

INSTITUT FÜR INFORMATIK

**Estimating The Makespan of The
Two-Valued Restricted Assignment
Problem**

Klaus Jansen, Kati Land, Marten Maack

Bericht Nr. 1505

Mai 2015

ISSN 2192-6247



CHRISTIAN-ALBRECHTS-UNIVERSITÄT
ZU KIEL

Estimating The Makespan of The Two-Valued Restricted Assignment Problem*

Klaus Jansen¹, Kati Land¹, and Marten Maack¹

Institute of Computer Science, University of Kiel, 24118 Kiel, Germany
{kj, kla, mmaa}@informatik.uni-kiel.de

Abstract We consider a special case of the scheduling problem on unrelated machines, namely the Restricted Assignment Problem with two different processing times. We show that the configuration LP has an integrality gap of at most $\frac{5}{3} + \frac{1}{156} \approx 1.6731$ for this problem. This allows us to estimate the optimal makespan within a factor of $\frac{5}{3} + \frac{1}{156}$, improving upon the previously best known estimation algorithm with ratio $\frac{11}{6} \approx 1.833$ due to Chakrabarty, Khanna, and Li [2].

1 Introduction

Scheduling on unrelated machines is a problem where we are given a set J of jobs and a set M of machines and the processing time of job $j \in J$ on machine $i \in M$ is given by p_{ij} . The task is to find an assignment $\sigma: J \rightarrow M$, called the schedule, that minimizes the makespan, i.e. the maximum load $\max_{i \in M} \sum_{j \in \sigma^{-1}(i)} p_{ij}$ of a machine. Lenstra, Shmoys, and Tardos [5] proved that it is NP-hard to approximate the makespan with a factor less than 1.5. On the other hand they have presented an algorithm with approximation ratio 2 which uses a special rounding procedure for the assignment LP. This was slightly improved to $2 - 1/|M|$ [6], matching the integrality gap of the assignment LP.

Closing this gap between approximability and inapproximability is a major open problem in scheduling theory. Since no progress in this regard has been made for more than 20 years, the focus has shifted towards special cases of the problem. One important special case is the so called Restricted Assignment Problem, where

* Research was in part supported by German Research Foundation (DFG) project JA 612 /15-1

for each job $j \in J$ there is a number p_j such that $\{p_{ij} \mid i \in M\} \subseteq \{p_j, \infty\}$. A natural interpretation is that for each job j there is a set $M(j) \subseteq M$ of machines on which j may be processed, and the processing time is the same on each of these machines. The restricted assignment case may look easier than the general problem, but the inapproximability bound of $1.5 - \varepsilon$ still holds, even if we further restrict that $|M(j)| \leq 2$ and $p_j \in \{1, 2\}$ for each job j [3]. A breakthrough was achieved by Svensson [8], who proved that the integrality gap of another LP formulation, the configuration LP, is at most $33/17 \approx 1.941$ for the Restricted Assignment Problem. Using a rounding technique [1], the optimal makespan can be estimated within a factor of $33/17 + \varepsilon$ in polynomial time, where $\varepsilon > 0$ is an arbitrary small constant. Unfortunately, no polynomial time rounding procedure for the configuration LP which matches the integrality gap is known, so a corresponding schedule may not be found.

Better results have been obtained when the instances have further restrictions. E.g. if $|M(j)| \leq 2$ for each $j \in J$, there is a 1.75-approximation [3]. We will in particular investigate the case of only two types of jobs: small jobs with processing time s and big jobs with processing time b . Svensson [8] proved that the integrality gap of the configuration LP in this case is at most $\frac{5}{3} + \frac{s}{b}$ if $b = \text{OPT}$. Kolliopoulos and Moysoglou [4] pointed out that this bound also holds for $\text{OPT} < 2b$. Note that if $\text{OPT} \geq 2b$, the approximation algorithm due to Lenstra, Shmoys, and Tardos [5] is already a 1.5 approximation. Combining this with a $2 - \frac{s}{b}$ -approximation [4,2], the makespan can be estimated within a factor of 1.883 [4]. Recently, Chakrabarty, Khanna, and Li [2] improved the estimation of the makespan to 1.833 and presented a constructive algorithm with approximation ratio $2 - \delta$ for a very small $\delta > 0$.

Our Contribution Our first result is an improved bound for the integrality gap of the configuration LP. In Section 3, we conduct a tighter analysis for Svensson's [8] local search algorithm in the case that $\text{OPT}_{\text{LP}} - b$ is a multiple of s . Then, we modify general instances in Section 4 to satisfy this requirement. To this end, we scale one of the processing times s and b to create a new instance, distinguishing several cases, based among other things on the value of $\frac{s}{b}$ and whether OPT_{LP} is a multiple of s . In each case, we bound the integrality gap

of the original instance in terms of the integrality gap of the new instance, and subsequently use our result from Section 3 on the new instance. With this technique we are able to show the following:

Theorem 1. *The integrality gap of the configuration LP for the Restricted Assignment Problem with two different processing times is at most $\frac{5}{3} + \frac{1}{156} \approx 1.6731$.*

Indeed, we prove the integrality gap is bounded by $\frac{5}{3}$ for almost all instances. This also allows us to estimate the optimal makespan within a factor of 1.6731, improving upon the previously best possible ratio 1.833.

Our second result, presented in Section 5, is an algorithm that finds a schedule with makespan at most $\text{OPT}_{\text{LP}} + b - s$ for the case that $\text{OPT} < 2b$. This is obtained by adding additional constraints to the assignment LP and applying a rounding technique due to Shmoys and Tardos [7]. As a corollary, the integrality gap of the augmented assignment LP and the configuration LP is bounded by $2 - \frac{s}{b}$. We should note that the approximation ratio $2 - \frac{s}{b}$ was independently obtained by Chakrabarty, Khanna, and Li [2] using slightly different methods.

Our new bounds on the integrality gap of the configuration LP are displayed in Fig. 1.

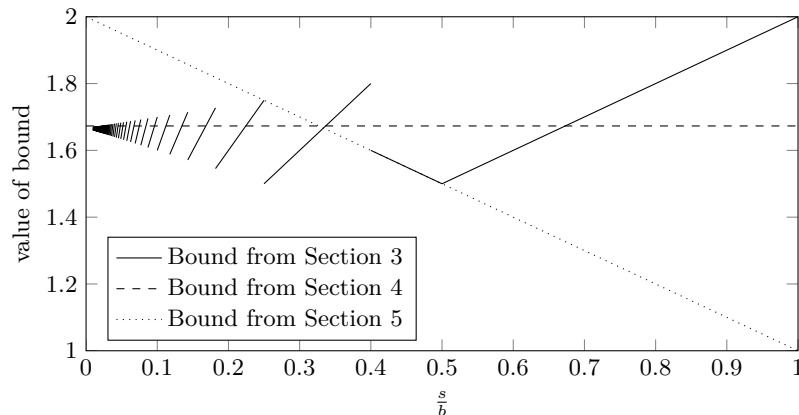


Figure 1. Comparison of bounds on the integrality gap.

2 Preliminaries

Let I be an instance to the restricted assignment problem and T be the desired makespan. Then the following LP, denoted by $\text{ALP}(T)$, is known as the assignment LP and has a feasible integer solution if and only if there is a schedule with makespan T :

$$\sum_{i \in M} x_{ij} = 1 \quad \text{for each } j \in J \quad (1)$$

$$\sum_{j \in J} p_{ij} x_{ij} \leq T \quad \text{for each } i \in M \quad (2)$$

$$x_{ij} = 0 \quad \text{for each } i \in M, j \in J \text{ with } p_{ij} > T \quad (3)$$

$$x_{ij} \geq 0 \quad \text{for each } i \in M, j \in J \quad (4)$$

Usually, the integrality gap of an integer linear program would be defined by $\sup_I \frac{\text{OPT}(I)}{\text{OPT}_{\text{LP}}(I)}$, where $\text{OPT}(I)$ and $\text{OPT}_{\text{LP}}(I)$ denote the optimal integer and fractional solutions of the LP. In this case however, we have feasibility programs. One therefore defines $\text{OPT}(I) = \min\{T \mid \text{ALP}(T) \text{ has a feasible integer solution}\}$ and $\text{OPT}_{\text{LP}}(I)$ analogously. Indeed, with this definition $\text{OPT}(I)$ is equal to the makespan of an optimal schedule and $\text{OPT}_{\text{LP}}(I)$ can be found by binary search in polynomial time [5].

The configuration LP is a stronger formulation for this problem. To introduce it, we require some notation. For each set $J' \subseteq J$ of jobs we define $p(J') = \sum_{j \in J'} p_j$. We also abbreviate $p(j) = p(\{j\})$ for a single job j . A configuration for a machine i is a set $C \subseteq J$ with $\sum_{j \in C} p_{ij} \leq T$. We denote the set of all configurations for machine i that respect the target makespan T by $\mathcal{C}(i, T)$. The configuration LP $\text{CLP}(T)$ is defined as

$$\sum_{C \in \mathcal{C}(i, T)} x_{i,C} \leq 1 \quad \text{for each } i \in M \quad (5)$$

$$\sum_{C \in \mathcal{C}(i, T)} \sum_{\substack{i \in M \\ j \in C}} x_{i,C} \geq 1 \quad \text{for each } j \in J \quad (6)$$

$$x_{i,C} \geq 0 \quad \text{for each } i \in M, C \in \mathcal{C}(i, T) \quad (7)$$

The first constraint enforces that at most one configuration is assigned to each machine, and with the second constraint we ensure each job is

completely assigned. From now on, we will use $\text{OPT}(I)$ and $\text{OPT}_{\text{LP}}(I)$ to denote the integral and fractional optima of the configuration LP, and even OPT or OPT_{LP} if the instance is clear from the context. Note that, while the number of configurations may be exponentially large, one can solve $\text{CLP}(T)$ in polynomial time via its dual [1]. If we interpret $\text{CLP}(T)$ as maximizing a zero objective function, the dual is given by

$$\min \sum_{i \in M} y_i - \sum_{j \in J} z_j \tag{8}$$

$$y_i \geq \sum_{j \in C} z_j \quad \text{for each } i \in M, C \in \mathcal{C}(i, T) \tag{9}$$

$$z_j \geq 0 \quad \text{for each } j \in J \tag{10}$$

Finding a violated constraint of the dual is equivalent to $|M|$ knapsack problems. In the general case, one uses an FPTAS for knapsack as separation oracle. The solution then may contain configurations C with $T < p(C) \leq (1 + \varepsilon)T$, where $\varepsilon > 0$ is the chosen precision. We can therefore approximate $\text{OPT}_{\text{LP}}(I)$ within a factor $1 + \varepsilon$ for arbitrary small in polynomial time ε [1].

In our case, it is even possible to solve the knapsack problems exactly. Remember that we have only two job sizes and each configuration may contain at most one big job. Thus we generate two solutions, one containing the most profitable big job, and one without big jobs. Both solutions are filled with small jobs in a greedy manner. The better of those solutions is optimal.

3 Bounding the Integrality Gap by Local Search

In this section we prove an improved bound on the integrality gap of the configuration LP. Our proof requires that each configuration in the optimal solution contains at most one big job and $\text{OPT}_{\text{LP}} - b$ is a multiple of s . To satisfy the first condition it is sufficient (but not necessary) that $\text{OPT} < 2b$. The argumentation also works for a restricted variant of the problem where only one big job per configuration is permitted.

We will show the following, see also Fig. 1:

Theorem 2. *If an instance of the restricted assignment problem only has jobs of two sizes $s < b$ such that each configuration in the optimal solution contains at most one big job and $\text{OPT}_{\text{LP}} - b$ is a multiple of s , then the integrality gap of the CLP is at most*

- (i) $1 + \frac{(b-s)}{\text{OPT}_{\text{LP}}}$ if $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$ and
- (ii) $1 + \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil \frac{s}{\text{OPT}_{\text{LP}}}$ otherwise.

For simplicity, we scale the processing times such that $\text{OPT}_{\text{LP}} = 1$. An upper bound on the values given in Theorem 2 is $\frac{5}{3} + \frac{1}{3}s$. We give a short proof of the upper bound: For (i): We have $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$, therefore $b \leq \frac{5}{2}s$ and $s < \frac{1}{2}b \leq \frac{1}{2}$, and it follows that

$$1 + b - s \leq 1 + \left(\frac{5}{2} - 1\right)s = 1 + \frac{4}{3}s + \frac{1}{6}s \leq \frac{5}{3} + \frac{1}{6}s \quad . \quad (11)$$

For (ii): Using $b \leq 1$ we obtain

$$\begin{aligned} 1 + \left\lceil \frac{2}{3} \left(\frac{b}{s} - 1 \right) \right\rceil s &\leq 1 + \left(\frac{2}{3} \left(\frac{b}{s} - 1 \right) + 1 \right) s \\ &= 1 + \left(\frac{2b}{3s} + \frac{1}{3} \right) s = 1 + \frac{2}{3}b + \frac{1}{3}s \quad (12) \\ &\leq \frac{5}{3} + \frac{1}{3}s \quad . \end{aligned}$$

We will use the rest of the section to prove Theorem 2. Recall that Svensson [8] proved the bound $\frac{5}{3} + s$. To prove Theorem 2, we utilize the local search technique due to Svensson [8], but with an improved analysis.

The high-level overview is as follows: We use a family $(A_R)_{R>0}$ of algorithms, where each member A_R takes a partial schedule σ , i.e. a feasible schedule for a subset $J' \subset J$ of the jobs, and a currently unscheduled job j_{new} as parameters. It should return a feasible schedule for $J' \cup \{j_{\text{new}}\}$. In addition, A_R maintains the invariant that the makespan is at most $1 + R$. Given an instance of the problem for that CLP(1) is feasible, iteratively applying A_R to each job, starting with an empty schedule, yields a schedule for all jobs with makespan at most $1 + R$. Note that we cannot give a polynomial bound on the running time of this procedure, but the mere existence of the resultant schedule proves that the integrality gap is bounded by $1 + R$.

The crucial point is indeed that the algorithm *successfully terminates at all*, and we can prove this if R meets certain requirements.

In the following we give a more detailed description of the algorithm. The main idea is to move jobs from their current machine to another machine while maintaining the invariant that the makespan is at most $1 + R$. In the beginning, we wish to move only the unassigned job j_{new} to some machine. Suppose that all machines in $M(j)$ have too much load, otherwise we are done. The algorithm then will try to reduce the load on some machine $i \in M(j_{\text{new}})$ by moving some job $j \in \sigma^{-1}(i)$ away from i . If such a move is again not immediately possible, the process repeats. Since we are trying to reduce the load on machine i , moving more jobs to i may be unhelpful, depending on the job's sizes. Thus the algorithm needs to store which jobs it currently tries to move and which machines it should not try to move jobs to, and it does so by the use of blockers. Whenever the algorithm decides that a move has the potential to be helpful but does not immediately lead to a schedule with makespan at most $1 + R$, a blocker is created. More formally, a move is a pair (j, i) , where j is a job and $i \in M(j) \setminus \{\sigma(j)\}$. We distinguish three types of moves: (j, i) is a small move if j is small, a big-to-small move if j is big and $\sigma^{-1}(i)$ contains only small jobs, and a big-to-big move if j is big and $\sigma^{-1}(i)$ contains a big job. If assigning $\sigma(j) = i$ yields a schedule of makespan at most $1 + R$, the move is called valid, otherwise it is invalid. When a potentially helpful move (j, i) is found to be invalid, a blocker B is created. B is a tuple consisting of a machine $m(B) = i$, the set $J(B)$ of jobs we wish to move away from $m(B)$, and the move $\text{mv}(B) = (j, i)$ that caused the creation of B . If (j, i) is a big-to-big move, we call B a big blocker and set $J(B) = \{j_{\text{big}}\}$, where j_{big} is the single big job on i . Creating this big blocker will prevent all attempts to move another big job to i . The intuition is that the move (j, i) only can become valid if j_{big} is moved away from i and no other big job replaces it. Note that we use the fact that no two big jobs can be on one machine. If (j, i) is a small or big-to-small move, we set $J(B) = \sigma^{-1}(i)$ and call B a small blocker. The algorithm will not try to move any job to machine i , increasing the likelihood that (j, i) becomes valid. All blockers are stored in a list $L = B_0, \dots, B_t$ in order of creation.

We proceed to describe when the algorithm deems a move potentially helpful. Let $J(L) = \{j_{\text{new}}\} \cup \bigcup_{k=0}^t J(B_k)$ be the set of all jobs we

wish to move. Define the set of machines that are contained in a big blocker by $M_B(L)$, the set of machines that are contained in a small blocker by $M_S(L)$, and $M(L) = M_S(L) \cup M_B(L)$. Furthermore denote the set of small jobs on machine i that cannot be moved to any other machine by $S_i := \{j \in \sigma^{-1}(i) \mid j \text{ is small and } M(j) \setminus \{i\} \subseteq M_S(L)\}$. We now define the potential moves. A small move (j, i) is a potential move when $j \in J(L)$ and $i \notin M_S(L)$. A big-to-small or big-to-big-move (j, i) is a potential move when $j \in J(L)$, $i \notin M(L)$, and $p(S_i) \leq 1 - b + R$. In the presence of several potential moves, the algorithm chooses one with minimum lexicographical value, defined as

$$\text{Val}(j, i) = \begin{cases} (0, 0) & \text{if } (j, i) \text{ is valid,} \\ (1, p(\sigma^{-1}(i))) & \text{if } (j, i) \text{ is small move,} \\ (2, p(\sigma^{-1}(i))) & \text{if } (j, i) \text{ is big-to-small move, and} \\ (3, 0) & \text{if } (j, i) \text{ is big-to-big move.} \end{cases} \quad (13)$$

The complete procedure is summarized in algorithm 1.

Algorithm 1: $A_R(\sigma, j_{\text{new}})$

```

1 Initialize  $L \leftarrow$  empty list
2 while  $\sigma(j_{\text{new}}) = \perp$  do
3   Choose a potential move  $(j, i)$  of minimum lexicographic value
4   if  $(j, i)$  is valid then
5     Let  $B_k$  be the blocker in  $L = B_0, \dots, B_t$  such that  $j \in J(B_k)$ 
6     Remove  $B_k$  and all blockers added after it from  $L$ :  $L \leftarrow B_0, \dots, B_{k-1}$ 
7     Update Schedule:  $\sigma(j) \leftarrow i$ 
8   else if  $(j, i)$  is small or big-to-small then
9     Create small blocker  $B$  with  $J(B) = \sigma^{-1}(i) \setminus J(L)$ ,  $m(B) = i$ ,
10     $mv(B) = (j, i)$ 
11    Append  $B$  to  $L$ 
12   else  $\#$   $(j, i)$  is big-to-big
13     Let  $j_{\text{big}}$  be the big job in  $\sigma^{-1}(i)$ 
14     Create big blocker  $B$  with  $J(B) = \{j_{\text{big}}\}$ ,  $m(B) = i$ ,  $mv(B) = (j, i)$ 
15     Append  $B$  to  $L$ 
16 return  $\sigma$ 

```

Svensson proved the following:

Lemma 1 ([8]). *Let $b = \text{OPT}$. $A_{\frac{2}{3}+s}$ always terminates, unless $\text{CLP}(1)$ is infeasible.*

We provide a specialized, but stronger, variant of Lemma 1.

Lemma 2. *Let $R \geq s$ and let each configuration in the optimal solution contain at most one big job. Define $k = \lfloor \frac{R}{s} \rfloor$ and $\delta = (k + 1)s - R \in (0, s]$. If there exists $0 \leq c \leq b$ such that*

$$b + s - R - \delta \leq c \leq R + \delta \tag{14}$$

$$2R + 2\delta + c + 2(1 - b) - s \geq 2 \tag{15}$$

$$2R + \delta - b - s \geq 0 \quad , \tag{16}$$

then A_R always terminates, unless $\text{CLP}(1)$ is infeasible.

Proof. The proof that A_R successfully terminates consists of two steps. The first step is to show that the algorithm always has a potential move to choose (line 3). In the second step one argues that the algorithm terminates under this condition (condition in line 2).

For the second step, associate to the list $L = B_0, \dots, B_t$ of blockers for each iteration the vector

$$(\text{Val}(\text{mv}(B_0)), \text{Val}(\text{mv}(B_1)), \dots, \text{Val}(\text{mv}(B_t)), \infty) \tag{17}$$

of corresponding values. Svensson showed that this vector strictly decreases (by lexicographical ordering) with every iteration and that there is only a finite number of such vectors [8, Lemma 3.10]. The same argument holds in our case.

For the rest of the proof, we concern ourselves with the first step, the existence of a potential move. Assume for the sake of contradiction that algorithm A_R reached an iteration where no more potential moves exist. We want to show that the $\text{CLP}(1)$ is infeasible. To this end we construct a solution (y, z) to the dual of $\text{CLP}(1)$ with negative value. This implies unboundedness of the dual, since (y, z) can be scaled by an arbitrary large value α to obtain a new solution $(\alpha y, \alpha z)$. By the duality theorem the primal is infeasible as desired.

We define $S = \cup_{i \in M} S_i$. Our solution (y, z) of the dual is given by

$$z_j = \begin{cases} c & \text{if } j \in J(L) \text{ is big} \\ s & \text{if } j \in J(L) \cup S \text{ is small} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

and

$$y_i = \begin{cases} 1 & \text{if } i \in M_S(L) \\ \sum_{j \in \sigma^{-1}(i)} z_j & \text{otherwise.} \end{cases} \quad (19)$$

CLAIM 1. (y, z) is feasible for the dual.

Proof of Claim 1. Let $i \in M$ and $C \in \mathcal{C}(i, 1)$. We show that the constraint of the dual of the CLP(1) is fulfilled: $y_i \geq \sum_{j \in C} z_j$. Note that it is sufficient to consider $C \subseteq J(L) \cup S$ since any $j \in J \setminus (J(L) \cup S)$ has $z_j = 0$.

Case 1: $i \in M_S(L)$. For $i \in M_S(L)$ we get $y_i = 1$ by the definition of our solution (y, z) of the dual. Then we have $y_i = 1 \geq \sum_{j \in C} p_j \geq \sum_{j \in C} z_j$, since $c \leq b$ and therefore $z_j \leq p_j$ for any job j .

Case 2: $i \notin M_S(L)$. For $i \notin M_S(L)$ we get $y_i = \sum_{j \in \sigma^{-1}(i)} z_j$ by the definition of our solution (y, z) .

Case 2.1: C contains a big job j_{big} .

Case 2.1.1: $i \in M_B(L)$. Because $i \in M_B(L)$ we know that i has a big job since big blockers only occur when we try to move a big job to a machine that already has a big job assigned. Let $\sigma_s^{-1}(i)$ be the small jobs scheduled on i . We have $y_i = c + \sum_{j \in \sigma_s^{-1}(i)} s = c + |\sigma_s^{-1}(i)|s$ and $\sum_{j \in C} z_j = c + (|C| - 1)s$. Now assume that $|\sigma_s^{-1}(i)| < |C| - 1$. Then there is a small job that is allowed on machine i because $j \in C \setminus \{j_{\text{big}}\}$. But because of $\sigma(j) \neq i$ the job j is not scheduled on i . There is no small blocker on machine i , so it is possible to move job j to machine i , that means (j, i) is a valid move, which by assumption cannot exist. Therefore $|\sigma_s^{-1}(i)| \geq |C| - 1$ and it follows $y_i = c + \sum_{j \in \sigma_s^{-1}(i)} s \geq c + (|C| - 1)s = \sum_{j \in C} z_j$.

Case 2.1.2: $i \notin M_B(L)$. Since j_{big} is allowed on i and (i, j_{big}) is not a potential move, the definitions of a potential big-to-big move and a potential big-to-small move say that $p(S_i) > 1 - b + R$. Since $p(S_i)$ and $1 - b$ are multiples of s and $R + \delta$ is the next multiple of s above R , we even have $p(S_i) \geq 1 - b + R + \delta$. It follows with (14) that

$$y_i = \sum_{j \in \sigma^{-1}(i)} z_j \geq p(S_i) \geq 1 - b + R + \delta \geq 1 - b + c \geq c + (|C| - 1)s = \sum_{j \in C} z_j \quad . \quad (20)$$

Case 2.2: C contains only small jobs. Let $j \in C$. Since $C \subseteq J(L) \cup S$ we have $j \in J(L) \cup S$. Then it must hold that $j \in \sigma^{-1}(i)$, otherwise (j, i) would be a potential move. Thus $y_i = \sum_{j \in \sigma^{-1}(i)} z_j \geq \sum_{j \in C} z_j$. \square (Claim 1)

CLAIM 2. The value of (y, z) is negative.

Proof of Claim 2. It follows from the definitions of y and z that

$$\sum_{i \in M} y_i = |M_S(L)| + \sum_{i \notin M_S(L)} \sum_{j \in \sigma^{-1}(i)} z_j \quad (21)$$

and

$$\sum_{j \in J} z_j = \left(\sum_{i \in M} \sum_{j \in \sigma^{-1}(i)} z_j \right) + z_{j_{\text{new}}} \quad (22)$$

Therefore the value of (y, z) is equal to

$$\sum_{i \in M} y_i - \sum_{j \in J} z_j = |M_S(L)| - \left(\sum_{i \in M_S(L)} \sum_{j \in \sigma^{-1}(i)} z_j \right) - z_{j_{\text{new}}} \quad (23)$$

Since $z_{j_{\text{new}}} > 0$, it is sufficient to show $\sum_{i \in M_S(L)} \sum_{j \in \sigma^{-1}(i)} z_j \geq |M_S(L)|$.

Let B_0, \dots, B_ℓ be the blockers of L and $B_t, t \in \{0, \dots, \ell\}$ a small blocker. Then B_t was added because a potential but invalid move (j_0, i_t) was chosen, and that move was either a potential small or a potential big-to-small move. Suppose that (j_0, i_t) was a small move. Since it was not valid, we have $p(j_0) + p(\sigma^{-1}(i_t)) > 1 + R$.

Case 1: i_t is assigned a big job. Then $p(\sigma^{-1}(i_t)) - b > 1 + R - b - p(j_0) = 1 - b - s + R$ and left hand side as well as $1 - b - s$ are multiples of s . Hence $p(\sigma^{-1}(i_t)) - b \geq 1 - b - s + R + \delta$. It follows that

$$\sum_{j \in \sigma^{-1}(i_t)} z_j = p(\sigma^{-1}(i_t)) - b + c \geq 1 - b - s + R + \delta + c \quad (24)$$

Case 2: i_t is not assigned a big job. Then

$$\sum_{j \in \sigma^{-1}(i_t)} z_j = p(\sigma^{-1}(i_t)) > 1 + R - s \quad (25)$$

If on the other hand (j_0, i_t) was a big-to-small move, then $p(\sigma^{-1}(i_t)) > 1 - b + R$. Again, since the left hand side and $1 - b$ are multiples of s , we have $p(\sigma^{-1}(i_t)) \geq 1 - b + R + \delta$. Therefore

$$\sum_{j \in \sigma^{-1}(i_t)} z_j = p(\sigma^{-1}(i_t)) \geq 1 - b + R + \delta \quad . \quad (26)$$

Note that the right hand sides of Eqs. (24) and (25) are at least 1, so only the blockers due to the big-to-small moves are a problem. Svensson [8] showed that if B_t is of this type, then B_{t+1} was added because of a small move. By pairing B_t and B_{t+1} we obtain

$$\sum_{j \in \sigma^{-1}(i_t)} z_j + \sum_{j \in \sigma^{-1}(i_{t+1})} z_j \geq 2R + 2\delta + c + 2(1 - b) - s \quad (27)$$

or

$$\sum_{j \in \sigma^{-1}(i_t)} z_j + \sum_{j \in \sigma^{-1}(i_{t+1})} z_j > 2 + 2R + \delta - b - s \quad . \quad (28)$$

In both cases (15) and (16) ensure that the right hand side is at least 2. Therefore $\sum_{i \in M_S(L)} \sum_{j \in \sigma^{-1}(i)} z_j \geq |M_S(L)|$. \square (Claim 2)

This concludes the proof of Lemma 2. \square

Note that we can obtain the termination of $A_{\frac{2}{3}+s}$ from Lemma 2 if we set $b = 1$, $R = \frac{2}{3} + s$ and $c = \frac{2}{3}$, but none of the constraints (14) to (16) are tight for these values. We now present values for R that satisfy the prerequisites of Lemma 2 and are best possible.

Lemma 3. *The following are the smallest values for R that satisfy the prerequisites of Lemma 2:*

- (i) $R = b - s$ if $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$ and
- (ii) $R = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil s$ otherwise.

Proof. We first show that the claimed values satisfy the prerequisites of Lemma 2.

In case of (i), set $R = b - s$ and define $k = \lfloor \frac{R}{s} \rfloor$ and $\delta = (k+1)s - R$. Note that $1 < \frac{R}{s} \leq \frac{3}{2}$ since $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$. Therefore $k = 1$ and $\delta = 3s - b$. Choose $c = R + \delta = 2s$. It is easily confirmed that $0 \leq c < b$ and Eqs. (14) to (16) are satisfied.

For (ii) we set $R = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil s$ and define $k = \lfloor \frac{R}{s} \rfloor$ and $\delta = (k + 1)s - R$ as before. Remember that for the processing times of the jobs it holds $0 < s < b \leq 1$. Obviously, we have $k = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil$ and $\delta = s$. Also note that

$$R = \left\lceil \frac{2}{3} \left(\frac{b}{s} - 1 \right) \right\rceil s \geq \frac{2}{3} \left(\frac{b}{s} - 1 \right) s = \frac{2}{3}(b - s) \quad . \quad (29)$$

Now set $c = \min\{b, R + \delta\}$. Then condition Eq. (14) is satisfied because

$$b - R + s - \delta = b - R \leq b - \frac{2}{3}(b - s) = \frac{1}{3}b + \frac{2}{3}s < \frac{2}{3}b + \frac{1}{3}s \quad (30)$$

$$= \frac{2}{3}(b - s) + s \leq R + s = R + \delta \quad (31)$$

and $b - R + s - \delta = b - R \leq 1 - R < 1$.

Considering Eq. (15) we have

$$2R + 2\delta + c + 2(1 - b) - s \geq \frac{4}{3}(b - s) + s + c + 2 - 2b = -\frac{2}{3}b - \frac{1}{3}s + c + 2 \quad , \quad (32)$$

and the latter is at least 2 if $c \geq \frac{2}{3}b + \frac{1}{3}s$ holds. This is true since $b > \frac{2}{3}b + \frac{1}{3}s$ and $R + \delta \geq \frac{2}{3}(b - s) + s = \frac{2}{3}b + \frac{1}{3}s$.

To finally prove Eq. (16), we consider three cases. Note that Eq. (16) simplifies to $R \geq \frac{1}{2}b$ since $\delta = s$.

Case 1: $\frac{s}{b} < \frac{1}{4}$. We have $R \geq \frac{2}{3}(b - s) > \frac{2}{3}(b - \frac{1}{4}b) = \frac{1}{2}b$.

Case 2: $\frac{1}{4} \leq \frac{s}{b} < \frac{2}{5}$. By the bounds on $\frac{s}{b}$ we get $k = 2$, so $R = 2s \geq 2\frac{1}{4}b = \frac{1}{2}b$.

Case 3: $\frac{s}{b} \geq \frac{1}{2}$. In this case $k = 1$ and $R = s \geq \frac{1}{2}b$.

Now we proof that no smaller values do so.

CLAIM 1. For any R that satisfies the prerequisites of Lemma 2, we have

- (i) $R \geq b - s$ if $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$ and
- (ii) $R \geq \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil s$ otherwise.

Let $R \geq s$, $0 \leq c \leq b$, $k = \lfloor \frac{R}{s} \rfloor$ and $\delta = (k + 1)s - R$ such that Eqs. (14) to (16) hold. For convenience we express $\delta = \gamma s$, i.e. we define $\gamma = \frac{\delta}{s} \in (0, 1]$. Then $R = (k + 1 - \gamma)s$.

To show (i) we consider two cases.

Case 1: $k = 1$. From Eq. (16) and $R = (k + 1 - \gamma)s = (2 - \gamma)s$ it follows that $\gamma \leq 3 - \frac{b}{s}$. Therefore

$$R = (2 - \gamma)s \geq \left(2 - \left(3 - \frac{b}{s}\right)\right)s = b - s \quad . \quad (33)$$

Case 2: $k \geq 2$. Since $\gamma \leq 1$ and $s \geq \frac{2b}{5} > \frac{b}{3}$ we have

$$R = (k + 1 - \gamma)s \geq ks \geq 2s = 3s - s > b - s \quad . \quad (34)$$

For (ii) consider that $R + \delta \geq c$ and Eq. (15) imply $R \geq \frac{2}{3}b + \frac{1}{3}s - \delta$. Inserting $R = (k + 1 - \gamma)s$ and $\delta = \gamma s$ yields $k \geq \frac{2}{3}\left(\frac{b}{s} - 1\right) \geq \left\lceil \frac{2}{3}\left(\frac{b}{s} - 1\right) \right\rceil$ since k is integral. The claim now follows by $R = (k + 1 - \gamma)s \geq ks$.

This concludes the proof of Lemma 3. And we get Theorem 2 \square

4 Improving the Bound by Scaling

In this section, we still demand that each configuration in the optimal solution contains at most one big job. The results of Section 3 only apply if additionally $\text{OPT}_{\text{LP}} - b$ is a multiple of s . To obtain a better bound and get rid of the latter restriction, we use a new scaling technique and consider several cases, depending various parameters such as the value of $\frac{s}{b}$. We denote the integrality gap of the configuration LP for the instance I by $IG(I)$ throughout this section. Assume in the following that the instance I satisfies $\frac{s}{b} < \frac{1}{3}$, otherwise we obtain that the integrality gap is at most $2 - \frac{s}{b} \leq \frac{5}{3}$ from Section 5. Define $k = \left\lceil \frac{2}{3}\left(\frac{b}{s} - 1\right) \right\rceil$ as in Section 3 so $IG(I) \leq 1 + R = 1 + ks$. Since $k \geq \frac{2}{3}\left(\frac{b}{s} - 1\right)$ we have $\frac{s}{b} \geq \frac{2}{3k+2}$. Similarly, $k - 1 < \frac{2}{3}\left(\frac{b}{s} - 1\right)$ and therefore $\frac{s}{b} < \frac{2}{3k-1}$. It follows that for k it holds $\frac{s}{b} \in \left\{\frac{2}{3k+2}, \frac{2}{3k-1}\right\}$. We now consider two main cases, namely whether $\text{OPT}_{\text{LP}}(I)$ is a multiple of s or not.

4.1 Case 1: $\text{OPT}_{\text{LP}}(I)$ is a multiple of s

The analysis in this case is split further into four cases.

Case 1.1: k is even or $\frac{s}{b} \geq \frac{2}{3k+1}$. We construct a modified instance I' of the restricted problem where each configuration may contain at most one big job. The instance I' arises from I by scaling b up to the next multiple b' of s . We know that $\frac{3k-1}{2}s < b \leq \frac{3k+2}{2}s$, therefore $b' = \frac{3k+1}{2}s$ if k is odd and $b' \in \{\frac{3k+2}{2}s, \frac{3k+0}{2}s\}$ if k is even. We have $\text{OPT}(I) \leq \text{OPT}(I')$ and $\text{OPT}_{\text{LP}}(I) = \text{OPT}_{\text{LP}}(I')$, therefore

$$IG(I) = \frac{\text{OPT}(I)}{\text{OPT}_{\text{LP}}(I)} \leq \frac{\text{OPT}(I')}{\text{OPT}_{\text{LP}}(I')} = IG(I') \quad . \quad (35)$$

On the other hand, $\text{OPT}_{\text{LP}}(I') - b'$ is a multiple of s and we can apply Theorem 2. Since $b' \in \{\frac{3k+2}{2}s, \frac{3k+1}{2}s, \frac{3k+0}{2}s\}$ we get $s \leq \frac{2b'}{3k}$, and this gives us $IG(I') \leq 1 + R = 1 + ks \leq 1 + k\frac{2b'}{3k} \leq \frac{5}{3}$, using $b' \leq 1$.

Case 1.2: k is odd and $\frac{2}{3k+\frac{7}{5}} \leq \frac{s}{b} < \frac{2}{3k+1}$. This time, we construct I' by scaling the length of small jobs up to $s' = \frac{2b}{3k+1}$, i.e. let I' be an instance of the restricted problem where each configuration may contain at most one big job. In particular, I' should be identical to I except that small jobs have processing time $s' = \frac{2b}{3k+1}$. Since $s \leq s'$ we have $\text{OPT}(I) \leq \text{OPT}(I')$. Define $\alpha = \frac{s'}{s} = \frac{2b}{(3k+1)s}$. We next prove that $\text{OPT}_{\text{LP}}(I') \leq \alpha \text{OPT}_{\text{LP}}(I)$. Let $T = \text{OPT}_{\text{LP}}(I)$ and consider a feasible solution x of $\text{CLP}(T, I)$. Then x is also a feasible solution of $\text{CLP}(\alpha T, I')$, since the processing time of each configuration increases at most by factor α in the modified instance I' . We therefore have

$$IG(I) = \frac{\text{OPT}(I)}{\text{OPT}_{\text{LP}}(I)} \leq \frac{\text{OPT}(I')}{\text{OPT}_{\text{LP}}(I')^{\frac{1}{\alpha}}} \leq \alpha IG(I') \quad . \quad (36)$$

To apply Theorem 2 we have to scale I' by $\beta = \frac{1}{\text{OPT}_{\text{LP}}(I')} \leq 1$ to I'' such that $\text{OPT}_{\text{LP}}(I'') = 1$. In I'' the processing times are $b'' = b \cdot \beta$ and $s'' = s' \cdot \beta \leq s'$. We now can compute

$$IG(I') = \frac{\text{OPT}(I')}{\text{OPT}_{\text{LP}}(I')} = \frac{\beta \cdot \text{OPT}(I')}{\beta \cdot \text{OPT}_{\text{LP}}(I')} = \frac{\text{OPT}(I'')}{\text{OPT}_{\text{LP}}(I'')} = IG(I'') \quad (37)$$

Since $b'' = \frac{3k+1}{2} \cdot s''$ we have $s'' \mid b''$. In both cases that $\text{OPT}_{\text{LP}}(I'')$ is a multiple of s'' and in the case that $\text{OPT}_{\text{LP}}(I'') = b'' + ys''$ for $y \in \mathbb{N}_0$ it holds $s \mid \text{OPT}_{\text{LP}}(I'')$. Now we can apply Theorem 2 and get

$$\begin{aligned}
IG(I) &\leq \alpha IG(I') = \alpha IG(I'') \leq \alpha(1 + ks'') \leq \alpha(1 + ks') \\
&\leq \frac{3k + \frac{7}{5}}{3k + 1} \left(1 + \frac{2bk}{3k + 1}\right) \leq \frac{3k + \frac{7}{5}}{3k + 1} \left(1 + \frac{2k}{3k + 1}\right) \\
&= \frac{15k^2 + 10k + \frac{7}{5}}{9k^2 + 6k + 1} = \frac{5k^2 + \frac{2}{3}k + \frac{7}{75}}{3k^2 + \frac{2}{3}k + \frac{1}{9}} < \frac{5k^2 + \frac{2}{3}k + \frac{1}{9}}{3k^2 + \frac{2}{3}k + \frac{1}{9}} \quad (38) \\
&= \frac{5}{3} \quad .
\end{aligned}$$

Case 1.3: k is odd, $\frac{2}{3k+2} \leq \frac{s}{b} < \frac{2}{3k+\frac{7}{5}}$, and $\text{OPT}(I)$ is a multiple of s . We scale b down to $b' = \frac{3k+1}{2}s$. Note that $b \leq \frac{3k+2}{2}s = b' + \frac{1}{2}s$ and $\text{OPT}(I') \leq \text{OPT}(I)$. Assume $\text{OPT}(I') < \text{OPT}(I)$. Since $s \mid \text{OPT}(I)$ and $s \mid b'$, we have $s \mid \text{OPT}(I')$. Therefore it holds $\text{OPT}(I') \leq \text{OPT}(I) - s$. Take an optimal solution of I' and scale it to a solution of I : Remember we scaled only the big jobs, therefore configurations with no big jobs have no changes. Since $b \leq b' + \frac{1}{2}s$ configurations with big jobs grow at most $\frac{1}{2}s$. It follows that we have a solution for I with length at most $\text{OPT}(I') + \frac{1}{2}s \leq \text{OPT}(I) - s + \frac{1}{2}s < \text{OPT}(I)$. This is a contradiction and it holds $\text{OPT}(I') = \text{OPT}(I)$. The same argumentation holds for $\text{OPT}_{\text{LP}}(I)$ and $\text{OPT}_{\text{LP}}(I')$ therefore $\text{OPT}_{\text{LP}}(I) = \text{OPT}_{\text{LP}}(I')$. Applying Theorem 2 to I' we get

$$IG(I) = IG(I') \leq 1 + ks \leq 1 + k \frac{2b}{3k + \frac{7}{5}} \leq \frac{5k + \frac{7}{5}}{3k + \frac{7}{5}} < \frac{5}{3} \quad . \quad (39)$$

Case 1.4: k is odd, $\frac{2}{3k+2} \leq \frac{s}{b} < \frac{2}{3k+\frac{7}{5}}$, and $\text{OPT}(I) = b + ys$ for $y \in \mathbb{N}$. Again, we scale b down to $b' = \frac{3k+1}{2}s$ and again it holds that $b \leq \frac{3k+2}{2}s = b' + \frac{1}{2}s$ and $\text{OPT}(I') \leq \text{OPT}(I)$. If we scale $\text{OPT}(I)$ we get a solution with length l' for I' . It holds $\text{OPT}(I) - \frac{1}{2} \cdot s \leq l' \leq \text{OPT}(I)$. Assume $\text{OPT}(I') < \text{OPT}(I) - \frac{1}{2}s$. Again we take an optimal solution of I' and scale it to a solution of I . Again only configurations with big jobs grow at most $\frac{1}{2}s$. We have a solution with length $\text{OPT}(I') + \frac{1}{2} \cdot s < \text{OPT}(I)$ which is a contradiction. Therefore $\text{OPT}(I') \geq \text{OPT}(I) - \frac{1}{2}s$. Because $\text{OPT}_{\text{LP}}(I)$ is a multiple of s as above it holds $\text{OPT}_{\text{LP}}(I) = \text{OPT}_{\text{LP}}(I')$. Applying Theorem 2

to I' we get

$$\begin{aligned} IG(I) &= \frac{\text{OPT}(I)}{\text{OPT}_{\text{LP}}(I)} \leq \frac{\text{OPT}(I') + \frac{1}{2}s}{\text{OPT}_{\text{LP}}(I')} \leq IG(I') + \frac{1}{2}s \leq 1 + ks + \frac{1}{2}s \\ &= 1 + \left(k + \frac{1}{2}\right)s \leq 1 + \left(k + \frac{1}{2}\right) \frac{2b}{3k + \frac{7}{5}} \leq \frac{5k + \frac{12}{5}}{3k + \frac{7}{5}} . \end{aligned} \tag{40}$$

For $k \geq 3$ this term attains its maximum $\frac{87}{52} = \frac{5}{3} + \frac{1}{156}$ at $k = 3$.

Therefore we have proven that the integrality gap is at most $\frac{5}{3} + \frac{1}{156}$ if $\text{OPT}_{\text{LP}}(I)$ is a multiple of s .

4.2 Case 2: $\text{OPT}_{\text{LP}}(I)$ is not a multiple of s

In this case, the longest configuration of the optimal fractional solution contains one big job, i.e there is $x \in \mathbb{Z}_{\geq 0}$ such that $\text{OPT}_{\text{LP}}(I) = b + xs$. Then Theorem 2 yields

$$IG(I) \leq 1 + ks \quad . \tag{41}$$

We can prove another bound by scaling s . We look at the bound from Section 3 (compare Fig. 1). Note that for $\frac{s}{b} \in [\frac{2}{3k+2}, \frac{2}{3k-1})$ the integrality gap increases as $\frac{s}{b}$ approaches $\frac{2}{3k-1}$ and jumps down again at $\frac{s}{b} = \frac{2}{3k-1}$. Therefore, if $\frac{s}{b}$ is slightly below $\frac{2}{3k-1}$, we can increase the processing time s of small jobs to s' such that $\frac{s'}{b} = \frac{2}{3k-1}$. Define $s_0 = -\frac{1}{2k} + \sqrt{\frac{1}{4k^2} + \frac{2b}{(3k-1)k} \cdot \left(1 + \frac{(k-1)2b}{3k-1}\right)}$. In Lemma 5 we will show, that scaling for all $s \geq s_0$ is beneficial.

Define I' accordingly and let $\alpha = \frac{s'}{s} = \frac{2b}{(3k-1)s}$. As in Section 4.1 we have

$$IG(I) \leq \alpha IG(I') \quad , \tag{42}$$

see also Eq. (36). We want to apply Theorem 2 to I' and therefore need to show that the prerequisites hold. In particular, we show that $\text{OPT}_{\text{LP}}(I') - b$ is a multiple of s' :

Lemma 4. $\text{OPT}_{\text{LP}}(I') = b + xs'$ if we scale $s \geq s_0$ to $s' = \frac{2b}{3k-1}$.

In the proof of Lemma 4 we will use that $s_0 > \frac{2b}{3k}$. We first give a proof of this: Remember that we assumed in Section 4 that $\frac{s}{b} < \frac{1}{3}$

and used Section 5 otherwise. From $\frac{s}{b} < \frac{1}{3}$ and the definition $k = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil$ it follows that $k \geq 2$. Using further that $\frac{1}{2} \leq b \leq 1$, it follows that

$$\begin{aligned}
& 48bk - 48b^2k + 24bk - 24b - 16b^2 > 0 \\
& \Leftrightarrow k^2(81 - 81 + 216b - 216b + 144b^2 - 144b^2) \\
& + k(-54 + 54 - 144b^2 + 144b - 72b + 96b^2) - 24b - 16b^2 > 0 \\
& \Leftrightarrow 81k^2 - 54k + 9 + 216bk^2 - 72bk + 144b^2k^2 - 144b^2k \\
& > 144b^2k^2 - 96b^2k + 16b^2 + 216bk^2 - 144bk + 24b + 81k^2 - 54k + 9 \\
& \Leftrightarrow 9(3k - 1)^2 + 72bk(3k - 1 + 2bk - 2b) > (16b^2 + 24b + 9)(3k - 1)^2 \\
& \Leftrightarrow \frac{1}{4} + \frac{2bk}{3k - 1} \left(\frac{3k - 1 + 2bk - 2b}{3k - 1} \right) > \frac{1}{36}(16b^2 + 24b + 9) \\
& \Leftrightarrow \sqrt{\frac{1}{4k^2} + \frac{2b}{(3k - 1)k} \left(1 + \frac{(k - 1)2b}{3k - 1} \right)} > \frac{4b + 3}{6k} \\
& \Leftrightarrow \sqrt{\frac{1}{4k^2} + \frac{2b}{(3k - 1)k} \left(1 + \frac{(k - 1)2b}{3k - 1} \right)} > \frac{2b}{3k} + \frac{1}{2k} \\
& \Leftrightarrow s_0 > \frac{2b}{3k}
\end{aligned} \tag{43}$$

Proof. Consider an optimal fractional solution to I . We show that this is a solution to I' with makespan at most $b + xs'$. Let C be any configuration that occurs in this solution. Denote by ℓ and ℓ' the length of C when small jobs have length s and s' , respectively. Then $\ell \leq b + xs$. If C contains a big job, we have $\ell = b + ys$ for some $y \leq x$. It follows that $\ell' = b + ys' \leq b + xs'$. Otherwise, C only contains small jobs and $\ell = ys$ for some $y \in \mathbb{N}$. Define $z = y - x \in \mathbb{Z}$. Since $\text{OPT}_{\text{LP}}(I)$ is not a multiple of s , we have $zs + xs = \ell < \text{OPT}_{\text{LP}}(I) = b + xs$ and therefore $zs < b$. This implies $z < \frac{b}{s}$. Remember that $s \geq s_0 > \frac{2b}{3k}$, so $\frac{s}{b} > \frac{2}{3k}$. This implies $z < \frac{b}{s} < \frac{3k}{2}$. Since z is integral, we also have $z \leq \frac{3k-1}{2}$. It follows that

$$\ell' = ys' = zs' + xs' \leq \frac{3k - 1}{2} \frac{2b}{3k - 1} + xs' = b + xs' \quad . \tag{44}$$

Therefore, $\text{OPT}_{\text{LP}}(I') \leq b + xs'$.

Now assume that there is a fractional solution for I' with makespan less than $b + xs'$. We will show that this implies that $\text{OPT}_{\text{LP}}(I) < b + xs$, a contradiction. Let C be any configuration occurring in the optimal solution and define ℓ and ℓ' as before. Then $\ell' < b + xs'$. If C contains a big job, we have $\ell = b + ys$ for some $y \in \mathbb{Z}_{\geq 0}$. We have $b + ys' = \ell' < b + xs'$, thus $y < x$ and $\ell = b + ys < b + xs$. Otherwise, C contains only small jobs and $\ell = ys$ for some $y \in \mathbb{N}$. Define again $z = y - x \in \mathbb{Z}$. Then $zs' + xs' = \ell' < b + xs'$, therefore $zs' < b$. This implies $\ell = ys = zs + xs < zs' + xs < b + xs$. \square

We have $k - 1 = \lceil \frac{2}{3} \frac{b}{s'} - 1 \rceil$. Also $\frac{s}{b} < \frac{1}{3}$ implies $k \geq 2$ and thus $\frac{s'}{b} = \frac{2}{3k-1} \leq \frac{2}{5}$. Again, we have to scale the processing times by $\frac{1}{\text{OPT}_{\text{LP}}(I')}$ to apply Theorem 2. As above we have $IG(I') = IG(I'')$ and can compute

$$IG(I) \leq \alpha IG(I') \leq \alpha(1 + (k - 1)s'') \leq \alpha(1 + (k - 1)s') \quad . \quad (45)$$

We will determine which of the two bounds (41) and (45) is better depending on the values of s and b .

Lemma 5. *Let I be the original instance and I' the modified instance. If $s \geq s_0$ we have $1 + ks \geq \alpha(1 + (k - 1)s')$.*

Proof. We have $s' = \frac{2b}{3k-1}$ and $\alpha = \frac{s'}{s}$ and therefore

$$\begin{aligned} 1 + ks &\geq \alpha(1 + (k - 1)s') \\ \Leftrightarrow 1 + ks &\geq \frac{2b}{(3k - 1)s} \cdot \left(1 + \frac{(k - 1)2b}{3k - 1}\right) \\ \Leftrightarrow s^2 + \frac{1}{k}s - \frac{2b}{(3k - 1)k} \left(1 + \frac{(k - 1)2b}{3k - 1}\right) &\geq 0 \quad . \end{aligned} \quad (46)$$

It is easily verified that s_0 is the only nonnegative root of the last inequation. \square

It turns out that Eqs. (41) and (45) can be combined if b is not too large.

Lemma 6. *If $\frac{s}{b} < \frac{1}{3}$, $\text{OPT}_{\text{LP}}(I) - b$ is a multiple of s , and $b \leq \frac{80}{81}$, the integrality gap of the CLP is at most $\frac{5}{3}$.*

Proof. We consider two cases.

Case 1: $s \leq s_0$. Then the integrality gap is at most $1 + ks \leq 1 + ks_0$ by Eq. (41).

Case 2: $s > s_0$. Remember that we scaled the processing time of small jobs to $s' = \alpha s = \frac{2b}{3k-1}$. One can easily see that the term

$$\alpha(1+(k-1)s') = \frac{2b}{(3k-1)s} \left(1 + \frac{(k-1)2b}{3k-1} \right) = \frac{1}{s} \left(\frac{2b}{3k-1} + \frac{(k-1)4b^2}{(3k-1)^2} \right) \quad (47)$$

is monotonically decreasing with respect to s . From Lemma 5 and $\alpha = \frac{s'}{s}$ we also know that $\frac{s'}{s_0}(1+(k-1)s') \leq 1 + ks_0$. We therefore have that the integrality gap is bounded by $\alpha(1+(k-1)s') \leq 1 + ks_0$ for all $s > s_0$.

For both cases we can compute

$$\begin{aligned} 1 + ks_0 &= 1 - \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{2bk}{(3k-1)} \cdot \left(1 + \frac{(k-1)2b}{3k-1} \right)} \\ &= \frac{1}{2} + \frac{1}{2(3k-1)} \cdot \sqrt{(3k-1)^2 + 8bk(3k-1 + (k-1)2b)} \quad . \end{aligned} \quad (48)$$

The last term attains its maximum $\frac{1}{4} \cdot (2 + \sqrt{18b+4})$ at $k = 2$. It is easy to verify that this is at most $\frac{5}{3}$ as long as $b \leq \frac{80}{81} \approx 0.988$. \square

In the remaining case that $b > \frac{80}{81}$, we scale the processing time of small jobs to $s' = \alpha s = \frac{2b}{3k-1}$ if $s > \frac{2b}{3k}$.

Lemma 7. *If $\frac{s}{b} < \frac{1}{3}$, $\text{OPT}_{\text{LP}}(I) - b$ is a multiple of s , and $b > \frac{80}{81}$, the integrality gap of the CLP is at most $\frac{5}{3}$.*

Proof. In our analysis, we again distinguish the two cases whether s is rounded or not. Remember that $\text{OPT}_{\text{LP}} = b + xs$ for $x \in \mathbb{N}$ and $s' = \frac{2b}{3k-1}$.

Case 1: $s \leq \frac{2b}{3k}$. We can directly apply Theorem 2 and obtain the bound

$$IG(I) \leq 1 + ks \leq 1 + k \frac{2b}{3k} \leq \frac{5}{3} \quad . \quad (49)$$

Case 2: $s > \frac{2b}{3k}$. We will first prove that $\text{OPT}_{\text{LP}}(I') \leq 1 + \frac{1}{79}s$.

Case 2.1: $x = 0$. Then $1 = \text{OPT}_{\text{LP}}(I) = b$. Consider any configuration C that occurs in the optimal fractional solution of I . Denote by ℓ and ℓ' the length of C when small jobs have length s and s' , respectively. Obviously, $\ell \leq \text{OPT}_{\text{LP}}(I) = b$. Either C contains a big job, which means that $\ell' = \ell = b$, or C contains only small jobs. In the latter case, $\ell = ys$ for some $y \in \mathbb{N}$. Since $\ell \leq \text{OPT}_{\text{LP}}(I) = b$, we have $y \leq \frac{b}{s} < \frac{3k}{2}$. Because y is integral, we even have $y \leq \frac{3k-1}{2}$. Now

$$\ell' = ys' \leq \frac{3k-1}{2} \cdot \frac{2b}{3k-1} = b = 1 \quad . \quad (50)$$

This means that the same solution, when used for I' , also has makespan b , i.e. $\text{OPT}_{\text{LP}}(I') \leq 1$.

Case 2.2: $x > 0$. Consider that $1 = \text{OPT}_{\text{LP}}(I) = b + xs$, therefore $xs < \frac{1}{81}$ and also $s < \frac{1}{81}$. Then, using $\alpha = \frac{s'}{s}$, we have

$$\frac{b}{\alpha} = \frac{3k-1}{2}s = \frac{3k+1}{2}s - s > \frac{3k+1}{2} \frac{2b}{3k} - s = \frac{3k+1}{3k}b - s > b - s \quad , \quad (51)$$

and consequently

$$\alpha < \frac{b}{b-s} = 1 + \frac{s}{b-s} \leq 1 + \frac{81}{79}s \quad . \quad (52)$$

Again, consider an arbitrary configuration C occurring in the optimal fractional solution of I and its lengths ℓ and ℓ' . If C contains a big job, then $\ell = b + ys$ for some $y \in \mathbb{Z}_{\geq 0}$, and $\ell \leq \text{OPT}_{\text{LP}}(I)$. It follows that $\ell' = b + ys' \leq b + xs'$ and

$$\begin{aligned} b + xs' &= b + x\alpha s < b + \left(1 + \frac{81}{79}s\right)xs \\ &= b + xs + \frac{81}{79}xs^2 < b + xs + \frac{81}{79}s \frac{1}{81} = 1 + \frac{1}{79}s \quad . \end{aligned} \quad (53)$$

Otherwise, C only contains small jobs and $\ell = ys$ for some $y \in \mathbb{N}$. Combining Eqs. (44) and (53), we see that $\ell' < 1 + \frac{1}{79}s$.

This concludes our proof that $\text{OPT}_{\text{LP}}(I') \leq 1 + \frac{1}{79}s$. Similar as in Eq. (42), we find that $IG(I) \leq \left(1 + \frac{1}{79}s\right)IG(I')$. As above we have to scale I' to I'' with $\text{OPT}_{\text{LP}}(I'') = 1$. Applying Theorem 2 to I'' and

using $s < \frac{2b}{3k-1}$ and $b \leq 1$, we obtain

$$\begin{aligned}
IG(I) &\leq \left(1 + \frac{1}{79}s\right)IG(I') \leq \left(1 + \frac{1}{79}s\right)IG(I'') \\
&\leq \left(1 + \frac{1}{79}s\right)\left(1 + (k-1)\frac{s'}{\text{OPT}_{\text{LP}}(I')}\right) \\
&\leq \left(1 + \frac{1}{79}s\right)(1 + (k-1)s') \tag{54} \\
&< \left(1 + \frac{2}{79(3k-1)}\right)\left(1 + \frac{2(k-1)}{3k-1}\right) \\
&= \frac{15k^2 - \frac{1096}{79}k + \frac{231}{79}}{9k^2 - 6k + 1}.
\end{aligned}$$

The last term can be seen to be monotonically increasing for $k \geq 1$ and has the limit $\frac{5}{3}$. \square

With both Lemmas 6 and 7, we have proven that the integrality gap is at most $\frac{5}{3}$ if $\text{OPT}_{\text{LP}}(I)$ is not a multiple of s .

5 An $(\text{OPT} + b - s)$ -Approximation

In this section we present an algorithm for the restricted assignment problem with two different processing times $s < b$, and in consequence bound the integrality gap of the CLP. W.l.o.g. we consider s and b to be integers. Our algorithm depends on a result by Shmoys and Tardos [7] for a variant of unrelated scheduling with costs. Their algorithm is based on solving and rounding the assignment LP $\text{ALP}(T)$. We strengthen the LP relaxation with additional constraints, apply their rounding procedure, and provide an improved analysis. In the following OPT_{LP} denotes the optimal makespan of the strengthened ALP.

Theorem 3. *For the restricted assignment problem with two different processing times $s < b$, there is a polynomial time approximation algorithm that produces a schedule of length at most $\min\{\text{OPT}_{\text{LP}} + b, \lfloor \text{OPT}_{\text{LP}}/s \rfloor s + \lfloor \text{OPT}_{\text{LP}}/b \rfloor (b - s)\}$ in general, yielding a bound of $\text{OPT}_{\text{LP}} + (b - s)$ for the case $b \leq \text{OPT} < 2b$. Furthermore the algorithm has a multiplicative performance guarantee of $(2 - \frac{s}{b})$ in general.*

The first bound of $\text{OPT}_{\text{LP}} + b$ is due to the analysis by Shmoys and Tardos, which still holds for our algorithm. The second bound of $\lfloor \text{OPT}_{\text{LP}}/s \rfloor s + \lfloor \text{OPT}_{\text{LP}}/b \rfloor (b - s)$ can be used to show the remaining ones: Since $b \leq \text{OPT} < 2b$ implies $\lfloor \text{OPT}_{\text{LP}}/b \rfloor \leq 1$, the second bound yields the additive guarantee in that case. Furthermore it holds that $\lfloor \text{OPT}_{\text{LP}}/b \rfloor (b - s) + \lfloor \text{OPT}_{\text{LP}}/s \rfloor s \leq (2 - s/b)\text{OPT}_{\text{LP}}$. Therefore we can also guarantee the multiplicative bound of $(2 - s/b)$ matching the one by [2]. However, note that the respective algorithm provides no additive guarantee. We will prove the second bound in the following.

Consider the ALP(T), the assignment LP for a given makespan T . For every machine i , let $a_{iq} = \sum_{j:p_{ij}=q} x_{ij}$ for $q \in \{s, b\}$ denote the fractional number of jobs of size q scheduled on i . We add two additional constraints and denote the resulting LP by $\text{ALP}^+(T)$:

$$a_{ib} \leq \left\lfloor \frac{T}{b} \right\rfloor \quad \forall i \in M \quad (55)$$

$$a_{is} \leq \left\lfloor \frac{T}{s} \right\rfloor - \left\lfloor \frac{b}{s} \right\rfloor a_{ib} \quad \forall i \in M \quad (56)$$

Both constraints will always be satisfied by integer solutions of the LP: There can be at most $\lfloor T/b \rfloor$ big jobs on every machine yielding (55), and if there are a_{ib} big jobs on machine i there can be at most $\lfloor (T - a_{ib}b)/s \rfloor$ small jobs. The second condition does not give a linear constraint and has to be rounded using basic observations concerning the floor function. Note that:

$$\forall x, y \in \mathbb{R} : \lfloor x - y \rfloor \leq \lfloor x \rfloor - \lfloor y \rfloor, \quad (57)$$

$$\text{and } \forall x, y \in \mathbb{R}_{\geq 0} : \lfloor xy \rfloor \leq \lfloor x \rfloor \lfloor y \rfloor \quad (58)$$

Applying this we get $\lfloor (T - a_{ib}b)/s \rfloor \leq \lfloor T/s \rfloor - \lfloor a_{ib}b/s \rfloor \leq \lfloor T/s \rfloor - a_{ib} \lfloor b/s \rfloor$ for $a_{ib} \in \mathbb{Z}_{>0}$.

If there is no solution to the LP, then there is no integral solution in particular. However, we will show that if there is a solution, a schedule with makespan at most $\lfloor T/s \rfloor s + \lfloor T/b \rfloor (b - s)$ can be obtained via the rounding procedure due to Shmoys and Tardos. Using binary search, this yields the desired algorithm—a technique widely used, e.g. in [3,5,7].

The rounding procedure consists of constructing a bipartite graph that is weighted on the edges (see also figure 2), and finding a

matching for this graph. We give a short description of the procedure (for details we refer to [7]). For every machine i let $a_i = \sum_j x_{ij}$ be the fractional number of jobs scheduled on i , and $k_i := \lceil a_i \rceil$. The bipartite graph is constructed as follows:

- (a) For every job there is one job-node.
- (b) For every machine there are k_i machine-nodes.
- (c) For every $x_{ij} > 0$ there are one or in some cases two edges with (summed up) weight x_{ij} , and incident to the corresponding job-node and one of the corresponding machine-nodes.
- (d) For a machine i the summed up weight of the edges incident to the machine-nodes equals a_i , and all but one node are incident to edges with summed up weight 1. Because of this, some edges have to be split.
- (e) For every machine the machine-nodes are sorted, so that the smallest (in terms of processing time) job-node adjacent to a machine-node is at least as big as the biggest job-node adjacent to the next machine-node.

The weights on the edges give a *fractional* matching, that exactly matches all of the job-nodes, due to the constraint (1). If there is a fractional matching, that exactly matches all the job-nodes, there is also a proper matching, that matches all the job nodes. Such a matching can be found in polynomial time and yields a schedule by assigning job j to machine i , if the corresponding job-node is matched with one of the corresponding machine-nodes.

Lemma 8. *Given an instance of the restricted assignment problem with two different processing times $s < b$ and a target makespan T , the rounding procedure by Shmoys and Tardos applied to $ALP^+(T)$, yields a schedule with the following properties:*

- (i) *There are at most k_i jobs scheduled on machine i .*
- (ii) *On every machine at most $\lfloor T/b \rfloor$ big jobs are scheduled.*
- (iii) *Every machine receives jobs with summed up processing time at most $\lfloor T/b \rfloor(b - s) + \lfloor T/s \rfloor s$.*

Proof. (i) follows directly from (b). Because of (e) all the job nodes corresponding to big jobs and adjacent to machine nodes of a machine i , are adjacent to consecutive machine nodes, and due to (d) and

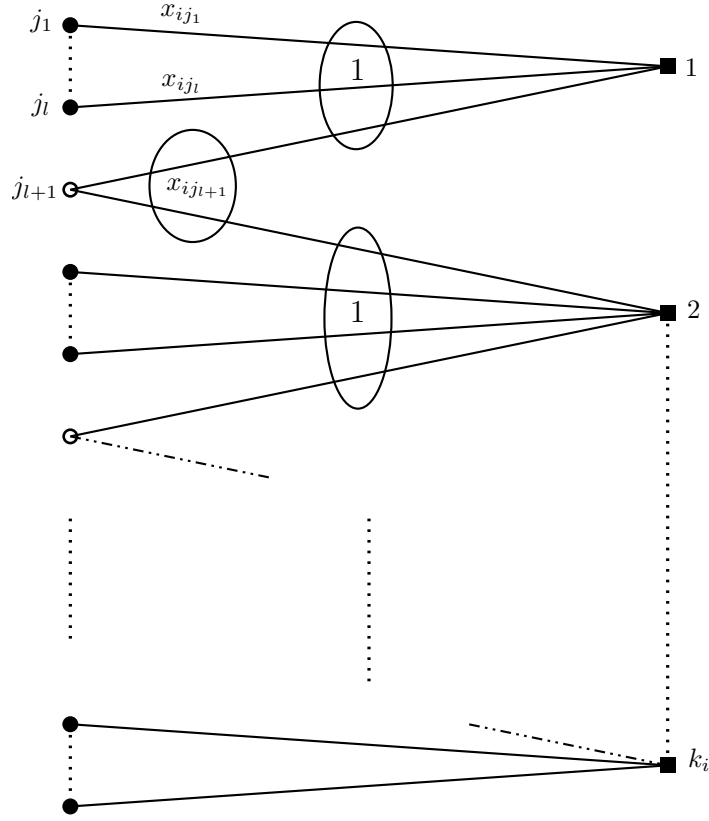


Figure 2. A sketch illustrating the rounding by Shmoys and Tardos for a fixed machine i . The values inside the ellipses are summed-up edge weights.

the constraint (55), there can be at most $\lfloor T/b \rfloor$ such machine-nodes, yielding (ii).

To proof (iii) we consider the number of small jobs k_{is} and the number of big jobs k_{ib} scheduled on machine i after the rounding. Note that in this notation we get $k_{ib} + k_{is} \leq k_i = \lceil a_i \rceil = \lceil a_{ib} + a_{is} \rceil$ due to (i), and $k_{ib} \leq \lfloor T/b \rfloor$ due to (ii). Furthermore, the makespan of i equals $k_{ib}b + k_{is}s$. By application of (56) we get:

$$k_i = \lceil a_{ib} + a_{is} \rceil \leq \left\lceil \left\lfloor \frac{T}{s} \right\rfloor + \underbrace{\left(1 - \left\lfloor \frac{b}{s} \right\rfloor\right)}_{\leq 0} a_{ib} \right\rceil \leq \left\lceil \left\lfloor \frac{T}{s} \right\rfloor \right\rceil = \left\lfloor \frac{T}{s} \right\rfloor$$

Hence:

$$k_{ib}b + k_{is}s \leq k_{ib}b + (k_i - k_{ib})s = k_{ib}(b - s) + k_is \leq \left\lfloor \frac{T}{b} \right\rfloor (b - s) + \left\lfloor \frac{T}{s} \right\rfloor s$$

□

As a corollary we can bound the integrality gap of the configuration LP.

Corollary 1. *The described algorithm can be modified to work with the CLP yielding the same bounds. In particular, if an instance of the restricted assignment problem has only two different processing times $s < b$, the integrality gap of the CLP is at most $2 - \frac{s}{b}$.*

Proof. Given a solution of the CLP, one can derive a solution of the ALP by setting $x_{ij} := \sum_{C:j \in C} x_{i,C}$. We will show that this solution satisfies the constraints (55) and (56), thus proving the corollary. For every $i \in M$, every configuration $C \in \mathcal{C}(i, T)$ corresponds to a feasible schedule of machine i . Hence $|\{j \in C | p_{ij} = b\}| \leq \lfloor T/b \rfloor$ and

$$\begin{aligned} |\{j \in C | p_{ij} = s\}| &\leq \left\lfloor \frac{1}{s} \left(T - b |\{j \in C | p_{ij} = b\}| \right) \right\rfloor \\ &\stackrel{(58)}{\leq} \left\lfloor \frac{T}{s} \right\rfloor - \left\lfloor \frac{b}{s} \right\rfloor |\{j \in C | p_{ij} = b\}| \end{aligned}$$

Therefore:

$$a_{ib} = \sum_{j:p_{ij}=b} x_{i,j} = \sum_{j:p_{ij}=b} \sum_{C:j \in C} x_{i,C} \quad (59)$$

$$= \sum_{C \in \mathcal{C}(i,T)} x_{i,C} |\{j \in C | p_{ij} = b\}| \leq \left\lfloor \frac{T}{b} \right\rfloor \sum_{C \in \mathcal{C}(i,T)} x_{i,C} \leq \left\lfloor \frac{T}{b} \right\rfloor \quad (60)$$

$$a_{is} = \sum_{j:p_{ij}=s} x_{i,j} = \sum_{j:p_{ij}=s} \sum_{C:j \in C} x_{i,C} \quad (61)$$

$$= \sum_{C \in \mathcal{C}(i,T)} x_{i,C} |\{j \in C | p_{ij} = s\}| \quad (62)$$

$$\leq \sum_{C \in \mathcal{C}(i,T)} x_{i,C} \left(\left\lfloor \frac{T}{s} \right\rfloor - \left\lfloor \frac{b}{s} \right\rfloor |\{j \in C | p_{ij} = b\}| \right) \quad (63)$$

$$= \left\lfloor \frac{T}{s} \right\rfloor \sum_{C \in \mathcal{C}(i,T)} x_{i,C} - \left\lfloor \frac{b}{s} \right\rfloor a_{i,b} \leq \left\lfloor \frac{T}{s} \right\rfloor - a_{i,b} \left\lfloor \frac{b}{s} \right\rfloor \quad (64)$$

□

Note that this can also be obtained using Lemma 8 and a more general argument due to Verschae and Wiese [9].

References

1. Bansal, N., Sviridenko, M.: The santa claus problem. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing, (STOC 2006). pp. 31–40 (2006)
2. Chakrabarty, D., Khanna, S., Li, S.: On $(1, \epsilon)$ -restricted assignment makespan minimization. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015). pp. 1087–1101 (2015)
3. Ebenlendr, T., Krčál, M., Sgall, J.: Graph balancing: A special case of scheduling unrelated parallel machines. *Algorithmica* 68(1), 62–80 (2014)
4. Kolliopoulos, S.G., Moysoglou, Y.: The 2-valued case of makespan minimization with assignment constraints. *Information Processing Letters* 113(1), 39–43 (2013)
5. Lenstra, J.K., Shmoys, D.B., Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming* 46(1-3), 259–271 (1990)
6. Shchepin, E.V., Vakhania, N.: An optimal rounding gives a better approximation for scheduling unrelated machines. *Operations Research Letters* 33(2), 127–133 (2005)
7. Shmoys, D.B., Tardos, É.: An approximation algorithm for the generalized assignment problem. *Mathematical Programming* 62(1-3), 461–474 (1993)
8. Svensson, O.: Santa claus schedules jobs on unrelated machines. In: Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC 2011). pp. 617–626 (2011)
9. Verschae, J., Wiese, A.: On the configuration-lp for scheduling on unrelated machines. In: *Algorithms-ESA 2011*, pp. 530–542. Springer (2011)