

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

Klaus Jansen<sup>1</sup>, Kati Land<sup>\*2</sup>, and Marten Maack<sup>3</sup>

- 1 Institute of Computer Science, University of Kiel, Kiel, Germany  
kj@informatik.uni-kiel.de
- 2 Institute of Computer Science, University of Kiel, Kiel, Germany  
kla@informatik.uni-kiel.de
- 3 Institute of Computer Science, University of Kiel, Kiel, Germany  
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  $\frac{5}{3} \approx 1.667$  for this problem. This allows us to estimate the optimal makespan within a factor of  $\frac{5}{3}$ , improving upon the previously best known estimation algorithm with ratio  $\frac{11}{6} \approx 1.833$  due to Chakrabarty, Khanna, and Li [2].

**1998 ACM Subject Classification** F.2.2 Sequencing and Scheduling, G.1.6 Optimization

**Keywords and phrases** unrelated scheduling, restricted assignment, configuration LP, integrality gap, estimation algorithm

**Digital Object Identifier** 10.4230/LIPIcs.SWAT.2016.24

## 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 [6] proved that it is NP-hard to approximate the makespan with a factor less than 1.5. On the other hand, they presented an algorithm with approximation ratio 2. The algorithm uses a rounding procedure for the following, natural linear programming formulation, which is commonly known as the *assignment linear program (LP)*:

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

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

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

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

Here,  $T$  is the desired makespan. We denote the above LP by  $ALP(T)$ . It is clear that there is a schedule with makespan  $T$  if and only if  $ALP(T)$  has an integral solution. Note that equation 3 strengthens the otherwise intuitive formulation by forbidding fractional assignments if the whole job cannot be feasibly processed on a machine.

---

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



Closing this gap between approximability and inapproximability is a major open problem in scheduling theory. Since 1990, the approximation ratio was slightly improved to  $2 - 1/|M|$  [7]. Because no substantial progress 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 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 [9], who considered another, stronger LP formulation, the *configuration LP*. 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

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

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

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

The first constraint enforces that at most one configuration is assigned to each machine, and the second constraint ensures that each job is completely assigned. Svensson [9] proved that the integrality gap of the configuration LP is at most  $33/17 \approx 1.941$  for the Restricted Assignment Problem.

Usually, the *integrality gap* of an integer linear program is 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 a feasibility program. One therefore defines  $\text{OPT}(I) = \min\{T \mid \text{CLP}(T) \text{ has a feasible integer solution}\}$  and  $\text{OPT}_{\text{LP}}(I)$  analogously. Indeed, with this definition,  $\text{OPT}(I)$  is equal to the makespan of an optimal schedule. We will write  $\text{OPT}$  or  $\text{OPT}_{\text{LP}}$  instead of  $\text{OPT}(I)$  and  $\text{OPT}_{\text{LP}}(I)$  if the instance is clear from the context.

Even though the number of variables in  $\text{CLP}(T)$  may be exponentially large, one can find an approximate solution in polynomial time via its dual [1]: If we interpret  $\text{CLP}(T)$  as maximizing a zero objective function, the dual is given by

$$\min \sum_{i \in M} y_i - \sum_{j \in J} z_j \quad (8)$$

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

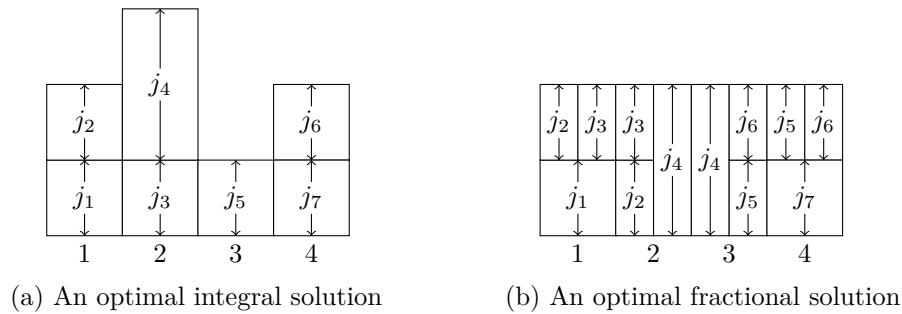
$$z_j \geq 0 \quad \text{for each } j \in J \quad (10)$$

Finding a violated constraint of the dual is equivalent to  $|M|$  knapsack problems, and we can find an approximate solution to these knapsack problems using an FPTAS. Using this FPTAS as separation oracle, a solution to  $\text{CLP}(T)$  can be found. This solution then may contain configurations  $C$  with  $T < p(C) \leq (1 + \varepsilon)T$ , where  $\varepsilon > 0$  is the chosen precision.

Performing a binary search for the best target makespan  $T$ , we can therefore estimate  $\text{OPT}_{\text{LP}}(I)$  within a factor  $1 + \varepsilon$  in polynomial time for arbitrary small  $\varepsilon > 0$ . Using Svenssons

■ **Table 1** An instance with integrality gap  $3/2$ .

Job	1	2	3	4	5	6	7
$p_j$	1	1	1	2	1	1	1
$M(j)$	1	1,2	1,2	2,3	3,4	3,4	4



■ **Figure 1** Solutions for the instance given in Table 1.

bound on the integrality gap then allows us to estimate the optimum makespan  $\text{OPT}(I)$  within a factor of  $33/17 + \varepsilon$  in polynomial time, where  $\varepsilon > 0$  is again an arbitrary small constant. It is a major open problem to find a polynomial rounding procedure whose approximation guarantee matches the integrality gap.

Better results have been obtained when the instances have further restrictions. For example, if  $|M(j)| \leq 2$  for each  $j \in J$ , there is a 1.75-approximation [3]. Recently, Huang and Ott [4] gave improved algorithms for the case that the constraint  $|M(j)| \leq 2$  only applies to big jobs.

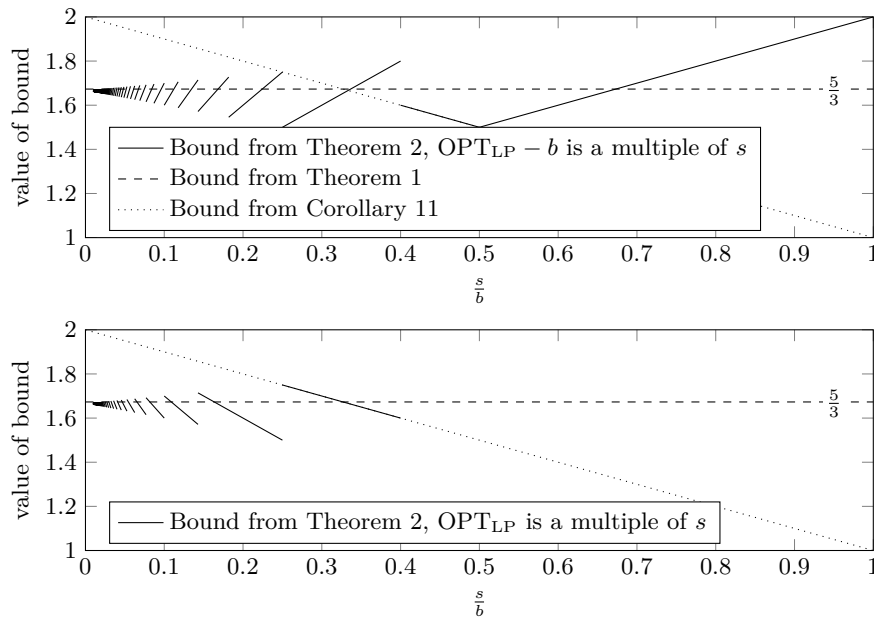
We will in particular investigate the case of only two types of jobs: small jobs with processing time  $s$  and big jobs with processing time  $b$ . Even in this case the integrality gap is at least  $\frac{3}{2}$ , as the instance given in Table 1 and its solutions depicted in Figure 1 show.

Svensson [9] proved that the integrality gap of the configuration LP in this case is at most  $\frac{5}{3} + s$  if  $b = \text{OPT}_{\text{LP}} = 1$ . Koliopoulos and Moysoglou [5] pointed out that this bound also holds for  $\text{OPT}_{\text{LP}} < 2b$  and generalizes to  $\frac{5}{3} + \frac{s}{\text{OPT}_{\text{LP}}}$  when  $b < \text{OPT}_{\text{LP}} < 2b$ . Note that if  $\text{OPT}_{\text{LP}} \geq 2b$ , the analysis of Lenstra, Shmoys, and Tardos [6] bounds the integrality gap of the assignment LP by 1.5, and the configuration LP is at least as strong.

Koliopoulos and Moysoglou [5] developed a  $(2 - \frac{s}{b})$ -approximation for the case that  $b$  is a multiple of  $s$ . By rounding general instances to this form when  $\frac{s}{b} \geq 0.2$  and using Svensson's result when  $\frac{s}{b} < 0.2$ , the makespan can be estimated within a factor of 1.883. Recently, Chakrabarty, Khanna, and Li [2] found a  $(2 - \frac{s}{b})$ -approximation for the general case and therefore improved the estimation ratio to 1.833. They also presented a constructive algorithm with approximation ratio  $2 - \delta$  for a very small  $\delta > 0$ .

### Our Contribution

In section 2, we conduct a tighter analysis of Svensson's [9] local search algorithm that depends on the structure of the fractional solution. In particular, we distinguish the cases that  $\text{OPT}_{\text{LP}} - b$  or  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$ . Note that either  $\text{OPT}_{\text{LP}} - b$  or  $\text{OPT}_{\text{LP}}$  must be a multiple of  $s$ , because a configuration of length  $\text{OPT}_{\text{LP}}$  either contains one big job, or only small jobs. As result, we present better bounds on the integrality gap of the configuration LP. A peculiarity of these bounds is that they are piecewise linear functions with infinitely



■ **Figure 2** Comparison of bounds on the integrality gap for  $\text{OPT}_{\text{LP}} = b$ .

many discontinuities. Our bounds are largest in the case  $\text{OPT}_{\text{LP}} = b$ , for which they are depicted in Figure 2.

We improve this bound further in Sections 3 and 4 and obtain our main result:

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

This bound also allows us to estimate the optimal makespan within a factor of  $\frac{5}{3} \approx 1.667$ , improving upon the previously best possible ratio 1.833 [2].

The proof of Theorem 1 is split in two parts. If  $\frac{s}{b} \leq \frac{1}{3}$ , we study the shape of the bounds we obtained in section 2 more closely in section 3. Our idea is to modify instances for which the bound from section 2 is larger than  $\frac{5}{3}$  by scaling one of the processing times  $s$  and  $b$  such that we get a better bound for the modified instance. We then bound the integrality gap of the original instance in terms of the integrality gap of the newly constructed instance.

In section 4, we present an algorithm with approximation ratio  $2 - \frac{s}{b}$ . The algorithm is based on solving an augmented assignment LP and rounding the solution using a technique due to Shmoys and Tardos [8]. As a corollary, the integrality gap of the augmented assignment LP and the (stronger) configuration LP is bounded by  $2 - \frac{s}{b}$ , which proves Theorem 1 if  $\frac{s}{b} \geq \frac{1}{3}$ . We should note that the approximation ratio  $2 - \frac{s}{b}$  was independently obtained by Chakrabarty, Khanna, and Li [2] using different methods, but they do not use LPs, and therefore cannot derive any bound on the integrality gap. Moreover, our algorithm has an additive approximation guarantee of  $\text{OPT}_{\text{LP}} + b - s$ , which might be of independent interest.

## 2 Bounding the Integrality Gap by Local Search

In this section we present an improved bound on the integrality gap of the configuration LP. Our proof requires that each configuration in the optimal fractional solution contains at most one big job. To satisfy the first condition, it is sufficient (but not necessary) that

$\text{OPT}_{\text{LP}} < 2b$ . The argumentation also works for a restricted variant of the problem where only one big job per configuration is permitted. We further distinguish whether  $\text{OPT}_{\text{LP}} - b$  or  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$ . Note that, if  $\text{OPT}_{\text{LP}} - b$  is not a multiple of  $s$ , then  $\text{OPT}_{\text{LP}}$  is, and vice-versa, so at least one of the cases always holds, and both hold exactly if  $b$  is a multiple of  $s$ .

We will show the following, see also Figure 2:

► **Theorem 2.** *Consider an instance of the restricted assignment problem with jobs of two sizes  $s < b$  such that each configuration in the optimal fractional solution contains at most one big job.*

1. *If  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$ , then the integrality gap of the CLP is at most*

(a)  $1 + \frac{(b-s)}{\text{OPT}_{\text{LP}}}$  if  $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$  and

(b)  $1 + \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil \frac{s}{\text{OPT}_{\text{LP}}}$  otherwise.

2. *If  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$  and  $\frac{s}{b} \leq \frac{2}{5}$ , then the integrality gap of the CLP is at most*

(a)  $1 + (b - \lfloor \frac{b}{3s} + \frac{2}{3} \rfloor \cdot s) \cdot \frac{1}{\text{OPT}_{\text{LP}}}$ .

An upper bound on the values given in Theorem 2 is  $\frac{5}{3} + \frac{1}{3} \frac{s}{\text{OPT}_{\text{LP}}}$ . Recall that Svensson [9] bound generalizes to  $\frac{5}{3} + \frac{s}{\text{OPT}_{\text{LP}}}$  for  $b < \text{OPT}_{\text{LP}} < 2b$  [5]. To prove Theorem 2, we utilize the local search technique due to Svensson [9], but with an improved analysis.

For simplicity, we scale the processing times such that  $\text{OPT}_{\text{LP}} = 1$  from now on. 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  $\sigma$ , i.e. a feasible schedule for a subset  $J' \subset J$  of the jobs, and a currently unscheduled job  $j_{\text{new}}$  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  $\text{CLP}(1)$  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. A more detailed description of the algorithm is given in the next section.

## 2.1 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  $j_{\text{new}}$  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  $mv(B) = (j, i)$  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  $j_{\text{big}}$  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  $j_{\text{big}}$  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

$$\text{Val}(j, i) = \begin{cases} (0, 0) & \text{if } (j, i) \text{ is valid,} \\ (1, p(\sigma^{-1}(i))) & \text{if } (j, i) \text{ is small move,} \\ (2, p(\sigma^{-1}(i))) & \text{if } (j, i) \text{ is big-to-small move, and} \\ (3, 0) & \text{if } (j, i) \text{ is big-to-big move.} \end{cases} \quad (11)$$

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  $J(B) = \sigma^{-1}(i) \setminus J(L)$ ,  $m(B) = i$ ,  $mv(B) = (j, i)$ 
10    Append  $B$  to  $L$ 
11  else # then  $(j, i)$  is a big-to-big move
12    Let  $j_{\text{big}}$  be the big job in  $\sigma^{-1}(i)$ 
13    Create big blocker  $B$  with  $J(B) = \{j_{\text{big}}\}$ ,  $m(B) = i$ ,  $mv(B) = (j, i)$ 
14    Append  $B$  to  $L$ 
15 return  $\sigma$ 

```

---

## 2.2 Proof of Termination

As already mentioned, the crucial step is to prove that the algorithm is successful in creating a feasible schedule. This is of course equivalent to the termination of the algorithm. Svensson proved the termination for the special case  $R = \frac{2}{3} + s$ .

► **Lemma 3** ([9]). *Let  $b = 1$ . Then  $A_{\frac{2}{3}+s}$  always terminates, unless  $\text{CLP}(1)$  is infeasible.*

We provide a stronger and more general variant of Lemma 3.

► **Lemma 4.** *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  $\delta = (k+1)s - R \in (0, s]$  and define  $l = \lfloor \frac{R-b}{s} \rfloor$  and  $\varepsilon = (l+1)s - (R-b) \in (0, s]$ .*

1. *If  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$  and there exists  $0 \leq c \leq b$  such that*

$$b + s - R - \delta \leq c \leq R + \delta \quad (12)$$

$$2R + 2\delta + c + 2(1-b) - s \geq 2 \quad (13)$$

$$2R + \delta - b - s \geq 0, \quad (14)$$

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

2. *If  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$  and there exists  $0 \leq c \leq b$  such that*

$$b + s - R - \varepsilon \leq c \leq R + \varepsilon \quad (15)$$

$$2R + 2\varepsilon + c + 2(1-b) - s \geq 2 \quad (16)$$

$$2R + \delta + \varepsilon - b - s \geq 0, \quad (17)$$

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

Note that Lemma 3 can be derived from Lemma 4 by setting  $b = 1$ ,  $R = \frac{2}{3} + s$  and  $c = \frac{2}{3}$ , but none of the constraints are tight for these values. The improvement is mainly due to the introduction of  $\varepsilon$  and  $\delta$  as the remainder of the division of  $R - b$  respectively  $R$  by  $s$ : in several parts of the analysis, one can see that some multiple of  $s$  is larger than  $R - b$  or  $R$ , so we know it is at least the next multiple of  $s$ , which we can now express as  $R - b + \varepsilon$  and  $R + \delta$ . Since the value of  $\varepsilon$  and  $\delta$  can be anywhere in the interval  $(0, s]$ , the gain varies greatly, and causes the discontinuities in the resulting bounds. For a more detailed proof of Lemma 4 we refer our readers to the full version of the paper.

Our second improvement is to not only consider  $R = \frac{2}{3} + s$  and  $c = \frac{2}{3}$ , but to allow larger values for  $c$ , and subsequently, smaller values for  $R$ , depending on  $s$  and  $b$ . We have determined values for  $R$  (and  $c$ ) that satisfy the prerequisites of Lemma 4. In the case that  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$ , we can prove that these values are best possible.

► **Lemma 5.**

1. *If  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$ , the following are the smallest values for  $R$  that satisfy the prerequisites of Lemma 4, case 1:*

(a)  $R = b - s$  if  $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$  and

(b)  $R = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil s$  otherwise.

2. *If  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$  and  $\frac{s}{b} \leq \frac{2}{5}$ , then  $R = b - \lfloor \frac{b}{3s} + \frac{2}{3} \rfloor s$  satisfies the prerequisites of Lemma 4, case 2.*

We here show that the claimed values for case 1 satisfy the prerequisites of Lemma 4. We omit the proofs for case 2 and the optimality in case 1.

► **Claim 1.** In case (a), i.e. if  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$  and  $\frac{2}{5} \leq \frac{s}{b} < \frac{1}{2}$ , the value  $R = b - s$  satisfies the prerequisites of Lemma 4, case 1.

**Proof.** Recall that  $k = \lfloor \frac{R}{s} \rfloor$  and  $\delta = (k+1)s - R$ . Also 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 conditions (12), (13), and (14) are satisfied. ◀

► **Claim 2.** In case (b), i.e. if  $\text{OPT}_{\text{LP}} - b$  is a multiple of  $s$  and  $\frac{s}{b} < \frac{2}{5}$  or  $\frac{s}{b} \geq \frac{1}{2}$ , the value  $R = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil s$  satisfies the prerequisites of Lemma 4, case 1.

**Proof.** Remember that processing times of the jobs satisfy  $0 < s < b \leq 1$ . Obviously, we have  $k = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil$  and  $\delta = s$ . Also note that

$$R = \left\lceil \frac{2}{3} \left( \frac{b}{s} - 1 \right) \right\rceil s \geq \frac{2}{3} \left( \frac{b}{s} - 1 \right) s = \frac{2}{3}(b - s). \quad (18)$$

Now set  $c = \min\{b, R + \delta\}$ . Then condition (12) is satisfied because

$$\begin{aligned} b - R + s - \delta &= b - R \leq b - \frac{2}{3}(b - s) = \frac{1}{3}b + \frac{2}{3}s < \frac{2}{3}b + \frac{1}{3}s \\ &= \frac{2}{3}(b - s) + s \leq R + s = R + \delta \end{aligned} \quad (19)$$

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

Considering condition (13) we have

$$2R + 2\delta + c + 2(1 - b) - s \geq \frac{4}{3}(b - s) + s + c + 2 - 2b = -\frac{2}{3}b - \frac{1}{3}s + c + 2, \quad (20)$$

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 condition (14), we consider three cases. Note that (14) 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  $R = s \geq \frac{1}{2}b$ . ◀

### 3 Improving the Bound by Scaling

In this section we prove our main result, Theorem 1. Remember that the result by Lenstra, Shmoys, and Tardos [6] shows that the integrality gap is at most  $\frac{3}{2}$  if  $\text{OPT}_{\text{LP}} \geq 2b$ . If  $\frac{s}{b} \geq \frac{1}{3}$ , we obtain that the integrality gap is at most  $2 - \frac{s}{b} \leq \frac{5}{3}$  from section 4. Thus we can restrict our attention to instances  $I$  with  $\frac{s}{b} < \frac{1}{3}$  and  $\text{OPT}_{\text{LP}} < 2b$ . To improve upon the bound from Theorem 2 for the remaining instances, we use a new scaling technique that considers several cases, depending on whether  $\text{OPT}_{\text{LP}} - b$  or  $\text{OPT}_{\text{LP}}$  is a multiple of  $s$  and the value of  $\frac{s}{b}$ . We denote the integrality gap of the configuration LP for an instance  $I$  by  $IG(I)$  throughout this section. Remember that we scaled the processing times such that  $\text{OPT}_{\text{LP}} = 1$ .

#### 3.1 Case 1: $\text{OPT}_{\text{LP}}(I) - b$ is a multiple of $s$

In this case, the longest configuration of the optimal fractional solution contains a big job, i.e. there is  $x \in \mathbb{Z}_{\geq 0}$  with  $\text{OPT}_{\text{LP}}(I) = b + xs$ . Define  $k = \lceil \frac{2}{3}(\frac{b}{s} - 1) \rceil$ . Then Theorem 2 yields

$$IG(I) \leq 1 + ks. \quad (21)$$

We can prove another bound by scaling  $s$ . First, we describe the connection between  $k$  and  $\frac{s}{b}$ . Since  $k \geq \frac{2}{3}(\frac{b}{s} - 1)$  we have  $\frac{s}{b} \geq \frac{2}{3k+2}$ . Similarly,  $k - 1 < \frac{2}{3}(\frac{b}{s} - 1)$  and therefore  $\frac{s}{b} < \frac{2}{3k-1}$ . It follows that  $\frac{s}{b} \in [\frac{2}{3k+2}, \frac{2}{3k-1})$ . This interval actually corresponds to one of the continuous segments of our bound, see also Figure 2. 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 the instance  $I'$  identical to  $I$ , but with small jobs having 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 first prove that  $\text{OPT}_{\text{LP}}(I') \leq \alpha \text{OPT}_{\text{LP}}(I)$ . For this, 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 a configuration increases at most by factor  $\alpha$  in the modified instance  $I'$ . We therefore have

$$IG(I) = \frac{\text{OPT}(I)}{\text{OPT}_{\text{LP}}(I)} \leq \frac{\text{OPT}(I')}{\text{OPT}_{\text{LP}}(I')^{\frac{1}{\alpha}}} \leq \alpha IG(I'). \quad (22)$$

Define

$$s_0 = -\frac{1}{2k} + \sqrt{\frac{1}{4k^2} + \frac{2b}{(3k-1)k} \cdot \left(1 + \frac{(k-1)2b}{3k-1}\right)}. \quad (23)$$

We will later show that scaling is beneficial if  $s \geq s_0$ . In order to apply Theorem 2 to  $I'$ , we need to prove that the prerequisites hold. In particular, we show that  $\text{OPT}_{\text{LP}}(I') - b$  is a multiple of  $s'$ :

► **Lemma 6.** *Let  $I$  be an instance with small processing time  $s \geq s_0$  and  $\text{OPT}_{\text{LP}}(I) = b + xs$ . Let  $I'$  be the modified instance with small processing time  $s' = \frac{2b}{3k-1}$ . Then  $\text{OPT}_{\text{LP}}(I') = b + xs'$ .*

**Proof.** We first claim without proof that  $s_0 > \frac{2b}{3k}$ . Now 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  $\ell$  and  $\ell'$  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

$$\ell' = ys' = zs' + xs' \leq \frac{3k-1}{2} \frac{2b}{3k-1} + xs' = b + xs'. \quad (24)$$

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  $\ell$  and  $\ell'$  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  $\ell = ys = zs + xs < zs' + xs < b + xs$ . ◀

To apply Theorem 2, we also have to scale  $I'$  by  $\beta = \frac{1}{\text{OPT}_{\text{LP}}(I')} \leq 1$  to obtain the instance  $I''$  with  $\text{OPT}_{\text{LP}}(I'') = 1$ . In  $I''$  the processing times are  $b'' = b \cdot \beta$  and  $s'' = s' \cdot \beta \leq s'$ . One can easily see that  $I'$  and  $I''$  have the same integrality gap:

$$IG(I') = \frac{\text{OPT}(I')}{\text{OPT}_{\text{LP}}(I')} = \frac{\beta \cdot \text{OPT}(I')}{\beta \cdot \text{OPT}_{\text{LP}}(I')} = \frac{\text{OPT}(I'')}{\text{OPT}_{\text{LP}}(I'')} = IG(I''). \quad (25)$$

## 24:10 Estimating The Makespan of The Two-Valued Restricted Assignment Problem

Also  $\frac{s}{b} < \frac{1}{3}$  implies  $k \geq 2$  and thus  $\frac{s'}{b} = \frac{2}{3k-1} \leq \frac{2}{5}$ . This finally allows us to apply Theorem 2. From the definition  $s' = \frac{2b}{3k-1}$  we can calculate  $k - 1 = \frac{2}{3}(\frac{b}{s'} - 1)$ . Since  $k$  is integral, we have  $\lceil \frac{2}{3}(\frac{b}{s'} - 1) \rceil = k - 1$ . The integrality gap of the original instance  $I$  is thus bounded by

$$IG(I) \leq \alpha IG(I') = \alpha IG(I'') \leq \alpha(1 + (k-1)s'') \leq \alpha(1 + (k-1)s'). \quad (26)$$

We will determine which of the two bounds (21) and (26) is better depending on the values of  $s$  and  $b$ .

► **Lemma 7.**  $1 + ks \geq \alpha(1 + (k-1)s')$  holds if and only if  $s \geq s_0$ .

For a proof we refer our readers to the full version of our work.

It turns out that inequalities (26) and (21) can be combined if  $b$  is not too large.

► **Lemma 8.** 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 inequality (21).

*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

$$\alpha(1 + (k-1)s') = \frac{2b}{(3k-1)s} \left( 1 + \frac{(k-1)2b}{3k-1} \right) = \frac{1}{s} \left( \frac{2b}{3k-1} + \frac{(k-1)4b^2}{(3k-1)^2} \right) \quad (27)$$

is monotonically decreasing with respect to  $s$ . From Lemma 7 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

$$\begin{aligned} 1 + ks_0 &= 1 - \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{2bk}{(3k-1)} \cdot \left( 1 + \frac{(k-1)2b}{3k-1} \right)} \\ &= \frac{1}{2} + \frac{1}{2(3k-1)} \cdot \sqrt{(3k-1)^2 + 8bk(3k-1 + (k-1)2b)}. \end{aligned} \quad (28)$$

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  $b \leq \frac{80}{81} \approx 0.988$ . ◀

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

► **Lemma 9.** 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 small jobs were rounded or not. We use the same scaled instance  $I'$  from above, where we scaled  $s$  to  $s' = \frac{2b}{3k-1}$ . Remember that  $\text{OPT}_{\text{LP}} = b + xs$  for  $x \in \mathbb{N}$ .

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

$$IG(I) \leq 1 + ks \leq 1 + k \frac{2b}{3k} \leq \frac{5}{3}. \quad (29)$$

*Case 2:*  $s > \frac{2b}{3k}$ . We claim without proof that  $\text{OPT}_{\text{LP}}(I') \leq 1 + \frac{1}{79}s$ .

Similar as in inequality (22), we find that  $IG(I) \leq (1 + \frac{1}{79}s)IG(I')$ . As above, we have to scale  $I'$  to  $I''$  such that  $\text{OPT}_{\text{LP}}(I'') = 1$ . Applying Theorem 2 to  $I''$  and using  $s < \frac{2b}{3k-1}$  and  $b \leq 1$ , we obtain

$$\begin{aligned}
IG(I) &\leq \left(1 + \frac{1}{79}s\right)IG(I') = \left(1 + \frac{1}{79}s\right)IG(I'') \\
&\leq \left(1 + \frac{1}{79}s\right)\left(1 + (k-1)\frac{s'}{\text{OPT}_{\text{LP}}(I'')}\right) \\
&\leq \left(1 + \frac{1}{79}s\right)(1 + (k-1)s') \\
&< \left(1 + \frac{2}{79(3k-1)}\right)\left(1 + \frac{2(k-1)}{3k-1}\right) \\
&= \frac{15k^2 - \frac{1096}{79}k + \frac{231}{79}}{9k^2 - 6k + 1}.
\end{aligned} \tag{30}$$

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

### 3.2 Case 2: $\text{OPT}_{\text{LP}}(I)$ is a multiple of $s$

Using Theorem 2, case (b) with  $\text{OPT}_{\text{LP}} = 1$  and  $k = \lfloor \frac{1}{3}\frac{b}{s} + \frac{2}{3} \rfloor$  we have

$$IG(I) \leq 1 + R = 1 + b - ks. \tag{31}$$

Since  $k \leq \frac{1}{3}\frac{b}{s} + \frac{2}{3}$ , we have  $\frac{s}{b} \leq \frac{1}{3k-2}$ . Similarly,  $k+1 > \frac{1}{3}\frac{b}{s} + \frac{2}{3}$  and therefore  $\frac{s}{b} > \frac{1}{3k+1}$ . It follows that  $\frac{s}{b} \in (\frac{1}{3k+1}, \frac{1}{3k-2}]$ .

Now assume that  $\frac{s}{b} \geq \frac{1}{3k}$ . Then the integrality gap  $IG(I)$  is bounded by

$$1 + b - ks \leq 1 + b - k\frac{b}{3k} = 1 + \frac{2}{3}b \leq \frac{5}{3}, \tag{32}$$

since  $b \leq 1$ .

Finally, we have the case that  $\frac{s}{b} \in (\frac{1}{3k+1}, \frac{1}{3k})$ . Our bound (31) increases when  $\frac{s}{b}$  approaches  $\frac{1}{3k+1}$  and jumps down when it reaches that value, see also Figure 2. So we create an instance  $I'$  by rounding the running time  $b$  of big jobs up to  $b' = (3k+1)s$ . Obviously,  $\text{OPT}(I') \geq \text{OPT}(I)$ . On the other hand, the largest configuration in the fractional optimum has the height  $xs$  for some  $x \in \mathbb{Z}_{\geq 1}$ , and any configuration that contains a big job has height  $b + ys \leq xs$  for some  $y \in \mathbb{Z}_{\geq 0}$ . We assume that  $b + ys < xs$ , because otherwise  $b$  and therefore  $\text{OPT}_{\text{LP}}(I) - b$  were also a multiples of  $s$ , and we could use the proof from section 3.1. Then, since  $b \in (3ks, 3ks + s)$ , we have

$$xs - (y + 3k)s > b + ys - ys - 3ks = b - 3ks > 0 \tag{33}$$

and therefore  $xs - (y + 3k)s \geq s$ . This implies

$$b' + ys = (3k+1)s + ys = (y + 3k)s + s \leq xs - s + s = xs, \tag{34}$$

i.e.  $\text{OPT}_{\text{LP}}(I') = \text{OPT}_{\text{LP}}(I)$ . In particular,

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

and  $\text{OPT}_{\text{LP}}(I')$  is a multiple of  $s$ . Note that

$$\left\lfloor \frac{1}{3}\frac{b'}{s} + \frac{2}{3} \right\rfloor = \left\lfloor \frac{1}{3}(3k+1) + \frac{2}{3} \right\rfloor = \lfloor k+1 \rfloor = k+1. \tag{36}$$

Theorem 2, case (b) now yields

$$\begin{aligned} IG(I) &\leq IG(I') \leq 1 + R = 1 + b' - (k+1)s \\ &= 1 + (3k+1)s - (k+1)s = 1 + 2ks < 1 + 2k\frac{b}{3k} = 1 + \frac{2}{3}b \\ &\leq 1 + \frac{2}{3} = \frac{5}{3}. \end{aligned} \tag{37}$$

#### 4 An $(OPT + b - s)$ -Approximation

In this section we present an algorithm for the restricted assignment problem with two different processing times, which also proves a bound on the integrality gap of the CLP. Here, we assume that the processing times  $s < b$  are positive integers. Our algorithm depends on a result by Shmoys and Tardos [8] for a variant of unrelated scheduling with costs. Their algorithm is based on solving and rounding the assignment LP. Consider the assignment LP  $ALP(T)$  for a given makespan  $T$ . For  $i \in \{1, \dots, m\}$  and  $q \in \{s, b\}$ , let  $a_{iq} = \sum_{j:p_{ij}=q} x_{ij}$  denote the fractional number of jobs of size  $q$  scheduled on machine  $i$ . We strengthen the LP relaxation by adding two classes of constraints

$$a_{ib} \leq \left\lfloor \frac{T}{b} \right\rfloor \quad \text{for each } i \in M \tag{38}$$

$$a_{is} \leq \left\lfloor \frac{T}{s} \right\rfloor - \left\lfloor \frac{b}{s} \right\rfloor a_{ib} \quad \text{for each } i \in M. \tag{39}$$

In the following,  $OPT_{LP}$  denotes the optimal makespan of the augmented ALP. Our algorithm solves the augmented ALP and applies the rounding procedure of Shmoys and Tardos [8] to the solution.

► **Theorem 10.** *For the restricted assignment problem with two different processing times  $s < b$  there is a polynomial time algorithm that produces a schedule of length at most  $\min\{OPT_{LP} + b, \lfloor OPT_{LP}/s \rfloor s + \lfloor OPT_{LP}/b \rfloor (b - s)\}$ , yielding a bound of  $OPT_{LP} + (b - s)$  for the case  $b \leq OPT_{LP} < 2b$ . Furthermore the algorithm has a multiplicative performance guarantee of  $(2 - \frac{s}{b})$ .*

We omit the analysis of the algorithm due to space restrictions. As a corollary we can bound the integrality gap of the configuration LP.

► **Corollary 11.** *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  $2 - \frac{s}{b}$ .*

---

#### References

- 1 Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, (STOC 2006)*, pages 31–40, 2006.
- 2 Deeparnab Chakrabarty, Sanjeev Khanna, and Shi Li. On  $(1, \epsilon)$ -restricted assignment makespan minimization. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, pages 1087–1101, 2015.
- 3 Tomáš Ebenlendr, Marek Krčál, and Jiří Sgall. Graph balancing: A special case of scheduling unrelated parallel machines. *Algorithmica*, 68(1):62–80, 2014.
- 4 Chien-Chung Huang and Sebastian Ott. A combinatorial approximation algorithm for graph balancing with light hyper edges. *CoRR*, abs/1507.07396, 2015.

- 5 Stavros G Kolliopoulos and Yannis Moysoglou. The 2-valued case of makespan minimization with assignment constraints. *Information Processing Letters*, 113(1):39–43, 2013.
- 6 Jan Karel Lenstra, David B Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3):259–271, 1990.
- 7 Evgeny V. Shchepin and Nodari Vakhania. An optimal rounding gives a better approximation for scheduling unrelated machines. *Operations Research Letters*, 33(2):127–133, 2005.
- 8 David B Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(1-3):461–474, 1993.
- 9 Ola Svensson. Santa claus schedules jobs on unrelated machines. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC 2011)*, pages 617–626, 2011.