Strong LP Formulations for Scheduling Splittable Jobs on Unrelated Machines*

José R. Correa[1], Alberto Marchetti-Spaccamela[2], Jannik Matuschke[1], Leen Stougie[3], Ola Svensson[4], Víctor Verdugo[1], and José Verschae[1]

[1] Departamento de Ingeniería Industrial, Universidad de Chile
[2] Department of Computer and System Sciences, Sapienza University of Rome
[3] Department of Econometrics and Operations Research, VU Amsterdam & CWI
[4] School of Computer and Communication Sciences, EPFL

Abstract. We study a natural generalization of the problem of minimizing makespan on unrelated machines in which jobs may be split into parts. The different parts of a job can be (simultaneously) processed on different machines, but each part requires a setup time before it can be processed. First we show that a natural adaptation of the seminal approximation algorithm for unrelated machine scheduling [11] yields a 3-approximation algorithm, equal to the integrality gap of the corresponding LP relaxation. Through a stronger LP relaxation, obtained by applying a lift-and-project procedure, we are able to improve both the integrality gap and the implied approximation factor to $1 + \phi$, where $\phi \approx 1.618$ is the golden ratio. This ratio decreases to 2 in the restricted assignment setting, matching the result for the classic version. Interestingly, we show that our problem cannot be approximated within a factor better than $\frac{e}{e-1} \approx 1.582$ (unless $\mathcal{P} = \mathcal{NP}$). This provides some evidence that it is harder than the classic version, which is only known to be inapproximable within a factor $1.5 - \varepsilon$. Since our $1 + \phi$ bound remains tight when considering the seemingly stronger machine configuration LP, we propose a new *job based* configuration LP that has an infinite number of variables, one for each possible way a job may be split and processed on the machines. Using convex duality we show that this infinite LP has a finite representation and can be solved in polynomial time to any accuracy, rendering it a promising relaxation for obtaining better algorithms.

1 Introduction

The unrelated machine scheduling problem, $R||C_{\max}$ in the three-field notation of [8], has attracted significant attention within the scientific community. The problem is to find a schedule of jobs with machine-dependent processing times

* This work was partially supported by Nucleo Milenio Información y Coordinación en Redes ICM/FIC P10-024F, by EU-IRSES grant EUSACOU, by the DFG Priority Programme "Algorithm Engineering" (SPP 1307), by the Berlin Mathematical School, by ERC Starting Grant 335288-OptApprox, and by FONDECYT project 3130407.

J. Lee and J. Vygen (Eds.): IPCO 2014, LNCS 8494, pp. 249–260, 2014.

that minimizes the makespan, i.e., the maximum machine load. In [11] a polynomial time linear programming based rounding algorithm was shown to give an approximation guarantee of 2, and a lower bound of 3/2 on the approximation ratio of any polynomial time algorithm was shown, assuming $\mathcal{P} \neq \mathcal{NP}$.

A natural generalization of this problem is to allow jobs to be *split* and processed on multiple machines simultaneously, where in addition a *setup* has to performed on every machine processing the job. This generalized scheduling problem finds applications in production planning, e.g., in textile and semiconductor industries [18,10], and disaster relief operations [21]. Formally, we are given a set of m machines M and a set of n jobs J with *processing times* $p_{ij} \in \mathbb{Z}_+$ and *setup times* $s_{ij} \in \mathbb{Z}_+$ for every $i \in M$ and $j \in J$. A *schedule* corresponds to a vector $x \in [0,1]^{M \times J}$, where x_{ij} denotes the fraction of job j that is assigned to machine i, satisfying $\sum_{i \in M} x_{ij} = 1$ for all $j \in J$. If job j is processed (partially) on machine i then a setup of length s_{ij} has to be performed on the machine. During the setup of a job, the machine is occupied and thus no other job can be processed nor be set up. This results in the definition of *load* of machine $i \in M$ as $\sum_{j : x_{ij} > 0} (x_{ij} p_{ij} + s_{ij})$. The objective is to minimize the *makespan*, the maximum load of the schedule. We denote this problem by $R|\text{split,setup}|C_{\max}$. Note that by setting $p_{ij} = 0$ and interpreting the setup times s_{ij} as processing requirements we obtain $R||C_{\max}$.

Related Work. Reducing the approximability gap for $R||C_{\max}$ is a prominent open question [23]. Since the seminal work by Lenstra et al. [11] there has been a considerable amount of effort leading to partial solutions to this question. In the *restricted assignment* problem, the processing times are of the form $p_{ij} \in \{p_j, \infty\}$ for all $i, j \in J$. A special case of this setting, in which each job can only be assigned to two machines, was considered by Ebenlendr et al. [6]. They note that while the lower bound of 3/2 still holds, a 7/4-approximation can be obtained. Svensson [19] shows that the general restricted assignment problem is approximable within a factor of $33/17 + \varepsilon \approx 1.9412 + \varepsilon$, breaking the barrier of 2. This algorithm is based on a *machine configuration* linear programming relaxation where each variable indicates the subset of jobs assigned to a given machine. On the other hand, this relaxation has an integrality gap of 2 for general unrelated machines [22]. Configuration LPs have also been studied extensively for the max-min version of the problem [22,3,7,9,2,14].

Most work concerned with scheduling splittable jobs focuses on heuristics. Theoretical results on the subject are not only scarce, but also restricted to the special case of identical machines. In particular, Xing and Zhang [24] describe a $(1.75 - 1/m)$-approximation for makespan minimization, that was later improved to 5/3 by Chen et al. [4]. The objective of minimizing the sum of completion times is studied by Schalekamp et al. [16], who gave a polynomial time algorithm in the case of 2 machines, and a 2.781-approximation algorithm for arbitrary m. This was later improved to $2 + \varepsilon$ in [5], even in the presence of weights.

Another setting that comes close to job splitting is preemptive scheduling with setup times [17,12,15], which does not allow simultaneous processing of parts of the same job. We also refer to the survey [1] and references therein for results on other scheduling problems with setup costs.

Our Contribution. Due to the novelty of the considered problem, our aim is to advance the understanding of its approximability, in particular in comparison to $R||C_{\max}$. We first study the integrality gap of a natural generalization of the LP relaxation by Lenstra et al. [11] to our setting and notice that their rounding technique does not work in our case. This is because it might assign a job with very large processing time to a single machine, while the optimal solution splits this job. On the other hand, assigning jobs by only following the fractional solution given by the LP might incur a large number of setups (belonging to different jobs) to a single machine. We get around these two extreme cases by adapting the technique from [11] so as to only round variables exceeding a certain threshold while guaranteeing that only one additional setup time is required per machine. This yields a 3-approximation algorithm presented in § 2. Additionally, we show that the integrality gap of this LP is exactly 3, and therefore our algorithm is best possible for this LP.

In § 3 we improve the approximation ratio by tightening our LP relaxation with a lift-and-project approach. We refine our previous analysis by balancing the rounding threshold, resulting in a $(1+\phi)$-approximation, where $\phi \approx 1.618$ is the golden ratio. Surprisingly, we can show that this number is best possible for this LP; even for the seemingly stronger machine configuration LP mentioned above. This suggests that considerably different techniques are necessary to match the 2-approximation algorithm for $R||C_{\max}$. Indeed, we also show in § 5 that it is $\mathcal{NP}$-hard to approximate within a factor $\frac{e}{e-1} \approx 1.582$, a larger lower bound than the 3/2 hardness result known for $R||C_{\max}$. For the restricted assignment case, where $s_{ij} \in \{s_j, \infty\}$ and $p_{ij} \in \{p_j, \infty\}$, we obtain a 2-approximation algorithm, matching the 2-approximation of [11] in § 4. We remark that the solutions produced by all our algorithms have the property that at most one split job is processed on each machine. This property may be desirable in practice since in manufacturing systems setups require labor causing additional expenses.

As the integrality gaps of all mentioned relaxations are no better than $1+\phi$, we propose a novel *job based* configuration LP relaxation in § 6 that has the potential to lead to better guarantees. Instead of considering machine configurations that assign jobs to machines, we introduce *job configurations*, describing the assignment of a particular job to the machines. The resulting LP cuts away worst-case solutions of the other LPs considered in this paper, rendering it a promising candidate for obtaining better approximation ratios. While the job configuration LP has an infinite set of variables, we show that we can restrict a priori to a finite subset. Applying discretization techniques we can approximately solve the LP within a factor of $(1+\varepsilon)$ by separation over the dual constraints. Finally, we study the projection of this polytope to the *assignment space* and derive an explicit set of inequalities that defines this polytope. An interesting open problem is to determine the integrality gap of the job configuration LP.

2 A 3-Approximation Algorithm

Our 3-approximation algorithm is based on a generalization of the LP by Lenstra, Shmoys, and Tardos [11]. Let C^* be a guess on the optimal makespan. Consider the following feasibility LP, whose variable x_{ij} denotes the fraction of job j assigned to machine i. Notice that the LP is a relaxation, since it allows the setups to be performed fractionally.

$$[\text{LST}]: \quad \sum_{i \in M} x_{ij} = 1 \quad \text{for all } j \in J, \tag{1}$$

$$\sum_{j \in J} x_{ij}(p_{ij} + s_{ij}) \le C^* \quad \text{for all } i \in M, \tag{2}$$

$$x_{ij} = 0 \quad \text{for all } i \in M,\ j \in J : s_{ij} > C^*, \tag{3}$$

$$x_{ij} \ge 0 \quad \text{for all } i \in M, j \in J.$$

Let x be a feasible extreme solution. We define the bipartite graph $G(x) = (J \cup M, E(x))$, where $E(x) = \{ij : 0 < x_{ij}\}$. Using the same arguments as in [11], not repeated here, we can show the following property.

Lemma 1. *For every extreme solution x of* [LST]*, each connected component of $G(x)$ is a* pseudotree*; a tree plus at most one edge that creates a single cycle.*

We show how to round an extreme solution x of [LST]. Let $E_+ = \{ij \in E(x) : x_{ij} > 1/2\}$ and $J_+ = \{j \in J : \text{there exists } i \in M \text{ with } ij \in E_+\}$, i.e., those jobs that the fractional solution x assigns to some machine by a factor of more than 1/2. In our rounding procedure each job $j \in J_+$ is completely assigned to the machine $i \in M$ if $x_{ij} > 1/2$. We now show how to assign the rest of the jobs.

Let us call $G'(x)$ the subgraph of $G(x)$ induced by $(J \cup M) \setminus J_+$. Notice that every edge ij in $G'(x)$ satisfies that $0 < x_{ij} \le 1/2$. Also, since $G'(x)$ is a subgraph of $G(x)$ every connected component of $G'(x)$ is a pseudotree.

Definition 1. *Given $A \subseteq E(G'(x))$, we say that a machine $i \in M$ is A-*balanced*, if there exists at most one job $j \in J \setminus J_+$ such that $ij \in A$. We say that a job $j \in J \setminus J_+$ is A-*processed *if there is at most one machine $i \in M$ such that $ij \notin A$ and $x_{ij} > 0$.*

In what follows we seek to find a subset $A \subseteq E(G'(x))$ such that each job $j \in J \setminus J_+$ is A-processed and each machine is A-balanced. We will show that this is enough for a 3-approximation, by assigning each job $j \in J \setminus J_+$ to machine i by a fraction of at most $2x_{ij}$ for each $ij \in A$, and not assigning it anywhere else. Since every job $j \in J \setminus J_+$ is A-processed and $x_{ij} \le 1/2$ for all $i \in M$, job j will be completely assigned. Also, since each machine is A-balanced, the load of each machine i will be affected by at most the setup-time of one job j. This setup time s_{ij} is at most C^* by restriction (3). This and the fact that the processing time of a job on each machine is at most doubled are the basic ingredients to show the approximation factor of 3.

Construction of the Set A. In the following, we denote by (T, r) a rooted tree T with root r. Consider a connected component T of $G'(x)$. Since $G'(x)$ is a subgraph of $G(x)$, Lemma 1 implies that T is a pseudotree. We denote by $C = j_1 i_1 j_2 i_2 \cdots j_\ell i_\ell j_1$ the only cycle of T (if it exists), which must be of even length. (If such a cycle does not exist we choose any path in T from j_1 to some i_ℓ.) Here the jobs are $J(C) = \{j_1, \ldots, j_\ell\}$ and the machines are $M(C) = \{i_1, \ldots, i_\ell\}$. In the cycle, we define the matching $K_C = \{(j_k, i_k) : k \in \{1, \ldots, \ell\}\}$. In the forest $T \setminus K_C$, we denote by (T_u, u) the tree rooted in u, for every $u \in M(C)$. Notice that by deleting the matching, no two vertices in $M(C)$ will be in the same component of $T \setminus K_C$.

For every $u \in M(C)$, directing the edges of (T_u, u) away from the root, we obtain the directed tree of which each level consists either entirely of machine-nodes or entirely of job-nodes. We delete all edges going out of machine nodes, i.e. all edges entering job-nodes. The remaining edges we denote by A_u. We define $A := K_C \cup \bigcup_{u \in M(C)} A_u$. We obtain the following to lemmas.

Lemma 2. *Every job $j \in J \setminus J_+$ is A-processed.*

Lemma 3. *Every machine $i \in M$ is A-balanced.*

Given set A, we apply the following rounding algorithm that constructs a new assignment $\tilde{x}$. The algorithm also outputs a binary vector $\tilde{y}_{ij} \in \{0, 1\}$ which indicates whether job j is (partially) assigned to machine i or not.

Algorithm 1. Rounding(x)

1: Construct the graphs $G(x)$, $G'(x)$, and the set A as above.
2: For all $ij \in E_+$, $\tilde{x}_{ij} \leftarrow 1$ and $\tilde{y}_{ij} \leftarrow 1$;
3: For all $ij \in A$, $\tilde{x}_{ij} \leftarrow \dfrac{x_{ij}}{\sum_{k:kj \in A} x_{kj}}$ and $\tilde{y}_{ij} \leftarrow 1$;
4: For all $ij \in E \setminus (E_+ \cup A)$, $\tilde{x}_{ij} \leftarrow 0$ and $\tilde{y}_{ij} \leftarrow 0$.

Theorem 1. *There exists a 3-approximation algorithm for $R|split, setup|C_{\max}$.*

Proof. Our algorithm first finds the smallest value C^* for which [LST] is feasible. This can be easily done with a binary search procedure. For that value C^*, let x be an extreme point of [LST], and consider the output $\tilde{x}, \tilde{y}$ of algorithm Rounding(x). Clearly $\tilde{x}, \tilde{y}$ can be computed in polynomial time. We show that the schedule that assigns a fraction $\tilde{x}_{ij}$ of job j to machine i has a makespan of at most $3C^*$. This implies the theorem since $C^* \leq \text{OPT}$.

First we show that $\tilde{x} \geq 0$ defines a valid assignment, i.e., $\sum_{i \in M} \tilde{x}_{ij} = 1$ for all j. Indeed, this directly follows by the algorithm Rounding(x): If $j \in J_+$, then there exists a unique machine $i \in M$ with $ij \in E_+$ and therefore j is completely assigned to machine i. If $j \notin J_+$, then $\sum_{i \in M} \tilde{x}_{ij} = 1$ by construction.

Now we show that the makespan of the solution is at most $3C^*$. First notice that for every $ij \in E_+$ we have that $1 = \tilde{x}_{ij} = \tilde{y}_{ij} \leq 2x_{ij}$, because $ij \in E_+$

implies that $x_{ij} > 1/2$. On the other hand, for every $j \in J \setminus J_+$ we have that $\sum_{k:kj\in A} x_{kj} \geq 1/2$, because at most one machine that processes j fractionally is not considered in A. We conclude that $\tilde{x} \leq 2x$. Then for every $i \in M$ it holds that

$$\begin{aligned}\sum_{j\in J}(\tilde{x}_{ij}p_{ij} + \tilde{y}_{ij}s_{ij}) &= \sum_{j:ij\in E_+}(\tilde{x}_{ij}p_{ij} + \tilde{y}_{ij}s_{ij}) + \sum_{j:ij\in A}(\tilde{x}_{ij}p_{ij} + \tilde{y}_{ij}s_{ij})\\ &\leq \sum_{j:ij\in E_+} 2x_{ij}(p_{ij} + s_{ij}) + \sum_{j:ij\in A}(2x_{ij}p_{ij} + s_{ij})\\ &\leq 2C^* + \sum_{j:ij\in A} s_{ij}.\end{aligned}$$

Recall that machine i is A-balanced, and therefore there is at most one job j with $ij \in A$. Also, $ij \in A$ implies that $ij \in E(x) = \{ij : x_{ij} > 0\}$, and hence, by (3), $s_{ij} \leq C^*$. We conclude that $\sum_{j:ij\in A} s_{ij} \leq C^*$, proving the theorem. □

We finish this section by noting that our analysis is tight. Specifically, it can be shown that the gap between the LP solution and the optimum can be arbitrarily close to 3.

Theorem 2. *For any $\varepsilon > 0$, there exists an instance such that $(3-\varepsilon)C^* \leq OPT$, where C^* is the smallest number such that* [LST] *is feasible.*

3 A $(1 + \phi)$-Approximation Algorithm

In this section we refine the previous algorithm and improve the approximation ratio. Since [LST] has a gap of 3, we strengthen it in order to obtain a stronger LP. To this end notice that inequalities (2) in [LST] are the LP relaxation of the following restrictions of the mixed integer linear program with binary variables y_{ij} for machine i and job j:

$$\sum_{j\in J}(x_{ij}p_{ij} + y_{ij}s_{ij}) \leq C^* \qquad \text{for all } i \in M, \tag{4}$$

$$x_{ij} \leq y_{ij} \qquad \text{for all } i \in M \text{ and } j \in J. \tag{5}$$

A stronger relaxation is obtained by applying a lift and project step [13] to the first inequality. For some fixed choice ij multiplying both sides of the i-th inequality (4) by the corresponding variable y_{ij} implies (by leaving out terms)

$$y_{ij}x_{ij}p_{ij} + y_{ij}^2 s_{ij} \leq y_{ij}C^*.$$

In case $C^* - s_{ij} > 0$, this inequality implies the valid linear inequality

$$x_{ij}\frac{p_{ij}}{C^* - s_{ij}} \leq y_{ij}, \tag{6}$$

since every feasible integer solution has $y_{ij}x_{ij} = x_{ij}$ and $y_{ij}^2 = y_{ij}$. Note that, in optimal solutions of the LP relaxation, y_{ij} attains the smallest value that satisfies (5) and (6). We define $\alpha_{ij} = \max\left\{1, \frac{p_{ij}}{C^*-s_{ij}}\right\}$ if $C^* > s_{ij}$, and $\alpha_{ij} = 1$ otherwise, and substitute y_{ij} by $\alpha_{ij}x_{ij}$ to obtain the strengthened LP relaxation

$$[\text{LST}_{\text{strong}}]: \quad \sum_{i\in M} x_{ij} = 1 \qquad \text{for all } j \in J, \tag{7}$$

$$\sum_{j\in J} x_{ij}(p_{ij} + \alpha_{ij}s_{ij}) \le C^* \qquad \text{for all } i \in M, \tag{8}$$

$$x_{ij} = 0 \qquad \text{for all } i \in M,\ j \in J : s_{ij} > C^*, \tag{9}$$

$$x_{ij} \ge 0 \qquad \text{for all } i \in M, j \in J.$$

Notice that this LP is at least as strong as [LST] since $\alpha_{ij} \ge 1$ and, therefore, the C^* values used in [LST] and [$\text{LST}_{\text{strong}}$] might differ. Again binary search allows us to find the minimum C^* for which [$\text{LST}_{\text{strong}}$] is feasible.

Let x be an extreme point solution of this LP. We use a rounding approach similar to the one in the previous section. Proofs that are the same as in that section will be skipped. Consider the graph $G(x)$. As before, each connected component of $G(x)$ is a pseudotree, using the same arguments that justified Lemma 1. Also, we define again a set of jobs J_+ that the LP assigns to one machine by a sufficiently large fraction. In the previous section this fraction was 1/2. Now we parameterize it by $\beta \in (1/2, 1)$, to be chosen later. We define $E_+ = \{j \in E(x) : x_{ij} > \beta\}$ and $J_+ = \{j \in J :$ there exists $i \in M$ with $ij \in E_+\}$.

Consider the subgraph $G'(x)$ of $G(x)$ induced by the set of nodes $(J \cup M) \setminus J_+$. Let A be a set constructed as in the previous section. Then every machine is A-balanced and every job is A-processed. Now we apply the algorithm Rounding(x) of the last section to obtain a new assignment $\tilde{x}, \tilde{y}$. We show that for $\beta = \phi - 1$ this is a solution with makespan $(1 + \phi)C^*$, where $\phi = (1 + \sqrt{5})/2 \approx 1.618$ is the golden ratio. We need the following technical lemma.

Lemma 4. *Let β be a real number such that $1/2 < \beta < 1$. Then*

$$\max_{0\le\mu\le1}\left\{\mu + \max\left\{\frac{1}{\beta}, \frac{1-\mu}{1-\beta}\right\}\right\} = \max\left\{\frac{1}{1-\beta}, 1 + \frac{1}{\beta}\right\}.$$

Theorem 3. *There exists a $(1 + \phi)$-approximation algorithm for the problem* $R|\text{split,setup}|C_{\max}$.

Proof. Let x be an extreme point solution of [$\text{LST}_{\text{strong}}$], and let $\tilde{x}, \tilde{y}$ be the output of algorithm Rounding(x) described in § 2. The fact that $\tilde{x}, \tilde{y}$ correspond to a feasible assignment follows from the same argument as in the proof of Theorem 1. We now show that the makespan of this solution is at most $(1+\phi)C^*$, which implies the approximation factor.

For any edge $ij \in E_+$, we have $x_{ij} > \beta$ and hence $1 = \tilde{x}_{ij} = \tilde{y}_{ij} \le 1/\beta \cdot x_{ij}$. Additionally, for every $j \in J \setminus J_+$, we have again, by the choice of A, that it is A-processed. Hence, $\sum_{k:kj\notin A} x_{kj} \le \beta$, because at most one machine that processes

j fractionally is not considered in A. Thus, $\sum_{k:kj\in A} x_{kj} \geq 1-\beta$, which implies that $\tilde{x}_{ij} \leq x_{ij}/(1-\beta)$. Hence, for machine i,

$$\begin{aligned}\sum_{j\in J}(\tilde{x}_{ij}p_{ij}+\tilde{y}_{ij}s_{ij}) &= \sum_{j:ij\in E_+}(\tilde{x}_{ij}p_{ij}+\tilde{y}_{ij}s_{ij}) + \sum_{j:ij\in A}(\tilde{x}_{ij}p_{ij}+\tilde{y}_{ij}s_{ij}) \\ &\leq \frac{1}{\beta}\sum_{j:ij\in E_+} x_{ij}(p_{ij}+s_{ij}) + \frac{1}{1-\beta}\sum_{j:ij\in A} x_{ij}p_{ij} + \sum_{j:ij\in A} s_{ij}.\end{aligned}$$

Since machine i is A-balanced, there exists at most one job j with $ij \in A$ (if there is no such job then i has load at most C^*/β). Let $j(i)$ be that job, and define $\mu_i = s_{ij(i)}/C^*$. Then notice that

$$\begin{aligned}x_{ij(i)}(p_{ij(i)}+\alpha_{ij(i)}s_{ij(i)}) &\geq x_{ij(i)}p_{ij(i)}\left(1+\frac{s_{ij(i)}}{C^*-s_{ij(i)}}\right) \\ &= x_{ij(i)}p_{ij(i)}\left(1+\frac{\mu_i}{1-\mu_i}\right) = x_{ij(i)}p_{ij(i)}\frac{1}{1-\mu_i}.\end{aligned}$$

Combining the last two inequalities we obtain that

$$\begin{aligned}\sum_{j\in J}(\tilde{x}_{ij}p_{ij}+\tilde{y}_{ij}s_{ij}) &\leq \frac{1}{\beta}\sum_{j:ij\in E_+} x_{ij}(p_{ij}+s_{ij}) + \frac{1}{1-\beta}x_{ij(i)}p_{ij(i)} + s_{ij(i)} \\ &\leq \frac{1}{\beta}\sum_{j:ij\in E_+} x_{ij}(p_{ij}+s_{ij}) + \frac{1-\mu_i}{1-\beta}x_{ij(i)}(p_{ij(i)}+\alpha_{ij(i)}s_{ij(i)}) + \mu_i C^* \\ &\leq \max\left\{\frac{1}{\beta},\frac{1-\mu_i}{1-\beta}\right\}\sum_{j\in J} x_{ij}(p_{ij}+\alpha_{ij}s_{ij}) + \mu_i C^* \\ &\leq C^*\left(\mu_i + \max\left\{\frac{1}{\beta},\frac{1-\mu_i}{1-\beta}\right\}\right).\end{aligned}$$

Therefore, by the previous lemma we have that the load of each machine is at most $C^* \cdot \max\{1/(1-\beta), 1+1/\beta\}$. The approximation factor is minimized when $1/(1-\beta) = 1+1/\beta$, hence $\beta = (-1+\sqrt{5})/2 = (1+\sqrt{5})/2 - 1 = \phi - 1$. Thus, the approximation ratio is $1+1/(\phi-1) = 1+\phi$. □

We close this section by showing that $1+\phi$ is the best approximation ratio achievable by $[\mathrm{LST_{strong}}]$.

Theorem 4. *For any $\varepsilon > 0$, there exists an instance such that $C^*(1+\phi-\varepsilon) \leq$ OPT, where C^* is the smallest number such that* $[\mathrm{LST_{strong}}]$ *is feasible.*

Proof. Consider the instance depicted in Fig. 1. It consists of two disjoint sets of jobs J and J'. Each job $j_\ell \in J$ forms a pair with its corresponding job $j'_\ell \in J'$. Each such pair is associated with a *parent machine* i_p^ℓ such that both j_ℓ and j'_ℓ can be processed on this machine with setup time $s_{i_p^\ell j_\ell} = s_{i_p^\ell j'_\ell} = \phi/2$ and $p_{i_p^\ell j_\ell} = p_{i_p^\ell j'_\ell} = 0$. Each job j of each pair is furthermore associated with a *child machine* $i_c(j)$ such that $s_{i_c(j)j} = 0$ and $p_{i_c(j)j} = \phi + 1$. □

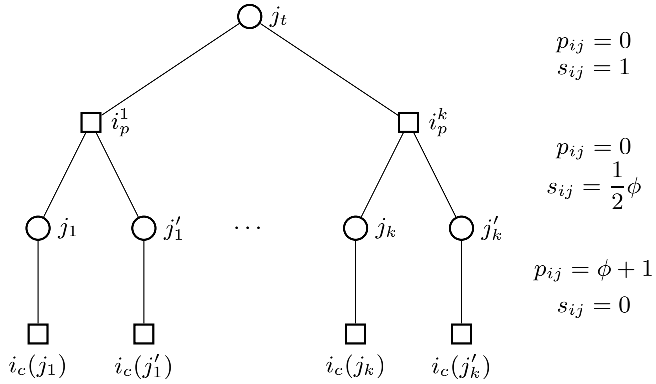

Fig. 1. Example showing that [$\text{LST}_{\text{strong}}$] has a gap of $1 + \phi$

4 A 2-Approximation Algorithm for Restricted Assignment

We also consider the restricted assignment case, where for every $j \in J$ there are values p_j and s_j such that $p_{ij} \in \{p_j, \infty\}$ and $s_{ij} \in \{s_j, \infty\}$ for all $i \in M$. For this setting we obtain an improved approximation ratio of 2, also based on rounding the [$\text{LST}_{\text{strong}}$] relaxation. After constructing the same graph $G(x)$, we distribute the processing requirement of each job to the machine corresponding to its child nodes. Although this might increase the processing requirement of a job on the child machines by more than a factor 2, we show that increasing the load of these machines by C^* suffices to completely process the job and its setup.

Theorem 5. *There exists a 2-approximation algorithm for scheduling splittable jobs on unrelated machines under restricted assignment.*

5 Hardness of Approximation

By reducing from MAX k-COVER, we derive an inapproximability bound of $e/(e-1) \approx 1.582$ for $R|\text{split,setup}|C_{\max}$, indicating that the problem might indeed be harder from an approximation point of view compared to the classic $R||C_{\max}$, for which 3/2 is the best known lower bound.

Theorem 6. *For any $\varepsilon > 0$, there is no $\left(\frac{e}{e-1} - \varepsilon\right)$-approximation algorithm for $R|split,setup|C_{\max}$ unless $\mathcal{P} = \mathcal{NP}$.*

6 A Job Configuration LP

A basic tool of combinatorial optimization is to design stronger linear programs based on certain configurations. These LPs often provide improved integrality gaps and thus lead to better approximation algorithms as long as they can be solved efficiently and be appropriately rounded. In machine scheduling the most widely used configuration LP uses as variables the possible configurations of jobs in a given machine. These *machine configuration* LPs have been successfully studied for the unrelated machine setting since the pioneering work of Bansal and Sviridenko [3]. Recent progress in the area includes [19,6,22,20].

Unfortunately, while there is a natural extension of the concept of machine configurations to $R|\text{split,setup}|C_{\max}$, this formulation surprisingly exhibits the same integrality gap of $1+\phi$ as already observed for $[\text{LST}_{\text{strong}}]$. Instead, we introduce a new family of configuration LPs, which we call *job configuration* LPs. A configuration f for a given job j specifies the fraction of j that is scheduled on each machine. The configuration consists of two vectors $x^f \in [0,1]^M$ and $y^f \in \{0,1\}^M$ such that $\sum_{i\in M} x_i^f = 1$ and $y_i^f = 1$ if and only if $x_i^f > 0$. On machine $i \in M$ configuration f requires time $t_i^f := p_{ij}x_i^f + s_{ij}y_i^f$. Let $\mathcal{F}_j$ be the set of configurations for job j with $t_i^f \le C$ for all $i \in M$. Then every feasible solution to $R|\text{split,setup}|C_{\max}$ with makespan C corresponds to an integer solution of

$$
\begin{aligned}
[\text{CLP}]: \qquad \sum_{f\in\mathcal{F}_j} \lambda_f &= 1 && \text{for all } j \in J,\\
\sum_{j\in J}\sum_{f\in\mathcal{F}_j} \lambda_f t_i^f &\le C && \text{for all } i \in M,\\
\lambda_f &\ge 0 && \text{for all } f \in \bigcup_{j\in J} \mathcal{F}_j.
\end{aligned}
$$

Note that this formulation has infinitely many variables. However, by investigating the separation problem of the convex dual of [CLP], we can show that we can restrict [CLP] without loss of generality to the finite subset of so-called maximal configurations. A configuration $f \in \mathcal{F}_j$ is *maximal*, if there is at most one machine $i \in M$ with $0 < x_i^f < x_{ij}^{\max}$, where $x_{ij}^{\max} := (C - s_{ij})/p_{ij}$.

Theorem 7. [CLP] *is feasible if and only if the restriction of* [CLP] *to maximal configurations is feasible.*

It can further be shown that after discretizing the configurations, the dual separation problem can be solved in polynomial time, implying that [CLP] can be solved efficiently up to a factor $(1+\varepsilon)$. Henceforth, we will restrict $\mathcal{F}_j$ to the set of maximal configurations for each job $j \in J$.

Projection of the Job Configuration LP. Observe that any convex combination of job configurations λ can be translated into a pair of vectors $x^\lambda, y^\lambda \in [0,1]^{M\times J}$ in the assignment space by setting

$$x_{ij}^\lambda := \sum_{f\in\mathcal{F}_j} \lambda_f x_i^f \quad \text{and} \quad y_{ij}^\lambda := \sum_{f\in\mathcal{F}_j} \lambda_f y_i^f.$$

We show that applying this projection to [CLP] leads to assignment vectors described by the following set of inequalities:

$$[\text{CLP}_{\text{proj}}]: \quad \sum_{j \in J}(p_{ij}x_{ij} + s_{ij}y_{ij}) \leq C \qquad \text{for all } \ i \in M, \tag{10}$$

$$\sum_{i \in M}(\beta_i x_{ij} + \gamma_i y_{ij}) \geq M(j, \beta, \gamma) \quad \text{for all } \ j \in J, \ \beta, \gamma \in \mathbb{R}^M, \tag{11}$$

with $M(j, \beta, \gamma) := \min \left\{ \sum_{i \in M}(\beta_i x_i^f + \gamma_i y_i^f) \ : \ f \in \mathcal{F}_j \right\}$.

Theorem 8. *If* $\lambda \in [\text{CLP}]$ *then* $(x^\lambda, y^\lambda) \in [\text{CLP}_{\text{proj}}]$. *Conversely, if* $(x, y) \in [\text{CLP}_{\text{proj}}]$ *then there exists* $\lambda \in [\text{CLP}]$ *such that* $x = x^\lambda$ *and* $y = y^\lambda$.

We conclude by showing that already a very special class of $[\text{CLP}_{\text{proj}}]$-inequalities is sufficient to eliminate the gap in the worst-case instances of $[\text{LST}_{\text{strong}}]$. For a set of machines $S \subseteq M$ let $L(j, S) := \sum_{i \in M \setminus S} \max \left\{ \frac{C - s_{ij}}{p_{ij}}, 0 \right\}$ be the maximum fraction of job j that can be processed within time C by the machines in $M \setminus S$. The following inequalities are satisfied by the vector x, y induced by any feasible solution to $R|\text{split,setup}|C_{\max}$ with makespan at most C.

$$\frac{\sum_{i \in S'} x_{ij}}{1 - L(j, S \cup S')} + \sum_{i \in S} y_{ij} \geq 1 \quad \text{for all } j \in J \text{ and } S, S' \subseteq M \text{ with } L(j, S \cup S') < 1.$$

Interestingly, these inequalities can be seen as a special case of inequalities (11) by setting $\beta_i = \frac{1}{1 - L(j, S \cup S')}$ for $i \in S'$ and $\gamma_i = 1$ for $i \in S$. Furthermore, consider the example instance given in the proof of Theorem 4 (cf. Fig. 1). If $C < 1 + \phi$, then $L(j, \{i_p(j)\}) = C/p_{i_c(j)j} < 1$ and therefore $y_{i_p(j)j} = 1$ for all $j \in J \cup J'$ in any feasible solution to $[\text{CLP}_{\text{proj}}]$. This immediately implies infeasibility of $[\text{CLP}_{\text{proj}}]$ for $C < 1 + \phi$. We also note that the exact same argument applies to the worst-case instance of the machine configuration LP.

References

1. Allahverdi, A., Ng, C., Cheng, T., Kovalyov, M.: A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. 187, 985–1032 (2008)
2. Asadpour, A., Feige, U., Saberi, A.: Santa claus meets hypergraph matchings. ACM Trans. Algorithms 24, 24:1–24:9 (2012)
3. Bansal, N., Sviridenko, M.: The Santa Claus problem. In: STOC, pp. 31–40 (2006)
4. Chen, B., Ye, Y., Zhang, J.: Lot-sizing scheduling with batch setup times. J. Sched. 9, 299–310 (2006)
5. Correa, J.R., Verdugo, V., Verschae, J.: Approximation algorithms for scheduling splitting jobs with setup times. In: Talk in MAPSP (2013)
6. Ebenlendr, T., Krčál, M., Sgall, J.: Graph balancing: A special case of scheduling unrelated parallel machines. Algorithmica (2012), doi:10.1007/s00453-012-9668-9
7. Feige, U.: On allocations that maximize fairness. In: SODA, pp. 287–293 (2008)
8. Graham, R., Lawler, E., Lenstra, J., Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math. 5, 287–326 (1979)

9. Haeupler, B., Saha, B., Srinivasan, A.: New constructive aspects of the Lovász Local Lemma. J. ACM 58, 28:1–28 (2011)
10. Kim, D.-W., Na, D.-G., Frank Chen, F.: Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. Robot. Com. -Int. Manuf. 19, 173–181 (2003)
11. Lenstra, J.K., Shmoys, D.B., Tardos, E.: Approximation algorithms for scheduling unrelated parallel machines. Math. Program. 46, 259–271 (1990)
12. Liu, Z., Cheng, T.C.E.: Minimizing total completion time subject to job release dates and preemption penalties. J. Sched. 7, 313–327 (2004)
13. Lovász, L., Schrijver, A.: Cones of matrices and set-functions and 0-1 optimization. SIAM J. Optimiz. 1, 166–190 (1991)
14. Polacek, L., Svensson, O.: Quasi-polynomial local search for restricted max-min fair allocation. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part I. LNCS, vol. 7391, pp. 726–737. Springer, Heidelberg (2012)
15. Potts, C.N., Wassenhove, L.N.V.: Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. J. Oper. Res. Soc. 43, 395–406 (1992)
16. Schalekamp, F., Sitters, R., van der Ster, S., Stougie, L., Verdugo, V., van Zuylen, A.: Split scheduling with uniform setup times. Arxiv (2012)
17. Schuurman, P., Woeginger, G.J.: Preemptive scheduling with job-dependent setup times. In: SODA, pp. 759–767 (1999)
18. Serafini, P.: Scheduling jobs on several machines with the job splitting property. Oper. Res. 44, 617–628 (1996)
19. Svensson, O.: Santa claus schedules jobs on unrelated machines. SIAM J. Comput. 41, 1318–1341 (2012)
20. Sviridenko, M., Wiese, A.: Approximating the configuration-lp for minimizing weighted sum of completion times on unrelated machines. In: Goemans, M., Correa, J. (eds.) IPCO 2013. LNCS, vol. 7801, pp. 387–398. Springer, Heidelberg (2013)
21. van der Ster, S.: The allocation of scarce resources in disaster relief. MSc-Thesis in Operations Research at VU University Amsterdam (2010)
22. Verschae, J., Wiese, A.: On the configuration-LP for scheduling on unrelated machines. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 530–542. Springer, Heidelberg (2011)
23. Williamson, D.P., Shmoys, D.B.: The Design of Approximation Algorithms. Cambridge University Press (2011)
24. Xing, W., Zhang, J.: Parallel machine scheduling with splitting jobs. Discrete Appl. Math. 103, 259–269 (2000)