Annals of Operations Research (2021) 298:469–495
https://doi.org/10.1007/s10479-018-2952-6

S.I.: CODIT2017-COMBINATORIAL OPTIMIZATION

Parametric analysis of the quality of single preemption schedules on three uniform parallel machines

Alan J. Soper[1] **· Vitaly A. Strusevich**[1]

Published online: 2 July 2018

Abstract

For a scheduling problem to minimize the makespan on three uniform parallel machines we present a parametric analysis of the quality of a schedule with at most one preemption compared to the global optimal schedule with any number of preemptions. A tight bound is derived as a function of the relative speeds of the machines, provided that two of the machines have the same speed.

Keywords Scheduling · Uniform parallel machines · Power of preemption

1 Introduction

In parallel machine scheduling, we are given the jobs of set $N = \{J_1, J_2, \ldots, J_n\}$ and m parallel machines $M_1, M_2, \ldots, M_m$. If a job $J_j \in N$ is processed on machine M_i alone, then its processing time is known to be p_{ij}. There are three main types of scheduling systems with 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 this paper, we focus on the problem of minimizing the *makespan,* i.e., the maximum completion time across all m machines. For a schedule S, the makespan is denoted by $C_{\max}(S)$. In a non-preemptive schedule, each job is processed on the machine it is assigned to 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. 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

✉ Vitaly A. Strusevich
V.Strusevich@greenwich.ac.uk

Alan J. Soper
A.J.Soper@greenwich.ac.uk

[1] Department of Mathematical Sciences, University of Greenwich, Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, UK

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^*_{np}.

The number of preemptions in an optimal schedule S^*_p need not exceed $m-1$ in the case of identical machines and $2(m-1)$ in the case of uniform machines, as proved by McNaughton (1959) and Gonzalez and Sahni (1978), respectively.

If any number of preemptions is allowed, then even in the most general settings with unrelated machines an optimal schedule can be found in polynomial time. Finding an optimal non-preemptive schedule is an NP-hard problem, even on two identical parallel machines. See a focused survey by Chen (2004) on parallel machine scheduling with the makespan objective for details and references.

For $m \geq 3$, the problem of finding an optimal preemptive schedule with at most $q \leq m-2$ preemptions on identical parallel machines is NP-hard due to Shchepin and Vakhania (2008). For $m \geq 3$ uniform machines, the problem of finding an optimal schedule with the number of preemptions q such that q is even and $q \leq 2(m-3)$ is shown to be NP-hard by Shachnai et al. (2005). Soper and Strusevich (2018) prove that finding an optimal schedule with at most one preemption on two unrelated parallel machines is NP-hard. On the other hand, it is shown by Soper and Strusevich (2018) that on two uniform machines an optimal schedule with at most one preemption can be found in polynomial time; we reproduce the corresponding algorithm in Sect. 2.2.

Consider an instance of a scheduling problem to minimize the makespan $C_{\max}$ on m parallel machines (identical, uniform or unrelated). For the corresponding problem, we measure the quality of a schedule with a limited number of preemptions by an upper bound $\rho_m^{(q)}$ on the ratio $C_{\max}(S^*_{(q)})/C_{\max}(S^*_p)$. The bound $\rho_m^{(q)}$ determines what can be gained regarding the maximum completion time if instead of at most q preemptions any number of preemptions is allowed. More formally, in order to determine the exact value of $\rho_m^{(q)}$ for a particular problem it has to be demonstrated that the inequality

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

holds for all instances of the problem at hand. To establish tightness of the bound $\rho_m^{(q)}$, instances of the problem have to be exhibited, for which (1) holds as equality.

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, regarding the makespan objective. The value $\rho_m^{(0)}$ is often called the power of preemption. Some of the results on determining $\rho_m^{(0)}$ are reviewed below. Among the results on the power of preemption measured with respect to objective functions other than the makespan, here we mention the recent studies on the single machine problem to minimize the weighted completion time by Epstein and Levin (2016) and on the problem on uniform parallel machines to minimize the total completion time by Epstein et al. (2017).

For the makespan objective, if the machines are identical parallel, then it is known that $\rho_m^{(0)} = 2 - 2/(m+1)$, as independently proved by Braun and Schmidt (2003) and Lee and Strusevich (2005). It is shown in Rustogi and Strusevich (2013) that the value of $\rho_m^{(0)}$ can be reduced for some instances that contain jobs with fairly large processing times.

According to Woeginger (2000), for m uniform parallel machines $\rho_m^{(0)} = 2 - 1/m$. In Soper and Strusevich (2014b), the necessary and sufficient conditions under which the global bound of $2 - 1/m$ is tight are given. If the makespan of an optimal preemptive schedule S^*_p

is equal to the ratio of the total processing time of $r < m$ longest jobs over the total speed of r fastest machines, it is shown in Soper and Strusevich (2014b) that the tight bound on the power of preemption $\rho_m^{(0)}$ is $2 - 1/\min\{r, m - r\}$.

For unrelated parallel machines, a tight bound $\rho_m^{(0)} = 4$ on the power of preemption is established in Correa et al. (2012).

For uniform machines, several papers conduct a parametric analysis of the power of preemption. For example, an analysis of $\rho_2^{(0)}$ with respect to the speed of the faster machine is independently performed in Jiang et al. (2014) and 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, 1 and $s \geq 1$; in other words, either there are two fast machines with speed s and one slow machine with speed 1 or there are two slow machines with speed 1 and one fast machine with speed s. Notice that the difference between the problem in this paper and the one studied in Soper and Strusevich (2014a) is that in the latter paper an optimal preemptive schedule is compared with an optimal schedule with no preemption.

Several publications compare optimal schedules with a limited number of preemptions to optimal preemptive schedules. For identical machines, Braun and Schmidt (2003) prove that $\rho_m^{(q)} = (2m) / (m + q + 1)$, where $0 \leq q \leq m - 1$, and that this bound is tight. For the problem on m uniform parallel machines, Soper and Strusevich (2018) show that $\rho_m^{(1)} = 2 - 2/m$, where $m \geq 3$.

Jiang et al. (2014) perform a parametric analysis of a single preemption for two uniform machines with speeds s' and s'', where $s' \geq s''$, from which it follows that

$$\rho_2^{(1)} = \frac{2\left(s'\right)^2 + s's'' - \left(s''\right)^2}{2\left(s'\right)^2}. \tag{2}$$

This function attains its maximum value of 9/8 when $s' = 2s''$.

In this paper, we focus on the problem on three uniform parallel machines. The bound $\rho_3^{(1)} = \frac{4}{3}$ follows from Soper and Strusevich (2018) and holds as a global bound for all instances, irrespective of relative speeds of the machines. The main purpose of this paper is to perform a parametric analysis of the value $\rho_3^{(1)}$. Although we establish bounds on $\rho_3^{(1)}$ with respect to arbitrary speeds on the machines, as a rule we are only able to demonstrate the tightness of these bounds, provided that there are either two fast machines or two slow machines.

The problem that we address may arise in producing schedules for multi-processor systems. We limit our analysis, at least as far as the tightness of the derived bounds is concerned, to the situations in which the speeds of the processors have two values, 1 (slow) and $s \geq 1$ (fast). Uniform machines with two distinct values of machine speeds are often used in attempts either to improve bounds of approximation algorithms or to demonstrate tightness of those bounds; see Gonzalez et al. (1977) as an example. Besides, having s as a parameter, allows performing a thorough parametric analysis, as is done for the problem of minimizing the makespan on two uniform machines when no preemption is allowed (see Mireault et al. (1997)) or a single preemption is allowed (see Jiang et al. (2014), Soper and Strusevich (2014a)). For these problems, as well as for the problems on three uniform machines in Soper and Strusevich (2014a) and in this paper, the results of the parametric analysis allow the user to perform an appropriate speed scaling in order to achieve a certain guaranteed performance.

The single-parameter analysis performed in Soper and Strusevich (2014a) and in this paper for three uniform machines requires considerable technical efforts, which give an estimate of a possible difficulty of extending a similar analysis to a larger number of machines. The latter can be seen as a goal for further research.

An additional interest in considering the three-machine problem with at most one preemption is due to (i) $m = 3$ is the smallest number of machines for which finding a schedule with at most one preemption is NP-hard, and (ii) for $m = 3$ allowing at most one preemption leads to the smallest loss of quality of the makespan compared to the global preemptive schedule.

The remainder of this paper is organized as follows. Section 2 presents two main tools that are used as subroutines in the subsequent parts of the paper. A tight bound on the quality of a schedule with a single preemption is derived as a function of the relative speeds of the machines in Sects. 3 and 4.

2 Main ingredients

We start this section by introducing a classification of the instances of the problem on m uniform parallel machines, based on the shape of the optimal preemptive schedule S_p^*.

In our analysis of the quality of schedules with a single preemption on three uniform parallel machines we rely on two tools: (i) a polynomial-time algorithm for finding an optimal schedule with a single preemption on two uniform parallel machines, and (ii) a parametrized partition algorithm that splits a set of jobs into two subsets with required properties. These two tools are discussed in the remainder of this section.

2.1 Class distinction

An instance I of the problem with n jobs and m uniform parallel machines is defined by the list $\mathcal{L}_n = (p_1, p_2, \dots, p_n)$ of the processing times of the jobs and the list $\mathcal{M}_m = (s_1, s_2, \dots, s_m)$ of the machine speeds. In what follows, we assume that both lists are non-increasing. In other words, the jobs are numbered in accordance with the LPT rule (6) and the machines are numbered in non-increasing order of their speeds, i.e., $s_1 \geq s_2 \geq \cdots \geq s_m$.

Feasible non-preemptive and preemptive schedules for an instance $I = (\mathcal{L}_n, \mathcal{M}_m)$ are denoted by $S_{np}(\mathcal{L}_n, \mathcal{M}_m)$ or $S_{np}(I)$, and by $S_p(\mathcal{L}_n, \mathcal{M}_m)$ or $S_p(I)$, respectively; the corresponding optimal non-preemptive and preemptive schedules are denoted by $S_{np}^*(\mathcal{L}_n, \mathcal{M}_m)$ or $S_{np}^*(I)$ and by $S_p^*(\mathcal{L}_n, \mathcal{M}_m)$ or $S_p^*(I)$, respectively. The reference to an instance may be omitted if it is clear which instance is being discussed.

For our analysis, we need precise expressions for the makespan of the preemptive schedules. The fastest algorithm for finding an optimal preemptive schedule on uniform parallel machines is due to Gonzalez and Sahni (1978) and requires $O(n + m \log m)$ time.

Given an instance $I = (\mathcal{L}_n, \mathcal{M}_m)$, for each u, $1 \leq u \leq m$, define the total speed of the u fastest machines $S_u = \sum_{i=1}^{u} s_i$. Besides, define the set of the u longest jobs $H_u = \{1, 2, \dots, u\}$.

Define $m' = \min\{n, m-1\}$ and

$$T_u = p(H_u)/S_u,\ 1 \leq u \leq m';\ T_m = p(N)/S_m. \tag{3}$$

It is well-known (see, e.g., Brucker (2007)) that for an optimal preemptive schedule $S_p^*(I)$ the makespan is equal to

$$C_{\max}(S_p^*(I)) = \max\left\{\max\left\{T_u | 1 \leq u \leq m'\right\}, T_m\right\}. \tag{4}$$

Definition 1 An instance $I = (\mathcal{L}_n, \mathcal{M}_m)$ is said to belong to Class r, $1 \leq r \leq m$, if $C_{\max}\left(S_p^*(I)\right) = T_r = \max\{T_u | 1 \leq u \leq m\}$.

In the case of three machines, we will distinguish between three classes

$$C_{\max}(S_p^*(I)) = \begin{cases} \frac{p_1}{s_1}, & \text{for Class 1 instances} \\ \frac{p_1+p_2}{s_1+s_2}, & \text{for Class 2 instances} \\ \frac{p(N)}{s_1+s_2+s_3} & \text{for Class 3 instances.} \end{cases} \tag{5}$$

Given an instance of the problem on three uniform machines, we determine its class by computing the values T_r, $1 \leq r \leq 3$, and the value of r associated with the largest T_r defines the class of the instance. In the case of ties, an instance may belong to more than one class; however, for our purposes instances of Class 3 are harder to handle and we assume without loss of generality that the ties are broken in favour of the smaller r.

Clearly $T_m = p(N)/S_m$ is a lower bound on $C_{\max}(S_p^*(I))$ even when simultaneous execution of jobs is allowed, since it is the average machine load, i.e., the time for processing all jobs, provided all machines are continuously busy.

For an instance I, a schedule $S_{(1)}(I)$ with exactly one preemption is defined by

(i) a job $J_k \in N$ which is processed with preemption on two machines $M_{k'}$ and $M_{k''}$ such that $1 \leq k' < k'' \leq m$; the actual processing times of job J_k on these machines are equal to $x_k/s_{k'}$ and $y_k/s_{k''}$, where $x_k + y_k = p_k$;

(ii) a partition of set $N \setminus \{J_k\}$ into m subsets $N_1, N_2, \ldots, N_m$, where the jobs of set N_i are assigned to be processed on machine M_i, $1 \leq i \leq m$.

Notice that even in an optimal schedule some of the subsets N_i can be empty, since it may be counterproductive to assign jobs to very slow machines. For a particular instance, it might be optimal not to preempt any job, in which case a schedule $S_{(1)}(I)$ is defined by a partition of set N into m subsets $N_1, N_2, \ldots, N_m$. If there is a preempted job J_k in schedule $S_{(1)}(I)$ then the jobs assigned to machines $M_{k'}$ and $M_{k''}$ must be arranged in a such a way that the two portions of job J_k do not overlap.

2.2 Single preemption on two uniform parallel machines

The algorithm that finds an optimal schedule with at most one preemption on two uniform parallel machines is used throughout this paper as a subroutine. This is why below we present it in a generic way, so that the two machines are denoted by M' and M'', while their speeds are s' and s'', respectively, where $s' \geq s''$. Assume that the jobs are numbered in accordance with the LPT rule, i.e., in non-increasing order of their processing times

$$p_1 \geq p_2 \geq \cdots \geq p_n. \tag{6}$$

For a non-empty subset $R \subseteq N$, define

$$p(R) = \sum_{j \in R} p_j,$$

and for completeness define $p(\varnothing) = 0$.

For two uniform parallel machines, instances can only be of either Class 1 or Class 2, i.e., $C_{\max}\left(S_p^*\right) = \max\left\{p_1/s', T_2\right\}$, where

$$T_2 = \frac{p(N)}{s' + s''}. \tag{7}$$

As follows from Gonzalez and Sahni (1978), if there is no restriction on the number of preemptions, then an optimal preemptive schedule S_p^* can be found in $O(n)$ time.

The algorithm below finds an optimal schedule with at most one allowed preemption. It is developed by Soper and Strusevich (2018).

Algorithm Q2Pr1

Step 1. Compute T_2 by (7). If

$$p_1 < s'T_2,$$

go to Step 2; otherwise, output a non-preemptive schedule $S_{(1)}$ with job J_1 on machine M' and the other jobs on machine M'' and stop.

Step 2. Scanning the jobs in the order of their numbering, find job J_k such that

$$\sum_{j=1}^{k-1} p_j < s'T_2, \quad \sum_{j=1}^{k} p_j \geq s'T_2.$$

Step 3. Compute

$$x_k = s'T_2 - \sum_{j=1}^{k-1} p_j, \ \ y_k = p_k - x_k.$$

If

$$\frac{y_k}{s''} > \frac{1}{s'}\sum_{j=1}^{k-1} p_j,$$

go to Step 4; otherwise, output the following schedule $S_{(1)}$ (see Fig. 1a): on M' the jobs $J_1, \ldots, J_{k-1}$ are processed in any order, followed by a part of job J_k for x_k/s' time units; on M'' process a part of job J_k for y_k/s'' time units, followed by an arbitrary sequence of jobs $J_{k+1}, \ldots, J_n$. Stop.

Step 4. For job J_k, compute the values $\tilde{x}_k$ and $\tilde{y}_k$ such that

$$\frac{\tilde{y}_k}{s''} = \frac{1}{s'}\sum_{j=1}^{k-1} p_j, \ \ \tilde{x}_k = p_k - \tilde{y}_k.$$

Output the following schedule $S_{(1)}$ (see Fig. 1b): on M' the jobs $J_1, \ldots, J_{k-1}$ are processed in any order, followed by a part of job J_k for $\tilde{x}_k/s'$ time units; on M'' process a part of job J_k for $\tilde{y}_k/s''$ time units, followed by an arbitrary sequence of jobs $J_{k+1}, \ldots, J_n$. Stop.

Soper and Strusevich (2018) show that Algorithm Q2Pr1 finds an optimal schedule and it requires $O(n)$ time, provided that the LPT numbering of the jobs is available. Obtaining such a numbering can be seen as a preprocessing stage. Thus, the overall time complexity of Algorithm Q2Pr1 is $O(n \log n)$.

On several occasions in this paper, we demonstrate that certain bounds are tight by exhibiting instances of the problem for which inequalities similar to (1) hold as equalities. In the

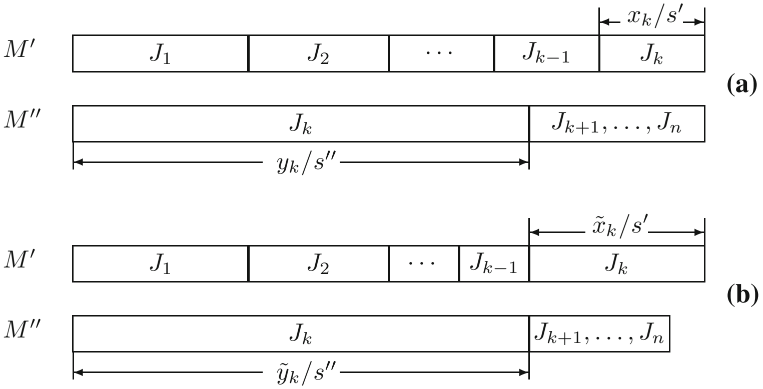

Fig. 1 **a** Schedule $S_{(1)}$ in Step 3; **b** schedule $S_{(1)}$ in Step 4

analysis of such instances, we quite often need the value of the makespan for a schedule found by Algorithm Q2Pr1 applied to two jobs of equal duration.

Notice that for schedule $S_{(1)}$ found in Step 4, we have that $\tilde{y}_k = \frac{s''}{s'}\sum_{j=1}^{k-1} p_j$ and $\tilde{x}_k = p_k - \frac{s''}{s'}\sum_{j=1}^{k-1} p_j$, so that we deduce that $\sum_{j=1}^{k-1} p_j + \tilde{x}_k = \left(\frac{s'-s''}{s'}\right)\sum_{i=1}^{k-1} p_j + p_k$. This implies that

$$C_{\max}\left(S_{(1)}\right) = \frac{1}{s'}\left(\left(\frac{s'-s''}{s'}\right)\sum_{i=1}^{k-1} p_j + p_k\right). \tag{8}$$

Lemma 1 *Let $S_{(1)}$ be a schedule found by Algorithm Q2Pr1 applied to the instance that contains two jobs J_1 and J_2 such that $p_1 = p_2 = w$. Then*

$$C_{\max}\left(S_{(1)}\right) = \frac{2s'-s''}{(s')^2}w. \tag{9}$$

Proof For the instance under consideration, Algorithm Q2Pr1 will output schedule $S_{(1)}$ found in Step 4. Applying (8) with $k = 2$, we obtain

$$C_{\max}\left(S_{(1)}\right) = \frac{1}{s'}\left(\left(\frac{s'-s''}{s'}\right)w + w\right),$$

and (9) follows. □

2.3 Parametric partition algorithm

Below, we present and analyze Procedure Split, which is used as a subroutine in the algorithms presented in Sect. 4. Although the procedure is used in this paper as an auxiliary tool, it is of interest in its own right and is applicable if one needs a partition with some prescribed properties. Such partitions are often required in approximation scheduling algorithms. For example, a simplified version of Procedure Split has been used in Soper and Strusevich (2014a) for the single-parameter analysis of the power of preemption on three uniform machines.

Procedure Split

INPUT: An integer $r \geq 3$, a set of jobs $N = \{J_1, J_2, \ldots, J_n\}$ with normalized processing times, $p(N) = 1$, numbered in the LPT order (at least for the $r-1$ longest jobs) and with $p_j \leq \frac{2}{r+1}$ for all $j \in N$

OUTPUT: A partition of set N into two subsets H' and H'' such that $p(H') \leq \frac{2}{r+1}$ and $p(H'') \leq 1 - \frac{1}{r}$

Step 1. Scanning the jobs in the order of their numbering, determine the job J_u, $u > 1$, such that

$$\sum_{j=1}^{u} p_j \leq \frac{2}{r+1}, \ \sum_{j=1}^{u} p_j + p_{u+1} > \frac{2}{r+1}$$

and define

$$U := \{J_1, \ldots, J_u\}. \tag{10}$$

If

$$p(U) \geq \frac{1}{r} \tag{11}$$

define

$$H' := U, \ H'' = N \backslash U;$$

and go to Step 4. Otherwise go to Step 2.

Step 2. Scanning the jobs in the order of their numbering, consider the sum $\sum_{j=1}^{r-1} p_j$. If

$$\sum_{j=1}^{r-1} p_j \geq 1 - \frac{2}{r+1} \tag{12}$$

then define $U := \{J_1, \ldots, J_{r-1}\}$,

$$H'' := U, \ H' = N \backslash U$$

and go to Step 4. If

$$p_{r-2} + p_{r-1} \geq \frac{1}{r} \tag{13}$$

then define $V := \{J_{r-2}, J_{r-1}\}$ and

$$H' := V, \ H'' = N \backslash V$$

and go to Step 4. Otherwise go to Step 3.

Step 3. Continue scanning the jobs until the first job J_u is reached such that

$$\sum_{j=1}^{u} p_j \geq 1 - \frac{2}{r+1}$$

Define $U := \{J_1, \ldots, J_u\}$

$$H'' := U, \ H' = N \backslash U.$$

Step 4. Stop.

The running time of Procedure Split is $O(n)$.

Lemma 2 *Procedure Split finds a required partition.*

Proof In Step 1,

$$p(U) \leq \frac{2}{r+1},$$

and under the condition (11) we have that

$$p(H') \leq \frac{2}{r+1}, \; p\left(H''\right) \leq 1 - \frac{1}{r},$$

as required.

We come to Step 2 if the inequality $p(U) < \frac{1}{r}$ holds for set U defined in Step 1 by (10). In fact, $p_1 < \frac{1}{r}$, i.e., the duration of each job is less than $\frac{1}{r}$. This implies that

$$\sum_{j=1}^{r-1} p_j \leq \frac{r-1}{r} = 1 - \frac{1}{r}.$$

If (12) holds, then we may set $H'' = \{J_1, \ldots, J_{r-1}\}$ and $H' = N \backslash H''$ since

$$p\left(H'\right) = \sum_{j=r}^{n} p_j \leq \frac{2}{r+1},$$

as required.

If (12) does not hold, then it follows from

$$\sum_{j=1}^{r-1} p_j \leq \frac{r-1}{r+1}$$

that the sum of the two smallest processing times $p_{r-2} + p_{r-1}$ does not exceed $\frac{2}{r+1}$. Thus, if $p_{r-2} + p_{r-1} \geq \frac{1}{r}$, we have

$$p\left(H'\right) = p_{r-2} + p_{r-1} \leq \frac{2}{r+1}, \; p\left(H''\right) = 1 - p\left(H'\right) \leq \frac{r-1}{r},$$

as required.

We come to Step 3 with $p_{r-2} + p_{r-1} < \frac{1}{r}$, which implies that p_{r-1} and the processing time p_j of each job $j \geq r$ is smaller then $\frac{1}{2r}$. Thus, for job J_u, $u \geq r$, found in Step 3, we deduce

$$1 - \frac{2}{r+1} \leq \sum_{j=1}^{u} p_j = \sum_{j=1}^{u-1} p_j + p_u \leq \left(1 - \frac{2}{r+1}\right) + \frac{1}{2r} \leq 1 - \frac{1}{r},$$

where the last inequality holds for all $r \geq 3$.

Thus,

$$p\left(H''\right) \leq 1 - \frac{1}{r}, \; p\left(H'\right) = 1 - p\left(H''\right) \leq \frac{2}{r+1},$$

as required. □

In this paper, Procedure Split is used in Sect. 4 and is applied with $r = 4$.

3 Single parameter analysis for three uniform machines: Classes 1 and 2

The remainder of this paper focuses on the case of three uniform machines, i.e., we consider instances $I = (\mathcal{L}_n, \mathcal{M}_3)$. We perform a class-based analysis, for each form of the optimal preemptive schedule, considering Classes 1 and 2 first and then the more intricate Class 3. Our results give tight bounds for each class as a function of a single parameter equal to the relative speeds of the machines.

In this section, we consider instances of the problem with three uniform machines that belong either to Class 1 or 2; see Definition 1. The Class 3 instances are studied in Sect. 4.

We create heuristic schedules with at most one preemption, that will deliver an upper bound on $\rho_3^{(1)}$ as a function of the speeds of the three machines. Although most of the presented arguments hold for arbitrary speeds, similarly to Soper and Strusevich (2014a) we derive our final conclusion for the models with two fast machines ($s_1 = s_2 = s \geq s_3 = 1$) and for those with two slow machines ($s_1 = s \geq s_2 = s_3 = 1$). This will allow us to deliver the bound on $\rho_3^{(1)}$ as a function of a single parameter s and to prove its tightness.

Consider instances of the form $I = (\mathcal{L}_n, \mathcal{M}_3)$ of the problem with three uniform machines M_1, M_2 and M_3 that belong to either Class 1 or Class 2. As above, it is assumed that the jobs are numbered in accordance with (6), while for the machines $s_1 \geq s_2 \geq s_3$ holds.

For an instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of Class r, $r \in \{1, 2\}$, let $\rho_3^{(1)}$ (Class r) denote the ratio of the makespan that is optimal for schedules with at most one preemption to the optimal makespan for preemptive schedules for instances of that class. We derive an upper bound on $\rho_3^{(1)}$ (Class r) as a function $\Phi\,(s_1, s_2, s_3)$ of the machine speeds.

The algorithm below takes as input an instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of either Class 1 or Class 2, and finds a schedule $S_{(1)}^{H}\,(\mathcal{L}_n, \mathcal{M}_3)$ with at most one preemption.

Algorithm Q3Pr1CL1&2

Step 1. If a given instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ belongs to Class 1, go to Step 2; otherwise, go to Step 3.

Step 2. Find schedule $S_{(1)}^{H}\,(\mathcal{L}_n, \mathcal{M}_3)$ in which job J_1 is assigned to be processed on machine M_1, while the remaining jobs are scheduled by Algorithm Q2Pr1 on machines $M' = M_2$ and $M'' = M_3$. Go to Step 4.

Step 3. Find schedule $S_{(1)}^{H}\,(\mathcal{L}_n, \mathcal{M}_3)$ in which jobs J_1 and J_2 are scheduled by Algorithm Q2Pr1 on machines $M' = M_1$ and $M'' = M_2$, while the remaining jobs are processed non-preemptively in any order on machine M_3. Go to Step 4.

Step 4. Stop.

Algorithm Q3Pr1CL1&2 is analyzed below. Part of the analysis is based on the parametric bound (2) developed in Jiang et al. (2014) for two uniform machines.

Theorem 1 *For an instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of Class r, $1 \leq r \leq 2$, Algorithm Q3Pr1CL1&2 takes $O\,(n \log n)$ time and finds a schedule $S_{(1)}^{H}\,(\mathcal{L}_n, \mathcal{M}_3)$ such that*

$$\frac{C_{\max}\left(S_{(1)}^{*}\,(\mathcal{L}_n, \mathcal{M}_3)\right)}{C_{\max}\left(S_{p}^{*}\,(\mathcal{L}_n, \mathcal{M}_3)\right)} \leq \frac{C_{\max}\left(S_{(1)}^{H}\,(\mathcal{L}_n, \mathcal{M}_3)\right)}{C_{\max}\left(S_{p}^{*}\,(\mathcal{L}_n, \mathcal{M}_3)\right)} \leq \Phi\,(s_1, s_2, s_3)\,, \qquad (14)$$

where

$$\Phi\left(s_1, s_2, s_3\right)=\left\{\begin{array}{l}\frac{2s_2^2+s_2s_3-s_3^2}{2s_2^2},\ \text{if } r=1\\ \frac{2s_1^2+s_1s_2-s_2^2}{2s_1^2},\ \text{if } r=2\end{array}\right\}.$$

Proof First, note that the running time of Algorithm Q3Pr1CL1&2 is determined by the running time of Algorithm Q2Pr1, and is therefore $O\left(n\log n\right)$.

For a given instance of Class r, $r\in\{1,2\}$, define the lists $\mathcal{L}_r$ and $\mathcal{M}_r$, that contain the r longest jobs and the r fastest machines, respectively. Also define the lists $\mathcal{L}_r'$ and $\mathcal{M}_r'$ obtained from the lists $\mathcal{L}_n$ and $\mathcal{M}_3$ by the removal of the r longest jobs and the r fastest machines, respectively. In other words, $\mathcal{L}_r'=\left(p_{r+1},\ldots,p_n\right)$ and $\mathcal{M}_r'=\left(s_{r+1},\ldots,s_3\right)$.

By definition of a Class r instance, we have that

$$C_{\max}\left(S_p^*\left(\mathcal{L}_r,\mathcal{M}_r\right)\right)\geq C_{\max}\left(S_p^*\left(\mathcal{L}_r^{'},\mathcal{M}_r^{'}\right)\right).$$

For schedule $S_{(1)}^H\left(\mathcal{L}_n,\mathcal{M}_3\right)$ found by Algorithm Q3Pr1CL1&2, we have that

$$C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_n,\mathcal{M}_3\right)\right)=\max\left\{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r,\mathcal{M}_r\right)\right),C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r^{'},\mathcal{M}_r^{'}\right)\right)\right\}.$$

It follows that

$$\begin{aligned}\frac{C_{\max}\left(S_{(1)}^*\left(\mathcal{L}_n,\mathcal{M}_3\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_n,\mathcal{M}_3\right)\right)}&=\frac{C_{\max}\left(S_{(1)}^*\left(\mathcal{L}_n,\mathcal{M}_3\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_r,\mathcal{M}_r\right)\right)}\\&\leq\frac{\max\left\{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r,\mathcal{M}_r\right)\right),C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r^{'},\mathcal{M}_r^{'}\right)\right)\right\}}{C_{\max}\left(S_p^*\left(\mathcal{L}_r,\mathcal{M}_r\right)\right)}\\&\leq\max\left\{\frac{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r,\mathcal{M}_r\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_r,\mathcal{M}_r\right)\right)},\frac{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_r^{'},\mathcal{M}_r^{'}\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_r^{'},\mathcal{M}_r^{'}\right)\right)}\right\}.\end{aligned}\tag{15}$$

For instances of Class 1, Step 2 of Algorithm Q3Pr1CL1&2 guarantees that

$$C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_1,\mathcal{M}_1\right)\right)=p_1/s_1,$$

i.e.,

$$\frac{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_1,\mathcal{M}_1\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_n,\mathcal{M}_3\right)\right)}=1.$$

Since the remaining jobs are scheduled on machines M_2 and M_3 optimally with at most one preemption, we have that $C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)=C_{\max}\left(S_{(1)}^*\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)$, so that

$$\frac{C_{\max}\left(S_{(1)}^H\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)}=\frac{C_{\max}\left(S_{(1)}^*\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)}{C_{\max}\left(S_p^*\left(\mathcal{L}_1^{'},\mathcal{M}_1^{'}\right)\right)}\leq\frac{2s_2^2+s_2s_3-s_3^2}{2s_2^2}$$

where the last inequality follows from (2) applied to the machines with speeds $s'=s_2$ and $s''=s_3$. Thus, (14) holds for $r=1$, since $\left(2s_2^2+s_2s_3-s_3^2\right)/2s_2^2\geq 1$.

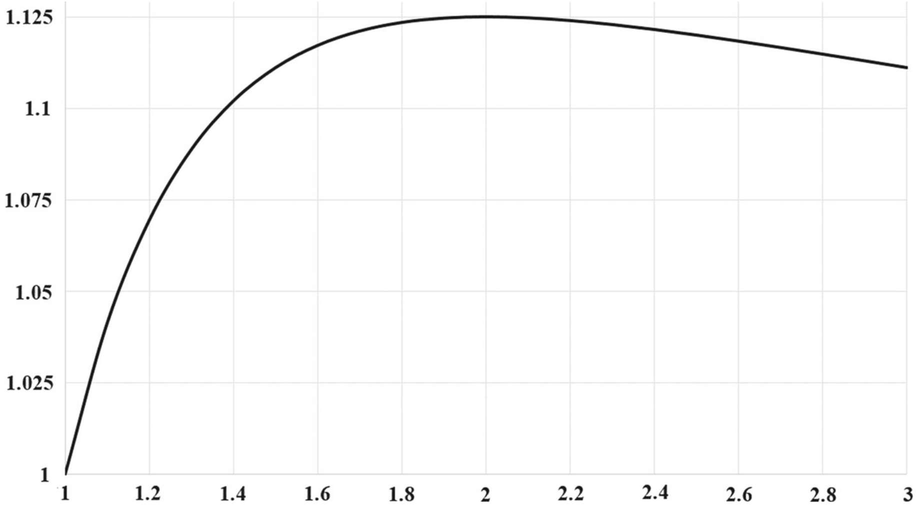

Fig. 2 $\Phi\left(s_1, s_2, s_3\right)$ as a function of speed ratio

Applying (15) to the instances of Class 2, notice that

$$\frac{C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_2^{'}, \mathcal{M}_2^{'}\right)\right)}{C_{\max}\left(S_p^{*}\left(\mathcal{L}_2^{'}, \mathcal{M}_2^{'}\right)\right)} = 1,$$

since the machine environment $\mathcal{M}_2^{'}$ consists only of machine M_3, so that $C_{\max}\left(S_p^{*}\left(\mathcal{L}_2^{'}, \mathcal{M}_2^{'}\right)\right)$ $= C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_2^{'}, \mathcal{M}_2^{'}\right)\right)$.

Step 3 of Algorithm Q3Pr1CL1&2 guarantees that

$$\frac{C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)}{C_{\max}\left(S_p^{*}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)} = \frac{C_{\max}\left(S_{(1)}^{*}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)}{C_{\max}\left(S_p^{*}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)},$$

and due to (2) applied with $s' = s_1$ and $s'' = s_2$, we conclude that

$$\frac{C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)}{C_{\max}\left(S_p^{*}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right)} \leq \frac{2s_1^2 + s_1 s_2 - s_2^2}{2s_1^2}.$$

Thus, (14) holds for $r = 2$. □

Figure 2 illustrates $\Phi\left(s_1, s_2, s_3\right)$ as a function of the speed ratio: s_2/s_3 for $r = 1$ or s_1/s_2 for $r = 2$. The maximum of the function is $\frac{9}{8}$, attained for the ratio equal to 2. That value provides a global numerical bound on $\rho_3^{(1)}$ for the instances of Classes 1 and 2.

Lemma 3 *Bound (14) is tight for instances of Class 2.*

Proof Take a Class 2 instance given by a set of three jobs with $\mathcal{L}_3 = \left(p, p, p\frac{2s_3}{s_1+s_2}\right)$ and $\mathcal{M}_3 = \left(s_1, s_2, s_3\right)$. It follows that $C_{\max}\left(S_p^{*}\left(\mathcal{L}_3, \mathcal{M}_3\right)\right) = C_{\max}\left(S_p^{*}\left(\mathcal{L}_2, \mathcal{M}_2\right)\right) =$

$2p/(s_1+s_2)$. Let $S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)$ be the schedule found by Algorithm Q3Pr1CL1&2. The algorithm schedules the identical jobs J_1 and J_2 on machines M_1 and M_2 and the remaining job on machine M_3. It follows that schedule $S_{(1)}^{H}(\mathcal{L}_2,\mathcal{M}_2)$ is in fact an optimal schedule $S_{(1)}^{*}(\mathcal{L}_2,\mathcal{M}_2)$ for processing the two longest jobs on the two fastest machines with at most one preemption. We deduce from (9) applied with $s'=s_1$, $s''=s_2$ and $w=p$ that

$$C_{\max}\left(S_{(1)}^{*}(\mathcal{L}_2,\mathcal{M}_2)\right)=\frac{2s_1-s_2}{s_1^2}p.$$

Furthermore,

$$C_{\max}\left(S_{(1)}^{*}(\mathcal{L}_3,\mathcal{M}_3)\right)=\max\left\{\frac{2s_1-s_2}{s_1^2}p,\ \frac{2p}{s_1+s_2}\right\}\geq\frac{2s_1-s_2}{s_1^2}p,$$

and therefore $\rho_3^{(1)}$ (Class 2) $=\frac{(2s_1-s_2)(s_1+s_2)}{2s_1^2}=\frac{2s_1^2+s_1s_2-s_2^2}{2s_1^2}=\Phi(s_1,s_2,s_3)$, as required. □

Lemma 4 *Bound (14) is tight for instances of Class 1 with two distinct values of speed.*

Proof Consider a Class 1 instance given by a set of three jobs such that $\mathcal{L}_3=\left(p,p\frac{s_2+s_3}{2s_1},p\frac{s_2+s_3}{2s_1}\right)$ and $\mathcal{M}_3=(s_1,s_2,s_3)$. It follows that $C_{\max}\left(S_p^{*}(\mathcal{L}_3,\mathcal{M}_3)\right)=C_{\max}\left(S_p^{*}(\mathcal{L}_1,\mathcal{M}_1)\right)=p/s_1$. Algorithm Q3Pr1CL1&2 creates the heuristic schedule $S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)$ by scheduling job J_1 on machine M_1 and the two remaining identical jobs on machines M_2 and M_3 in accordance with Algorithm Q2Pr1. It follows from (9) applied with $s'=s_2$, $s''=s_3$ and $w=p\frac{s_2+s_3}{2s_1}$ that

$$C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_1',\mathcal{M}_1'\right)\right)=\frac{(2s_2-s_3)}{s_2^2}\frac{(s_2+s_3)}{2s_1}p=\frac{2s_2^2+s_2s_3-s_3^2}{2s_1s_2^2}p>\frac{p}{s_1}.$$

Thus, $C_{\max}\left(S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)\right)=C_{\max}\left(S_{(1)}^{H}\left(\mathcal{L}_1',\mathcal{M}_1'\right)\right)$, and therefore

$$\frac{C_{\max}\left(S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)\right)}{C_{\max}\left(S_p^{*}(\mathcal{L}_3,\mathcal{M}_3)\right)}=\frac{2s_2^2+s_2s_3-s_3^2}{2s_2^2}.$$

Bound (14) is tight for the instance under consideration if schedule $S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)$ is in fact an optimal schedule with at most one preemption. However, unlike for the Class 2 instances discussed in Lemma 3, this does not happen for all combinations of speeds. To demonstrate this, assume that $s_1=s$, $s_2=\frac{s}{2}$ and $s_3=1$. It follows that

$$C_{\max}\left(S_{(1)}^{H}(\mathcal{L}_3,\mathcal{M}_3)\right)=p(s+2)\frac{s-1}{s^3}.$$

However, for such an instance an optimal schedule with exactly one preemption can be found by splitting job J_3 between machines M_1 and M_3. In this schedule $S_{(1)}^{*}(\mathcal{L}_3,\mathcal{M}_3)$, jobs J_1 and J_2 are processed without preemption on machines M_1 and M_2, respectively. Job J_3 is processed in the time interval $[0,p/s]$ on machine M_3 and then starting from time p/s on machine M_1, so that the makespan of this schedule becomes

$$C_{\max}\left(S_{(1)}^{*}(\mathcal{L}_3,\mathcal{M}_3)\right)=\frac{p}{s}+\frac{p}{s}\left(\frac{\frac{s}{2}+1}{2s}-\frac{1}{s}\right)=p\frac{5s-2}{4s^2}.$$

It is easily seen that $\frac{5s-2}{4s^2} < (s+2)\frac{s-1}{s^3}$ for $2 < s < 4$.

Still, bound (14) is tight if there are two different values of speeds only, i.e., either $s_1 = s > s_2 = s_3 = 1$ (two slow machines) or $s_1 = s_2 = s > s_3 = 1$ (two fast machines).

In the former case, the smallest makespan of p/s is achieved for a non-preemptive schedule, and that cannot be improved by allowing any number of preemptions, so that $\rho_3^{(1)}$ (Class 1) becomes equal to 1, which complies with (14).

In the latter case, schedule $S_{(1)}^{H}(\mathcal{L}_3, \mathcal{M}_3)$ is in fact an optimal schedule with a single preemption, which implies that $\rho_3^{(1)}$ (Class 1) $= \frac{2s_2^2+s_2s_3-s_3^2}{2s_2^2} = \Phi(s_1, s_2, s_3)$, as required. Note that for two slow machines $s_2 = s_3$, the bound $\rho_3^{(1)}$ (Class 1) becomes equal to 1.

Notice that the example above demonstrates that for the Class 2 instances the bound (14) is tight for arbitrary machines speeds, in particular for instances with two fast and two slow machines. □

Since our main intention is to perform a single parameter analysis of the quality of schedules with at most one single preemption on three uniform parallel machines, below we formulate the results of this section in terms of the instances with two fast machines and with two slow machines.

Corollary 1 *For the instances of the problem with two fast machines ($s_1 = s_2 = s > 1 = s_3$) the following bounds*

$$\rho_3^{(1)}\,(\text{Class 1}) \le \frac{2s^2+s-1}{2s^2};$$
$$\rho_3^{(1)}\,(\text{Class 2}) = 1$$

hold and are tight. For the instances of the problem with two slow machines ($s_1 = s > 1 = s_2 = s_3$) the following bounds

$$\rho_3^{(1)}\,(\text{Class 1}) = 1;$$
$$\rho_3^{(1)}\,(\text{Class 2}) \le \frac{2s^2+s-1}{2s^2}$$

hold and are tight.

The fact that the bound $\rho_3^{(1)}$ is equal to 1 for some classes of instances complies with the fact that for those instances the optimal preemptive schedule requires at most one preemption.

4 Single parameter analysis for Class 3 instances on three uniform machines

It remains to consider instances of Class 3, i.e., instances for which $C_{\max}\left(S_p^*(I)\right) = T_3$. Our analysis is presented in two subsections, starting with instances for which the ratio between the fastest and the slowest speeds is limited to 2 and followed by instances with larger relative speeds.

4.1 Limited relative machine speeds $s_2 \le s_1 \le 2s_3$

Temporarily assume that the feasibility of schedules with a single preemption is ignored, i.e., simultaneous processing of pieces of the same (preempted) job on different machines is allowed. Let C be the makespan of an optimal schedule of this type. Comparing C with the optimal makespan $C_{\max}\left(S_p^*(I)\right) = T_3$, define an upper bound $\Phi(s_1, s_2, s_3)$ such that the inequality $C \le \Phi(s_1, s_2, s_3)\,T_3$ holds across all instances of Class 3. Obtaining the required heuristic schedule simplifies to an appropriate split of the jobs into two sets, and an allocation of one of these sets to be processed on a single machine, and placing the other on the remaining pair of machines, using the preemption to share the load appropriately between the pair. The splitting of the jobs is performed by Procedure Split defined and analyzed in Sect. 2.3.

During the analysis of the Class 3 instances we will assume that the jobs of any instance have normalized processing times, $p(N) = 1$, so that $T_3 = \frac{1}{s_1+s_2+s_3}$. Define

$$\Phi(s_1, s_2, s_3)\,T_3 = \max\left\{\frac{2}{5s_1}, \frac{3}{4(s_1+s_2)}, \frac{1}{s_1+2s_3}, \frac{1}{2s_2+s_3}\right\}. \tag{16}$$

Algorithm OneSplit below partitions the set of jobs into two subsets N_u and $N_{v,w}$ such that for some triple (u, v, w) of the machine indices 1, 2 and 3 the inequalities

$$p(N_u) \le \Phi T_3 s_u; \tag{17}$$

$$p\left(N_{v,w}\right) \le \Phi T_3 (s_v + s_w). \tag{18}$$

hold, provided that $s_2 \le 2s_3$. In what follows, without loss of generality we assume that $v < w$, i.e., (u, v, w) is one of the triples $(1, 2, 3)$, $(2, 1, 3)$ or $(3, 1, 2)$.

Algorithm OneSplit

INPUT: An instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of Class 3, with normalized processing times, $p(N) = 1$, numbered in the LPT order and a set of speeds satisfying $s_3 \le s_2 \le s_1 \le 2s_3$

OUTPUT: Subsets N_u and $N_{v,w}$ that satisfy (17) and (18) for some triple (u, v, w)

```
Step 1. Compute $T_3$ and $\Phi T_3$. If $p_1 > 2/5$ then output $N_1 = \{J_1\}$ and $N_{2,3} = N \setminus \{J_1\}$ and
        stop. Otherwise, run Procedure Split with $r = 4$, to produce two sets of jobs $H'$
        and $H''$ satisfying $p(H') \le \frac{2}{5}$ and $p(H'') \le \frac{3}{4}$.
Step 2. Identify a triple $(u, v, w) \in \{(1,2,3), (2,1,3), (3,1,2)\}$ such that $p(H') \in \left[1 - \Phi T_3 (s_v + s_w), \Phi T_3 s_u\right]$
        and output $N_u = H'$, $N_{v,w} = H''$.
```

Lemma 5 *Algorithm OneSplit splits the jobs of set N into two subsets N_u and $N_{v,w}$ such that* (17) *and* (18) *hold for some triple* (u, v, w).

Proof For a partition N_1 and $N_{2,3}$ found in Step 1, we deduce that

$$p_1 \le T_3 s_1 \le \Phi T_3 s_1,$$

since $\Phi \ge 1$ and our instance is of Class 3, so that $\frac{p_1}{s_1} \le T_3$. It follows from $p(N_1) = p_1 > 2/5$ that $p\left(N_{2,3}\right) < \frac{3}{5}$. Using the fourth term in the right-hand side of (16), we derive

$$\Phi T_3 (s_2 + s_3) \ge \frac{s_2 + s_3}{2s_2 + s_3}.$$

The right-hand side of the above inequality decreases in s_2, so that $s_2 \le 2s_3$ implies $\Phi T_3 (s_2 + s_3) \ge \frac{3}{5}$, as required by (18). Thus, the lemma holds for $(u, v, w) = (1, 2, 3)$.

We carry out Step 2 if $p_1 \le 2/5$, so that the conditions for running Procedure Split are satisfied and the two sets of jobs H' and H'' will be found with $p(H') \le \frac{2}{5}$ and $p(H'') \le \frac{3}{4}$. This additionally bounds $p(H')$ from below, since $p(H') = 1 - p(H'') \ge \frac{1}{4}$.

If for some triple (u, v, w) we have that $p\left(H'\right) \in [1 - \Phi T_3\left(s_u + s_v\right), \Phi T_3 s_w]$ then the lemma holds for such a triple. Thus, we need to prove that such a triple always exists.

Consider the three intervals $[1 - \Phi T_3\left(s_1 + s_2\right), \Phi T_3 s_3]$, $[1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$, $[1 - \Phi T_3\left(s_2 + s_3\right), \Phi T_3 s_1]$. It is straightforward to check that the right endpoint of each interval is no less than its left endpoint, since $\Phi \geq 1$, so that the intervals are well defined. We can consider the intervals in the natural order: $[1 - \Phi T_3\left(s_1 + s_2\right), \Phi T_3 s_3]$, $[1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$, $[1 - \Phi T_3\left(s_2 + s_3\right), \Phi T_3 s_1]$, since in this sequence the right endpoints of the intervals are nondecreasing, as are their left endpoints.

We show that the value $p\left(H'\right)$ such that $\frac{1}{4} \leq p(H') \leq \frac{2}{5}$ belongs to at least one of these intervals.

Using each of the components of Φ in the order of their appearance in the right-hand side of (16), we deduce that

- the right endpoint of $[1 - \Phi T_3\left(s_2 + s_3\right), \Phi T_3 s_1]$ is no less than 2/5 since $\Phi T_3 s_1 \geq \frac{2}{5}$;
- the left endpoint of $[1 - \Phi T_3\left(s_1 + s_2\right), \Phi T_3 s_3]$ is at most 1/4 since $\Phi T_3\left(s_1 + s_2\right) \geq \frac{3}{4}$;
- the intervals $[1 - \Phi T_3\left(s_1 + s_2\right), \Phi T_3 s_3]$ and $[1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$ overlap since $\Phi T_3 s_3 \geq 1 - \Phi T_3\left(s_1 + s_3\right)$ due to $\Phi T_3\left(s_1 + 2s_3\right) \geq 1$;
- the intervals $[1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$ and $[1 - \Phi T_3\left(s_2 + s_3\right), \Phi T_3 s_1]$ overlap since $\Phi T_3 s_2 \geq 1 - \Phi T_3\left(s_2 + s_3\right)$ due to $\Phi T_3\left(2s_2 + s_3\right) \geq 1$.

Suppose that $p\left(H'\right)$ belongs to neither the interval $[1 - \Phi T_3\left(s_1 + s_2\right), \Phi T_3 s_3]$ nor to $[1 - \Phi T_3\left(s_2 + s_3\right), \Phi T_3 s_1]$; otherwise, the lemma holds either for $(u, v, w) = (3, 1, 2)$ or for $(u, v, w) = (1, 2, 3)$, respectively. This is only possible if the two intervals do not intersect and

$$\Phi T_3 s_3 < p\left(H'\right) < 1 - \Phi T_3\left(s_2 + s_3\right).$$

However, since each of these two intervals intersects with $[1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$, we obtain that

$$1 - \Phi T_3\left(s_1 + s_3\right) \leq \Phi T_3 s_3 < 1 - \Phi T_3\left(s_2 + s_3\right) \leq \Phi T_3 s_2,$$

so that $p\left(H'\right) \in [1 - \Phi T_3\left(s_1 + s_3\right), \Phi T_3 s_2]$, and the lemma holds for $(u, v, w) = (2, 1, 3)$. □

Algorithm OneSplit is used as a splitting procedure in the following algorithm for finding a schedule with a single preemption.

Algorithm Q3Pr1CL3Lim

INPUT: An instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of Class 3, with its jobs having normalized processing times, $p\left(N\right) = 1$, numbered in the LPT order

OUTPUT: A schedule $S_{(1)}^{H}$ with at most one preemption

Step 1. If $p_1 \geq p_2 > \Phi T_3 s_2$ then define $(u, v, w) = (3, 1, 2)$, $N_{v,w} = \{J_1, J_2\}$, $N_u = N \setminus \{J_1, J_2\}$ and go to Step 4; otherwise go to Step 2.

Step 2. If $p_2 > \Phi T_3 s_3$ then define $(u, v, w) = (2, 1, 3)$, $N_u = \{J_2\}$, $N_{v,w} = N \setminus \{J_2\}$ and go to Step 4; otherwise go to Step 3.

Step 3. Run Algorithm OneSplit that outputs sets N_u and $N_{v,w}$ for a triple of indices (u, v, w).

Step 4. Run Algorithm Q2Pr1 to find a schedule $S_{v,w}$ for processing the jobs of set $N_{v,w}$ on machines $M' = M_v$ and $M'' = M_w$, provided that at most one preemption is allowed. Create the following schedule $S_{(1)}^{H}$: schedule the jobs of set N_u nonpreemptively on machine M_u and process the remaining jobs in accordance with schedule $S_{v,w}$. Stop.

In the analysis below, the following notation is used. For schedule $S_{(1)}^H$ found by Algorithm Q3Pr1CL3Lim that is associated with a triple (u, v, w) of indices, let $C_u = p(N_u)/s_u$ denote the makespan of processing the jobs of set N_u non-preemptively on machine M_u, while $C_{v,w}$ denotes the makespan of processing the jobs of set $N_{v,w}$ with a single preemption on machines M_v and M_w. Thus, $C_{\max}\left(S_{(1)}^H\right) = \max\left\{C_u, C_{v,w}\right\}$.

Lemma 6 *For schedule $S_{(1)}^H$ associated with the sets $N_{1,2}$ and N_3 found in Step 1 of Algorithm Q3Pr1CL3Lim the following bound*

$$\frac{C_{\max}\left(S_{(1)}^*\right)}{C_{\max}\left(S_p^*\right)} \le \frac{C_{\max}\left(S_{(1)}^H\right)}{T_3} \le \max\left\{\Phi, \frac{2s_1^2 + s_1 s_2 - s_2^2}{2s_1^2}\right\}$$

holds, where Φ is defined by (16).

Proof Since in Step 1, we have that $p_1 \ge p_2 \ge \Phi T_3 s_2$, it follows from (16) that

$$p\left(N\setminus\{J_1, J_2\}\right) = 1 - p_1 - p_2 \le 1 - 2\Phi T_3 s_2 \le 1 - \frac{2s_2}{2s_2 + s_3} = \frac{s_3}{2s_2 + s_3},$$

which implies that $\frac{p(N\setminus\{J_1,J_2\})}{s_3} \le \frac{1}{2s_2+s_3} \le \Phi T_3$. Therefore, $C_3 \le \Phi T_3$ as required. According to (8), the processing of jobs J_1 and J_2 will complete at time

$$C_{1,2} = \frac{1}{s_1}\max\left\{p_1, \frac{s_1 - s_2}{s_1}p_1 + p_2\right\}.$$

If $C_{1,2} = \frac{p_1}{s_1} = T_1$ then $C_{1,2} \le T_3 \le \Phi T_3$, since we deal with instances of Class 3 (and not of Class 1) and $\Phi \ge 1$. Further, since we are considering instances of Class 3 (and not of Class 2), it follows from $T_2 = \frac{p_1+p_2}{s_1+s_2} \le T_3 = \frac{1}{s_1+s_2+s_3}$ that $p_1 \le \frac{s_1+s_2}{s_1+s_2+s_3} - p_2$, which implies that $p_2 \le \frac{s_1+s_2}{2(s_1+s_2+s_3)}$ due to $p_2 \le p_1$. We deduce

$$\begin{aligned} C_{1,2} &\le \frac{1}{s_1}\left(\frac{s_1 - s_2}{s_1}\left(\frac{s_1 + s_2}{s_1 + s_2 + s_3} - p_2\right) + p_2\right) \\ &= \frac{1}{s_1}\left(\frac{s_1 - s_2}{s_1}\left(\frac{s_1 + s_2}{s_1 + s_2 + s_3}\right) + \frac{s_2}{s_1}p_2\right) \\ &\le \frac{1}{s_1}\left(\frac{s_1 - s_2}{s_1}\left(\frac{s_1 + s_2}{s_1 + s_2 + s_3}\right) + \frac{s_2}{s_1}\frac{s_1 + s_2}{2(s_1 + s_2 + s_3)}\right) = \frac{(2s_1 - s_2)(s_1 + s_2)}{2s_1^2(s_1 + s_2 + s_3)}. \end{aligned} \tag{19}$$

This proves the lemma. □

Lemma 7 *For schedule $S_{(1)}^H$ associated with the sets N_u and $N_{v,w}$ found either in Step 2 or in Step 3 of Algorithm Q3Pr1CL3Lim the following bound*

$$\begin{aligned} \frac{C_{\max}\left(S_{(1)}^*\right)}{C_{\max}\left(S_p^*\right)} &\le \frac{C_{\max}\left(S_{(1)}^H\right)}{T_3} \\ &\le \begin{cases} \max\left\{\Phi, \frac{(2s_1 - s_3)(s_1+s_2+s_3)}{3s_1^2}\right\} & \text{if } S_{(1)}^H \text{ is found in Step 2,} \\ \Phi & \text{otherwise,} \end{cases} \end{aligned}$$

holds, where Φ is defined by (16).

Proof We only need to analyze the instances for which $p_2 \le \Phi T_3 s_2$. Since for any triple (u, v, w) we have that $v \le 2$, it follows that $p_2 \le \Phi T_3 s_v$.

In our proof, we need to distinguish between the following three options: the sets N_u and $N_{v,w}$ are found

Option 1: in Step 2 of Algorithm Pr1CL3Lim, so that $\Phi T_3 s_3 < p_2 \le \Phi T_3 s_2$ and $(u, v, w) = (2, 1, 3)$, $N_u = \{J_2\}$, $N_{v,w} = N \setminus \{J_2\}$;

Option 2: by running Step 1 of Algorithm OneSplit, so that $p_1 > \frac{2}{5}$, $p_2 \le \Phi T_3 s_3$ and $(u, v, w) = (1, 2, 3)$, $N_u = \{J_1\}$, $N_{v,w} = N \setminus \{J_1\}$;

Option 3: by running Step 2 of Algorithm OneSplit, so that $p_1 \le \frac{2}{5}$, $p_2 \le \Phi T_3 s_3$ and $p(N_u) \in [1 - \Phi T_3 (s_v + s_w), \Phi T_3 s_u]$.

First, we prove that $C_u \le \Phi T_3$. For Option 1, the inequality $C_u = p_2/s_2 \le \Phi T_3$ follows immediately. For Options 2 and 3, set N_u is found by Algorithm OneSplit, so that (17) holds due to Lemma 5.

From now on, we focus on establishing an upper bound on $C_{v,w}$. Let $p = \max \left\{ p_j | J_j \in N_{v,w} \right\}$ denote the duration of the longest job in set $N_{v,w}$, and $\sigma = (\sigma(1), \sigma(2), \ldots)$ be the sequence of indices for jobs of set $N_{v,w}$ taken in the LPT order, so that $p_{\sigma(1)} = p$. Adapting (8), the makespan $C_{v,w}$ can be written as

$$C_{v,w} = \max \left\{ \frac{p\left(N_{v,w}\right)}{s_v + s_w}, \frac{p}{s_v}, \frac{1}{s_v} \left(\left(\frac{s_v - s_w}{s_v} \right) \sum_{i=1}^{k-1} p_{\sigma(i)} + p_{\sigma(k)} \right) \right\}, \tag{20}$$

where in the last term the preempted job $J_{\sigma(k)}$, such that $\sigma(k) \ge 2$, is identified by Algorithm Q2Pr1.

We start with the case that $C_{v,w}$ is determined by the average load on machines M_v and M_w. Again, for Options 2 and 3, set $N_{v,w}$ is found by Algorithm OneSplit, so that (18) holds due to Lemma 5 and $C_{v,w} \le \Phi T_3$. For Option 1, it follows from $\Phi T_3 s_3 < p_2 \le \Phi T_3 s_2$ that $p(N_u) = p_2 \in [1 - \Phi T_3 (s_1 + s_3), \Phi T_3 s_2]$, since the intervals $[1 - \Phi T_3 (s_1 + s_2), \Phi T_3 s_3]$ and $[1 - \Phi T_3 (s_1 + s_3), \Phi T_3 s_2]$ overlap; see the proof of Lemma 5. Thus, for Option 1, we deduce that $p(N_u) \in [1 - \Phi T_3 (s_v + s_w), \Phi T_3 s_u]$, so that

$$C_{v,w} = \frac{p(N_{v,w})}{s_v + s_w} = \frac{1 - p(N_u)}{s_v + s_w} \le \frac{1 - (1 - \Phi T_3 (s_v + s_w))}{s_v + s_w} = \Phi T_3,$$

and the lemma holds.

Now we look at the case that $C_{v,w} = \frac{p}{s_v}$, so that schedule $S_{v,w}$ is non-preemptive and its makespan is determined by the duration of processing the longest job in set $N_{v,w}$ on the fastest machine M_v. For Option 1, $p = \max \left\{ p_j | J_j \in N_{1,3} \right\} = p_1$ and if $C_{v,w} = C_{1,3} = \frac{p_1}{s_1}$, we obtain $\frac{p_1}{s_1} = T_1 \le T_3 \le \Phi T_3$. For Option 2, clearly $p = p_2$. For Option 3, set $N_{v,w}$ is found by Procedure Split run with $r = 4$. It follows from Step 1 of that procedure that either job J_1 belongs to set N_u and therefore $p \le p_2$, or $p \le p_1 \le \frac{1}{4}$.

It follows from (16) that

$$\Phi T_3 \ge \frac{1}{s_1 + 2s_3} \ge \frac{1}{4s_3},$$

since $s_2 \le s_1 \le 2s_3$. This guarantees that $\Phi T_3 s_3 \ge \frac{1}{4}$ for all speeds of the range under consideration. If $p \le p_2$ then the inequality $p_2 \le \Phi T_3 s_v$ ensures $C_{v,w} \le \Phi T_3$. Alternatively, $p \le \frac{1}{4}$ and we have that

$$\frac{p}{s_v} \le \frac{1}{4s_v} \le \frac{1}{4s_3} \le \Phi T_3. \tag{21}$$

We are now left to consider the case that $C_{v,w}$ is given by the third term in the right-hand side of (20). It can be derived that $C_{\max}\left(S_{v,w}\right)$ does not exceed the actual processing time of the preempted job $J_{\sigma(k)}$ on the slower machine, i.e., $C_{v,w} \le \frac{p_{\sigma(k)}}{s_w}$. Since (8) implies that job $J_{\sigma(k)}$ is at most the second longest in set $N_{v,w}$, it follows that $p_{\sigma(k)} \le p_2$. In the case of Options 2 and 3, the condition $p_2 \le \Phi T_3 s_3$ implies that

$$C_{v,w} \le \frac{p_{\sigma(k)}}{s_w} \le \frac{p_2}{s_w} \le \frac{p_2}{s_3} \le \Phi T_3,$$

so that we only need to consider Option 1. If $p_{\sigma(k)} \le \frac{1}{4}$ then we apply (21) to deduce

$$C_{v,w} \le \frac{p_{\sigma(k)}}{s_w} \le \frac{1}{4s_3} \le \Phi T_3.$$

In the remainder of this proof, we assume $p_{\sigma(k)} > \frac{1}{4}$, i.e., job $J_{\sigma(k)}$, as the second longest job of set $N_{1,3}$ is job J_3.

$$C_{v,w} = \frac{1}{s_1}\left(\left(\frac{s_1 - s_3}{s_1}\right) p_1 + p_3\right).$$

It is clear that $p_1 \le (1 - p_2 - p_3) \le (1 - 2p_3)$, and $p_3 \le \frac{1}{3}$. We deduce

$$\begin{aligned} C_{v,w} &= \frac{1}{s_1}\left(\left(\frac{s_1 - s_3}{s_1}\right) p_1 + p_3\right) \le \frac{1}{s_1}\left(\left(\frac{s_1 - s_3}{s_1}\right)(1 - 2p_3) + p_3\right) \\ &= \frac{1}{s_1}\left(\left(\frac{s_1 - s_3}{s_1}\right) + \left(\frac{2s_3 - s_1}{s_1}\right) p_3\right). \end{aligned}$$

Since $p_3 \le \frac{1}{3}$ and $s_1 \le 2s_3$ for the range of speeds under consideration, we finally derive

$$C_{v,w} \le \frac{1}{3s_1^2}\left(2s_1 - s_3\right),$$

as required. □

Lemmas 6 and 7 together provide a global upper bound on $\rho_3^{(1)}$ (Class 3). We now use this to provide tight bounds for the cases of two slow and two fast machines.

We first notice that the component of Φ given by $\frac{s_1+s_2+s_3}{s_1+2s_3}$ is dominated by $\frac{(2s_1-s_3)(s_1+s_2+s_3)}{3s_1^2}$ found in Lemma 7 since $\frac{(2s_1-s_3)}{3s_1^2} \ge \frac{1}{s_1+2s_3}$ if $(s_1 - 2s_3)(s_3 - s_1) \ge 0$. The latter condition holds for $s_3 \le s_1 \le 2s_3$, the speed range of interest. Hence we define $\tilde{\Phi}$ to be Φ with the component $\frac{s_1+s_2+s_3}{s_1+2s_3}$ removed.

Corollary 2 *For schedule $S_{(1)}^H$ on two slow machines $(2 \ge s_1 = s \ge 1 = s_2 = s_3)$ found by Algorithm Q3Pr1CL3Lim the bound*

$$\frac{C_{\max}\left(S_{(1)}^H\right)}{C_{\max}\left(S_p^*\right)} \le \rho_3^{(1)}\text{ (Class 3)} \le \left\{\begin{array}{ll} \frac{2(s+2)}{5s}, & \text{if } 1 \le s \le \frac{8}{7} \\ \frac{3(s+2)}{4(s+1)}, & \text{if } \frac{8}{7} \le s \le \frac{5}{4} \\ \frac{1}{3}(s+2), & \text{if } \frac{5}{4} \le s \le 2 \end{array}\right\}, \qquad (22)$$

holds and this bound is tight.

Proof For the case of two slow machines, $T_3 = \frac{1}{s+2}$ and we derive from (16) that

$$\tilde{\Phi} = \frac{1}{T_3}\max\left\{\frac{2}{5s_1}, \frac{3}{4\left(s_1 + s_2\right)}, \frac{1}{2s_2 + s_3}\right\} = \max\left\{\frac{2(s+2)}{5s}, \frac{3(s+2)}{4\left(s+1\right)}, \frac{s+2}{3}\right\}.$$

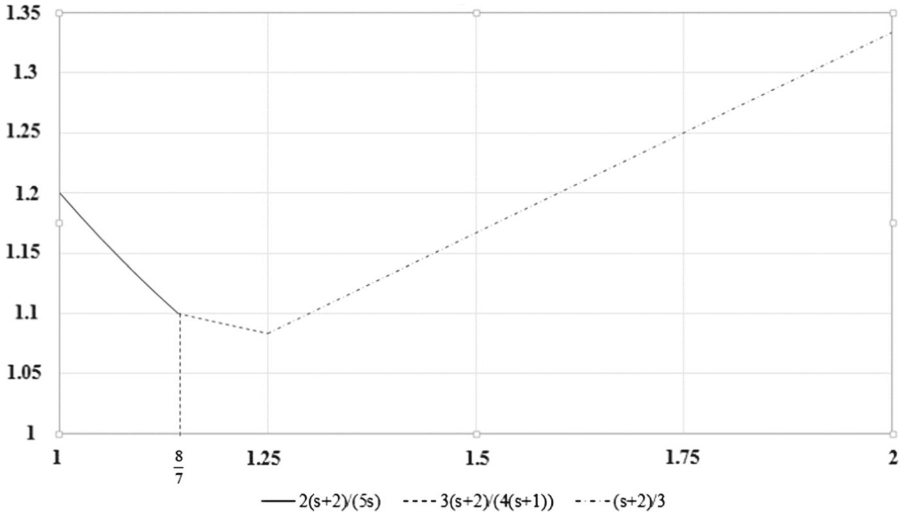

Fig. 3 Graph of Graph of $\rho_3^{(1)}$ (Class 3) for two slow machines as a function of s

If the makespan of schedule $S_{(1)}^H$ is dominated by preemptive processing, then the expressions given in Lemma 6 and in Lemma 7, respectively, simplify as

$$\frac{(2s_1 - s_2)(s_1 + s_2)}{2s_1^2} = \frac{(2s-1)(s+1)}{2s^2};$$
$$\frac{(2s_1 - s_3)(s_1 + s_2 + s_3)}{3s_1^2} = \frac{(2s-1)(s+2)}{3s^2}.$$

Since $\frac{s+1}{2} > \frac{s+2}{3}$ for $s \geq 1$, it follows that in the two expressions above the first one dominates the second. On other other hand, we can compare $\frac{(2s-1)(s+1)}{2s^2}$ with the second and the third components of $\tilde{\Phi}$. We see that $\frac{(2s-1)(s+1)}{2s^2} \leq \frac{3(s+2)}{4(s+1)} \leq \tilde{\Phi}$ for $s \leq \sqrt[3]{2}$, while $\frac{(2s-1)(s+1)}{2s^2} \leq \frac{s+2}{3} \leq \tilde{\Phi}$ for $(s-1)\left(2s^2-3\right) \geq 0$, which holds if $s \geq \sqrt{\frac{3}{2}}$. Since for each $s \in [1,2]$ either $s \leq \sqrt[3]{2} = 1.2599\ldots$ or $s \geq \sqrt{\frac{3}{2}} = 1.2247\ldots$, we deduce that $\frac{(2s-1)(s+1)}{2s^2} \leq \tilde{\Phi}$, so that $\rho_3^{(1)}$ (Class 3) $\leq \tilde{\Phi}$. The break-points are easily seen to be as in (22).

Figure 3 shows the graph of $\rho_3^{(1)}$ (Class 3), as a function of $s \in [1,2]$.

Tight instances are represented by five jobs of equal length for $1 \leq s \leq \frac{8}{7}$, four jobs of equal length for $\frac{8}{7} \leq s \leq \frac{5}{4}$ and three jobs of equal length for $\frac{5}{4} \leq s \leq 2$; see rows 1–3 of Table 1. In this table, the column labelled $S_{(1)}^*$ shows the assignment of the jobs to the machines; if a job appears in two of the columns labelled M_1, M_2 or M_3, then the job is processed on the corresponding machines with a single preemption.

For the instance in row 1 of Table 1, we have that $\tilde{\Phi} = \frac{2(s+2)}{5s}$. The schedule shown in Fig. 4a assigns two jobs to the fast machine M_1 and processes the remaining three jobs on two slow machines with a single preemption of job J_4. Notice that $\frac{2}{5s} > \frac{3}{10}$ for $s < \frac{4}{3}$. This schedule is in fact an optimal schedule for the instance. Indeed, with a single preemption, we need to schedule four full jobs and two parts of the fifth job on three machines. In any case,

Table 1 Tightness examples—limited relative machine speeds

#	$C_{\max}(S_p^*)$	s	p_1	p_2	p_3	p_4	p_5	$S_{(1)}^*$			$C_{\max}(S_{(1)}^*)$
	$= T_3$							M_1	M_2	M_3	
1	$\frac{1}{s+2}$	$1 \le s \le \frac{8}{7}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	J_1, J_2	J_3, J_4	J_4, J_5	$\frac{2}{5s}$
2		$\frac{8}{7} \le s \le \frac{5}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	–	J_1, J_2	J_2, J_3	J_4	$\frac{3}{4(s+1)}$
3		$\frac{5}{4} \le s \le 2$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	–	–	J_1	J_2	J_3	$\frac{1}{3}$
4	$\frac{1}{2s+1}$	$1 \le s \le \frac{5}{4}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	J_1, J_2	J_3, J_4	J_5	$\frac{2}{5s}$
5		$\frac{5}{4} \le s \le 2$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$			J_1, J_3	J_2	J_3	$\frac{2s-1}{3s^2}$

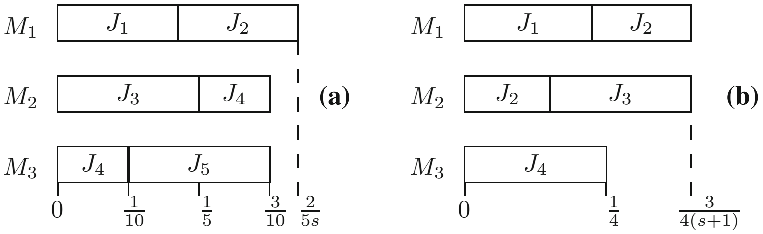

Fig. 4 Schedule $S_{(1)}^*$ for the instances in Table 1: **a** row 1, **b** row 2

there will be a machine that processes at least two full jobs, so that the makespan cannot be smaller than $\frac{2}{5s}$.

For the instance in row 2, we have that $\tilde{\Phi} = \frac{3(s+2)}{4(s+1)}$. The schedule shown in Fig. 4b processes three jobs on machines M_1 and M_2, while machine M_3 processes a single job. The preempted job J_2 is processed on M_2 in the time interval $\left[0, \frac{2-s}{4(s+1)}\right]$, where $\frac{2-s}{4(s+1)} < \frac{1}{4s}$ for all s. Job J_2 is also processed on machine M_1 in the time interval $\left[\frac{1}{4s}, \frac{3}{4(s+1)}\right]$. There is no overlap in the processing of this job. Machine M_2 performs $\frac{2-s}{s+1}$ of the job, while machine M_1 performs $\frac{2s-1}{s+1} = 1 - \frac{2-s}{s+1}$ of the job, i.e., job J_2 is processed in full. Since $\frac{3}{4(s+1)} \ge \frac{1}{4}$ for $s \le 2$, the makespan of this schedule is equal to $\frac{3}{4(s+1)}$, which is the average machine load for processing three jobs on two non-identical machines, i.e., this schedule is the best schedule, provided that the fast machine is involved in the processing of the preempted job. In the alternative class of schedules, exactly one job is processed on machine M_1, while the remaining three jobs are processed with a single preemption on machines M_2 and M_3, similarly to the schedule shown in Fig. 4a. However, the makespan of the latter schedule is equal to $\frac{3}{8} \ge \frac{3}{4(s+1)}$. Thus, the schedule in Fig. 4b is optimal.

For the instance in row 3, we have that $\tilde{\Phi} = \frac{s+2}{3}$. Consider a schedule in which each machine receives exactly one job, which is processed with no preemption. The makespan of the resulting schedule is $\frac{1}{3}$. If no job is processed in full on a slow machine, then the average load of the fast and a single slow machine is equal to $\frac{1}{s+1}$, which is no less than $\frac{1}{3}$ for $s \le 2$.

This completes the proof. □

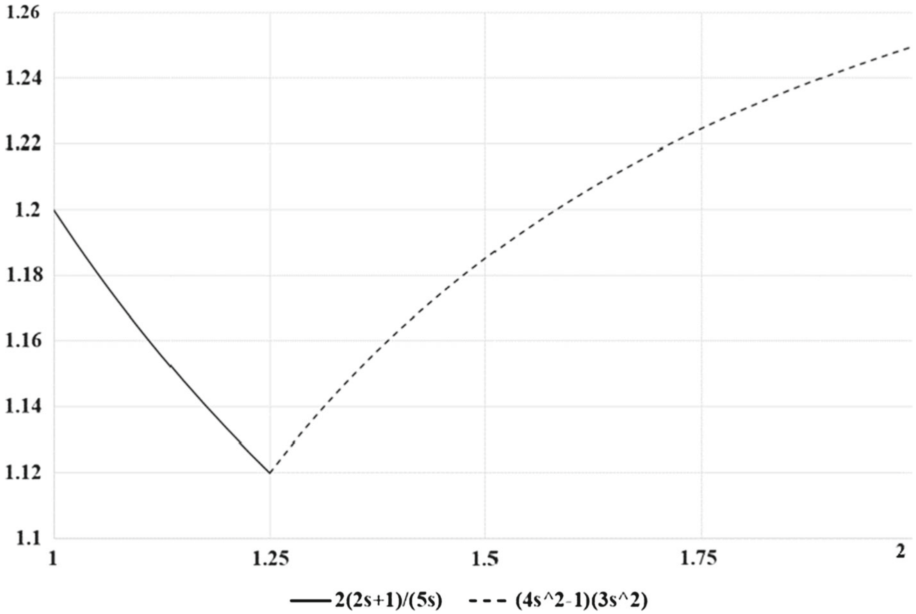

Fig. 5 Graph of $\rho_3^{(1)}$ (Class 3) for two fast machines as a function of s

Corollary 3 *For schedule* $S_{(1)}^H$ *on two fast machines* $(2 \geq s_1 = s_2 = s \geq 1 = s_3)$ *found by Algorithm Q3Pr1CL3Lim the bound*

$$\frac{C_{\max}\left(S_{(1)}^H\right)}{C_{\max}\left(S_p^*\right)} \leq \rho_3^{(1)}\text{ (Class 3)} \leq \left\{ \begin{array}{ll} \frac{2(2s+1)}{5s}, & \text{if } 1 \leq s \leq \frac{5}{4} \\ \frac{4s^2-1}{3s^2}, & \text{if } \frac{5}{4} \leq s \leq 2 \end{array} \right\}, \tag{23}$$

holds and this bound is tight.

Proof For the case of two fast machines, $T_3 = 2s + 1$ and we derive from (16) that

$$\begin{aligned} \tilde{\Phi} &= \frac{1}{T_3} \max\left\{\frac{2}{5s_1}, \frac{3}{4(s_1+s_2)}, \frac{1}{2s_2+s_3}\right\} \\ &= \max\left\{\frac{2(2s+1)}{5s}, \frac{3(2s+1)}{8s}, 1\right\} = \frac{2(2s+1)}{5s}, \end{aligned}$$

where the last equality holds since $\frac{2}{5s}$ dominates the other components for $s \leq 2$.

If the makespan of schedule $S_{(1)}^H$ is dominated by preemptive processing, then the expressions given in Lemma 6 and in Lemma 7, respectively, simplify as

$$\begin{aligned} \frac{(2s_1 - s_2)(s_1+s_2)}{2s_1^2} &= \frac{(2s-s)\,2s}{2s^2} = 1; \\ \frac{(2s_1 - s_3)(s_1+s_2+s_3)}{3s_1^2} &= \frac{(2s-1)(2s+1)}{3s^2} \geq 1. \end{aligned}$$

Thus,

$$\rho_3^{(1)}\text{ (Class 3)} \leq \max\left\{\frac{2(2s+1)}{5s}, \frac{(2s-1)(2s+1)}{3s^2}\right\},$$

and the break-point is easily seen to be as in (23).

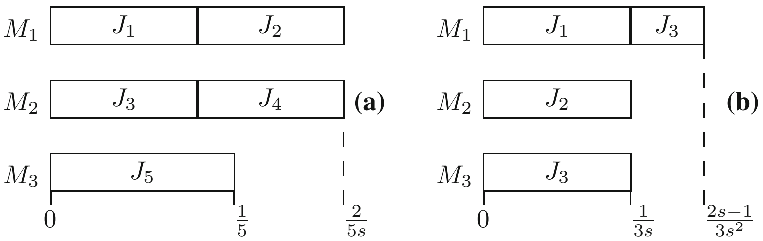

Fig. 6 Schedule $S^*_{(1)}$ for the instances in Table 1: **a** row 4, **b** row 5

Figure 5 shows the graph of $\rho_3^{(1)}$ (Class 3), as a function of $s \in [1, 2]$.

Tight instances consist of five jobs of equal length for $1 \le s \le \frac{5}{4}$, and of three jobs of equal length for $\frac{5}{4} \le s \le 2$; see rows 4–5 of Table 1.

For the instance in row 4 of Table 1, we have that $\tilde{\Phi} = \frac{2(2s+1)}{5s}$. The schedule shown in Fig. 6a assigns two jobs to each of the fast machines M_1 and M_2, while the fifth job is processed on M_3. The makespan of this schedule is $\frac{2}{5s} \ge \frac{1}{5}$ for $s \le 2$. This schedule is in fact an optimal schedule for the instance. Indeed, with a single preemption, we need to schedule four full jobs and two parts of the fifth job on three machines. In any case, there will be a machine that processes at least two full jobs, so that the makespan cannot be smaller than $\frac{2}{5s}$.

For the instance in row 5, the schedule shown in Fig. 6b assigns one job to be processed non-preemptively on each fast machine M_1 and M_2, while job J_3 is processed with a single preemption: on the slow machine M_3 in the time interval $\left[0, \frac{1}{3s}\right]$ and on machine M_1 in the interval $\left[\frac{1}{3s}, \frac{2s-1}{3s^2}\right]$. There is no overlap in the processing of this job. Machine M_3 performs $\frac{1}{s}$ of the job, while machine M_1 performs $\frac{s-1}{s} = 1 - \frac{1}{s}$ of the job, i.e., job J_3 is processed in full. If machine M_3 processes at least one full job, then it completes no earlier than time $\frac{1}{3}$, which is no less than $\frac{2s-1}{3s^2}$ for all $s \ge 1$. If M_3 processes no jobs at all, then the three jobs are to be processed on two identical fast machines M_1 and M_2, which implies the makespan of at least $\frac{1}{2s}$, the latter value being no less than $\frac{2s-1}{3s^2}$ for $s \le 2$. Thus, the schedule in Fig. 6b is an optimal schedule $S^*_{(1)}$. □

4.2 Unlimited relative machine speeds ($s_1 \ge 2s_3$)

In this section, we assume that $s_1 \ge 2s_3$. The analysis of the value of $\rho_3^{(1)}$ (Class 3) is focused on the instances of the three-machine problem with either two fast or two slow machines. For these situations, we derive tight upper bounds on the value $\rho_3^{(1)}$ (Class 3).

The algorithm below schedules all jobs with at most one preemption on the first two machines, the slowest machine M_3 is not used at all.

Algorithm Q3Pr1CL3UnLim

Input: An instance $I = (\mathcal{L}_n, \mathcal{M}_3)$ of Class 3, with its jobs having normalized processing times, $p(N) = 1$, numbered in their LPT order; among the three machines there are either two fast ($s_1 = s_2 = s \ge 2, s_3 = 1$) or two slow ($s_1 = s \ge 2, s_2 = s_3 = 1$)

Output: A schedule $S^H_{(1)}$ with at most one preemption

Step 1. Create schedule $S_{(1)}^{H}$ in which all jobs are scheduled in accordance with Algorithm Q2Pr1 on machines $M' = M_1$ and $M'' = M_2$. Stop.

Lemma 8 *For schedule $S_{(1)}^{H}$ found in Step 1 of Algorithm Q3Pr1CL3UnLim, the inequality*

$$\rho_3^{(1)}\,(\text{Class }3) \le \frac{(3s-2)\,(s+2)}{3s^2}$$

holds for two slow machines, while the inequality

$$\rho_3^{(1)}\,(\text{Class }3) \le \frac{2s+1}{2s}$$

holds for two fast machines. Both bounds are tight.

Proof Algorithm Q2Pr1 will produce a schedule with the makespan equal to the maximum of one of three expressions similarly to (20), i.e.,

$$C_{\max}\left(S_{(1)}^{H}\right) = \max\left\{\frac{1}{s_1+s_2}, \frac{p_1}{s_1}, \frac{1}{s_1}\left(\left(\frac{s_1-s_2}{s_1}\right)\sum_{i=1}^{k-1} p_i + p_k\right)\right\}. \tag{24}$$

The second expression $\frac{p_1}{s_1}$ cannot dominate, since $\frac{p_1}{s_1} = T_1 \le T_3 = \frac{1}{s_1+s_2+s_3} < \frac{1}{s_1+s_2}$ for Class 3 instances.

In the case of two fast machines, in order to create schedule $S_{(1)}^{H}$, two identical machines of speed s are used and $C_{\max}\left(S_{(1)}^{H}\right) = \frac{1}{s_1+s_2} = \frac{1}{2s}$ for any Class 3 instance. Hence, $\rho_3^{(1)}\,(\text{Class }3) = \frac{C_{\max}\left(S_{(1)}^{H}\right)}{T_3} = \frac{2s+1}{2s}$ as claimed. This function is decreasing and attains its maximum of $\frac{5}{4}$ at $s = 2$; see Fig. 7.

For two slow machines, the makespan $C_{\max}\left(S_{(1)}^{H}\right)$ is given by either the average machine load on the two machines $\frac{1}{s_1+s_2} = \frac{1}{s+1}$, or is defined by the total actual processing time of the preempted job J_k. In the latter case, the third expression of (24) for the two slow machines simplifies to

$$C_{\max}\left(S_{(1)}^{H}\right) = \frac{\left(\frac{s-1}{s}\right)\sum_{i=1}^{k-1} p_j + p_k}{s}. \tag{25}$$

If $k = 2$, then in schedule $S_{(1)}^{H}$ the jobs J_1 and J_2 are scheduled exactly as in the schedule analyzed in Lemma 6. Thus, (19) holds, which for the two slow machines simplifies to

$$C_{\max}\left(S_{(1)}^{H}\right) \le \frac{(2s_1-s_2)\,(s_1+s_2)}{2s_1^2\,(s_1+s_2+s_3)} = \frac{(2s-1)\,(s+1)}{2s^2\,(s+2)}.$$

If for the preempted job J_k we have that $k \ge 3$, then substituting $\sum_{i=1}^{k-1} p_j \le 1 - p_k$ and $p_k \le 1/k$ into (25) gives

$$C_{\max}\left(S_{(1)}^{H}\right) \le \frac{\frac{s-1}{s} + \frac{p_k}{s}}{s} \le \frac{\frac{s-1}{s} + \frac{1}{ks}}{s}.$$

The right-hand side expression above achieves its maximum for $k = 3$, which implies

$$C_{\max}\left(S_{(1)}^{H}\right) \le \frac{3s-2}{3s^2}.$$

Hence

$$C_{\max}\left(S_{(1)}^{H}\right) \le \max\left\{\frac{1}{s+1}, \frac{(2s-1)\,(s+1)}{2s^2\,(s+2)}, \frac{3s-2}{3s^2}\right\}.$$

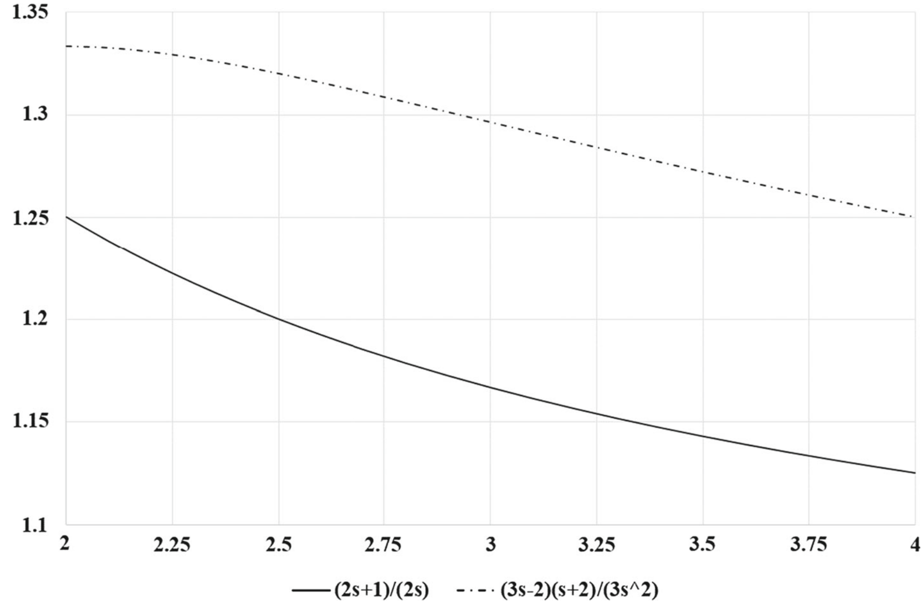

Fig. 7 Graphs of $\rho_3^{(1)}$ (Class 3) as a function of s

It is easily checked that $\frac{(2s-1)(s+1)}{2s^2(s+2)} \le \frac{3s-2}{3s^2}$ for $s \ge 1$ and $\frac{1}{s+1} \le \frac{3s-2}{3s^2}$ for $s \ge 2$, so that $\frac{3s-2}{3s^2}$ dominates, and

$$\rho_3^{(1)}\text{ (Class 3)} \le \frac{C_{\max}\left(S_{(1)}^H\right)}{T_3} \le \frac{(3s-2)(s+2)}{3s^2},$$

as claimed. The maximum value of $\rho_3^{(1)}$ (Class 3) for the two slow machines is $4/3$, which is attained at $s=2$; see Fig. 7. Notice that this maximum value is equal to the global bound $2-2/m$, since $m=3$.

For both situations, the tightness of the bounds is provided by the instance with three identical jobs, $p_1 = p_2 = p_3 = \frac{1}{3}$. In the schedules discussed below the slow machine M_3 is assigned no jobs.

In the case of two fast machines, the schedule shown in Fig. 8a processes the jobs on two identical machines of speed s with a single preemption, which gives the makespan of $\frac{1}{2s}$. This schedule is in fact an optimal schedule for the instance. Indeed, if machine M_3 processes a full job, then that job would complete at time $\frac{1}{3} > \frac{1}{2s}$ for $s \ge 2$. If there is a job which is processed with preemption on M_3 and on one of the fast machines, then we get a schedule shown in Fig. 6b, with the makespan $\frac{2s-1}{3s^2} \ge \frac{1}{2s}$ for $s \ge 2$.

In the case of two slow machines, consider the schedule shown in Fig. 8b, in which two full jobs are processed on machine M_1, and job J_3 is processed with a single preemption, on the slow machine M_2 in the time interval $\left[0, \frac{2}{3s}\right]$ and on machine M_1 in the interval $\left[\frac{2}{3s}, \frac{3s-2}{3s^2}\right]$. There is no overlap in the processing of this job. Machine M_2 performs $\frac{2}{s}$ of the job, while machine M_1 performs $\frac{s-2}{s} = 1 - \frac{2}{s}$ of the job, i.e., job J_3 is processed in full. If a slow machine processes at least one full job, then it completes no earlier than time $\frac{1}{3}$, which is no less than $\frac{3s-2}{3s^2}$ for $s \ge 2$. Thus, the schedule in Fig. 8b is an optimal schedule $S_{(1)}^*$. □

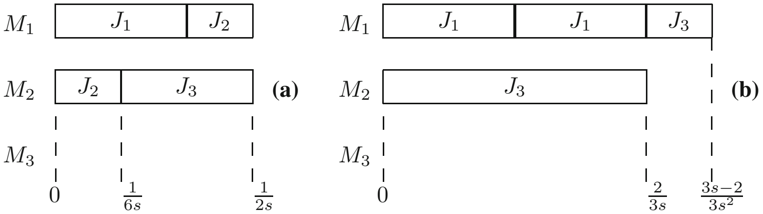

Fig. 8 Schedule $S^*_{(1)}$ in the case of **a** two fast machines; **b** two slow machines

5 Conclusion

In this paper, for the problem on three uniform machines to minimize the makespan we have performed a parametric analysis of the quality of a schedule with at most one preemption compared to the optimal preemptive schedule. We have derived tight bounds in terms of a single parameter, the relative speed of the machines, for the cases of two fast and two slow machines, respectively.

We hope that Procedure Split, simple as it, may appear useful as part of various scheduling algorithms that require partitioning of jobs into two subsets with prescribed properties.

For the general problem on $m \geq 3$ uniform machines, it is an interesting research goal to deduce a quality measure of schedules on uniform machines, provided any fixed number of preemptions $2 \leq q \leq 2(m-1)$ is allowed. Possible extensions may also include an objective function other than the makespan and/or a more general machine environment, e.g., unrelated parallel machines.

References

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

Brucker, P. (2007). *Scheduling algorithms* (5th ed.). Berlin: Springer.

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. *Mathematical Operations Research*, *37*, 379–398.

Epstein, L., & Levin, A. (2016). The benefit of preemption for single machine scheduling so as to minimise 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., Ibarra, O. H., & Sahni, S. (1977). Bounds for LPT schedules on uniform processors. *SIAM Journal on Computing*, *6*, 155–166.

Gonzalez, T. F., & Sahni, S. (1978). Preemptive scheduling of uniform processor systems. *Journal of the ACM*, *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.

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

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

Mireault, P., Orlin, J. B., & Vohra, R. V. (1997). A parametric worst case analysis of the LPT heuristic for two uniform machines. *Operations Research*, *45*, 116–125.

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

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

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

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)* (Vol. 28, pp. 392–402). https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.392.

Soper, A. J., & Strusevich, V. A. (2018). Power of a single preemption on uniform parallel machines. *Discrete Applied Mathematics* https://doi.org/10.1016/j.dam.2018.03.007. (**in Press**).

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