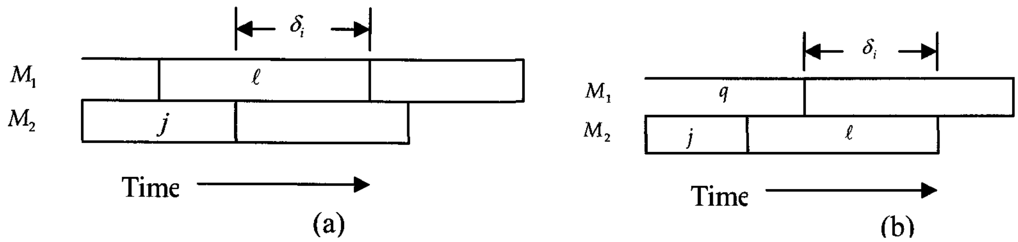

Figure 1. Examples of δ_i where the last two jobs processed in batch i are j and ℓ : (a) batch i is processed by both machines ; (b) batch i is processed only by machine 1 and q is the last job to complete in batch k (Yang and Posner [17]).

some of the preliminary results concerning δ_i^{BC}. These results are used in the analysis of BC for problem $P2 \| \Sigma C_{B_i}$ in Yang and Posner [17]. We use them in the analysis of BC for problem $P2 | P_i = \bar{P} | \Sigma C_{B_i}$ since they are of the same heuristic.

For the proofs of the following two lemmas, see Yang and Posner [17]. The next lemma establishes upper bounds for δ_i^{BC} according to completion times of schedules γ^i and $\bar{\gamma}^i$ and an upper bound of δ's from two consecutively processed batches.

Lemma 2 : (Yang and Posner [17]) *If* $C_i^L = \bar{C}_i$ *for* $i-1,\ i \in B$, *then* $\delta_i^{BC} \le \sum_{l=1}^{b} \delta_l^* + \delta_{i-1}^{BC}$. *Alternatively, if* $C_i^L > \bar{C}_i$ *for* $i-1,\ i \in B$, *then* $\delta_i^{BC} \le \max\{\delta_{i-1}^{BC},\ \bar{P}/5\}$. *Also,* $\delta_{i-1}^{BC} + \delta_i^{BC} \le \bar{P}$ *for* $i-1,\ i \in B$.

Much of our analysis is based on which job is the last one to complete in a given batch in σ^{BC}. The following lemma establishes some preliminary results for various possible final jobs.

Lemma 3 : (Yang and Posner [17]) (1) *If in* σ^{BC}, β_i *is the last job to complete in batch* $i \in B$, *then* $\delta_{i-1}^{BC} + \delta_i^{BC} \le \sum_{l=1}^{b} \delta_l^*$; *(2) If in* σ^{BC}, $\beta_i + 1$ *is the last job to complete in batch* $i \in B$, *then* $\delta_i^{BC} \le \delta_{i-1}^{BC}$; *(3) If in* σ^{BC}, $\beta_i + j$ *is the last job to complete in batch* $i \in B$, *then* $\delta_i^{BC} \le \bar{P}/(j+1)$.

In σ^*, let the batches complete in the order of $v_1,\ v_2,\ \cdots,\ v_b$. From (1),

$$\frac{z^{BC}}{z^*}=\frac{\sum_{i=1}^{b} C_i(\sigma^{BC})}{\sum_{i=1}^{b} C_i(\sigma^*)}$$
$$=\frac{b\overline{P}+\delta_1^{BC}+(b-1)\overline{P}+\delta_2^{BC}+\cdots+2\overline{P}+\delta_{b-1}^{BC}+\overline{P}+\delta_b^{BC}}{b\overline{P}+\delta_{v_1}^*+(b-1)\overline{P}+\delta_{v_2}^*+\cdots+\overline{P}+\delta_{v_b}^*}$$

$$=\frac{b\overline{P}+(b-1)\overline{P}+\cdots+\overline{P}+\sum_{i=1}^{b}\delta_i^{BC}}{b\overline{P}+(b-1)\overline{P}+\cdots+\overline{P}+\sum_{i=1}^{b}\delta_i^*}. \tag{2}$$

We review the following remark that is used to establish the bound of BC.

Remark 1. (Graham [9]) *The worst case bound of the LPT rule for the problem* $P\,||\,C_{\max}$ *is* $(4m-1)/(3m)$.

The following four propositions are used to establish a worst case bound of BC. Each proposition finds the bound for a set of problems with the different number of batches. We begin with the case where there exists only one batch.

Proposition 1: *If* $b=1$, *then* $z^{BC}/z^*\le 7/6$ *for problem* $P2\,|\,P_i=\overline{P}\,|\,\Sigma C_{B_i}$.

Proof. If $b=1$, then problem $P2\,|\,P_i=\overline{P}\,|\,\Sigma C_{B_i}$ becomes identical to problem $P2\,||\,C_{\max}$, and as a result of Remark 1, $z^{BC}/z^*\le 7/6$. □

Proposition 2: *If* $b=2$, *then* $z^{BC}/z^*\le 7/6$ *for problem* $P2\,|\,P_i=\overline{P}\,|\,\Sigma C_{B_i}$.

Proof. From (2), we need to prove that

$$\frac{z^{BC}}{z^*}=\frac{3\overline{P}+\delta_1^{BC}+\delta_2^{BC}}{3\overline{P}+\delta_1^*+\delta_2^*}\le\frac{7}{6}. \tag{3}$$

Suppose in δ^{BC}, batch 2 completes when job β_2 completes, then Lemma 3 implies that $\delta_1^{BC}+\delta_2^{BC}\le\delta_1^*+\delta_2^*$. Hence, from (3),

$$\frac{3\overline{P}+\delta_1^{BC}+\delta_2^{BC}}{3\overline{P}+\delta_1^*+\delta_2^*}=1.$$

Alternatively, if in δ^{BC}, batch 2 completes when job β_2+1 or larger finishes processing, then we consider two cases. First, if $C_1^L = \bar{C}_1$, then Lemma 2 implies $\delta_1^{BC} \le \delta_1^* + \delta_2^*$. Also, from Lemma 3, $\delta_2^{BC} \le \bar{P}/2$. Hence, from (3),

$$\frac{3\bar{P}+\delta_1^{BC}+\delta_2^{BC}}{3\bar{P}+\delta_1^*+\delta_2^*} \le 1+\frac{\delta_2^{BC}}{3\bar{P}}$$
$$\le 1+\frac{\bar{P}/2}{3\bar{P}} = \frac{7}{6}.$$

On the other hand, if $C_1^L > \bar{C}_1$, then Lemma 2 implies $\delta_1^{BC} \le \bar{P}/5$. Also, from Lemma 3, $\delta_2^{BC} \le \delta_1^{BC}$. Hence, from (3),

$$\frac{3\bar{P}+\delta_1^{BC}+\delta_2^{BC}}{3\bar{P}+\delta_1^*+\delta_2^*} \le 1+\frac{\delta_1^{BC}+\delta_2^{BC}}{3\bar{P}}$$
$$\le 1+\frac{2\bar{P}/5}{3\bar{P}} = \frac{17}{15} < \frac{7}{6}.$$

Hence, the result is established from (2). □

Proposition 3 : *If* $b=3$, *then* $z^{BC}/z^* \le 7/6$ *for problem* $P2 \mid P_i = \bar{P} \mid \sum C_{B_i}$.

Proof. From (2), we need to prove that

$$\frac{z^{BC}}{z^*} = \frac{6\bar{P}+\delta_1^{BC}+\delta_2^{BC}+\delta_3^{BC}}{6\bar{P}+\delta_1^*+\delta_2^*+\delta_3^*} \le \frac{7}{6}. \tag{4}$$

Note that if in δ^{BC}, batch 3 completes when job β_3+1 or larger finishes processing, then from Lemma 3, $\delta_3^{BC} \le \bar{P}/2$. Further, since $\delta_3^{BC}/(3\bar{P}) \le (\bar{P}/2)/(3\bar{P}) = 1/6$, $(3\bar{P}+\delta_3^{BC})/(3\bar{P}) = 7/6$. Proposition 2 implies that

$$\frac{3\bar{P}+\delta_1^{BC}+\delta_2^{BC}}{3\bar{P}+\delta_1^*+\delta_2^*+\delta_3^*} \le \frac{7}{6}.$$

Hence, we have the result.

Now, we only need to consider the case where in δ^{BC}, batch 3 completes when job β_3 completes. From Lemma 3, $\delta_2^{BC}+\delta_3^{BC} \le \delta_1^*+\delta_2^*+\delta_3^*$, and from Lemma 2,

$\delta_1^{BC} \le \overline{P}$. Hence, from (4),

$$\frac{6\overline{P}+\delta_1^{BC}+\delta_2^{BC}+\delta_3^{BC}}{6\overline{P}+\delta_1^{*}+\delta_2^{*}+\delta_3^{*}} \le 1+\frac{\delta_3^{BC}}{6\overline{P}} \le \frac{7}{6}.$$

Hence, the result is established from (2). □

Proposition 4 : *If* $b \ge 4$, *then* $z^{BC}/z^* \le 7/6$ *for problem* $P2 \mid P_i = \overline{P} \mid \sum C_{B_i}$.

Proof. If $b=4$, then Lemma 2 implies that $\delta_3^{BC}+\delta_4^{BC} \le \overline{P}$. Since $(\delta_3^{BC}+\delta_4^{BC})/(6\overline{P}) \le \overline{P}/(6\overline{P}) = 1/6$, Proposition 2 implies $z^{BC}/z^* \le 7/6$. Similarly, if $b=5$, then Lemma 2 implies that $\delta_4^{BC}+\delta_5^{BC} \le \overline{P}$. Since $(\delta_4^{BC}+\delta_5^{BC})/(6\overline{P}) \le \overline{P}/(6\overline{P}) = 1/6$, Proposition 3 implies $z^{BC}/z^* \le 7/6$. We can repeat this argument for the other cases where $b \ge 6$. □

Finally, the following theorem establishes the worst case bound of the heuristic.

Theorem 3 : *For problem* $P2 \mid P_i = \overline{P} \mid \sum C_{B_i}$, $z^{BC}/z^* \le 7/6$ *and this bound is tight.*

Proof. The result follows from Propositions 1 ~ 4. The bound is tight because LPT rule for $P2 \,||\, C_{\max}$ has a tight bound of 7/6 [9]. □

4.2 The Set LPT Heuristic

In this section, we present a new heuristic procedure to find a schedule for problem $P2 \mid P_i = \overline{P} \mid \sum C_{B_i}$. This heuristic has a worst case bound on relative error of $1+(b+6)/\{6(b^2+b)\}$ where b is the number of batches. First, the LPT rule is applied to each batch. Then, batches are reindexed such that $C_1^L \le C_2^L \le \cdots \le C_b^L$ and are sequenced in their index order. Each batch is partitioned into two sets using the LPT rule. The partitioning process assumes that processing of a batch can start on both machines at time zero. The set with the larger total processing time is assigned to the machine that becomes available first. The set which has smaller processing time is assigned to the other machine. Now, we formally describe the heuristic.

Heuristic H2

0. For $i = 1, 2, \cdots, b$, reindex the jobs so that $p_j \ge p_{j+1}$ if $j, j+1 \in B_i$.

For $v = 1, 2, \cdots, b$, set $F'_{1v} = F'_{2v} = 0$.

Set $i = j = v = 1$ and $F_1 = F_2 = 0$.

1. Assign job j in batch v to the first available machine $k = \arg\min\{F'_{1v}, F'_{2v}\}$.

   Set $F'_{kv} = F'_{kv} + p_j$ and $j = j+1$.

   Repeat Step 1 until all jobs in batch v are scheduled.
2. If $v < b$, then set $v = v+1$ and go to Step 1.
3. Reindex batches so that $\max\{F'_{11}, F'_{21}\} \le \max\{F'_{12}, F'_{22}\} \le \cdots \le \max\{F'_{1b}, F'_{2b}\}$.
4. Find ℓ and u such that $\ell = \arg\min\{F_1, F_2\}$ and $u = \arg\min\{F'_{1i}, F'_{2i}\}$.

   Set $F_\ell = F_\ell + F'_{3-u,i}$ and $F_{3-\ell} = F_{3-\ell} + F'_{ui}$.
5. Set $C_i = \max\{F_1, F_2\}$.
6. If $i < b$, then go to Step 4.

Otherwise, output $\sum_{i=1}^{b} C_i$ and stop.

In Step 0, reindexing the jobs in each batch $i = 1, 2, \cdots, b$ requires $O(\sum_{i=1}^{b} n_i \log n_i)$ time. In Step 3, reindexing batches requires $O(b \log b)$ time. Since all other operations require $O(n)$ time, the time requirement of H2 is $O(n \log n)$.

For the rest of this paper, we assume without loss of generality that when both machines are available for processing at the same time, a job is scheduled on machine 1 first in a schedule. For notational convenience, let σ^{H2} be the schedule found by H2 and z^{H2} be total completion time of this schedule. Also, let $\delta_i^{H2} = \delta_i(\sigma^{H2})$ for $i \in B$. Throughout this section, we assume that in σ^{H2}, batches complete in their index order and $p_j \ge p_{j+1}$ if $j, j+1 \in B_i$. In the following lemmas, we prove that $\delta_i^L \le \bar{\delta}_i + \bar{P}/6$ for $i = 1, 2, \cdots, b$. This result is a critical part of the proof for the worst case bound.

Lemma 4 : *If* $n_i \le 4$ *for* $i \in B$, *then* $\bar{C}_i = C_i^L$.

Proof. Recall that $\bar{\gamma}^i$ is the schedule corresponding to $\bar{C}_i$ and γ^i is the schedule corresponding to C_i^L. Observe that if $n_i \le 3$, then $\bar{\gamma}^i$ must be the same as γ^i.

If $n_i = 4$, then we consider two cases. First, suppose that $p_{\beta_i} \ge p_{\beta_i+1} + p_{\beta_i+2}$. Then, an optimal schedule is $\bar{\gamma}^i = (\bar{\gamma}_1^i, \bar{\gamma}_2^i) = ((\beta_i), (\beta_i+1, \beta_i+2, \beta_i+3))$, and this is the same as an LPT schedule. Second, suppose that $p_{\beta_i} < p_{\beta_i+1} + p_{\beta_i+2}$. Then, an optimal schedule is $\bar{\gamma}^i = ((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2))$, and this is identical to an LPT schedule. Therefore, $\bar{C}_i = C_i^L$. □

Lemma 5 : *If* $n_i = 5$ *for* $i \in B$ *and* $p_{\beta_i} \geq p_{\beta_i+1} + p_{\beta_i+2}$, *then* $\bar{C}_i = C_i^L$.

Proof. If $p_{\beta_i} \geq p_{\beta_i+1} + p_{\beta_i+2} + p_{\beta_i+3}$, then an optimal schedule $\bar{\gamma}^i = (\bar{\gamma}_1^i,\ \bar{\gamma}_2^i) = ((\beta_i),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+3,\ \beta_i+4))$, and this is the same as an LPT schedule. Alternatively, if $p_{\beta_i} < p_{\beta_i+1} + p_{\beta_i+2} + p_{\beta_i+3}$, then an optimal schedule $\bar{\gamma}^i = ((\beta_i,\ \beta_i+4),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+3))$, and this is identical to an LPT schedule. Therefore, $\bar{C}_i = C_i^L$. □

Lemma 6 : *If* $n_i = 5$, $p_{\beta_i} < p_{\beta_i+1} + p_{\beta_i+2}$, *and in* $\bar{\gamma}^i$, *the first two jobs in batch* i *are processed on the same machine for* $i \in B$, *then* $\delta_i^L \leq \bar{\delta}_i + \bar{P}/6$.

Proof. Without loss of generality, we assume that in $\bar{\gamma}^i$, jobs β_i and β_i+1 are processed on machine 1. Then, there exists only one possible optimal schedule $\bar{\gamma}^i = (\bar{\gamma}_1^i,\ \bar{\gamma}_2^i) = ((\beta_i,\ \beta_i+1),\ (\beta_i+2,\ \beta_i+3,\ \beta_i+4))$. This is due to $p_{\beta_i} + p_{\beta_i+1} \geq p_{\beta_i+2} + p_{\beta_i+3}$. Suppose $p_{\beta_i} > p_{\beta_i+2} + p_{\beta_i+3}$. Then, a better schedule is obtained by scheduling jobs β_i+1 and β_i+4 on machines 2 and 1, respectively. Contradiction. Hence, $p_{\beta_i} \leq p_{\beta_i+2} + p_{\beta_i+3}$.

For the case where $p_{\beta_i} \leq p_{\beta_i+2} + p_{\beta_i+3}$, we consider two cases. First, if $p_{\beta_i} + p_{\beta_i+1} > p_{\beta_i+2} + p_{\beta_i+3} + p_{\beta_i+4}$, then $p_{\beta_i+2} + p_{\beta_i+3} + p_{\beta_i+4} < \bar{P}/2$. Hence, $p_{\beta_i+4} < \bar{P}/6$, and the result holds. Second, if $p_{\beta_i} + p_{\beta_i+1} \leq p_{\beta_i+2} + p_{\beta_i+3} + p_{\beta_i+4}$, then $\bar{\delta}_i = p_{\beta_i+2} + p_{\beta_i+3} + p_{\beta_i+4} - p_{\beta_i} - p_{\beta_i+1}$. Since $p_{\beta_i} + p_{\beta_i+1} \leq p_{\beta_i+2} + p_{\beta_i+3} + p_{\beta_i+4}$, $p_{\beta_i} + p_{\beta_i+3} + p_{\beta_i+4} \geq p_{\beta_i+1} + p_{\beta_i+2}$ and $p_{\beta_i+1} + p_{\beta_i+2} + p_{\beta_i+4} \geq p_{\beta_i} + p_{\beta_i+4}$. Also, since $p_{\beta_i} < p_{\beta_i+1} + p_{\beta_i+2}$, γ^i must be either $((\beta_i,\ \beta_i+3,\ \beta_i+4)$, $(\beta_i+1,\ \beta_i+2))$ or $((\beta_i,\ \beta_i+3), (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$.

If $\gamma^i = ((\beta_i, \beta_i+3, \beta_i+4), (\beta_i+1, \beta_i+2))$, then $\delta_i^L = p_{\beta_i} + p_{\beta_i+3} + p_{\beta_i+4} - p_{\beta_i+1} - p_{\beta_i+2}$. Hence, $\delta_i^L - \bar{\delta}_i = 2(p_{\beta_i} - p_{\beta_i+2})$. Note that $p_{\beta_i+1} + p_{\beta_i+2} > p_{\beta_i} + p_{\beta_i+3}$ and $p_{\beta_i} + p_{\beta_i+1} \leq \bar{P}/2$. If $p_{\beta_i+4} \leq \bar{P}/6$, then the result holds from Lemma 3. Alternatively, if $p_{\beta_i+4} > \bar{P}/6$, then $p_{\beta_i+3} > \bar{P}/6$. Thus, $2(p_{\beta_i} - p_{\beta_i+2}) \leq 2(p_{\beta_i+1} - p_{\beta_i+3}) < 2(\bar{P}/4 - \bar{P}/6) \leq \bar{P}/6$. Hence, the result holds from Lemma 3.

Alternatively, if $\gamma^i = ((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$, then $\delta_i^L = p_{\beta_i+1} + p_{\beta_i+2} + p_{\beta_i+4}$

$-p_{\beta_i}-p_{\beta_i+3}$. Hence, $\delta_i^L-\overline{\delta}_i=2(p_{\beta_i+1}-p_{\beta_i+3})$. Note that $p_{\beta_i}+p_{\beta_i+3}\ge p_{\beta_i+1}+p_{\beta_i+2}$. If $p_{\beta_i+1}>\overline{P}/4$, then $p_{\beta_i}>\overline{P}/4$ and $p_{\beta_i+4}\le\overline{P}/6$. In this case, since $p_{\beta_i+4}\le\overline{P}/6$, the result holds from Lemma 3. Then, we only need to consider the case where $p_{\beta_i+1}\le\overline{P}/4$ and $p_{\beta_i+4}>\overline{P}/6$. Since $p_{\beta_i+4}>\overline{P}/6$, $p_{\beta_i+3}>\overline{P}/6$. Then, $2(p_{\beta_i+1}-p_{\beta_i+3})$ $<2(\overline{P}/4-\overline{P}/6)$. Hence, the result holds. □

Lemma 7 : *If $n_i=5$, $p_{\beta_i}<p_{\beta_i+1}+p_{\beta_i+2}$, and in $\overline{\gamma}^i$, the first two jobs in batch i are processed on the different machines for $i\in B$, then $\delta_i^L\le\overline{\delta}_i+\overline{P}/6$.*

Proof. Without loss of generality, we assume that in $\overline{\gamma}^i$, jobs β_i and β_i+1 are processed on machines 1 and 2, respectively. Then, there are three possible schedules of $\overline{\gamma}^i$.

They are $\overline{\gamma}^i=(\overline{\gamma}_1^i,\ \overline{\gamma}_2^i)=((\beta_i,\ \beta_i+3,\ \beta_i+4),\ (\beta_i+1,\ \beta_i+2))$; $((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$; $((\beta_i,\ \beta_i+2),\ (\beta_i+1,\ \beta_i+3,\ \beta_i+4))$.

Case 1 : $\overline{\gamma}^i=((\beta_i,\ \beta_i+3,\ \beta_i+4),\ (\beta_i+1,\ \beta_i+2))$.

Note that $p_{\beta_i}+p_{\beta_i+3}\le p_{\beta_i+1}+p_{\beta_i+2}$. Otherwise, a better schedule can be obtained by scheduling job β_i+4 on machine 2, contradiction. Hence, $\overline{\gamma}^i$ is the same as γ^i, and $\overline{C}_i=C_i^L$.

Case 2 : $\overline{\gamma}^i=((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$.

Note that $p_{\beta_i}+p_{\beta_i+3}>p_{\beta_i+1}+p_{\beta_i+2}$. Otherwise, a better schedule can be obtained by scheduling job β_i+4 on machine 1, contradiction. Hence, $\overline{\gamma}^i$ is the same as γ^i, and $\overline{C}_i=C_i^L$.

Case 3 : $\overline{\gamma}^i=((\beta_i,\ \beta_i+2),\ (\beta_i+1,\ \beta_i+3,\ \beta_i+4))$

Suppose that $p_{\beta_i}+p_{\beta_i+2}>p_{\beta_i+1}+p_{\beta_i+3}+p_{\beta_i+4}$. Then, $p_{\beta_i+1}+p_{\beta_i+3}+p_{\beta_i+4}\le\overline{P}/2$. Since $p_{\beta_i+4}\le p_{\beta_i+3}\le p_{\beta_i+1}$, $p_{\beta_i+4}\le\overline{P}/6$. The result holds from Lemma 3. Alternatively, suppose that $p_{\beta_i}+p_{\beta_i+2}\le p_{\beta_i+1}+p_{\beta_i+3}+p_{\beta_i+4}$. Then, $\overline{\delta}_i=p_{\beta_i+1}+p_{\beta_i+3}+p_{\beta_i+4}-p_{\beta_i}-p_{\beta_i+3}$. Also, since $p_{\beta_i}<p_{\beta_i+2}+p_{\beta_i+3}$, γ^i must be either $((\beta_i,\ \beta_i+3,\ \beta_i+4),\ (\beta_i+1,\ \beta_i+2))$ or $((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$.

Suppose $\gamma^i=((\beta_i,\ \beta_i+3,\ \beta_i+4),\ (\beta_i+1,\ \beta_i+2))$. If $p_{\beta_i}+p_{\beta_i+3}+p_{\beta_i+4}<p_{\beta_i+1}+p_{\beta_i+2}$, then $p_{\beta_i}+p_{\beta_i+3}+p_{\beta_i+4}\le\overline{P}/2$. Hence, $p_{\beta_i+4}\le\overline{P}/6$ and the result holds from Lemma 3.

Alternatively, if $p_{\beta_i}+p_{\beta_i+3}+p_{\beta_i+4}\geq p_{\beta_i+1}+p_{\beta_i+2}$, then $\delta_i^L=p_{\beta_i}+p_{\beta_i+3}+p_{\beta_i+4}-p_{\beta_i+1}-p_{\beta_i+2}$. Note that $p_{\beta_i+1}+p_{\beta_i+2}\geq p_{\beta_i}+p_{\beta_i+3}$. If $p_{\beta_i+4}\leq \overline{P}/6$, then the result holds from Lemma 3. Alternatively, if $p_{\beta_i+4}>\overline{P}/6$, then $p_{\beta_i+3}>\overline{P}/6$ and $p_{\beta_i+2}\leq(1/3)(\overline{P}-\overline{P}/3)=2\overline{P}/9$. Thus, $\delta_i^L-\overline{\delta}_i=2(p_{\beta_i}-p_{\beta_i+1})\leq 2(p_{\beta_i+2}-p_{\beta_i+3})<2(2\overline{P}/9-\overline{P}/6)=\overline{P}/9$. Hence, the result holds.

Alternatively, suppose that $\gamma^i=((\beta_i,\ \beta_i+3),\ (\beta_i+1,\ \beta_i+2,\ \beta_i+4))$. If $p_{\beta_i}+p_{\beta_i+3}>p_{\beta_i+1}+p_{\beta_i+2}+p_{\beta_i+4}$, then $p_{\beta_i+1}+p_{\beta_i+2}+p_{\beta_i+4}<\overline{P}/2$. Hence, $p_{\beta_i+4}<\overline{P}/6$ and the result holds from Lemma 3. On the other hand, if $p_{\beta_i}+p_{\beta_i+3}\leq p_{\beta_i+1}+p_{\beta_i+2}+p_{\beta_i+4}$, then $\delta_i^L=p_{\beta_i+1}+p_{\beta_i+2}+p_{\beta_i+4}-p_{\beta_i}-p_{\beta_i+3}$. If $p_{\beta_i+4}\leq\overline{P}/6$, then the result holds from Lemma 3. Alternatively, if $p_{\beta_i+4}>\overline{P}/6$, then $p_{\beta_i+3}>\overline{P}/6$ and $p_{\beta_i+2}\leq(1/3)(\overline{P}-\overline{P}/3)=2\overline{P}/9$. Thus, $\delta_i^L-\overline{\delta}_i=2(p_{\beta_i+2}-p_{\beta_i+3})<2(2\overline{P}/9-\overline{P}/6)=\overline{P}/9$. Hence, the result holds.

From Cases 1, 2, 3, and 4, we have the bound. □

Lemma 8 : *If* $n_i\geq 6$ *for* $i\in B$, *then* $\delta_i^L\leq\overline{\delta}_i+\overline{P}/6$.

Proof. If β_i+j is the last job to complete in batch $i\in B$, then $\delta_i^{BC}\leq\overline{P}/(j+1)$ (Lemma 3). When $i=1$, a schedule by BC is the same as that by H2. Hence, the result holds. □

From the five lemmas above, an upper bound of δ_i^L is established in the following proposition.

Proposition 5 : *For* $i\in B$, $\delta_i^L\leq\overline{\delta}_i+\overline{P}/6$.

Proof. Lemma 4 establishes an upper bound of δ_i^L for the case where $n_i\leq 4$ for $i\in B$. Lemmas 5, 6, and 7 establish the bound for the case where $n_i=5$ for $i\in B$. Finally, Lemma 8 establishes the bound for the case where $n_i\geq 6$ for $i\in B$. □

The following lemma establishes an upper bound of $\sum_{i=1}^{b}\delta_i^{H2}$.

Lemma 9 : *For Heuristic H2,* $\sum_{i=1}^{b}\delta_i^{H2}\leq(\delta_b^L+\sum_{i=1}^{b}\delta_i^L)/2$.

Proof. By the construction of H2, $\delta_1^L\leq\delta_2^L\leq\cdots\leq\delta_b^L$. Observe that $\delta_1^{H2}=\delta_1^L$, $\delta_2^{H2}=\delta_2^L-\delta_1^L$, $\delta_3^{H2}=\delta_3^L-\delta_2^L+\delta_1^L$, $\delta_4^{H2}=\delta_4^L-\delta_3^L+\delta_2^L-\delta_1^L$, and so on. In order to calculate $\sum_{i=1}^{b}\delta_i^{H2}$, we consider two cases. First, if b is an even number, then

$$\sum_{i=1}^{b}\delta_i^{H2}=\delta_2^L+\delta_4^L+\cdots+\delta_b^L\le(\delta_1^L+\delta_3^L+\cdots+\delta_{b-1}^L)+\delta_b^L.$$

On the other hand, if b is an odd number, then

$$\sum_{i=1}^{b}\delta_i^{H2}=\delta_1^L+\delta_3^L+\cdots+\delta_b^L\le(\delta_2^L+\delta_4^L+\cdots+\delta_{b-1}^L)+\delta_b^L.$$

Therefore, $\sum_{i=1}^{b}\delta_i^{H2}\le(\delta_b^L+\sum_{i=1}^{b}\delta_i^L)/2$. □

The next lemma establishes a lower bound for $\sum_{i=1}^{b}\delta_i^*$.

Lemma 10 : *For problem* $P2\,|\,P_i=\overline{P}\,|\,\Sigma C_{B_i}$, $\sum_{i=1}^{b}\delta_i^*\ge\sum_{i=1}^{b}\overline{\delta}_i/2$.

Proof. Suppose in σ^*, batches complete in their index order. Let P_i^k be the sum of the processing times of jobs in batch i on machine k in σ^* for $i\in B$ and $k\in\{1,2\}$. Observe that $|P_1^i-P_2^i|\ge\overline{\delta}_i$ for $i=1,2,\cdots,b$. Otherwise, C_i^L can not be an optimal makespan for jobs in batch i. Also, note that for $i=2,3,\cdots,b$, $\delta_{i-1}^*+\delta_i^*\ge|P_1^i-P_2^i|$, and $\delta_1^*=|P_1^i-P_2^i|$.

Then,

$$\begin{aligned}2\sum_{i=1}^{b}\delta_i^*&\ge2(\delta_1^*+\delta_2^*+\cdots+\delta_{b-1}^*)+\delta_b^*\\&\ge\sum_{i=1}^{b}|P_1^i-P_2^i|\\&\ge\sum_{i=1}^{b}\overline{\delta}_i.\end{aligned}$$

Therefore, $\sum_{i=1}^{b}\delta_i^*\ge\sum_{i=1}^{b}\overline{\delta}_i/2$. □

From (1) and (2),

$$z^*=\frac{b(b+1)\overline{P}}{4}+\frac{\sum_{i=1}^{b}\delta_i^*}{2}$$

and

$$z^{H2}=\frac{b(b+1)\overline{P}}{2}+\frac{\sum_{i=1}^{b}\delta_i^{H2}}{2}.$$

Hence, we have

$$z^{H2} - z^* = \frac{\sum_{i=1}^{b} \delta_i^{H2}}{2} - \frac{\sum_{i=1}^{b} \delta_i^*}{2} \tag{5}$$

Now, we prove the worst case bound of H2.

Theorem 4 : *For problem* $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, $z^{H2} / z^* \leq 1 + (b+6)/\{6(b^2+b)\}$.

Proof. From Lemma 10, $\sum_{i=1}^{b} \delta_i^* \geq \sum_{i=1}^{b} \overline{\delta}_i / 2$. Also, from Lemma 9 and Proposition 5,

$$\begin{aligned}\sum_{i=1}^{b} \delta_i^{H2} &\leq \frac{\delta_b^L + \sum_{i=1}^{b} \delta_i^L}{2} \\ &\leq \frac{\overline{P}}{2} + \frac{\sum_{i=1}^{b} \delta_i^L}{2} \\ &\leq \frac{\overline{P}}{2} + \frac{\sum_{i=1}^{b} (\overline{\delta}_i + \overline{P}/6)}{2} \\ &\leq \frac{b\overline{P} + 6\overline{P}}{12} + \frac{\sum_{i=1}^{b} \overline{\delta}_i}{2}.\end{aligned}$$

From Lemma 10 and (5),

$$\begin{aligned}z^{H2} - z^* &= \frac{b\overline{P} + 6\overline{P}}{24} + \frac{\sum_{i=1}^{b} \overline{\delta}_i}{4} - \frac{\sum_{i=1}^{b} \overline{\delta}_i}{4} \\ &\leq \frac{b\overline{P} + 6\overline{P}}{24}.\end{aligned} \tag{6}$$

If we divide both side of the inequality (6) by z^*, which is $\geq \{b(b+1)\overline{P}\}/4$, then

$$\begin{aligned}\frac{z^{H2}}{z^*} &\leq 1 + \frac{\frac{b\overline{P} + 6\overline{P}}{24}}{\frac{b(b+1)\overline{P}}{4}} \\ &= 1 + \frac{b+6}{6b(b+1)}.\end{aligned}$$

This proves the result. □

Even though the bound is not tight, it can be seen that it is fairly strong compared to the bound of BC. For example, for $b=3$, the bound is 1.125 < 7/6 and for $b=10$, the bound is ≈ 1.0242. Also, the bound converges to 1 as b goes to ∞.

The following remark shows that H2 and BC do not dominate each other.

Remark 2 : *For problem* $P2 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, *Heuristics H2 and BC do not dominate each other.*

Proof. We first present an example where BC performs better than H2. Consider the instance where $b=3$, $n_1=n_2=n_3=2$, $p_1=p_5=6$, $p_2=p_6=4$, $p_3=7$, and $p_4=3$. Since $C_1^L=C_2^L=6<C_3^L=7$, $\sigma^{H2}=(\sigma_1^{H2}, \sigma_2^{H2})$ where $\sigma_1^{H2}=(1, 6, 3)$ and $\sigma_2^{H2}=(2, 5, 4)$. The solution value is $z^{H2}=6+10+17=33$. However, $\sigma^{BC}=(\sigma_1^{BC}, \sigma_2^{BC})$ where $\sigma_1^{BC}=(1, 4, 5)$ and $\sigma_2^{BC}=(2, 3, 6)$. The solution value is $z^{BC}=6+11+15=32$.

Next, we present an example where H2 performs better than BC. Consider the instance where $b=2$, $n_1=1$, $n_2=2$, $p_1=2$, and $p_2=p_3=1$. Since $C_1^L=2>C_2^L$, $\sigma^{H2}=((2, 1), (3))$ and the solution value is $z^{H2}=1+3=4$. However, $\sigma^{BC}=((1, 2), (3))$ and the solution value is $z^{BC}=2+3=5$. □

5. More Than Two Parallel Machines

In this section, H2 is extended and applied to problem with more than two parallel machines. Recall from Theorem 2 that the recognition version of problem $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$ is unary NP-complete.

5.1 Problem $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$

We first extend H2 for problem $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$ and establish a worst case bound of the heuristic. We call this heuristic H. For the completeness, we formally describe the heuristic.

Heuristic H

0. For $i=1, 2, \cdots, b$, reindex the jobs so that $p_j \geq p_{j+1}$ if $j, j+1 \in B_i$.

For $v = 1, 2, \cdots, b$, set $F'_{1v} = F'_{2v} = \cdots = F'_{mv} = 0$.

Set $i = j = v = 1$ and $F_1 = F_2 = \cdots = F_m = 0$.

1. Assign job j in batch v to the first available machine $k = \arg\min\{F'_{1v}, F'_{2v}\}$.

   Set $F'_{kv} = F'_{kv} + p_j$ and $j = j + 1$.

   Repeat Step 1 until all jobs in batch v are scheduled.

2. If $v < b$, then set $v = v + 1$ and go to Step 1.
3. Reindex batches so that $\max_{\ell_1 \in M}\{F'_{\ell_1,1}\} \le \max_{\ell_2 \in M}\{F'_{\ell_2,2}\} \le \cdots \le \max_{\ell_m \in M}\{F'_{\ell_m,b}\}$.
4. Reindex machines so that $F_1 \le F_2 \le \cdots \le F_m$.

   Also, reindex $F'_{\ell i}$ for $\ell = 1, 2, \cdots, m$ so that $F'_{1i} \ge F'_{2i} \ge \cdots \ge F'_{mi}$.

   Set $F_\ell = F_\ell + F'_{\ell i}$ for $\ell = 1, 2, \cdots, m$.

5. Set $C_i = \max_{\ell \in M}\{F_\ell\}$.
6. If $i < b$, then go to Step 4.

   Otherwise, output $\sum_{i=1}^{b} C_i$ and stop.

In Step 0, reindexing the jobs in each batch $i = 1, 2, \cdots, b$ requires $O(\sum_{i=1}^{b} n_i \log n_i)$ time. In Step 3, reindexing batches and finding maximum makespan for each batch require $O(b \log b + mb)$ time. In Step 4, reindexing machines for each batch requires $O(bm \log m)$. Since all other operations require $O(n)$ time, the time requirement of H is $O(n \log n + bm \log m)$.

As described, the LPT rule is applied for each batch to group jobs to m partitions. The LPT makespan for each batch also determines the order of batches in the final schedule so that a batch with the smallest makespan is processed first and so on. Then, each partition is assigned to a specific machine so that a partition with the largest processing time is assigned to a machine with the smallest completion time. The heuristic repeats this step until no partition is left for scheduling.

In order to establish the bound, we define some new notation. The notation is an extended version of the notation introduced for $P2 \mid P_i = \bar{P} \mid \Sigma C_{B_i}$ in the previous section.

Suppose that in σ, batches complete in their index order. Let P_ℓ^k be sum of processing time of batch ℓ on machine k for $\ell \in B$ and $k \in M$ in σ. For $i \in B$, suppose that batch i is the last batch to complete in σ. Then, after batch i is scheduled, the total processing time on machine k is $\sum_{j=1}^{i} P_j^k$ for $k \in M$. For notational convenience, reindex machines so that $\sum_{j=1}^{i} P_j^1 \geq \sum_{j=1}^{i} P_j^2 \geq \cdots \geq \sum_{j=1}^{i} P_j^m$. Let

$$\delta_{ki}(\sigma) = \sum_{j=1}^{i} P_j^1 - \sum_{j=1}^{i} P_j^{(m-k+1)} \tag{7}$$

where $k = 1, 2, \cdots, m-1$. Notice that (7) is an extension of δ_i in the previous section to the case where $m \geq 3$. From (7), $\delta_{1i}(\sigma) \geq \delta_{2i}(\sigma) \geq \cdots \delta_{m-1,i}(\sigma)$ for each $i \in B$. We use $\delta_{ki}(\sigma)$ for $k = 1, 2, \cdots, m-1$ to provide a description of the completion time of batch i. If $G \subseteq B$ is the set of batches that complete no later than batch i, then

$$C_i(\sigma) = \frac{\sum_{\ell \in G} P_\ell + \sum_{k=1}^{m-1} \delta_{ki}(\sigma)}{m}. \tag{8}$$

Observe that since $P_\ell = \overline{P}$ for $\ell = 1, 2, \cdots, i$, $C_i(\sigma)$ depends only on the size of $\delta_{ki}(\sigma)$ for $k = 1, 2, \cdots, m-1$. Also, observe that (8) is a direct extension of (1). For notational convenience, let σ^H be the schedule found by Heuristic H and z^H be total completion time of this schedule. Also, let $\delta_i^H = \delta_i(\sigma^H)$ for $i \in B$.

The following lemma establishes an upper bound of δ_{1i}^H.

Lemma 11 : *For problem* $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, $\delta_{1i}^H \leq \overline{P}$ *for* $i \in B$.

Proof. By the construction of H and from the definition of δ_{1i}, the result holds. □

Now, we prove the worst case bound for H.

Theorem 5 : *For problem* $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, $z^H / z^* \leq 1 + 2(m-1)/(b+1)$ *where* m *is the number of machines.*

Proof. From (8) and Lemma 11,

$$\begin{aligned} z^H &\leq \frac{b(b+1)\overline{P}/2 + \sum_{i=1}^{b} \sum_{k=1}^{m-1} \delta_{ki}^H}{m} \\ &\leq \frac{b(b+1)\overline{P}/2 + (m-1)b\overline{P}}{m}. \end{aligned} \tag{9}$$

Also, from (8),

$$z^* \geq \frac{b(b+1)\overline{P}/2}{m}. \tag{10}$$

From (9) and (10),

$$\begin{aligned} \frac{z^H}{z^*} &\leq \frac{b(b+1)\overline{P}/2 + (m-1)b\overline{P}}{b(b+1)\overline{P}/2} \\ &\leq 1 + \frac{2(m-1)}{b+1}. \quad \square \end{aligned} \tag{11}$$

Note that with fixed m, the worst case bound (11) goes to 1 as the number of batches goes to ∞. In the next section, we tighten the bound for the three parallel machine case.

5.2 Problem $P3 \mid P_i = \overline{P} \mid \sum C_{B_i}$

We now apply H to problem $P3 \mid P_i = \overline{P} \mid \sum C_{B_i}$ and establish the worst case bound of the heuristic. The worst case bound of $1+2/b$ is tighter than (11) which is $1+2(m-1)/(b+1) = 1+4/(b+1)$.

For notational convenience, we call this heuristic H3. Let σ^{H3} be the schedule found by Heuristic H3 and z^{H3} be total completion time of this schedule. Also, let $\delta_i^{H3} = \delta_i(\sigma^{H3})$ for $i \in B$. Throughout this section, we assume that in σ^{H3}, batches complete in their index order and $p_j \geq p_{j+1}$ if $j,\ j+1 \in B_i$.

The following lemma establishes an upper bound of $\delta_{1i}^{H3} + \delta_{2i}^{H3}$.

Lemma 12 : *For problem* $P3 \mid P_i = \overline{P} \mid \sum C_{B_i}$, $\delta_{1i}^{H3} + \delta_{2i}^{H3} \leq 2\overline{P}$ *for* $i \in B$.

Proof. From the definition of δ_{2i} and Lemma 11, the result holds. $\square$

Suppose that batch j is processed later than batch i in a schedule for $i,\ j \in B$. We say δ_{ki} for $k \in \{1, 2, 3\}$ *is covered by* jobs $v_1, v_2, \cdots, v_\ell$ in batch j for $\ell \in n_j$ if $p_{v_1} + p_{v_2} + \cdots + p_{v_\ell} \geq \delta_{ki}$. For notational convenience, let P_i^k be sum of processing times of jobs in batch i on machine k in σ^{H3} for $k \in M$. Then, the next two results establish an important property of completion times in σ^{H3} for two consecutively processed batches.

Lemma 13 : *For problem* $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, *suppose that batches* i *and* j *are processed consecutively in* σ^{H3} *and* $P_j^{k_1} \geq P_j^{k_2} \geq P_j^{k_3}$ *for* $\{k_1, k_2, k_3\} \in M$ *and* $i, j \in B$. *Then,* $C_j(\sigma_{k_1}^{H3}) \geq C_i(\sigma^{H3})$.

Proof. We assume without loss of generality that batches are ordered in their index order in σ^{H3}. Suppose $j=2$ and $P_2^{k_1} \geq P_2^{k_2} \geq P_2^{k_3}$ for $\{k_1, k_2, k_3\} \in M$. Since $C_1^L \leq C_2^L$, $\delta_{11}^{H3} \leq \max_{k \in M}\{P_1^k\} \leq P_2^{k_1}$. Hence, $C_2(\sigma_{k_1}^{H3}) \geq C_1(\sigma^{H3})$.

Now, suppose that $j=3$ and $P_3^{\ell_1} \geq P_3^{\ell_2} \geq P_3^{\ell_3}$ for $\{\ell_1, \ell_2, \ell_3\} \in M$. Note that since $C_2^L \leq C_3^L$, $P_2^{k_1} \leq P_3^{\ell_1}$. Since $\delta_{12}^{H3} \geq \delta_{22}^{H3}$ and $P_3^{\ell_1} \geq P_3^{\ell_2} \geq P_3^{\ell_3}$, $\delta_{12}^{H3} \leq P_3^{\ell_1}$ implies that $C_3(\sigma_{k_1}^{H3}) \geq C_2(\sigma^{H3})$. To show $\delta_{12}^{H3} \leq P_3^{\ell_1}$, we consider three cases according to the machine where batch j completes.

First, suppose that $C_2(\sigma^{H3}) = C_2(\sigma_{k_1}^{H3})$. Then, $\delta_{12}^{H3} \leq P_2^{k_1} \leq P_3^{\ell_1}$. Hence, the result holds. Second, suppose that $C_2(\sigma^{H3}) = C_2(\sigma_{k_2}^{H3})$. Since $C_2(\sigma_{k_1}^{H3}) \geq C_1(\sigma^{H3})$, $\delta_{12}^{H3} \leq P_2^{k_1}$ $\leq P_3^{\ell_1}$. Hence, the result holds. Finally, suppose that $C_2(\sigma^{H3}) = C_2(\sigma_{k_3}^{H3})$. Since $C_2(\sigma_{k_1}^{H3}) \geq C_1(\sigma^{H3})$, $\delta_{12}^{H3} = \delta_{21}^{H3} - P_2^{k_2} + P_2^{k_3}$. Hence, $\delta_{12}^{H3} \leq \delta_{21}^{H3} \leq P_3^{\ell_1}$, and the result holds.

Then, we can repeat this argument for $j = 4, 5, \cdots, b$. □

Lemma 14 : *For problem* $P3 \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, *suppose that batches* i, j, *and* k *are processed consecutively in* σ^{H3} *for* $i, j, k \in B$. *Then,* $\delta_{1i}^{H3} + \delta_{2i}^{H3} + \delta_{1j}^{H3} + \delta_{2j}^{H3} + \delta_{1k}^{H3} + \delta_{2k}^{H3} \leq 3\overline{P}$.

Proof. We consider four different cases according to different combinations of completion times of any two consecutively processed batches.

Case 1 : $C_i \leq \min_{r \in M}\{C_j(\sigma_r^{H3})\}$ and $C_j \leq \min_{r \in M}\{C_k(\sigma_r^{H3})\}$.

Since $C_i \leq \min_{r \in M}\{C_j(\sigma_r^{H3})\}$, $\delta_{1i}^{H3} + \delta_{2i}^{H3} \leq \overline{P}$. Also, δ_{1j}^{H3} and δ_{2j}^{H3} are covered by jobs in batch k, and the portion of batch k, which does not cover δ_{1j}^{H3} and δ_{2j}^{H3}, is greater than or equal to δ_{1k}^{H3}. Hence, $\delta_{1j}^{H3} + \delta_{2j}^{H3} + \delta_{1k}^{H3} \leq \overline{P}$. From Lemma 11, $\delta_{2k}^{H3} \leq \overline{P}$.

Therefore, the result holds.

Case 2 : $C_i \le \min_{r\in M}\{C_j(\sigma_r^{H3})\}$ and $C_j > \min_{r\in M}\{C_k(\sigma_r^{H3})\}$.

Note that δ_{1i}^{H3} and δ_{2i}^{H3} are covered by jobs in batch j, and the portion of batch j, which does not cover δ_{1i}^{H3} and δ_{2i}^{H3},, is greater than or equal to δ_{1j}^{H3}. Hence, $\delta_{1i}^{H3}+\delta_{2i}^{H3}+\delta_{1k}^{H3} \le \bar{P}$.

If batch k completes on the machine where job β_k+1 is processed, then $C_j \le \min_{r\in M}\{C_k(\sigma_r^{H3})\}$ due to Lemma 13. Contradiction. Hence, batch k completes on the machine where either job β_k or β_k+2 is processed. Without loss of generality, we assume that jobs β_k and β_k+2 are processed on machines 1 and 3, respectively.

Suppose batch k completes on the machine where job β_k is processed. By the construction of H3, jobs in batch k on machine 1 covers δ_{1j}^{H3}, and the portion of those jobs in batch k on machine 1, which does not cover δ_{1j}^{H3}, is equal to δ_{2k}^{H3}. Hence, $\delta_{2j}^{H3}+\delta_{2k}^{H3} \le \delta_{1j}^{H3}+\delta_{2k}^{H3} \le \bar{P}$. From Lemma 11, $\delta_{1k}^{H3} \le \bar{P}$. Hence, the result holds.

Alternatively, suppose that batch k completes on the machine where job β_k+2 is processed. By the construction of H3 and from Lemma 13, $\delta_{1j}^{H3}+\delta_{2k}^{H3} \le P_k^1+P_k^3$. Hence, $\delta_{2j}^{H3}+\delta_{2k}^{H3} \le \delta_{1j}^{H3}+\delta_{2k}^{H3} \le \bar{P}$. From Lemma 11, $\delta_{1k}^{H3} \le \bar{P}$. Therefore, the result holds.

Case 3 : $C_i > \min_{r\in M}\{C_j(\sigma_r^{H3})\}$ and $C_j \le \min_{r\in M}\{C_k(\sigma_r^{H3})\}$.

Note that δ_{1j}^{H3} and δ_{2j}^{H3} are covered by jobs in batch k, and the portion of batch k, which does not cover δ_{1j}^{H3} and δ_{2j}^{H3}, is greater than or equal to δ_{1k}^{H3}. Hence, $\delta_{1j}^{H3}+\delta_{2j}^{H3}+\delta_{1k}^{H3} \le \bar{P}$.

If batch j completes on the machine where job β_j+1 is processed, then $C_i \le \min_{r\in M}\{C_j(\sigma_r^{H3})\}$ due to Lemma 13. Contradiction. Hence, batch j completes on the machine where either job β_j or β_j+2 is processed. Without loss of generality, we assume that jobs β_j and β_j+2 are processed on machines 1 and 3, respectively.

Suppose batch j completes on the machine where job β_j is processed. Then, δ_{1i}^{H3} is completely covered by jobs in batch j on machine 1, and δ_{2i}^{H3} is partially covered by remaining jobs in batch j. Uncovered portion of δ_{2i}^{H3} is covered by jobs in batch k, and the uncovered portion of δ_{2i}^{H3} is smaller than or equal to δ_{1j}^{H3}. Since $\delta_{1j}^{H3}+\delta_{2j}^{H3}+\delta_{1k}^{H3}\le \overline{P}$, $\delta_{1j}^{H3}+\delta_{2k}^{H3}\le \overline{P}$. Hence, the result holds. A similar proof can be used for the case where batch j completes on the machine where job β_j+2 is processed.

Case 4 : $C_i > \min_{r\in M}\{C_j(\sigma_r^{H3})\}$ and $C_j > \min_{r\in M}\{C_k(\sigma_r^{H3})\}$.

Note that batch k completes on the machine where either job β_k or β_k+2 is processed. Similarly, batch j completes on the machine where either job β_j or β_j+2 is processed. Without loss of generality, we assume that jobs β_j and β_j+2 are processed on machines 1 and 3, respectively. Also, we assume that jobs β_k and β_k+2 are processed on machines k_1 and k_3 for k_1, $k_3\in\{1, 2, 3\}$ and $k_1\ne k_3$, respectively.

First, δ_{1i}^{H3} is completely covered by jobs in batch j on machine 1. If batch j completes on the machine where job β_j is processed, then remaining portion of those jobs in batch j on machine 1, which does not cover δ_{1i}^{H3}, is equal to δ_{2j}^{H3}. Alternatively, if batch j completes on the machine where job β_j+2 is processed, then jobs in batch j on machine 3 is greater than or equal to δ_{2j}^{H3}. The remaining jobs of batch j partially covers δ_{2i}^{H3}, and uncovered portion of δ_{2i}^{H3} is completely covered by jobs in batch k on machine k_1.

If batch k completes on the machine where job β_k is processed, then the remaining portion of those jobs in batch k on machine k_1, which does not cover δ_{2i}^{H3}, is greater than or equal to δ_{1k}^{H3}. Also, jobs in batch k on machine k_1 can completely cover δ_{1j}^{H3}, and the remaining portion of those jobs in batch k on machine k_1, which does not cover δ_{1j}^{H3}, is equal to δ_{2k}^{H3}.

Alternatively, if batch k completes on the machine where job β_k+2 is processed, then jobs in batch k on machine k_3 and the remaining portion of those jobs in batch k on machine k_1, which does not cover δ_{2i}^{H3}, are greater than or equal to δ_{1k}^{H3}. Also, jobs in batch k on machine k_1 can completely cover δ_{1j}^{H3} from Lemma 13, and jobs in batch k on machine k_3 is greater than or equal to δ_{2k}^{H3}.

To cover all δ's, we use batch j once and batch k twice. Hence, the result holds. From Cases 1, 2, 3, and 4, we have the result. □

The next theorem establishes the worst case bound of H3.

Theorem 6 : *For problem* $P3 \mid P_i = \overline{P} \mid \sum C_{B_i}$, $z^{H3}/z^* \le 1 + 2/b$.

Proof. We first establish an upper bound of z^{H3}. We consider the following three cases of b. First, suppose that b is a multiple of three such that $b = 3\ell$ where $\ell = 1, 2, \cdots, b/3$. From Lemma 14 and (8),

$$\begin{aligned} z^{H3} &\le \frac{b(b+1)\overline{P}/2 + \sum_{i=1}^{b} \delta_{1i}^{H3}}{3} \\ &\le \frac{b(b+1)\overline{P}/2 + 3\ell\overline{P}}{3} \\ &\le \frac{b(b+1)\overline{P}/2 + b\overline{P}}{3}. \end{aligned} \tag{12}$$

Second, suppose that $b = 3\ell + 1$ where $\ell = 1, 2, \cdots, \lfloor b/3 \rfloor$. From Lemmas 12 and 14, and (8),

$$\begin{aligned} z^{H3} &\le \frac{b(b+1)\overline{P}/2 + 3\ell\overline{P} + 2\overline{P}}{3} \\ &\le \frac{b(b+1)\overline{P}/2 + (b+1)\overline{P}}{3}. \end{aligned} \tag{13}$$

Finally, suppose that $b = 3\ell + 2$ where $\ell = 1, 2, \cdots, \lfloor b/3 \rfloor$. From Lemma 14 and (8),

$$z^{H3} \le \frac{b(b+1)\overline{P}/2 + 3\ell\overline{P} + 3\overline{P}}{3}$$

$$\leq \frac{b(b+1)\overline{P}/2+(b+1)\overline{P}}{3}. \tag{14}$$

From (12), (13), and (14),

$$z^{H3} \leq \frac{b(b+1)\overline{P}/2+(b+1)\overline{P}}{3}. \tag{15}$$

Therefore, from (8) and (15),

$$\frac{z^{H3}}{z^*} \leq \frac{b(b+1)\overline{P}/2+(b+1)\overline{P}}{3}$$
$$\leq 1+\frac{2}{b}. \quad \square \tag{16}$$

Even though bound (16) is not tight, it can be seen that the bound decreases as b increases. Also, the bound converges to 1 as b goes to ∞. Also, when $m=3$, (16) is always less than or equal to (11).

6. Computational Study

We empirically evaluate BC, H2, and H (including H3) by comparing solution values generated by heuristics with a lower bound value. A lower bound value can be calculated by using the linear programming (LP) relaxation of problem $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$. Let z^L be the lower bound value. Then,

$$z^L = \frac{b\overline{P}}{m} + \frac{(b-1)\overline{P}}{m} + \cdots + \frac{\overline{P}}{m}.$$

In the LP relaxation of problem $P \mid P_i = \overline{P} \mid \Sigma C_{B_i}$, a job can be split into pieces of any size and processed, simultaneously if desired, on multiple machines. A lower bound z^L is used instead of an optimal solution value z^* because z^* is very difficult to obtain. As performance indicators of BC, H2, and H, we use upper bounds on relative errors z^{BC}/z^L, z^{H2}/z^L and z^H/z^L, respectively.

In this computational study, we compare the performances of BC, H2, and H under various conditions. We also observe the impact of different factors such as b, m, and $\overline{P}$ on the performances of BC, H2, and H.

For each problem instance, $p_j \sim DU[p^{LB}, p^{UB}]$ for $j \in N$ where p^{LB} and p^{UB} are parameters and where $DU[\ell, u]$ represents a discrete random variable uniformly distributed between ℓ and u. We use $\ell = 1$ and $u = 99$. Since $P_i = \overline{P}$ for all batches, the algorithm, which generates problem instances, checks whether accumulated p_j for each batch is greater than $\overline{P}$ while generating p_j's. If the finally added job makes accumulated p_j greater than $\overline{P}$, then the last p_j is reduced to make accumulated p_j is equal to $\overline{P}$.

We generate 1,440 test problems under 48 conditions. To test the effects of varying the number of batches b, we consider four different values of b : 1, 5, 10, and 50. To determine whether different $\overline{P}$'s have an impact on the performance of the heuristics, we consider three different values of $\overline{P}$: 250, 500, 2500. To test the effects of varying number of machines m, we consider four different values of m : 2, 3, 5, and 10. The two machine case is for BC and H2. Heuristic H(including H3) is tested for 3, 5, and 10 machine cases. For each combination of the different factors, we solve 30 problems. Table 1 presents a summary of the design for the computational study.

Table 1. Design for the Computational Study

$DU[1, 99]$							
$m = 2$		$m = 3$		$m = 5$		$m = 10$	
b	$\overline{P}$	b	$\overline{P}$	b	$\overline{P}$	b	$\overline{P}$
1	250	1	250	1	250	1	250
1	500	1	500	1	500	1	500
1	2500	1	2500	1	2500	1	2500
5	250	5	250	5	250	5	250
5	500	5	500	5	500	5	500
5	2500	5	2500	5	2500	5	2500
10	250	10	250	10	250	10	250
10	500	10	500	10	500	10	500
10	2500	10	2500	10	2500	10	2500
50	250	50	250	50	250	50	250
50	500	50	500	50	500	50	500
50	2500	50	2500	50	2500	50	2500

We now summarize the results of our study, and they are presented in Table 2. The average relative error is the average of the ratios of the solution value of a heuristic to z^L. The average relative error is calculated over the 30 replications of each test problem. Overall, the results suggest that all the heuristics perform well except for the cases where both b and $\overline{P}$ are small and m is large. Note that when $b=1$, problem $P||C_{\max}$ reduces to $P|P_i=\overline{P}|\Sigma C_{B_i}$, and from Remark 1, $z^H/z^* \leq (4m-1)/(3m)=1.3$. However, $z^H/z^L>3$, $\overline{P}=250$ and $m=10$. Remember that in LP relaxation problem, a job can be split into pieces of any size and can be processed simultaneously. Hence, as m increases, the gap between z^L and z^* increases. Hence, we strongly suspect that a large values of relative errors are due to using z^L instead of z^*.

Table 2. Performance of the Heuristics

$p_j \sim DU[1,99]$		$m=2$		$m=3$	$m=4$	s
b	$\overline{P}$	BC	H2	H(H3)	H	H
1	250	1.04187	1.04187	1.10320	1.10320	1.10320
	500	1.01107	1.01107	1.03100	1.07300	1.76667
	2500	1.00077	1.00077	1.00200	1.00487	1.02040
5	250	1.01801	1.01150	1.02043	1.13396	1.49707
	500	1.00438	1.00307	1.00756	1.01980	1.16240
	2500	1.00021	1.00014	1.00041	1.00122	1.00444
10	250	1.00940	1.00553	1.01146	1.06824	1.25726
	500	1.00246	1.00146	1.00380	1.01070	1.08592
	2500	1.00011	1.00007	1.00020	1.00054	1.00225
50	250	1.00208	1.00108	1.00230	1.01472	1.05430
	500	1.00054	1.00029	1.00081	1.00215	1.01764
	2500	1.00002	1.00001	1.00004	1.00012	1.00045

Even though H2 and BC do not dominate each other in theory (Remark 2), the results corresponding to the case where $m=2$ indicates that H2 consistently performs better than BC for all combinations of cases. Furthermore, the detailed results show that for all 420 problem instances where $m=2$, H2 performs no worse than BC.

While there may be errors due to the use of z^L instead of the optimal value, as $\overline{P}$ increases, performance of all the heuristics seems to improve. The same thing happens as b increases. Finally, for H, the larger the number of machines, the bigger the average error bound.

7. Discussion and Further Research

We have explored problem $P \mid P_i = \overline{P} \mid \sum C_{B_i}$. Even though condition $P_i = \overline{P}$ seems restrictive, the problem is more complicated than $P \mid \mid C_{\max}$ from a research perspective and has a practical application. Simple reductions show that $P2 \mid P_i = \overline{P} \mid \sum C_{B_i}$ is at least binary NP-complete and $P3 \mid P_i = \overline{P} \mid \sum C_{B_i}$ is unary NP-complete. For the two parallel machine case, we find a tight worst case bound for a known heuristic BC. Also, for a new heuristic H2, we establish a strong worst case bound which goes to 1 as the number of batches goes to infinity. In theory, neither heuristic is superior for all instances, but the computational study suggests that H2 may perform better than BC for practical problems. Heuristic H2 is extended to an arbitrary number of parallel machine case. For a fixed number of machines, we find a worst case bound which goes to 1 as the number of batches goes to infinity. Finally, we tighten the bound for the three parallel machine case.

There are several extensions of our research that might be considered. We can study different machine speeds such as proportional and unrelated parallel machines. Also, different shop environments can be considered, such as job shop, open shop, and flow shop. These different shop environments have a variety of realistic applications. Finally, it still has to be determined whether $P2 \mid P_i = \overline{P} \mid \sum C_{B_i}$ is binary or unary NP-Complete. We also leave this to future research.

References

[1] Ahmadi, R., U. Bagchi, and T. A. Roemer, "Coordinated Scheduling of Customer Orders for Quick Response," *Naval Research Logistics* 52 (2005), 493-512.

[2] Baker, K. R., "Scheduling the Production of Components at a Common Facility," *IIE Transactions* 20 (1988), 32-35.

[3] Blocher, J. D. and D. Chhajed, "The Customer Order Lead Time Problem on Parallel Machines," *Naval Research Logistics* 43 (1996), 629-654.

[4] Blocher, J. D., D. Chhajed, and M. Leung, "Customer Order Scheduling in a General Job Shop Environment," *Decision Science* 29 (1998), 951-981.

[5] Bruno, J., E. G. Coffman and R. Sethi, "Scheduling Independent Tasks to Reduce Mean Finishing Time," *Communications of the ACM* 17 (1974), 382-387.

[6] Coffman, E. G., A. Nozari, and M. Yannakakis, "Optimal Scheduling of Products with Two Subassemblies on a Single Machine," *Operations Research* 37 (1989), 426-436.

[7] Ding, F. Y., "A Pairwise Interchange Solution Procedure for a Scheduling Problem with Production of Components at a Single Facility," *Computers and Industrial Engineering* 18 (1990), 325-331.

[8] Gerodimos, A. E., C. A. Glass, and C. N. Potts, "Scheduling the Production of Two-Component Jobs on a Single Machine," *European Journal of Operational Research* 120 (2000), 250-259.

[9] Graham, R. L., "Bounds on Multiprocessing Timing Anomalies," *SIAM Journal on Applied Mathematics* 17 (1969), 416-429.

[10] Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Machine Scheduling: A Survey," *Annals of Discrete Mathematics* 5 (1979), 287-326.

[11] Gupta, J. N. D., J. C. Ho, and A. A. van der Veen, "Single Machine Hierarchical Scheduling with Customer Orders and Multiple Job Classes," *Annals of Operations Research* 70 (1997), 127-143.

[12] Julien, F. M. and M. J. Magazine, "Scheduling Customer Orders: An Alternative Production Scheduling Approach," *Journal of Manufacturing and Operations Management* 3 (1990), 177-199.

[13] Liao, C. J., "Optimal Scheduling of Products with Common and Unique Components," *International Journal of Systems Science* 27 (1996), 361-366.

[14] Roemer, T. A. and R. Ahmadi, "Complexity of Scheduling Customer Orders," Working Paper, Anderson School at UCLA (1997), USA.

[15] Santos, C. and M. Magazine, "Batching in Single Operation Manufacturing Systems," *Operations Research Letters* 4 (1985), 99-103.

[16] Yang, J., "The Complexity of Customer Order Scheduling Problems on Parallel Machines," *Computers and Operations Research* 32 (2005), 1921-1939.

[17] Yang, J. and M. E. Posner, "Scheduling Parallel Machines for the Customer Order Problem," *Journal of Scheduling* 8 (2005), 49-74.