Preemptive and Non-Preemptive Scheduling on Two Unrelated Parallel Machines

Alan J. Soper and Vitaly A. Strusevich
School of Computing and Mathematical Sciences, University of Greenwich,
Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, U.K.
e-mail: {A.J.Soper,V.Strusevich}@greenwich.ac.uk

Abstract

In this paper, for the problem of minimizing the makespan on two unrelated parallel machines we compare the quality of preemptive and non-preemptive schedules.

It is known that there exists an optimal preemptive schedule with at most two preemptions. We show that the power of preemption, i.e., the ratio of the makespan computed for the best non-preemptive schedule to the makespan of the optimal preemptive schedule is at most 3/2. We also show that the ratio of the makespan computed for the best schedule with at most one preemption to the makespan of the optimal preemptive schedule is at most 9/8. For both models, we present polynomial-time algorithms that find schedules of the required quality. The established bounds match those previously known for a less general problem with two uniform machines. We have found one point of difference between the uniform and unrelated machines: if an optimal preemptive schedule contains exactly one preemption then the ratio of the makespan computed for the best non-preemptive schedule to the makespan of the optimal preemptive schedule is at most 4/3 if the two machines are uniform and remains 3/2 if the machines are unrelated.

Keywords: Unrelated parallel machines; power of preemption; quality of a single preemption

1 Introduction

In parallel machine scheduling, we are given the jobs of set $N = \{J_1, J_2, \dots, J_n\}$ and m parallel machines $M_1, M_2, \dots, M_m$. If a job $J_j \in N$ is processed on machine M_i alone, then its processing time is known to be p_{ij}. In the scheduling literature, there is a distinction between the following three types of parallel machines: (i) *identical* parallel machines, for which the processing times are machine-independent, i.e., $p_{ij} = p_j$; (ii) *uniform* parallel machines, which have different speeds, so that $p_{ij} = p_j/s_i$, where s_i denotes the *speed* of machine M_i; and (iii) *unrelated* parallel machines, for which the processing time of a job depends on the machine assignment.

In a non-preemptive schedule, each job is assigned to a machine and is processed there without interruption. In a preemptive schedule, the processing of a job on a machine can be interrupted at any time and then resumed either on this or on any other machine, provided that the job is not processed on two or more machines at a time, and the amount of processing assigned to each machine guarantees that the job is fully completed.

In all problems considered in this paper the objective is to minimize the *makespan,* i.e., the maximum completion time across all m machines. For a schedule S, the makespan is denoted by $C_{\max}(S)$. For an instance of a scheduling problem on parallel machines, let $S^*_{(q)}$ and S^*_p denote an optimal schedule with at most q preemptions, and an optimal preemptive schedule which uses an unlimited number of preemptions, respectively. We will refer to schedules with an unlimited number of preemptions as simply preemptive. The case $q = 0$ corresponds to a non-preemptive schedule, and an optimal non-preemptive schedule is denoted by $S^*_{(0)}$.

For a scheduling problem to minimize the makespan $C_{\max}$ on m parallel machines (identical, uniform or unrelated), we measure the quality of a schedule with at most q preemptions by establishing a tight upper bound $\rho_m^{(q)}$ on the ratio $C_{\max}(S^*_{(q)})/C_{\max}(S^*_p)$ across all instances of the problem at hand. The value of $\rho_m^{(q)}$ determines what can be gained regarding the maximum completion time if instead of at most q preemptions any number of preemptions is allowed. For $q = 0$ the ratio $\rho_m^{(0)}$ is often called the *power of preemption*.

In this paper, we mainly focus on the problem with two unrelated parallel machines, i.e., $m = 2$. Section 2 provides an overview of known relevant results. In Section 3, we show that $\rho_2^{(0)} = 3/2$, i.e., for the problem on two unrelated machines the power of preemption coincides with the value established for a less general problem on two uniform machines; see (Woeginger, 2000), as well as (Jiang et al., 2014; Soper and Strusevich, 2014a,b). We also show that for $m \geq 3$ parallel machine the power of preemption on unrelated machines is larger than that on uniformly related machines. In Section 4, we prove that for the problem with two unrelated parallel machines the equality $\rho_2^{(1)} = 9/8$ holds, i.e., an optimal schedule with at most one preemption has a makespan that is at most 9/8 times worse than the makespan in the optimal preemptive schedule S^*_p. Again, the value of $\rho_2^{(1)}$ is the same for the less general problem on two uniform machines; see (Jiang et al., 2014; Soper and Strusevich, 2018, 2019). If an optimal preemptive schedule S^*_p contains one preempted job, for two unrelated machines the power of preemption $\rho_2^{(0)}$ remains 3/2, however, as we demonstrate in Section 5, it decreases to 4/3 if the machines are uniformly related. Section 6 contains concluding remarks.

2 Preliminaries

For a scheduling problem on parallel machines, we compare the quality of an optimal schedule with at most q preemptions $S^*_{(q)}$ and a schedule with any number of preemptions S^*_p. In this section, we briefly review the relevant results on determining value $\rho_m^{(q)}$, an upper bound on the ratio $C_{\max}(S^*_{(q)})/C_{\max}(S^*_p)$.

We start with discussing the complexity of the corresponding problems. Recall that finding an optimal non-preemptive schedule on parallel machines is NP-hard even for the case of two identical machines. If any number of preemptions is allowed, then all these problems are polynomially solvable, even in the most general setting with unrelated machines. See a focused survey by Chen (2004) on parallel machine scheduling with the makespan objective for details and references. The number of preemptions in an optimal schedule S^*_p need not exceed $m - 1$ in the case of identical machines, as proved by McNaughton (1959), and $2(m - 1)$ in the case of uniform machines; see (Gonzalez and Sahni, 1978). For unrelated machines, it has been shown in (Lawler and Labetoulle, 1978) that an optimal preemptive schedule requires no more than $O(m^2)$ preemptions.

For $m = 2$, the number of preemptions in an optimal schedule S^*_p is at most one if the machines are identical and at most two, if the machines are either uniform or unrelated; see (Gonzalez et al., 1990). For the latter settings, an optimal preemptive schedule can be found in $O(n)$ time.

If the makespan is minimized in the class of schedules with a restricted number of preemptions, then for $m = 2$, the problem of finding an optimal schedule with at most one preemption on uniform machines is polynomially solvable, while in the case of unrelated machine it is NP-hard; see (Soper and Strusevich, 2019). For $m \geq 3$, the problem of finding an optimal schedule $S^*_{(q)}$ with at most $q \leq m-2$ preemptions on m parallel machines is NP-hard due to Shchepin and Vakhania (2008), even if the machines are identical.

In order to determine the exact value of $\rho_m^{(q)}$ for a particular problem and to give the concept some practical meaning, the following should be done:

(i) demonstrate that the inequality

$$\frac{C_{\max}(S^*_{(q)})}{C_{\max}\left(S^*_p\right)} \leq \rho_m^{(q)} \tag{1}$$

holds for all instances of the problem;

(ii) exhibit instances of the problem for which (1) holds as equality, i.e., show that the value of $\rho_m^{(q)}$ is tight; and

(iii) develop a polynomial-time algorithm that finds a heuristic schedule $S_{(q)}$ with at most q preemptions such that

$$\frac{C_{\max}(S^*_{(q)})}{C_{\max}\left(S^*_p\right)} \leq \frac{C_{\max}\left(S_{(q)}\right)}{C_{\max}\left(S^*_p\right)} \leq \rho_m^{(q)}. \tag{2}$$

Most of the known results in this area address the situation of $q = 0$, i.e., are aimed at comparing an optimal non-preemptive schedule with an optimal preemptive schedule in order to determine the power of preemption $\rho_m^{(0)}$.

If the machines are identical parallel, then $\rho_m^{(0)} = 2 - 2/(m+1)$, as independently proved in (Braun and Schmidt, 2003) and in (Lee and Strusevich, 2005). It is shown by Rustogi and Strusevich (2013) that the value of $\rho_m^{(0)}$ can be reduced for some instances that contain jobs that are longer than the average machine load.

According to Woeginger (2000), for m uniform parallel machines $\rho_m^{(0)} = 2 - 1/m$. Soper and Strusevich (2014b) give the necessary and sufficient conditions under which the global bound of $2 - 1/m$ is tight. For two uniform machines, a parametric analysis of the power of preemption $\rho_2^{(0)}$ with respect to the speed of the faster machine is independently performed in (Jiang et al., 2014) and in (Soper and Strusevich, 2014a). For $m = 3$, a similar analysis is contained in (Soper and Strusevich, 2014a), provided that the machine speeds take at most two values.

Before we pass to discussing the power of preemption on unrelated machines, we briefly review relevant results on approximation algorithms for the problem of minimizing the makespan on m unrelated machines with no preemption allowed. Recall that for that problem a polynomial-time heuristic algorithm is called a ρ-approximation algorithm if it creates a non-preemptive schedule S^H such that $C_{\max}\left(S^H\right) \leq \rho C_{\max}(S^*_{(0)})$; here ρ is called the worst-case performance ratio. As shown in (Lawler and Labetoulle, 1978), finding an optimal preemptive schedule S_p^* on m parallel unrelated machines can be done by the following two-phase method: in the first phase, a linear programming (LP) problem is solved and its solution defines the optimal makespan $C_{\max}\left(S_p^*\right)$ and total durations of processing each job on each machine; in the second phase, schedule S_p^* is found by solving an artificial open shop scheduling problem. The LP problem to be solved in the first phase can be formulated as

$$\begin{array}{lll} \text{minimize} & T & \\ \text{subject to} & \sum_{j=1}^{n} p_{ij}x_{ij} \le T, & 1 \le i \le m; \\ & \sum_{i=1}^{m} p_{ij}x_{ij} \le T, & 1 \le j \le n; \\ & \sum_{i=1}^{m} x_{ij} = 1 & 1 \le j \le n; \\ & 0 \le x_{ij} \le 1 & 1 \le i \le m, 1 \le j \le n. \end{array} \tag{3}$$

Here, the minimum value of T corresponds to the optimal makespan $C_{\max}\left(S_p^*\right)$, while x_{ij} denotes the portion of job J_j processed on machine M_i.

The link of the problem of finding S_p^* to linear programming forms the basis of most approximation algorithms for finding a non-preemptive schedule. As shown in (Potts, 1985), finding an optimal non-preemptive schedule $S_{(0)}^*$ reduces to the following integer linear programming problem with Boolean variables x_{ij}:

$$\begin{array}{lll} \text{minimize} & T & \\ \text{subject to} & \sum_{j=1}^{n} p_{ij}x_{ij} \le T, & 1 \le i \le m \\ & \sum_{i=1}^{m} x_{ij} = 1 & 1 \le j \le n, \\ & x_{ij} \in \{0,1\} & 1 \le i \le m, 1 \le j \le n. \end{array} \tag{4}$$

If problem (4) is solved and the optimal value T^* of the objective function and the optimal values x_{ij}^* of the decision variables are found, then there exists an optimal non-preemptive $S_{(0)}^*$ such that $C_{\max}(S_{(0)}^*) = T^*$ and job J_j is processed on machine M_i if and only if $x_{ij}^* = 1$. Potts (1985) introduces a relaxation LP problem by replacing the integrality constraint $x_{ij} \in \{0,1\}$ by the inequality $x_{ij} \ge 0$ for each decision variable. Based on a solution to this relaxation problem, Potts (1985) presents a rounding procedure which leads to a heuristic non-preemptive schedule S^{LPE} such that $C_{\max}\left(S^{LPE}\right)/C_{\max}(S_{(0)}^*) \le 2$; however, the running time of such an algorithm is not polynomial with respect to m. An improved algorithm based on a rounding scheme applied to another relaxation LP problem is given in (Lenstra et al., 1990), it is a 2-approximation algorithm that requires polynomial time. In the same paper it is proved that for this problem the existence of an approximation algorithm with a worst-case bound $\rho < \frac{3}{2}$ would imply that $P = NP$. A rounding procedure in (Shchepin and Vakhania, 2005) applied to the relaxation of (4) leads to a $(2 - 1/m)$-approximation algorithm.

In the case of $m = 2$ unrelated machines, the best known approximation algorithms for finding a non-preemptive schedule guarantee a worst-case performance ratio $\rho = 3/2$; see a purpose-built algorithm in (Potts, 1985) and a general algorithm in (Shchepin and Vakhania, 2005) applied to $m = 2$.

None of the quoted results on approximation algorithms for finding a non-preemptive schedule for the problem with unrelated machines can be directly used for determining the power of preemption for that machine environment. The reason is that in all quoted papers the used lower bounds on the optimal makespan $C_{\max}(S^*_{(0)})$ for non-preemptive schedules are not lower bounds on an optimal makespan $C_{\max}\left(S^*_p\right)$ for preemptive schedules.

The first result that provides a bound on the power of preemption $\rho_m^{(0)}$ on m unrelated parallel machines is due to Lin and Vitter (1992). The authors take a solution to the LP problem (3) and convert it to a solution that satisfies the constraints

$$\begin{aligned} &\sum_{j=1}^{n} p_{ij}x_{ij} \le 2C_{\max}(S^*_p), && 1 \le i \le m \\ &\sum_{i=1}^{m} x_{ij} = 1, && 1 \le j \le n, \\ &0 \le x_{ij} \le 1, && 1 \le i \le m, 1 \le j \le n. \end{aligned}$$

Then, using the rounding scheme by Lenstra et al. (1990) this solution can be converted into a solution that satisfies the constraints in problem (4) with $T = 4C_{\max}(S^*_p)$. An alternative presentation of the same approach is contained in (Correa et al., 2012) and is based on the rounding scheme by Shmoys and Tardos (1993) applicable to the generalized assignment problem. This implies that $\rho_m^{(0)}$ on m unrelated parallel machines is at most 4. Correa et al. (2012) develop an iterative procedure which for any $\beta \in [2, 4)$ generates an instance such that $C_{\max}(S^*_{(0)}) \ge \beta C_{\max}\left(S^*_p\right)$. Therefore, the bound of 4 on $\rho_m^{(0)}$ cannot be improved for an arbitrary m.

Only a few studies compare optimal schedules $S^*_{(q)}$ with at most q preemptions to optimal preemptive schedules S^*_p. For identical parallel machines, Braun and Schmidt (2003) prove that $\rho_m^{(q)} = (2m)\,/\,(m + q + 1)$, where $0 \le q \le m - 1$. Jiang et al. (2014) show that in the case of two uniform machines $\rho_2^{(1)} = 9/8$. For m uniform machines, Soper and Strusevich (2019) prove that $\rho_m^{(1)} = 2 - 2/m$. A parametric analysis of the quality of optimal schedules with a single preemption is performed in (Jiang et al., 2014) for two uniform machines and in (Soper and Strusevich, 2018) for three uniform machines.

Regarding objective functions other than the makespan, the power of preemption for the problems of minimizing the weighted total completion time is determined in (Epstein and Levin, 2016) for a single machine and in (Sitters, 2017) for m unrelated machines. For the problem of minimizing the total completion time on m uniform parallel machines the power of preemption is derived in (Epstein et al., 2017).

3 Power of Preemption

The power of preemption $\rho_m^{(0)} = 4$ on m unrelated machines holds for sufficiently large values of m. Finding the exact values of the power of preemption for small values of m is of interest. In this section, we prove that for the problem of minimizing the makespan on two unrelated

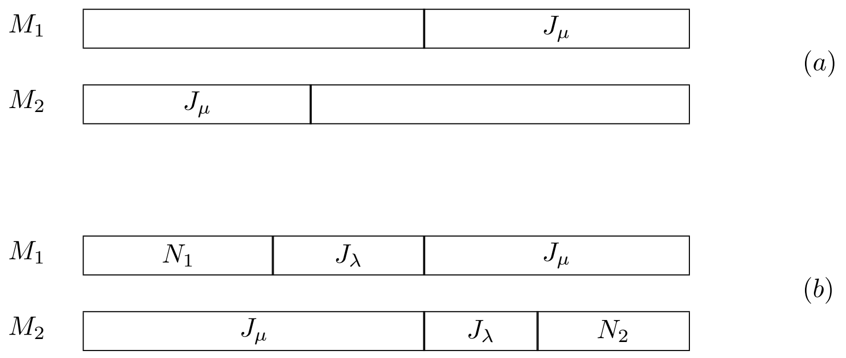

Figure 1: (a) schedule S_p^* with one preempted job J_μ, (b) schedule S_p^* with two preempted jobs J_μ and J_λ

machines the power of preemption is at most 3/2, i.e.,

$$\frac{C_{\max}(S_{(0)}^*)}{C_{\max}\left(S_p^*\right)} \leq \rho_2^{(0)} = \frac{3}{2}, \tag{5}$$

and this bound is tight. We also show how to find a non-preemptive schedule S^H such that

$$\frac{C_{\max}\left(S^H\right)}{C_{\max}\left(S_p^*\right)} \leq \frac{3}{2}. \tag{6}$$

According to Gonzalez et al. (1990), in an optimal schedule either one job or two jobs need be processed with preemption, see Figure 1. In Figure 1(b), the sets of jobs processed without preemption on machines M_1 and M_2 are labeled as N_1 and N_2, respectively.

3.1 One Preemption in an Optimal Schedule

Assume that job J_μ is processed with preemption. Without loss of generality, assume that $p_{1\mu} \leq p_{2\mu}$; otherwise, the machines can be appropriately renamed. Then

$$C_{\max}\left(S_p^*\right) \geq p_{1\mu}. \tag{7}$$

Let $x_{i\mu}$ be the portion of job J_μ processed on machine M_i, $i \in \{1,2\}$, where

$$x_{1\mu} + x_{2\mu} = 1. \tag{8}$$

The total processing time of job J_μ on machine M_i is $x_{i\mu}p_{i\mu}$, so that

$$C_{\max}\left(S_p^*\right) \geq x_{1\mu}p_{1\mu} + x_{2\mu}p_{2\mu}. \tag{9}$$

Consider the following algorithm for transforming an optimal preemptive schedule S_p^* into a schedule S^H, in which job J_μ is processed without preemption.

Algorithm 1

Step 1. Find schedule S_p^* with job J_μ being a preempted job. Create schedule $S^{\mu 1}$ by keeping all jobs as they are assigned to the machines in schedule S_p^* except job J_μ is entirely processed on machine M_1.

Step 2. Similarly, transform schedule S_p^* to a non-preemptive schedule $S^{\mu 2}$ by assigning job J_μ to be processed on machine M_2.

Step 3. Compute $C_{\max}\left(S^{\mu 1}\right)$ and $C_{\max}\left(S^{\mu 2}\right)$. Output schedule S^H as that of schedules $S^{\mu 1}$ and $S^{\mu 2}$ which has the smaller makespan.

Since finding schedule S_p^* requires $O(n)$ time, the running time of Algorithm 1 is linear in n.

Lemma 1 *Given an optimal preemptive schedule S_p^* in which job J_μ is processed with preemption. Algorithm 1 finds schedule S^H in which the number of preemptions is reduced by 1 compared to schedule S_p^* and for which the bound (6) holds.*

Proof: It follows that

$$\begin{aligned} C_{\max}\left(S^{\mu 1}\right) &= C_{\max}\left(S_p^*\right) + x_{2\mu} p_{1\mu}; \\ C_{\max}\left(S^{\mu 2}\right) &= C_{\max}\left(S_p^*\right) + x_{1\mu} p_{2\mu}, \end{aligned}$$

so that

$$C_{\max}\left(S^H\right) = \min\left\{C_{\max}\left(S^{\mu 1}\right), C_{\max}\left(S^{\mu 2}\right)\right\} = C_{\max}\left(S_p^*\right) + \min\left\{x_{2\mu} p_{1\mu}, x_{1\mu} p_{2\mu}\right\}.$$

Assume that the lemma does not hold, i.e., $C_{\max}\left(S^H\right) > \frac{3}{2} C_{\max}\left(S_p^*\right)$. Then

$$\min\left\{x_{2\mu} p_{1\mu}, x_{1\mu} p_{2\mu}\right\} > \frac{1}{2} C_{\max}\left(S_p^*\right),$$

which due to (8) and (9) implies that

$$x_{2\mu} p_{1\mu} + x_{1\mu} p_{2\mu} = \left(1 - x_{1\mu}\right) p_{1\mu} + x_{1\mu} p_{2\mu} > C_{\max}\left(S_p^*\right) \geq x_{1\mu} p_{1\mu} + \left(1 - x_{1\mu}\right) p_{2\mu}.$$

This yields

$$x_{1\mu}\left(p_{2\mu} - p_{1\mu}\right) - \left(1 - x_{1\mu}\right)\left(p_{2\mu} - p_{1\mu}\right) > 0,$$

so that $x_{1\mu} > \frac{1}{2}$. But then

$$C_{\max}\left(S^H\right) \leq C_{\max}\left(S^{\mu 1}\right) = C_{\max}\left(S_p^*\right) + \left(1 - x_{1\mu}\right) p_{1\mu} < C_{\max}\left(S_p^*\right) + \frac{1}{2} p_{1\mu} \leq \frac{3}{2} C_{\max}\left(S_p^*\right),$$

where the last inequality is due to (7). ■

Notice that Lemma 1 holds irrespective of the number of preemptions in schedule S_p^*. Applying the lemma to an optimal schedule S_p^*, in which job J_μ is the only job that is processed with preemption, we obtain the following statement.

Lemma 2 *Given an optimal preemptive schedule S_p^* in which the only job processed with preemption is job J_μ, Algorithm 1 finds a non-preemptive schedule S^H for which the bound (6) holds and that bound is tight.*

	J_1	J_2	J_3
M_1	1	2	2
M_2	2	1	2

Table 1: Tightness example for Lemma 2

Proof: The fact that (6) holds, immediately follows from Lemma 1. To see that the bound (6) is tight, consider the instance of the problem on two unrelated machines with three jobs and the processing times given in Table 1.

It is easy to verify that in an optimal preemptive schedule S_p^* in the time interval $[0, 1]$ machine M_1 processes job J_1 while machine M_2 processes job J_3; in the time interval $[1, 2]$ machine M_1 completes processing of job J_3, while machine M_2 processes job J_2. In this schedule, $J_\mu = J_3$, $x_{1\mu} = x_{2\mu} = 0.5$ and $C_{\max}\left(S_p^*\right) = 2$. On the other hand, schedule S^H is an optimal non-preemptive schedule for this instance. In such a schedule machine M_1 processes job J_1, machine M_2 processes job J_2 and job J_3 is either assigned to machine M_1 (as in schedule $S^{\mu 1}$) or to machine M_2 (as in schedule $S^{\mu 2}$). Since $C_{\max}\left(S^H\right) = 3$, we see that for this instance (6) holds as equality. ■

3.2 Two Preemptions in an Optimal Schedule

Consider an optimal schedule S_p^* with two preempted jobs, J_λ and J_μ, so that the equalities (8) and

$$x_{1\lambda} + x_{2\lambda} = 1 \tag{10}$$

hold. According to Gonzalez et al. (1990), we may assume that the jobs and the machines are numbered in such a way that (9) holds as equality, i.e.,

$$C_{\max}\left(S_p^*\right) = x_{1\mu}p_{1\mu} + x_{2\mu}p_{2\mu} \tag{11}$$

and additionally the inequalities

$$\begin{aligned} C_{\max}\left(S_p^*\right) &\geq x_{1\lambda}p_{1\lambda} + x_{2\lambda}p_{2\lambda}; & (12)\\ \frac{p_{1\lambda}}{p_{2\lambda}} &\leq \frac{p_{1\mu}}{p_{2\mu}} \leq 1; & (13)\\ x_{1\lambda}p_{1\lambda} &\leq x_{2\mu}p_{2\mu}; & (14)\\ x_{2\lambda}p_{2\lambda} &\leq x_{1\mu}p_{1\mu}. & (15) \end{aligned}$$

hold. The schedule of such a structure is shown in Figure 1(b).

The algorithm below transforms an optimal preemptive schedule S_p^* by trying all options to process both preempted jobs without interruption.

Algorithm 2

Step 1. Having found schedule S_p^* with jobs J_λ and J_μ processed with preemption, create a non-preemptive schedule $S^{\lambda 1 \mu 1}$ by keeping all jobs as they are assigned to the machines in schedule S_p^* except jobs J_λ and J_μ are entirely processed on machine M_1.

Step 2. Similarly, transform schedule S_p^* to a non-preemptive schedule $S^{\lambda 2 \mu 2}$ by assigning both jobs J_λ and J_μ to be processed on machine M_2.

Step 3. Transform schedule S_p^* to a non-preemptive schedule $S^{\lambda 1 \mu 2}$ by assigning job J_λ to be processed on machine M_1 and J_μ to be processed on machine M_2.

Step 4. Transform schedule S_p^* to a non-preemptive schedule $S^{\lambda 2 \mu 1}$ by assigning job J_λ to be processed on machine M_2 and J_μ to be processed on machine M_1.

Step 5. Compute the makespan of all four found schedules. Output schedule S^H as that of schedules $S^{\lambda 1 \mu 1}, S^{\lambda 2 \mu 2}, S^{\lambda 1 \mu 2}$ and $S^{2 \mu 1}$ which has the smaller makespan.

The running time of Algorithm 2 is $O(n)$.

Lemma 3 *For schedule S^H found by Algorithm 2 the bound (6) holds, and that bound is tight.*

Proof: Throughout the proof, denote

$$C^* = C_{\max}\left(S_p^*\right). \tag{16}$$

It follows from the actions taken in Steps 1-4 of Algorithm 2 that

$$\begin{aligned}
C_{\max}\left(S^{\lambda 1 \mu 1}\right) &= C^* + x_{2\mu}p_{1\mu} + x_{2\lambda}p_{1\lambda}; \\
C_{\max}\left(S^{\lambda 2 \mu 2}\right) &= C^* + x_{1\mu}p_{2\mu} + x_{1\lambda}p_{2\lambda}; \\
C_{\max}\left(S^{\lambda 1 \mu 2}\right) &= C^* + \max\left\{x_{2\lambda}p_{1\lambda} - x_{1\mu}p_{1\mu}, x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda}\right\}; \\
C_{\max}\left(S^{\lambda 2 \mu 1}\right) &= C^* + \max\left\{x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}, x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu}\right\}.
\end{aligned}$$

Since $x_{1\mu}p_{1\mu} \geq x_{2\lambda}p_{2\lambda}$ due to (15) and $p_{2\lambda} \geq p_{1\lambda}$ due to (13) it follows that $x_{2\lambda}p_{1\lambda} - x_{1\mu}p_{1\mu} \leq 0$, so that

$$C_{\max}\left(S^{\lambda 1 \mu 2}\right) = C^* + x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda}.$$

We split our further consideration into two parts, depending on the structure of schedule $S^{\lambda 2 \mu 1}$.

Part 1: Assume that

$$\max\left\{x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}, x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu}\right\} = x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}.$$

Since $x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda} < x_{2\mu}p_{1\mu}$, it follows that $C_{\max}\left(S^{\lambda 2 \mu 1}\right) < C_{\max}\left(S^{\lambda 1 \mu 1}\right)$. Similarly, since $x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda} < x_{1\mu}p_{2\mu}$, it follows $C_{\max}\left(S^{\lambda 1 \mu 2}\right) < C_{\max}\left(S^{\lambda 2 \mu 2}\right)$, so that

$$\begin{aligned}
C_{\max}\left(S^H\right) &= \min\left\{C_{\max}\left(S^{\lambda 1 \mu 2}\right), C_{\max}\left(S^{\lambda 2 \mu 1}\right)\right\} \\
&= C^* + \min\left\{x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda}, x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}\right\}.
\end{aligned}$$

Recall that the proof of Lemma 1 applied to an arbitrary optimal preemptive schedule guarantees that

$$\min\left\{x_{2\mu}p_{1\mu}, x_{1\mu}p_{2\mu}\right\} \le \frac{1}{2}C^*,$$

which immediately yields that

$$\min\left\{x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda}, x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}\right\} \le \min\left\{x_{2\mu}p_{1\mu}, x_{1\mu}p_{2\mu}\right\} \le \frac{1}{2}C^*,$$

and therefore (6) holds.

Part 2. In the remainder of this proof we assume that

$$\max\left\{x_{2\mu}p_{1\mu} - x_{1\lambda}p_{1\lambda}, x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu}\right\} = x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu}.$$

Since $x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu} < x_{1\lambda}p_{2\lambda}$, it follows that $C_{\max}\left(S^{\lambda 2\mu 1}\right) < C_{\max}\left(S^{\lambda 2\mu 2}\right)$, so that

$$\begin{aligned} C_{\max}\left(S^H\right) &= \min\left\{C_{\max}\left(S^{\lambda 1\mu 1}\right), C_{\max}\left(S^{\lambda 1\mu 2}\right), C_{\max}\left(S^{\lambda 2\mu 1}\right)\right\} \\ &= C^* + \min\left\{x_{2\lambda}p_{1\lambda} + x_{2\mu}p_{1\mu}, x_{1\mu}p_{2\mu} - x_{2\lambda}p_{2\lambda}, x_{1\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu}\right\}. \end{aligned}$$

In order to prove the lemma, we need to show that the increment, i.e., the difference $I = C_{\max}\left(S^H\right) - C^*$, does not exceed $\frac{1}{2}C^*$. To simply the forthcoming algebraic transformations, define

$$\begin{aligned} a_\mu &= x_{2\mu}p_{2\mu}, a_\lambda = x_{2\lambda}p_{2\lambda}; && (17) \\ s_\mu &= \frac{p_{2\mu}}{p_{1\mu}}, s_\lambda = \frac{p_{2\lambda}}{p_{1\lambda}}. && (18) \end{aligned}$$

Notice that (13) implies

$$s_\lambda \ge s_\mu \ge 1. \qquad (19)$$

Using (11), rewrite

$$x_{1\mu}p_{2\mu} = s_\mu\left(x_{1\mu}p_{1\mu}\right) = s_\mu\left(C^* - a_\mu\right).$$

Using (12) we obtain $x_{1\lambda}p_{2\lambda} \le s_\lambda\left(C^* - a_\lambda\right)$ and due to (14) we deduce that $x_{1\lambda}p_{2\lambda} = s_\lambda\left(x_{1\lambda}p_{1\lambda}\right) \le s_\lambda a_\mu$.

With this new notation, the increment I can be bounded as

$$I \le \min\left\{\frac{a_\mu}{s_\mu} + \frac{a_\lambda}{s_\lambda}, s_\mu\left(C^* - a_\mu\right) - a_\lambda, s_\lambda\left(C^* - a_\lambda\right) - a_\mu, \left(s_\lambda - 1\right)a_\mu\right\}. \qquad (20)$$

Define

$$\begin{aligned} I_1 &= \min\left\{\frac{a_\mu}{s_\mu} + \frac{a_\lambda}{s_\lambda}, s_\mu\left(C^* - a_\mu\right) - a_\lambda, s_\lambda\left(C^* - a_\lambda\right) - a_\mu\right\}; && (21) \\ I_2 &= \min\left\{s_\mu\left(C^* - a_\mu\right) - a_\lambda, s_\lambda\left(C^* - a_\lambda\right) - a_\mu, \left(s_\lambda - 1\right)a_\mu\right\}, && (22) \end{aligned}$$

i.e., with respect to the four terms in the right-hand side of (20), I_1 is the smallest of the first three terms and I_2 is the smallest of the last three terms. It is clear that

$$I \le \min\left\{I_1, I_2\right\}.$$

For I_1, solve the system of simultaneous equations

$$\begin{array}{c}\frac{a_\mu}{s_\mu}+\frac{a_\lambda}{s_\lambda}=s_\mu\left(C^*-a_\mu\right)-a_\lambda\\ \frac{a_\mu}{s_\mu}+\frac{a_\lambda}{s_\lambda}=s_\lambda\left(C^*-a_\lambda\right)-a_\mu\end{array}$$

to obtain

$$\begin{aligned}a_\lambda &= \frac{s_\mu^2 s_\lambda^2+s_\lambda^2-s_\mu^2 s_\lambda-s_\mu s_\lambda}{s_\mu^2+s_\lambda^2+s_\mu^2 s_\lambda^2-s_\mu s_\lambda-s_\mu-s_\lambda}C^*;\\ a_\mu &= \frac{s_\mu^2 s_\lambda^2+s_\mu^2-s_\mu s_\lambda^2-s_\mu s_\lambda}{s_\mu^2+s_\lambda^2+s_\mu^2 s_\lambda^2-s_\mu s_\lambda-s_\mu-s_\lambda}C^*.\end{aligned}$$

Substituting the found a_λ and a_μ into the right-hand side of (21), we deduce that $I_1 \le F_1 C^*$, where

$$F_1=\frac{s_\mu^2 s_\lambda-s_\mu^2+s_\mu s_\lambda^2-s_\lambda^2}{s_\mu^2 s_\lambda^2+s_\mu^2+s_\lambda^2-s_\mu s_\lambda-s_\mu-s_\lambda}.$$

Similarly, for I_2, solve the system of simultaneous equations

$$\begin{array}{c}\left(s_\lambda-1\right)a_\mu=s_\mu\left(C^*-a_\mu\right)-a_\lambda\\ \left(s_\lambda-1\right)a_\mu=s_\lambda\left(C^*-a_\lambda\right)-a_\mu\end{array}$$

to obtain

$$a_\lambda=\frac{s_\lambda-1}{s_\mu+s_\lambda-2}C^*,\ a_\mu=\frac{s_\mu-1}{s_\mu+s_\lambda-2}C^*.$$

Substituting the found a_λ and a_μ into the right-hand side of (22), we deduce that $I_2 \le F_2 C^*$, where

$$F_2=\frac{\left(s_\mu-1\right)\left(s_\lambda-1\right)}{s_\mu+s_\lambda-2}.$$

To complete the proof of (6), it suffices to demonstrate that

$$\min\left\{F_1,F_2\right\}\le\frac{1}{2}. \tag{23}$$

The proof of (23) is by case analysis. Introduce

$$\alpha=s_\mu s_\lambda,\ \beta=s_\mu+s_\lambda.$$

Notice that

$$\beta^2=\left(s_\mu+s_\lambda\right)^2=s_\mu^2+s_\lambda^2+2s_\mu s_\lambda\ge 4s_\mu s_\lambda=4\alpha. \tag{24}$$

Case 1: $\beta \le 4$.

Notice that $\left(s_\mu-1\right)\left(s_\lambda-1\right)=s_\mu s_\lambda-s_\lambda-s_\mu+1$. Using (24), rewrite

$$F_2=\frac{\left(s_\mu-1\right)\left(s_\lambda-1\right)}{s_\mu+s_\lambda-2}\le\frac{\frac{1}{4}\beta^2-\beta+1}{\beta-2}=\frac{1}{4}\beta-\frac{1}{2}.$$

Since $\beta \le 4$, it follows that $F_2 \le \frac{1}{2}$.

Case 2: $\beta > 4$.

We split our consideration into two subcases.

Case 2.1: $\alpha \leq \beta$.

Rewrite
$$F_2 = \frac{(s_\mu - 1)(s_\lambda - 1)}{s_\mu + s_\lambda - 2} = \frac{\alpha - \beta + 1}{\beta - 2}.$$

To prove that $F_2 \leq \frac{1}{2}$, we need to show
$$2(\alpha - \beta + 1) \leq \beta - 2,$$
which is equivalent
$$3\beta \geq 2\alpha + 4.$$

The latter inequality holds by the conditions of Case 2.1, since
$$2(\beta - \alpha) + (\beta - 4) > 0$$

Case 2.2: $\alpha \geq \beta > 4$.

Rewrite
$$F_1 = \frac{s_\mu^2 s_\lambda - s_\mu^2 + s_\mu s_\lambda^2 - s_\lambda^2}{s_\mu^2 s_\lambda^2 + s_\mu^2 - s_\mu s_\lambda - s_\mu + s_\lambda^2 - s_\lambda} = \frac{\alpha\beta - \left(\beta^2 - 2\alpha\right)}{\alpha^2 - 3\alpha + \beta^2 - \beta}$$

To show that
$$\frac{\alpha\beta - \left(\beta^2 - 2\alpha\right)}{\alpha^2 - 3\alpha + \beta^2 - \beta} \leq \frac{1}{2},$$
we prove the inequality
$$2\alpha\beta - 2\left(\beta^2 - 2\alpha\right) - \alpha^2 + 3\alpha - \beta^2 + \beta \leq 0,$$
or, equivalently,
$$\alpha^2 - 2\alpha\beta - 7\alpha + 3\beta^2 - \beta \geq 0.$$

Since $\alpha^2 + \beta^2 - 2\alpha\beta \geq 0$, we have
$$\alpha^2 - 2\alpha\beta - 7\alpha + 3\beta^2 - \beta = \left(\alpha^2 + \beta^2 - 2\alpha\beta\right) + 2\beta^2 - 7\alpha - \beta \geq 2\beta^2 - 7\alpha - \beta.$$

Using (24) and the conditions of Case 2.2, we deduce that
$$2\beta^2 - 7\alpha - \beta \geq 8\alpha - 7\alpha - \alpha = 0,$$
as required.

Since the proved ratio $C_{\max}(S^*_{(0)})/C_{\max}\left(S^*_p\right) \leq C_{\max}\left(S^H\right)/C_{\max}\left(S^*_p\right) \leq 3/2$ coincides with the power of preemption established for two uniform machines, below we present the tightness example for the latter settings. Suppose that the speed of machine M_1 is 2, while the speed of machine M_2 is 1. There are two jobs, J_1 and J_2 such that $p_1 = p_2 = 3$. It is clear that in an optimal non-preemptive schedule $S^*_{(0)}$ either one of these jobs is assigned to the slow machine or both jobs are assigned to the fast machine, and in any case $C_{\max}(S^*_{(0)}) = 3$. On the other hand, in an optimal preemptive schedule S^*_p both jobs are processed with preemption, so that in the time interval $[0, 1]$ machine M_1 processes part of job J_1 and machine M_2 processes part of job J_2, while in the time interval $[1, 2]$ machine M_1 processes the remaining part of job J_2 and machine M_2 processes the remaining part of job J_1. It follows that $C_{\max}\left(S_p^*\right) = 2$, so that $C_{\max}(S_{(0)}^*)/C_{\max}\left(S_p^*\right) = 3/2$. ■

The results of this section can be summarized as the following statement.

Theorem 1 *For the scheduling problem of minimizing the makespan on two unrelated parallel machines the power of preemption* $\rho_2^{(0)}$ *is equal to* $3/2$.

3.3 More Than Two Machines

Theorem 1 asserts that the power of preemption on two unrelated parallel machines is equal to the power of preemption on two uniform machines. In this subsection we demonstrate that for $m \geq 3$ this phenomenon does not take place.

Lemma 4 *There exists an instance of the scheduling problem of minimizing the makespan on* $m \geq 3$ *unrelated parallel machines such that*

$$\frac{C_{\max}(S_{(0)}^*)}{C_{\max}\left(S_p^*\right)} \geq 1 + \frac{\sqrt{m-2}}{\sqrt{m-1}}. \tag{25}$$

Proof: We build the following instance of the problem on $m \geq 3$ unrelated parallel machines. There are $2\,(m-1)$ jobs, which can be seen as split into two groups, the $A-$jobs and the $B-$jobs. The group of the $A-$jobs consists of jobs $A_1,\ A_2, \ldots, A_{m-1}$, while the group of the $B-$jobs consists of jobs $B_1,\ B_2, \ldots B_{m-1}$. For each job A_j, $1 \leq j \leq m-1$, its processing time on machine M_m is equal to p time units and on machine M_j the processing time is sp time units, where $s > 1$. The processing times on other machines for the $A-$jobs are infinite. Each job B_j, $1 \leq j \leq m-1$, can be processed only on machine M_j for xp time units; the value of $x < 1$ depends on s and will be defined below.

Let us start with making an optimal preemptive schedule of the $A-$jobs alone. For each of these jobs, machine M_m can be seen as the faster machine of speed $s > 1$. Let x_j, $0 < x_j < 1$, be the portion of job A_j processed on machine M_m, so that the total processing time of that job on M_m is $x_j p$; the remaining portion of job J_j is processed on machine M_j for $(1 - x_j)\, sp$ time units. Since the values of processing times of the $A-$jobs are numerically the same, it follows that in an optimal schedule the values x_j are also the same, and we denote that common value by x. A possible structure of an optimal schedule S_A for the processing of the $A-$jobs is as follows. Machine M_m processes job A_j in the time interval $[(j-1)\,xp, jxp]\,, 1 \leq j \leq m-1$. On machine M_j, job A_j is processed in the time intervals outside the interval $[(j-1)\,xp, jxp]\,, 1 \leq j \leq m-1$. The value of x is chosen in such a way that it is guaranteed that all machines complete their jobs simultaneously. The completion time on machine M_m is equal to $(m-1)\,xp$ and the completion time on machine M_j is equal to $xp + (1-x)\,sp$. The equality

$$(m-1)\,xp = xp + (1-x)\,sp$$

is achieved for

$$x = \frac{s}{m-2+s}, \tag{26}$$

so that $C_{\max}(S_A) = \frac{m-1}{m-2+s} sp$. We use the value of x in (26) to define the finite processing time for the $B-$jobs.

To convert schedule S_A into an optimal schedule for all jobs, notice that each job B_j, $1 \le j \le m-1$, has to be processed on machine M_j for xp time units, and in schedule S_A there is an idle interval on each machine M_j of length exactly xp. Thus, the $B-$jobs can be allocated to the corresponding intervals. We obtain an optimal preemptive schedule S_p^* with $C_{\max}(S_p^*) = C_{\max}(S_A)$.

Now we turn to finding an optimal non-preemptive schedule for the instance under consideration. A possible structure of such a schedule is to assign all $A-$jobs to machine M_m and job B_j, $1 \le j \le m-1$, to machine M_j. The makespan of such a schedule is equal to $(m-1)p$. An alternative structure of an optimal non-preemptive schedule assigns at least one of the jobs A_j, $1 \le j \le m-1$, to be processed on machine M_j. Then, the makespan cannot be smaller than $sp + xp$, the completion time of the two jobs A_j and B_j on machine M_j. Thus, we deduce that

$$\begin{aligned} C_{\max}(S_{(0)}^*) &= \min\{(m-1)p, sp + xp\} \\ &= \min\left\{m-1, s + \frac{s}{m-2+s}\right\} p. \end{aligned}$$

Compute the ratio

$$\begin{aligned} \frac{C_{\max}(S_{(0)}^*)}{C_{\max}(S_p^*)} &= \frac{1}{(m-1)s} \min\{(m-1)(m-2+s), s + s(m-2+s)\} \\ &= \min\left\{\frac{m-2+s}{s}, 1 + \frac{s}{m-1}\right\}. \end{aligned}$$

The maximum in the last expression is achieved if the two terms are equal, i.e., if $s = \sqrt{(m-1)(m-2)}$. This implies that the ratio $C_{\max}(S_{(0)}^*)/C_{\max}(S_p^*)$ cannot be smaller than $1 + \frac{\sqrt{m-2}}{\sqrt{m-1}}$, i.e., (25) holds. ■

Lemma 4 implies that for the problem with $m \ge 3$ unrelated parallel machines the power of preemption is at least $1 + \frac{\sqrt{m-2}}{\sqrt{m-1}}$. For each $m \ge 3$, this value is strictly larger than $2 - 1/m$, which is the power of preemption on m uniformly related parallel machines.

4 Quality of Schedules with a Single Preemption

In this section, we turn back to the problem on two unrelated machines and compare the best schedule $S_{(1)}^*$ with at most one preemption to an optimal preemptive schedule S_p^*. We prove that

$$\frac{C_{\max}(S_{(1)}^*)}{C_{\max}(S_p^*)} \le \rho_2^{(1)} = \frac{9}{8}, \tag{27}$$

and this bound is tight. We also show how to find a schedule $S_{(1)}^H$ with a single preemption such that

$$\frac{C_{\max} C_{\max}(S_{(1)}^H)}{C_{\max}(S_p^*)} \le \frac{9}{8}. \tag{28}$$

Let S_p^* be an optimal preemptive schedule in which two jobs J_λ and J_μ satisfy the conditions outlined in Section 3.2, i.e., the equalities (8), (10) and (11), as well as the inequalities (12)–(15); see Figure 1(b). For $i \in \{1,2\}$, let $p_i(N_i)$ denote the total processing time of the jobs of set N_i on machine M_i.

Below, we present an algorithm that moves either job J_λ or job J_μ to be entirely processed on one of the machines, accompanied by an adjustment of the actual processing times of the other preempted job (i.e., job J_μ or job J_λ, respectively).

Algorithm 3

Step 1. Having found schedule S_p^* with jobs J_λ and J_μ processed with preemption, create a schedule $S^{\lambda 1}$ by moving job J_λ to be entirely processed on machine M_1. Increase the actual processing time of job J_μ on machine M_2 from $x_{2\mu}p_{2\mu}$ by $x_{2\lambda}p_{1\lambda}$ and decrease the actual processing time of job J_μ on machine M_1 from $x_{1\mu}p_{1\mu}$ by $x_{2\lambda}\frac{p_{1\lambda}p_{1\mu}}{p_{2\mu}}$. In schedule $S^{\lambda 1}$ job J_μ is processed on machine M_2 in the time interval $[0, x_{2\mu}p_{2\mu} + x_{2\lambda}p_{1\lambda}]$ and on machine M_1 starting from time $x_{2\mu}p_{2\mu} + x_{2\lambda}p_{1\lambda}$.

Step 2. If the inequality

$$p_1(N_1) + p_{1\mu} \le p_{2\lambda} + p_2(N_2) \tag{29}$$

holds, then go to Step 3. Otherwise, transform schedule S_p^* into a schedule $S^{\lambda 2}$ in which job J_λ is processed on machine M_2 without preemption, while job J_μ is preempted in such a way that in schedule $S^{\lambda 2}$ both machines complete simultaneously, i.e., the portion $x'_{1\mu}$ of job J_μ to be processed on machine M_1 in schedule $S^{\lambda 2}$ is given by

$$x'_{1\mu} = x_{1\mu} + x_{1\lambda}\frac{p_{1\lambda} + p_{2\lambda}}{p_{1\mu} + p_{2\mu}}. \tag{30}$$

Compute the makespan of the two found schedules. Output schedule $S_{(1)}^H$ as that of schedules $S^{\lambda 1}$ and $S^{\lambda 2}$, which has the smaller makespan.

Step 3. If $p_{2\lambda} \le p_1(N_1) + p_{1\mu}$, then go to Step 4. Otherwise, transform schedule S_p^* to a schedule $\tilde{S}^{\mu 1}$ by moving job J_μ to be entirely processed on machine M_1. Process job J_λ on machine M_2 in the time interval $[0, p_1(N_1) + p_{1\mu}]$ and on machine M_1 starting from time $p_1(N_1) + p_{1\mu}$. If schedule $\tilde{S}^{\mu 1}$ terminates on machine M_1, output schedule $S_{(1)}^H$ as that of schedules $S^{\lambda 1}$ and $\tilde{S}^{\mu 1}$, which has the smaller makespan; otherwise, go to Step 4.

Step 4. Transform schedule S_p^* to a schedule $S^{\mu 1}$ in which job J_μ is processed on machine M_1 without preemption, while job J_λ is preempted in such a way that in schedule $S^{\mu 1}$ both machines complete simultaneously, i.e., the portion $x'_{1\lambda}$ of job J_λ to be processed on machine M_1 in schedule $S^{\mu 1}$ is given by

$$x'_{1\lambda} = x_{1\lambda} - x_{2\mu}\frac{p_{1\mu} + p_{2\mu}}{p_{1\lambda} + p_{2\lambda}}. \tag{31}$$

Compute the makespan of the two found schedules. Output schedule $S_{(1)}^H$ as that of schedules $S^{\lambda 1}$ and $S^{\mu 1}$, which has the smaller makespan.

The running time of Algorithm 3 is $O(n)$. First, we prove that all found schedules are feasible under the corresponding conditions.

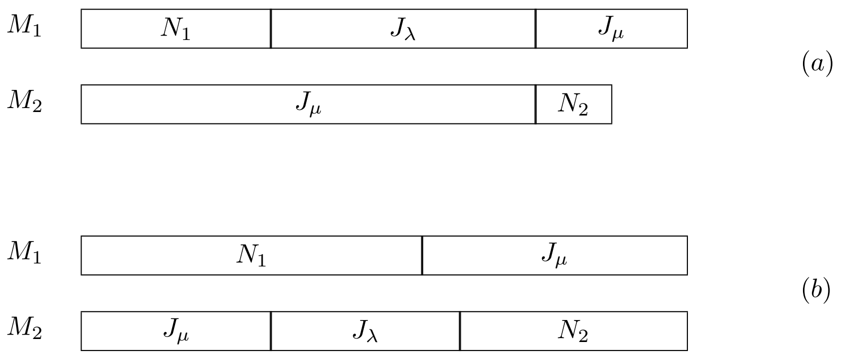

Figure 2: (a) schedule $S^{\lambda 1}$ found in Step 1; (b) schedule $S^{\lambda 2}$ found in Step 2

Lemma 5 *Schedules found by Algorithm 3 are feasible.*

Proof: To verify that each of the four schedules created by Algorithm 3 is feasible, we must demonstrate that on each machine a portion of the preempted job is a number between 0 and 1, and the sum of these portions is equal to 1. Besides, the two portions of the preempted job must not overlap.

In the original schedule S_p^* the actual processing time $x_{2\mu}p_{2\mu}$ of job J_μ on machine M_2 is equal to $p_1(N_1) + x_{1\lambda}p_{1\lambda}$, so that

$$p_1(N_1) = x_{2\mu}p_{2\mu} - x_{1\lambda}p_{1\lambda}. \tag{32}$$

Similarly,

$$p_2(N_2) = x_{1\mu}p_{1\mu} - x_{2\lambda}p_{2\lambda}. \tag{33}$$

We start with schedule $S^{\lambda 1}$ found in Step 1. Compared to schedule S_p^*, in schedule $S^{\lambda 1}$ the total processing of job J_λ on machine M_1 increases by $x_{2\lambda}p_{1\lambda}$ and decreases on machine M_2 by $x_{2\lambda}p_{2\lambda}$. The portion of job J_μ processed on machine M_1 decreases from $x_{1\mu}$ by $x_{2\lambda}\frac{p_{1\lambda}}{p_{2\mu}}$, and that on machine M_2 increases from $x_{2\mu}$ by $x_{2\lambda}\frac{p_{1\lambda}}{p_{2\mu}}$, so that the sum of the portions of job J_μ is schedule $S^{\lambda 1}$ remains equal to 1. It follows from (15) and (13) that

$$x_{2\lambda}\frac{p_{1\lambda}}{p_{2\mu}} = x_{2\lambda}\frac{p_{1\lambda}p_{2\lambda}}{p_{2\mu}p_{2\lambda}} \leq x_{1\mu}\frac{p_{1\lambda}p_{1\mu}}{p_{2\mu}p_{2\lambda}} \leq x_{1\mu},$$

as required. The actual processing time of job J_μ processed on machine M_2 increases by $x_{2\lambda}p_{1\lambda}$, i.e., becomes equal to $p_1(N_1) + p_{1\lambda}$, and the processing of job J_μ on machine M_1 may start exactly when the machine finishes the processing of job J_λ and simultaneously job J_μ ceases to be processed on machine M_2. See Figure 2(a).

In Step 2, the condition $p(N_1) + p_{1\mu} > p_{2\lambda} + p_2(N_2)$ holds, which due to (32) and (33) can be rewritten as $x_{2\mu}p_{2\mu} - x_{1\lambda}p_{1\lambda} + p_{1\mu} > p_{2\lambda} + x_{1\mu}p_{1\mu} - x_{2\lambda}p_{2\lambda}$, which is equivalent to

$$x_{2\mu}(p_{1\mu} + p_{2\mu}) > x_{1\lambda}(p_{1\lambda} + p_{2\lambda}). \tag{34}$$

For schedule $S^{\lambda 2}$, the value $x'_{1\mu}$ is chosen to guarantee that in that schedule both machines complete simultaneously, i.e., that

$$p_1(N_1) + x'_{1\mu}p_{1\mu} = \left(1 - x'_{1\mu}\right)p_{2\mu} + p_{2\lambda} + p_2(N_2).$$

Due to (32) and (33), this implies that

$$\begin{aligned} x'_{1\mu} &= \frac{p_{2\lambda} + p_2(N_2) - p_1(N_1) + p_{2\mu}}{p_{1\mu} + p_{2\mu}} \\ &= \frac{p_{2\lambda} + (x_{1\mu}p_{1\mu} - x_{2\lambda}p_{2\lambda}) - (x_{2\mu}p_{2\mu} - x_{1\lambda}p_{1\lambda}) + p_{2\mu}}{p_{1\mu} + p_{2\mu}}, \end{aligned}$$

which is equivalent to (30). To guarantee the feasibility of schedule $S^{\lambda 2}$ we need to prove that $x'_{1\mu} < 1$. Using (30), consider

$$x'_{1\mu} - 1 = -x_{2\mu} + x_{1\lambda}\frac{p_{1\lambda} + p_{2\lambda}}{p_{1\mu} + p_{2\mu}},$$

which is negative due to (34).

In schedule $S^{\lambda 2}$ job J_μ ceases to be processed on machine M_2 at time $\left(1 - x'_{1\mu}\right)p_{2\mu} = \left(x_{2\mu} - x_{1\lambda}\frac{p_{1\lambda}+p_{2\lambda}}{p_{1\mu}+p_{2\mu}}\right)p_{2\mu}$. To guarantee that there is no overlap in the processing of job J_μ, we need to prove that this time is at most $p_1(N_1)$. It follows from the structure of schedule S_p^* that $x_{2\mu}p_{2\mu} = p_1(N_1) + x_{1\lambda}p_{1\lambda}$. The difference

$$\left(x_{2\mu} - x_{1\lambda}\frac{p_{1\lambda} + p_{2\lambda}}{p_{1\mu} + p_{2\mu}}\right)p_{2\mu} - p_1(N_1) = x_{1\lambda}\left(p_{1\lambda} - \frac{p_{1\lambda} + p_{2\lambda}}{p_{1\mu} + p_{2\mu}}p_{2\mu}\right),$$

is non-positive since

$$p_{1\lambda}(p_{1\mu} + p_{2\mu}) - p_{2\mu}(p_{1\lambda} + p_{2\lambda}) = p_{1\lambda}p_{1\mu} - p_{2\lambda}p_{2\mu} \le 0,$$

where the last inequality is due to (13). See Figure 2(b).

We now address schedule $\tilde{S}^{\mu 1}$ found in Step 3. By the condition of this step, the inequality $p(N_1) + p_{1\mu} < p_{2\lambda}$ holds. By construction, the portion $x'_{2\lambda}$ of job J_λ processed on machine M_2 is equal to $(p_1(N_1) + p_{1\mu})/p_{2\lambda}$, which makes this schedule feasible.

If schedule $\tilde{S}^{\mu 1}$ terminates on machine M_2 then its makespan can be reduced by transferring some of the processing of job J_λ from machine M_2 to machine M_1 until both machines complete simultaneously. The resulting schedule is essentially schedule $S^{\mu 1}$ found in Step 4. This explains why in further analysis we only need to consider schedule $\tilde{S}^{\mu 1}$ that terminates on machine M_1. See Figure 3(a).

Finally, in Step 4, the condition (29) holds, which due to (32) and (33) is equivalent to

$$x_{2\mu}(p_{1\mu} + p_{2\mu}) \le x_{1\lambda}(p_{1\lambda} + p_{2\lambda}). \tag{35}$$

For schedule $S^{\mu 1}$, the value $x'_{1\lambda}$ is chosen to guarantee that in that schedule both machines complete simultaneously, i.e., that

$$p_1(N_1) + p_{1\mu} + x'_{1\lambda}p_{1\lambda} = \left(1 - x'_{1\lambda}\right)p_{2\lambda} + p_2(N_2).$$

Due to (32) and (33), this implies that

$$\begin{aligned} x'_{1\lambda} &= \frac{p_{2\lambda} + p_2(N_2) - p_1(N_1) - p_{1\mu}}{p_{1\lambda} + p_{2\lambda}} \\ &= \frac{p_{2\lambda} + (x_{1\mu}p_{1\mu} - x_{2\lambda}p_{2\lambda}) - (x_{2\mu}p_{2\mu} - x_{1\lambda}p_{1\lambda}) - p_{1\mu}}{p_{1\lambda} + p_{2\lambda}}, \end{aligned}$$

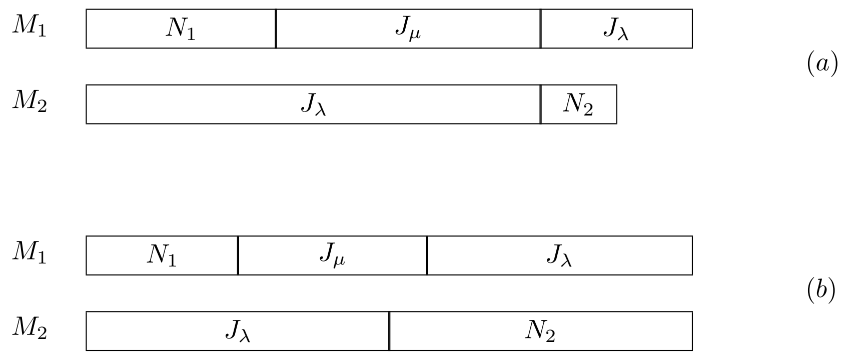

Figure 3: (a) schedule $\tilde{S}^{\mu 1}$ found in Step 3; (b) sschedule $S^{\mu 1}$ found in Step 4

which is equivalent to (31). The feasibility of schedule $S^{\mu 1}$ is guaranteed by the inequality $x'_{1\lambda} > 0$, which follows from (31) and (35).

We come to Step 4 either because the inequality $p(N_1) + p_{1\mu} \geq p_{2\lambda}$ holds or because schedule $\tilde{S}^{\mu 1}$ terminates on machine M_2. In either case, the completion of the portion of job J_λ on machine M_2 occurs earlier than time $p(N_1) + p_{1\mu}$, so that the processing of the preempted job J_λ does not overlap. See Figure 3(b). ■

We now estimate the makespan of each schedule found by Algorithm 3.

Lemma 6 *For schedule $S^{\lambda 1}$ the equality*

$$C_{\max}\left(S^{\lambda 1}\right) = C^* + x_{2\lambda} p_{1\lambda}\left(1 - \frac{p_{1\mu}}{p_{2\mu}}\right) \tag{36}$$

holds, and for each schedule $S \in \left\{S^{\lambda 2}, S^{\mu 1}, \tilde{S}^{\mu 1}\right\}$ the inequality

$$C_{\max}(S) \leq C^* + x_{2\mu} p_{1\mu}\left(1 - \frac{p_{1\lambda}}{p_{2\lambda}}\right) \tag{37}$$

holds, where C^* is defined by (16).

Proof: For schedule $S^{\lambda 1}$ found in Step 1 of Algorithm 3, compared to schedule S_p^* the total processing time on machine M_2 changes by $x_{2\lambda}p_{1\lambda} - x_{2\lambda}p_{2\lambda}$, which is non-positive due to (13). Thus, the makespan of this schedule is determined by the completion time on machine M_1. The net change in total processing time on machine M_1 is equal to $x_{2\lambda}p_{1\lambda}\left(1 - \frac{p_{1\mu}}{p_{2\mu}}\right)$, so that (36) holds.

For schedules $S^{\lambda 2}$ and $S^{\mu 1}$, the proof is based on checking the sign of the same value Δ, introduced below.

Consider schedule $S^{\lambda 2}$ found in Step 2 of Algorithm 3. Compared to schedule S_p^*, in schedule $S^{\lambda 2}$ the total processing time on machine M_1 changes by $x_{1\lambda}p_{1\mu}\frac{p_{1\lambda}+p_{2\lambda}}{p_{1\mu}+p_{2\mu}} - x_{1\lambda}p_{1\lambda}$.

Under the conditions of Step 2 the inequality (34) holds, so that

$$\begin{aligned} x_{1\lambda}\left(p_{1\mu}\frac{p_{1\lambda}+p_{2\lambda}}{p_{1\mu}+p_{2\mu}}-p_{1\lambda}\right) &\le \frac{x_{2\mu}\left(p_{1\mu}+p_{2\mu}\right)}{p_{1\lambda}+p_{2\lambda}}\left(p_{1\mu}\frac{p_{1\lambda}+p_{2\lambda}}{p_{1\mu}+p_{2\mu}}-p_{1\lambda}\right) \\ &= x_{2\mu}p_{1\mu}\left(1-\frac{\left(p_{1\mu}+p_{2\mu}\right)p_{1\lambda}}{\left(p_{1\lambda}+p_{2\lambda}\right)p_{1\mu}}\right). \end{aligned}$$

For schedule $S^{\mu 1}$ found in Step 4 of Algorithm 3, compared to schedule S_p^*, the total processing time on machine M_1 changes by $x_{2\mu}p_{1\mu}-x_{2\mu}p_{1\lambda}\frac{p_{1\mu}+p_{2\mu}}{p_{1\lambda}+p_{2\lambda}}$.

In order to prove (37) for $S\in\left\{S^{\lambda 2},S^{\mu 1}\right\}$, introduce the difference

$$\begin{aligned} \Delta &= x_{2\mu}p_{1\mu}\left(1-\frac{\left(p_{1\mu}+p_{2\mu}\right)p_{1\lambda}}{\left(p_{1\lambda}+p_{2\lambda}\right)p_{1\mu}}\right)-x_{2\mu}p_{1\mu}\left(1-\frac{p_{1\lambda}}{p_{2\lambda}}\right) \\ &= x_{2\mu}p_{1\mu}\left(\frac{p_{1\lambda}}{p_{2\lambda}}-\frac{\left(p_{1\mu}+p_{2\mu}\right)p_{1\lambda}}{\left(p_{1\lambda}+p_{2\lambda}\right)p_{1\mu}}\right) \end{aligned} \tag{38}$$

and show that it is non-positive. For this purpose, it is convenient to use notation (18), so that

$$\Delta = x_{2\mu}p_{1\mu}\left(\frac{1}{s_\lambda}-\frac{1+s_\mu}{1+s_\lambda}\right)\le 0,$$

since

$$\frac{1}{s_\lambda}-\frac{1+s_\mu}{1+s_\lambda}=-\frac{s_\lambda s_\mu-1}{s_\lambda\left(s_\lambda+1\right)}$$

and $s_\lambda\ge s_\mu\ge 1$ due to (13).

Finally, take schedule $\tilde{S}^{\mu 1}$ found in Step 3 of Algorithm 3. Recall that we only need to consider the situation that schedule $\tilde{S}^{\mu 1}$ terminates on machine M_1. By construction, the portion $x'_{2\lambda}$ of job J_λ processed on machine M_2 is equal to

$$x'_{2\lambda}=\frac{p_1\left(N_1\right)+p_{1\mu}}{p_{2\lambda}}.$$

Thus, for schedule $\tilde{S}^{\mu 1}$, compared to schedule S_p^*, the total processing time on machine M_1 changes by $x_{2\mu}p_{1\mu}-x_{1\lambda}p_{1\lambda}+\left(1-x'_{2\lambda}\right)p_{1\lambda}$. Transforming and applying (32), the overall change is given by

$$\begin{aligned} x_{2\mu}p_{1\mu}-x_{1\lambda}p_{1\lambda}+\left(1-x'_{2\lambda}\right)p_{1\lambda} &= x_{2\mu}p_{1\mu}-x_{1\lambda}p_{1\lambda}+\left(1-\frac{\left(p_1\left(N_1\right)+p_{1\mu}\right)}{p_{2\lambda}}\right)p_{1\lambda} \\ &= x_{2\mu}p_{1\mu}+\frac{p_{1\lambda}}{p_{2\lambda}}\left(p_{2\lambda}-\left(p_1\left(N_1\right)+p_{1\mu}\right)-x_{1\lambda}p_{2\lambda}\right) \\ &= x_{2\mu}p_{1\mu}+\frac{p_{1\lambda}}{p_{2\lambda}}\left(x_{2\lambda}p_{2\lambda}-\left(x_{2\mu}p_{2\mu}-x_{1\lambda}p_{1\lambda}+p_{1\mu}\right)\right) \\ &= x_{2\mu}p_{1\mu}+\frac{p_{1\lambda}}{p_{2\lambda}}\left(x_{2\lambda}p_{2\lambda}-x_{2\mu}p_{2\mu}+x_{1\lambda}p_{1\lambda}-p_{1\mu}\right). \end{aligned}$$

In order to prove (37) for $S=\tilde{S}^{\mu 1}$, introduce the difference

$$\Delta = x_{2\mu}p_{1\mu}+\frac{p_{1\lambda}}{p_{2\lambda}}\left(x_{2\lambda}p_{2\lambda}-x_{2\mu}p_{2\mu}+x_{1\lambda}p_{1\lambda}-p_{1\mu}\right)-x_{2\mu}p_{1\mu}\left(1-\frac{p_{1\lambda}}{p_{2\lambda}}\right)$$

and show that it is non-positive. Indeed,

$$\begin{aligned}\Delta &= \frac{p_{1\lambda}}{p_{2\lambda}}\left(x_{2\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu} + x_{1\lambda}p_{1\lambda} - p_{1\mu} + x_{2\mu}p_{1\mu}\right)\\ &= \frac{p_{1\lambda}}{p_{2\lambda}}\left(x_{2\lambda}p_{2\lambda} - x_{2\mu}p_{2\mu} + x_{1\lambda}p_{1\lambda} - x_{1\mu}p_{1\mu}\right) \le 0,\end{aligned}$$

since $x_{1\lambda}p_{1\lambda} + x_{2\lambda}p_{2\lambda} \le C_{\max}\left(S_p^*\right) = x_{1\mu}p_{1\mu} + x_{2\mu}p_{2\mu}$ due to (11) and (12); see Figure 1(b).

This proves the lemma. ■

Lemma 7 *For schedule $S_{(1)}^H$ found by Algorithm 3 the bound (28) holds, and that bound is tight.*

Proof: Due to (36) and (37), we need to prove that

$$\min\left\{x_{2\lambda}p_{1\lambda}\left(1-\frac{p_{1\mu}}{p_{2\mu}}\right), x_{2\mu}p_{1\mu}\left(1-\frac{p_{1\lambda}}{p_{2\lambda}}\right)\right\} \le \frac{1}{8}C^*,$$

where as above C^* is defined by (16).

Using notation (18), rewrite

$$\begin{aligned}x_{2\lambda}p_{1\lambda}\left(1-\frac{p_{1\mu}}{p_{2\mu}}\right) &= x_{2\lambda}p_{1\lambda}\left(1-\frac{1}{s_\mu}\right) = \frac{x_{2\lambda}p_{2\lambda}}{s_\lambda}\left(1-\frac{1}{s_\mu}\right);\\ x_{2\mu}p_{1\mu}\left(1-\frac{p_{1\lambda}}{p_{2\lambda}}\right) &= x_{2\mu}p_{1\mu}\left(1-\frac{1}{s_\lambda}\right) = \frac{x_{2\mu}p_{2\mu}}{s_\mu}\left(1-\frac{1}{s_\lambda}\right).\end{aligned}$$

Further, using notation (17), define

$$\begin{aligned}I_1 &= \frac{C^* - a_\mu}{s_\lambda}\left(1-\frac{1}{s_\mu}\right);\\ I_2 &= \frac{a_\mu}{s_\mu}\left(1-\frac{1}{s_\lambda}\right),\end{aligned}$$

so that (15) and (11) imply

$$\begin{aligned}\frac{x_{2\lambda}p_{2\lambda}}{s_\lambda}\left(1-\frac{1}{s_\mu}\right) &\le \frac{x_{1\mu}p_{1\mu}}{s_\lambda}\left(1-\frac{1}{s_\mu}\right) = \frac{C^*-a_\mu}{s_\lambda}\left(1-\frac{1}{s_\mu}\right) = I_1;\\ x_{2\mu}\frac{p_{2\mu}}{s_\mu}\left(1-\frac{1}{s_\lambda}\right) &= \frac{a_\mu}{s_\mu}\left(1-\frac{1}{s_\lambda}\right) = I_2.\end{aligned}$$

Solve the equation

$$\left(C^* - a_\mu\right)s_\mu\left(1-\frac{1}{s_\mu}\right) = a_\mu s_\lambda\left(1-\frac{1}{s_\lambda}\right)$$

to obtain

$$a_\mu = \frac{s_\mu - 1}{s_\lambda + s_\mu - 2}C^*,$$

so that

$$\min\left\{I_1, I_2\right\} \le \frac{\left(s_\lambda - 1\right)\left(s_\mu - 1\right)}{s_\lambda s_\mu\left(s_\lambda + s_\mu - 2\right)}C^*.$$

It can be verified that the maximum value of the function

$$f(x,y) = \frac{(x-1)(y-1)}{xy(x+y-2)}$$

subject to the constraint $x \geq y \geq 1$ is equal to $\frac{1}{8}$, which is achieved for $x = y = 2$. Applying this result with $x = s_\lambda$ and $y = s_\mu$ we deduce that compared to the value $C^* = C_{\max}\left(S_p^*\right)$ the makespan of schedule S^H does not increase by more than $\frac{1}{8}C^*$, so that (28) holds.

Since the proved ratio $C_{\max}(S_{(1)}^*)/C_{\max}\left(S_p^*\right) \leq C_{\max}(S_{(1)}^H)/C_{\max}\left(S_p^*\right) \leq 9/8$ coincides with the ratio established for two uniform machines, below we present the tightness example for the latter setting. Suppose that the speed of machine M_1 is 2, while the speed of machine M_2 is 1. There are two jobs, J_1 and J_2, such that $p_1 = p_2 = 12$. It is clear that if at most one preemption is allowed then in the corresponding schedule $S_{(1)}^*$ one of these jobs, e.g., job J_1, is fully processed on the fast machine M_1. Thus, in schedule $S_{(1)}^*$ in the time interval $[0,6]$ machine M_1 processes job J_1 and machine M_2 processes part of job J_2, while the remaining part of job J_2 is completed on machine M_1 in the time interval $[6,9]$, so that $C_{\max}(S_{(1)}^*) = 9$. On the other hand, in an optimal preemptive schedule S_p^* both jobs are processed with preemption, so that in the time interval $[0,4]$ machine M_1 processes part of job J_1 and machine M_2 processes part of job J_2, while in the time interval $[4,8]$ machine M_1 processes the remaining part of job J_2 and machine M_2 processes the remaining part of job J_1. It follows that $C_{\max}\left(S_p^*\right) = 8$, so that $C_{\max}(S_{(1)}^*)/C_{\max}\left(S_p^*\right) = 9/8$. ■

The results of this section can be summarized in the following statement.

Theorem 2 *For the scheduling problem of minimizing the makespan on two unrelated machines the upper bound $\rho_2^{(1)}$ on the ratio $C_{\max}(S_{(1)}^*)/C_{\max}\left(S_p^*\right)$ is equal to* $9/8$.

5 Comparison with Two Uniform Machines

It can be observed that most of the results obtained in Sections 3 and 4 for the problem with two unrelated machines coincide with those earlier known for the problem with two uniform machines. Indeed, if an optimal preemptive schedule S_p^* contains two jobs processed with preemption, then Theorem 1 asserts that $\rho_2^{(0)} = 3/2$ for two unrelated machines, and the same holds for two uniform machines as shown in (Woeginger, 2000). Besides, Theorem 2 asserts that $\rho_2^{(1)} = 9/8$ for two unrelated machines, and the same holds for two uniform machines; see (Jiang et al., 2014). All this is rather counter-intuitive, given that the system with unrelated machines is much more general compared to that with uniform machines. Besides, recall that in general for m parallel machines the values of power of preemption are $\rho_m^{(0)} = 2 - 1/m$ if the machines are uniform (see (Woeginger, 2000; Soper and Strusevich, 2014a)) and $\rho_m^{(0)} = 4$ if the machines are unrelated (see (Correa et al., 2012)), with a large gap between these two values.

In our attempt to find a point of difference between the systems with two uniform and two unrelated parallel machines, we turn to the instances for which an optimal preemptive schedule contains exactly one job that is processed with preemption. If the two machines are unrelated, Lemma 2 asserts that the power of preemption $\rho_2^{(0)}$ is equal to $3/2$. The problem with two uniform machines under the same assumption has not been earlier studied from the power of preemption prospective. Below, we show that for the latter problem the power of preemption $\rho_2^{(0)}$ is equal to $4/3$.

Recall that for two uniform machines, finding an optimal schedule with at most one preemption can be done in polynomial time, as proved in (Soper and Strusevich, 2019), however such a schedule does not have to be the global optimal schedule, and a schedule with two preemptions may deliver a smaller makespan. In fact, we know from (Jiang et al., 2014) that the makespan $C_{\max}(S^*_{(1)})$ can be as large as $\rho_2^{(1)} C_{\max}\left(S^*_p\right) = \frac{9}{8} C_{\max}\left(S^*_p\right)$. From now on in this section, we consider instances of the problem for which $C_{\max}(S^*_{(1)}) = C_{\max}\left(S^*_p\right)$.

For two uniform parallel machines, the speed of machine M_1 is $s \geq 1$ and that of machine M_2 is 1. For each job $J_j \in N = \{J_1, J_2, \cdots, J_n\}$, its processing time on the slow machine is known to be equal to p_j, while if it is processed entirely on the fast machine M_1, then its actual processing time is p_j/s. Given a non-empty set of jobs $Q \subseteq N$, denote

$$p\left(Q\right) = \sum_{J_j \in Q} p_j,$$

and for completeness, denote $p\left(\emptyset\right) = 0$. Assume that there is exactly one preemption in an optimal preemptive schedule, i.e., $S^*_{(1)} = S^*_p$, and J_μ is the only preempted job in that schedule. Denote the optimal makespan by C^*, where $C^* = C_{\max}(S^*_{(1)}) = C_{\max}\left(S^*_p\right)$. It follows from (Gonzalez and Sahni, 1978) and (Soper and Strusevich, 2014a) that

$$C^* = \frac{p\left(N\right)}{s+1} \tag{39}$$

and an optimal schedule $S^*_{(1)}$ has the following structure. Both machines have no intermediate idle time and are busy in the time interval $[0, C^*]$. Machine M_1 processes the set of jobs $N_\mu = \{J_j \in N \setminus \{J_\mu\} \,|p_j \geq p_\mu\}$ in the time interval $[0, p\left(N_\mu\right)/s]$, followed by processing part of job J_μ for xp_μ/s time units, where $x = x_{1\mu}$ as defined by (8). Machine M_2 processes the other part of job J_μ in the time interval $[0, \left(1-x\right)p_\mu]$, followed by an arbitrary sequence of the remaining jobs.

The algorithm for transforming schedule S^*_p into a non-preemptive schedule S^H described below is very similar to Algorithm 1.

Algorithm 4

Step 1. Having found schedule S^*_p with job J_μ being the preempted job, create schedule S^1 by keeping all jobs as they are assigned to the machines in schedule S^*_p except job J_μ is entirely processed on machine M_1.

Step 2. Similarly, transform schedule S^*_p to a non-preemptive schedule S^2 by assigning job J_μ to be fully processed on machine M_2.

Step 3. Create a non-preemptive schedule S^3, in which all jobs except job J_μ are processed on machine M_1, while job J_μ alone is processed on machine M_2.

Step 4. Compute the values of makespan for all three found schedules. Output schedule S^H as that of schedules S^1, S^2 and S^3 which has the smallest makespan.

Lemma 8 *For the problem on two uniform machines, let S^H be a schedule found by Algorithm 4 applied to schedule S^*_p with a single preempted job J_μ. Then the bound*

$$\frac{C_{\max}\left(S^H\right)}{C_{\max}\left(S^*_p\right)} \leq \frac{4}{3} \tag{40}$$

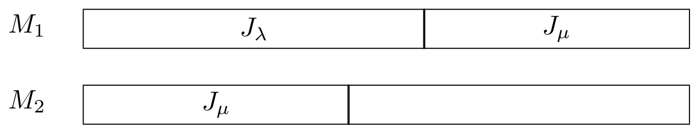

Figure 4: Schedule S_p^*

holds and this bound is tight.

Proof: Let S^1 and S^2 be the two schedules created in Steps 1 and 2, respectively, of Algorithm 1. Suppose that the lemma does not hold. In particular, assume that

$$\min\left\{C_{\max}\left(S^1\right), C_{\max}\left(S^2\right)\right\} > \frac{4}{3}C^*,$$

where C^* is defined by (39). Since $C_{\max}\left(S^1\right) = C^* + (1-x)\,p_\mu/s$ and $C_{\max}\left(S^2\right) = C^* + xp_\mu$, it follows that

$$\min\left\{(1-x)\,p_\mu/s, xp_\mu\right\} > \frac{1}{3}C^*.$$

Due to (39), this implies that

$$p_\mu = (1-x)\,p_\mu + xp_\mu > \frac{(s+1)\,C^*}{3} = p\,(N)\,/3.$$

We can now clarify the structure of schedule S_p^*. Since each non-preempted job processed on machine M_1 is at least as long as job J_μ, we deduce that exactly one non-preempted job, say, job J_λ is processed on M_1 and $p_\lambda \geq p_\mu > p\,(N)\,/3$. See Figure 4.

Consider schedule S^3. If the makespan of that schedule is determined by the completion time of machine M_1, we derive

$$\begin{aligned} C_{\max}\left(S^3\right) &= \frac{1}{s}\left(p\,(N) - p_\mu\right) < \frac{1}{s}\left(p\,(N) - \frac{1}{3}p\,(N)\right) \\ &= \frac{2}{3s}p\,(N) = \frac{2\,(s+1)\,C^*}{3s} \leq \frac{4C^*}{3}. \end{aligned}$$

Thus, in what follows we assume that $C_{\max}\left(S^3\right) = p_\mu$. Then

$$\min\left\{C_{\max}\left(S^1\right), C_{\max}\left(S^3\right)\right\} = C^* + \min\left\{(1-x)\,p_\mu/s, p_\mu - C^*\right\} \tag{41}$$

The structure of schedule S_p^* is such that

$$C^* \geq (1-x)\,p_\mu + xp_\mu/s,$$

so that

$$p_\mu \leq \frac{C^*s}{s+x-sx}.$$

Substituting this inequality into (41), we obtain

$$\min\left\{C_{\max}\left(S^1\right), C_{\max}\left(S^3\right)\right\} \leq C^* + C^*\min\left\{\frac{1-x}{s+x-sx}, \frac{x\,(s-1)}{s+x-sx}\right\}. \tag{42}$$

It can be checked that $\frac{1-x}{s+x-sx}$ is decreasing in x and $\frac{x(s-1)}{s+x-sx}$ does not decrease in x for $s \geq 1$. Solving the equation yields

$$\frac{1-x}{s+x-sx} = \frac{x\,(s-1)}{s+x-sx}$$

with respect to x we obtain $x = 1/s$. Substituting this into (42)

$$\min\left\{C_{\max}\left(S^1\right), C_{\max}\left(S^3\right)\right\} \leq C^* + C^* \frac{s-1}{s^2-s+1}.$$

In turn, for $s \geq 1$ the expression $\frac{s-1}{s^2-s+1}$ reaches its maximum of $1/3$ at $s = 2$. This implies that

$$\min\left\{C_{\max}\left(S^1\right), C_{\max}\left(S^3\right)\right\} \leq \frac{4}{3}C^*,$$

which proves the required bound (40).

To see that the bound (40) is tight, consider the following instance with two uniform machines. The speed of machine M_1 is 2, while the speed of machine M_2 is 1. There are three jobs, J_1, J_2 and J_3, such that $p_1 = p_2 = 4$ and $p_3 = 1$. In the optimal schedule S_p^* one preemption of a longer job is needed and the other longer job, e.g., job J_1, is fully processed on the fast machine M_1. Thus, in schedule $S_{(1)}^*$ in the time interval $[0, 2]$ machine M_1 processes job J_1 and machine M_2 processes part of job J_2, while in the time interval $[2, 3]$ machine M_1 processes the remaining part of job J_2 and machine M_2 processes job J_3, so that $C_{\max}(S_{(1)}^*) = 3$. On the other hand, in schedule S^H, which is also an optimal non-preemptive schedule $S_{(0)}^*$, either the shorter job J_3 is processed on machine M_2, while both longer jobs are assigned to the fast machine M_1 or the fast machine processes the short job and one of the longer jobs, while the remaining longer job alone is processed on M_2. In any case, $C_{\max}\left(S^H\right) = C_{\max}(S_{(0)}^*) = 4$, so that $C_{\max}(S_{(0)}^*)/C_{\max}\left(S_p^*\right) = 4/3$. ■

6 Conclusion

In this paper, we show that the power of preemption on two unrelated machines is 3/2 irrespective of the the number of preemptions in an optimal preemptive schedule S_p^*, i.e., is the same as on two uniform machines, provided that there are two preemptions in S_p^*. If, however, for two uniform machines at most one preemption in S_p^* is needed, then the power of preemption reduces to 4/3. This is the only difference that we have established between two unrelated and two uniform machines, since the quality of a schedule with a single preemption turns out to be the same for both machine environments, with the makespan being at most 9/8 times the makespan of schedule S_p^*.

Given that for systems with m parallel machines the values of the power of preemption differ considerably ($2-1/m$ for uniform machines against 4 for unrelated machines), the fact that in the case $m = 2$ no difference is observed is counter-intuitive. We prove, however, that this only holds for two machines, and for $m \geq 3$ parallel machines the power of preemption on unrelated machines is larger than that on uniformly related machines.

In the case of uniform machines the bound of $2 - 1/m$ on the power of preemption holds for each fixed m, including $m = 2$. For unrelated machines, the bound of 4 is only achieved as a limit, with growing m, and it is not surprising that for $m = 2$ a much smaller value of 3/2 is proved. A similar phenomenon has been established in (Epstein et al., 2017), where for the problem of minimizing the sum of the completion times on m uniform machines the limit value of the power of preemption is shown to be 1.39795, while for $m = 2$ it is 6/5.

Please also notice that the combination of Algorithms 1 and 2 described in this paper can be taken as a linear-time 3/2-approximation algorithm for the problem of finding an optimal non-preemptive schedule on two unrelated machines. Indeed, for a non-preemptive schedule S^H found by Algorithms 1 and 2 we prove that $C_{\max}\left(S^H\right)/C_{\max}\left(S_p^*\right) \leq 3/2$. It remains to be seen whether the bound of 3/2 is tight with respect to an optimal non-preemptive schedule, i.e., whether there exists an instance of the problem such that for schedule S^H the equality $C_{\max}\left(S^H\right) = \frac{3}{2}C_{\max}(S_{(0)}^*)$ holds. On the other hand, it is not impossible that $C_{\max}\left(S^H\right)/C_{\max}(S_{(0)}^*) \leq \rho$ for some $\rho < 3/2$. Previously known (3/2)-approximation algorithms in (Potts, 1985) and in (Shchepin and Vakhania, 2005) use lower bounds on $C_{\max}C_{\max}(S_{(0)}^*)$ which are larger than $C_{\max}\left(S_p^*\right)$ used in this paper.

References

Braun, O., & Schmidt, G. (2003). Parallel processor scheduling with limited number of preemptions. *SIAM Journal on Computing, 32*, 671–680.

Chen, B. (2004). Parallel machine scheduling for early completion. In: J.Y-T. Leung (Ed.) *Handbook of scheduling: Algorithms, models and performance analysis* (pp. 9-175–9-184). London: Chapman & Hall/CRC.

Correa, J.R., Skutella, M., & Verschae, J. (2012). The power of preemption on unrelated machines and applications to scheduling orders. *Mathematics of Operations Research, 37*, 379–398 .

Epstein, L., & Levin, A. (2016) The benefit of preemption for single machine scheduling so as to minimize total weighted completion time. *Operations Research Letters, 44,* 772–774.

Epstein, L., Levin, A., Soper, A.J., & Strusevich, V.A. (2017) Power of preemption for minimizing total completion time on uniform parallel machines. *SIAM Journal on Discrete Mathematics, 31*, 101–123.

Gonzalez, T.F., Lawler, E.L., & Sahni, S. (1990). Optimal preemptive scheduling of two unrelated processors. *ORSA Journal on Computing, 2(3)*, 219–224.

Gonzalez, T.F., & Sahni, S. (1978). Preemptive scheduling of uniform processor systems. *Journal of the Association for Computing Machinery, 25*, 92–101.

Jiang, Y., Weng, Z., & Hu, J. (2014) Algorithms with limited number of preemptions for scheduling on parallel machines. *Journal of Combinatorial Optimization, 27*, 711–723.

Lawler, E.L., & Labetoulle, J. (1978). On preemptive scheduling of unrelated parallel processors by linear programming. *Journal of the Association for Computing Machinery, 25(4)*, 612–619.

Lee, C.-Y., & Strusevich, V.A. (2005) Two-machine shop scheduling with an uncapacitated interstage transporter. *IIE Transactions, 37*, 725–736.

Lenstra, J.K., Shmoys, D.B., & Tardos, E. (1990) Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming, 46*, 259–271.

Lin, J.-H., & Vitter, J.S. (1992). $\varepsilon-$approximations with minimum packing constraint violation. In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC) (pp. 771–782). New York: ACM.

McNaughton, R. (1959). Scheduling with deadlines and loss functions. *Management Science, 6*, 1–12.

Potts, C.N. (1985) Analysis of a linear programming heuristic for scheduling unrelated parallel machines. *Discrete Applied Mathematics, 10*, 155–164.

Rustogi, K., & Strusevich, V.A. (2013). Parallel machine scheduling: Impact of adding extra machines, *Operations Research, 61*, 1243–1257.

Shchepin, E., & Vakhania, N. (2005). An optimal rounding gives a better approximation for scheduling unrelated machines. *Operations Research Letters, 33*, 127–133.

Shchepin, E., & Vakhania, N. (2008) On the geometry, preemptions and complexity of multiprocessor and shop scheduling. *Annals of Operations Research, 159*, 183–213.

Shmoys, D.B., & Tardos, E. (1993). An approximation algorithm for the generalized assignment problem. *Mathematical Programming, 62*, 461–474.

Sitters, R.A. (2017). Approximability of average completion time scheduling on unrelated parallel machines. *Mathematical Programming, 161*, 135–158.

Soper, A.J., & Strusevich, V.A. (2014a). Single parameter analysis of power of preemption on two and three uniform machines. *Discrete Optimization, 12*, 26–46.

Soper, A.J., & Strusevich, V.A. (2014b). Power of preemption on uniform parallel machines, In: 17th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14) / 18th International Workshop on Randomization and Computation (RANDOM'14). *Leibniz International Proceedings in Informatics (LIPIcs) 28*, 392–402.

Soper, A.J., & Strusevich, V.A. (2019). Schedules with a single preemption on uniform parallel machines. *Discrete Applied Mathematics, 261*, 332–343.

Soper, A.J., & Strusevich, V.A. (2021). Parametric analysis of the quality of single preemption schedules on three uniform parallel machines. *Annals of Operations Research, 298,* 469–495.

Shachnai, H., Tamir, T., & Woeginger, G.J. (2005). Minimizing makespan and preemption costs on a system of uniform machines. *Algorithmica, 42*, 309–334.

Woeginger, G.J. (2000). A comment on scheduling on uniform machines under chain-like precedence constraints. *Operations Research Letters, 26*, 107–109.