

Weighted Completion Time Minimization for Unrelated Machines via Iterative Fair Contention Resolution*

[Extended Abstract]

Sungjin Im[†]

Maryam Shadloo[‡]

Abstract

We give a 1.488-approximation for the classic scheduling problem of minimizing total weighted completion time on unrelated machines. This is a considerable improvement on the recent breakthrough of $(1.5 - 10^{-7})$ -approximation (STOC 2016, Bansal-Srinivasan-Svensson) and the follow-up result of $(1.5 - 1/6000)$ -approximation (FOCS 2017, Li). Bansal et al. introduced a novel rounding scheme yielding strong negative correlations for the first time and applied it to the scheduling problem to obtain their breakthrough, which resolved the open problem if one can beat out the long-standing 1.5-approximation barrier based on independent rounding. Our key technical contribution is in achieving significantly stronger negative correlations via *iterative fair contention resolution*, which is of independent interest. Previously, Bansal et al. obtained strong negative correlations via a variant of pipage type rounding and Li used it as a black box.

1 Introduction

The unrelated machines setting is a classic scheduling model that has been widely used to model fully heterogeneous parallel machines. In this setting, there is a set $\mathcal{M}$ of m machines and a set $\mathcal{J}$ of n jobs to be scheduled on the machines. Machines are unrelated in the sense that each job $j \in \mathcal{J}$ has an arbitrary size/processing time p_{ij} on each machine $i \in \mathcal{M}$. Further, each job j has weight w_j .¹ In this paper we consider non-preemptive—and therefore non-migratory—scheduling, which means that each job j must be executed without interruption on one of the machines. In this paper we seek to optimize one of the most popular objectives, namely minimizing total weighted completion time, i.e., $\sum_{j \in \mathcal{J}} w_j C_j$ where C_j denotes job j 's completion

time.

The problem we consider, denoted as $R||\sum_j w_j C_j$ using the common three-field notation, is known to be strongly NP-hard and APX-hard [22]. For this problem, more than fifteen years ago, Schulz and Skutella [33] gave an $1.5 + \epsilon$ -approximation based on a time indexed LP. Later, Skutella [36] and Sethuraman and Squillante [35] gave 1.5-approximations based on novel convex programming. It had been a long-standing open problem whether there exists a better than 1.5-approximation for the problem [13, 33, 28, 40, 34] until it was recently answered by Bansal et al. [8] in the affirmative.

The breakthrough by Bansal et al. had two important technical ingredients. First they introduced a novel SDP (semi-definite programming) to capture the pairwise interaction between jobs. Intuitively, this is important as the weighted completion time objective is not linear in job sizes. This is because a job j' can delay another job j if j' starts its execution before j on the same machine. Another contribution was developing a novel rounding scheme. All the aforementioned previous works used an independent rounding that randomly assigns each job j to machine i with probability x_{ij} , which is obtained by solving linear or convex programming. Since the 1.5-approximation factor is the best one can hope for using independent rounding, they had to develop a new rounding scheme. Their rounding not only ensures negative correlation between any pair of jobs assigned to the same machine but also ensures that any pair of jobs grouped together on the same machine subject to a capacity constraint are *strongly negatively* correlated. Their rounding was based on a variant of pipage style rounding. Using a SDP relaxation and the new rounding with a delicate grouping of jobs, they were able to obtain a $1.5 - 10^{-7}$ -approximation.

Later, Li observed that a time-indexed LP can be used instead of a SDP [31]. A time-indexed LP (fractionally) encodes when each job starts and ends on each machine. Using the special structure of the time-indexed LP solution, he was able to use Bansal et al.'s strong negative correlation rounding with a different grouping and obtained a better $1.5 - 1/6000$ -approximation.

Up to date, the only way to obtain a better than 1.5-

*Supported in part by NSF grants CCF-1409130, 1617653, and 1844939.

[†]Electrical Engineering and Computer Science, University of California, 5200 N. Lake Road, Merced CA 95344. sim3@ucmerced.edu.

[‡]Electrical Engineering and Computer Science, University of California, 5200 N. Lake Road, Merced CA 95344. mshadloo@ucmerced.edu.

¹We can handle more general weights w_{ij} , which depend on machines, but we assume each job's weight is the same on all machines following the convention.

approximation for $R|| \sum_j w_j C_j$ has been based on using the novel dependent rounding scheme by Bansal et al., which introduced the notion of strong negative correlations for the first time.

1.1 Our Results

THEOREM 1.1. *For minimizing total weighted completion time on unrelated machines, $R|| \sum_j w_j C_j$, there exists a randomized 1.488-approximation.*

As mentioned, this is a considerable improvement over the previous approximation ratios, $1.5 - 10^{-7}$ and $1.5 - 1/6000$ if we measure the improvement by the margin over the 1.5-approximations [36, 35] that are based on independent rounding. The improvement primarily comes from our new randomized dependent rounding that achieves strong negative correlations. We formally state what our randomized rounding guarantees as follows. We defer to Section 1.3 an overview of how we apply this theorem to $R|| \sum_j w_j C_j$, which is similar at a high-level to how Li [31] applied Bansal et al.'s strong negative correlation rounding to the same scheduling problem.

THEOREM 1.2. *Suppose we are given a set $\mathcal{M}$ of machines and a set $\mathcal{J}$ of jobs together with a fractional assignment $\{x_{ij}\}_{i \in \mathcal{M}, j \in \mathcal{J}}$ of jobs to machines (meaning that job j is assigned to machine i by x_{ij}) such that $x_{ij} \in [0, 1]$ for all $i \in \mathcal{M}, j \in \mathcal{J}$ and $\sum_{i \in \mathcal{M}} x_{ij} = 1$ for all $j \in \mathcal{J}$. For each machine $i \in \mathcal{M}$, select any family $\mathcal{G}_i$ of disjoint subsets of jobs such that $\sum_{j \in G} x_{ij} \leq 1$ for all $G \in \mathcal{G}_i$. Then, there is a randomized rounding algorithm that achieves the following properties:*

1. (Feasible Integer Assignment) Each job $j \in \mathcal{J}$ is assigned to exactly one machine $i \in \mathcal{M}$, which is denoted as $i \leftarrow j$.
2. (Preserving Marginal Probabilities) For every $i \in \mathcal{M}$ and $j \in \mathcal{J}$, $\Pr[i \leftarrow j] = x_{ij}$.
3. (Negative Correlation) For every $i \in \mathcal{M}$ and $j \neq j' \in \mathcal{J}$ such that no $G \in \mathcal{G}_i$ has j and j' simultaneously, $\Pr[i \leftarrow j \wedge i \leftarrow j'] \leq x_{ij}x_{ij'}$.
4. (Strong Negative Correlation²) For every $i \in \mathcal{M}$ and $j \neq j' \in \mathcal{J}$ such that $j, j' \in G$ for some $G \in \mathcal{G}_i$, $\Pr[i \leftarrow j \wedge i \leftarrow j'] \leq \frac{1}{1+e}(e^{x_{ij}} + e^{x_{ij'}})x_{ij}x_{ij'} \leq x_{ij}x_{ij'}$.

The algorithm terminates in $O(mn \log n)$ time in expectation and w.h.p.

²As noted in [8], it is impossible to impose strong negative correlation on every pair of jobs; thus, it was suggested to obtain strong negative correlations only between jobs in the same group.

The theorem states the properties our randomized rounding guarantees when rounding a fractional assignment of jobs to machines to an integer assignment. The first property says the rounding always outputs a feasible integer assignment when it terminates. The second property ensures that each job j is assigned to machine i with probability exactly x_{ij} , thus preserving the marginal probabilities. The third property says that two jobs $j \neq j'$ that are not grouped together on machine i are assigned to the same machine i with probability at most $x_{ij}x_{ij'}$; thus, the assignments are negatively correlated. All these properties can be achieved by simple independent rounding.

The last property, which is most interesting, says that for any two jobs grouped together on machine i , the probability they are simultaneously assigned to machine i is at most $\frac{\exp(x_{ij}) + \exp(x_{ij'})}{e+1} x_{ij}x_{ij'} \leq x_{ij}x_{ij'}$ as $x_{ij} + x_{ij'} \leq 1$. So, if $x_{ij} + x_{ij'}$ is close to 1, the theorem doesn't guarantee strong negative correlation. However, it seems that we need strong negative correlation only when x_{ij} and $x_{ij'}$ are sufficiently small, as will be discussed shortly. In the best scenario, we have $\lim_{x_{ij}, x_{ij'} \rightarrow 0} \frac{\exp(x_{ij}) + \exp(x_{ij'})}{e+1} = \frac{2}{e+1} < 0.5379$.

We note that Theorem 1.2 is identical to Theorem 1.2 in [8] except that they have $\frac{107}{108}x_{ij}x_{ij'}$ instead of $\frac{1}{1+e}(e^{x_{ij}} + e^{x_{ij'}})x_{ij}x_{ij'}$ as the upper bound of the fourth property. So, they have strong negative correlations regardless of value of $x_{ij} + x_{ij'}$. However, as alluded above, Bansal et al. grouped j and j' together on machine i only when $x_{ij}, x_{ij'} \leq 1/10$ and Li [31] did only when $x_{ij}, x_{ij'} \leq 1/100$. Under their grouping of jobs, the coefficient of $x_{ij}x_{ij'}$ in the fourth property of Theorem 1.2 becomes $\frac{2 \exp(0.1)}{e+1} \simeq 0.5945$ and $\frac{2 \exp(0.01)}{e+1} \simeq 0.5433$, respectively, meaning much stronger negative correlations when the theorem is actually needed.

Our method to achieve strong negative correlations is completely different from the method of Bansal et al. They used a very clever variant of pipage rounding. In contrast, we use a novel *iterative fair contention resolution*. While our method is inspired by the fair contention resolution scheme by Feige and Vondrák [17], our method is considerably different and perhaps easier to understand, which we believe enables the iterative application of fair contention resolution. We discuss this in more detail in the following subsection.

1.2 Our Techniques We first discuss two previous works that are most closely related to our new rounding (Theorem 1.2).

A Variant of Pipage Rounding [8]. As mentioned before, Bansal et al. [8] achieved the first strong negative correlations using a variant of pipage rounding. Note that the fractional assignment $\{x_{ij}\}_{i \in \mathcal{M}, j \in \mathcal{J}}$ described in Theorem 1.2 can be thought of as a fractional matching of a bipartite graph over $(\mathcal{M}, \mathcal{J})$ saturating all jobs in $\mathcal{J}$. While there are many variations and extensions [2, 3, 14, 18, 25], a typical pipage

rounding works as follows: In every iteration, the rounding finds a path or cycle only consisting of edges (i, j) with fractional values, i.e., $x_{ij} \in (0, 1)$. Then, it either maximally increases the weight of all odd (even, resp.) edges and simultaneously decreases the weight of all even (odd, resp.) edges by the same maximum margin—this choice is made at random in a way to preserve the marginal probabilities. This update makes at least one more edge have value either 0 or 1; and the value of such edges remains fixed afterwards. Intuitively, two edges (i, j) and (i, j') incident to the same machine i are chosen with negative correlation because the rounding never increases x_{ij} and $x_{ij'}$ simultaneously. To obtain strong negative correlations additionally, in each iteration Bansal et al. carefully chose paths of length 4 based on a random 2-coloring of the edges.

Fair Contention Resolution [17]. While our theorem statement is very similar to Bansal et al.’s corresponding theorem in [8], our rounding scheme is completely different and is inspired by the fair contention resolution by Feige and Vondrák [17]. To obtain a better than $1 - 1/e$ -approximation for the Maximum Submodular Welfare problem³ and related problems, they developed a novel fair contention resolution scheme: Suppose each player a claims item b independently with probability $q_{a,b}$. Now we need to assign each item that was claimed by one or more players to exactly one of them. They gave an elegant contention resolution scheme where each player a gets an item b with probability equal to $\frac{1 - \prod_{a' \neq a} (1 - q_{a',b})}{\sum_{a'} q_{a',b}}$, conditioned on a having claimed b . In the Maximum Submodular Welfare problem, if each player claims a subset of items according to her own distribution, the approximation guarantee was shown to be determined by the lowest probability that the player receives item b conditioned on her having claimed it. The contention resolution scheme attempts to maximize the lowest probability across all players and items; thus, the name fair contention resolution comes.

As an attempt to obtain a theorem like Theorem 1.2 using the fair contention resolution scheme, think of each group G defined in Theorem 1.2 as a player. Then, we let each group $G \in \mathcal{G}_i$ claim a job $j \in G$ with probability x_{ij} independently, hoping that this will help two different jobs in G less likely be assigned to the same machine i with the aid of the fair contention resolution. The resolution scheme guarantees that j is assigned to machine i with probability at least $\frac{1 - \prod_{i' \neq i} (1 - x_{i'j})}{\sum_{i'} x_{i'j}} x_{ij} \geq (1 - \prod_{i'} e^{-x_{i'j}}) x_{ij} = (1 - 1/e) x_{ij}$. This seems like a good sign as the contention resolution preserves each group’s choice up to $1 - 1/e$ factor.

Unfortunately, there are several issues. First, their rounding satisfies none of the desired properties claimed

in Theorem 1.2. It is not difficult to modify the scheme to satisfy the first two properties. However, their rounding scheme has no guarantees on the third and fourth properties. At a high-level, their algorithm focuses on the best guarantee on the first moment (recall that their goal was to give a better than $1 - 1/e$ -approximation for some assignment problems) and their analysis is remarkably accurate. However, because of the very reason their algorithm and analysis don’t seem to readily extend to satisfy negative and strong negative correlations.

Our Approach: Iterative Continuous Fair Contention Resolution. We develop a new contention resolution scheme of a continuous flavor which we believe is perhaps more intuitive.⁴ Thus, while the analysis is non-trivial, we are able to analyze its iterative application, thereby obtaining Theorem 1.2.

Here, we sketch how we develop our fair contention resolution along with the intuitions behind. To gain some intuitions, let’s first focus on each job j . Instead of assigning j to a machine i with probability x_{ij} as in independent rounding, we would like to have machine i claim job j , taking other jobs into account, with probability x_{ij} in order to have a better control on jobs assigned to each machine. Then, an obvious issue is that job j may be claimed by multiple machines (or by no machine). Therefore, we need to resolve this contention for job j among machines. Towards this end, we generate N_{ij} (lottery) tickets for job j on machine i , where $N_{ij} \sim \text{Pois}(x_{ij})$, a Poisson distribution with mean value x_{ij} . Note that this is equivalent to generating one ticket for job j on machine i with probability ϵ independently, for each ϵ unit of x_{ij} . Thus, this way of generating tickets allows us to view the problem more continuously. If no tickets are generated for j across machines, which happens with probability $1/e$, job j is not assigned; otherwise, we choose one ticket of job j uniformly at random and assign the job to the machine from which the ticket originated. It is an easy exercise to see that j is assigned to machine i with probability exactly $(1 - 1/e)x_{ij}$.

Now, we want to impose strong negative correlations between jobs grouped together on each machine. Towards this end, we let each group $G \in \mathcal{G}_i$ recommend one job j (or none) with probability x_{ij} (if $j \in G$); here we use the fact that $\sum_{j \in G} x_{ij} \leq 1$. The idea is to ensure that j has tickets on machine i only when G recommends job j – by doing so, if $j, j' \in G$, j having tickets on i will be negatively correlated with j' having tickets on i . To continue to have the above nice contention resolution based on tickets, our goal is to ensure $N_{ij} \sim \text{Pois}(x_{ij})$ and $N_{ij} = 0$ unless G

³The goal of the Maximum Submodular Problem is to allocate items to players so to maximize the total utility where each player has a monotone submodular utility function.

⁴However, our method doesn’t give as strong guarantee on the first moment as [17]. That is, using our method, each job j is assigned to machine i with probability exactly $(1 - 1/e)x_{ij}$ (in the first iteration). In contrast, the probability can be strictly greater in the method of [17] when $\{x_{ij}\}_i$ are not all tiny.

recommends j on machine i . To achieve this, we use a simple trick. Let $\tilde{N}_{ij}$ denote the number of *potential* tickets sampled from a distribution whose probability mass for each value $k > 0$ is exactly $1/x_{ij}$ times that of $\text{Pois}(x_{ij})$. Then, we set $N_{ij} := B_{ij} \cdot \tilde{N}_{ij}$ and have $N_{ij} \sim \text{Pois}(x_{ij})$. Here, B_{ij} is an indicator random variable that has value one iff j is recommended by the group where it belongs. In words, j has $N_{ij} = \tilde{N}_{ij}$ *real* tickets only when $B_{ij} = 1$ occurs.

Using the above observation that each job is assigned to some machine with probability exactly $1 - 1/e$, we can repeat the whole process, excluding jobs that have already been assigned, until all jobs are assigned.

The actual proof of the third and fourth properties is quite non-trivial, particularly the third property. This is because the random process of where two jobs j and j' are assigned depends on whether the two jobs are grouped together on each machine and how much they are assigned in the fractional solution. At a high-level, we show that the worst case for us happens when the two jobs j and j' are not grouped together on any machines possibly except on machine i —then the proof becomes relatively easy. To prove this we take a sequence of careful steps conditioning and deconditioning on some random variables. Perhaps proving negative correlations of our method is significantly more challenging than proving negative correlations of pipage rounding because our method assigns $(1 - 1/e)$ -fraction of remaining jobs in each iteration unlike pipage rounding that assigns one job wlog in each iteration. Thus, we need to take a global view of the random process considering how each pair of jobs are grouped on all machines.

1.3 Applying Theorem 1.2 to the Scheduling Problem

To obtain Theorem 1.1 by applying Theorem 1.2 to our scheduling problem, we borrow some important ideas from Li's approach [31]. Here we give a sketch of our rounding and briefly discuss the difference between our approach and his. The time-indexed integer programming (IP) has an indicator variable x_{ijs} which is 1 if and only if j starts its execution on machine i at time s . Note that since we consider non-preemptive scheduling, if $x_{ijs} = 1$, then j completes at time $s + p_{ij}$. Let T be a sufficiently large upper bound on the number of time steps we need to consider. We assume T is polynomially bounded in the input size since it was shown that this assumption is wlog with a loss of $(1 + \epsilon)$ factor in the approximation ratio [24]. The time-indexed IP considered in [31] is presented below.

Here, the first constraint ensures that every job is scheduled on some machine. The second constraint ensures that every machine processes at most one job at each time. The third constraint enforces that all jobs must complete by time T . By relaxing the last constraint into $x_{ijs} \geq 0$, we obtain a valid LP relaxation. Let x denote the optimal LP solution. Then, for each $x_{ijs} > 0$, it will be convenient to think that we

have a rectangle of height x_{ijs} starting at time s and ending at time $s + p_{ij}$, denoted as R_{ijs} .

$$(1.1) \quad \min \sum_{j \in \mathcal{J}} w_j \sum_{i \in \mathcal{M}, s \in [T]} x_{ijs}(s + p_{ij})$$

$$(1.2) \quad \sum_{i \in \mathcal{M}, s \in [T]} x_{ijs} = 1 \quad \forall j \in \mathcal{J}$$

$$(1.3) \quad \sum_{j \in \mathcal{J}, s \in (t - p_{ij}, t]} x_{ijs} \leq 1 \quad \forall i \in \mathcal{M}, t \in [T]$$

$$(1.4) \quad x_{ijs} = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{J}, s > T - p_{ij}$$

$$(1.5) \quad x_{ijs} \in \{0, 1\} \quad \forall i \in \mathcal{M}, j \in \mathcal{J}, s \in [T]$$

We first review how the independent rounding gives a 1.5-approximation: for each job j , we choose a rectangle R_{ijs} independently with probability x_{ijs} —then, j is assigned to machine i . Next, we sample a random offset τ_j for each job uniformly at random from $[0, p_{ij}]$. Then, we set $\theta_j = \tau_j + s$ conditioned on R_{ijs} being chosen for j . Now schedule jobs assigned to the same machine in increasing order of θ_j . To upper bound the expected completion time of job j , we need to know the expected size of jobs that are assigned to i and have smaller θ values than job j . Due to the linearity of expectation, we can focus on the expected size of each job j' that is scheduled prior to j on machine i , which we call the expected delay j' causes to job j on machine i . Thanks to the uniform choice of θ, τ values, one can show that fixing θ_j , the expected delay j' causes to job j on machine i is exactly the area of rectangle $R_{ij's'}$ up to time θ_j , if $R_{ij's'}$ is chosen for j' . This is because the probability that j' has a smaller θ value than j on the condition is exactly $\frac{\max(\theta_j - s', 0)}{p_{ij'}}$, which is exactly the horizontal length of $R_{ij's'}$ up to the time θ_j , divided by $p_{ij'}$. Since the total area of rectangles till time θ_j is at most $1 \cdot \theta_j = \theta_j$ (recall at most one unit of job can be scheduled at each time), we have $\mathbb{E}[C_j \mid \theta_j, R_{ijs}] \leq \theta_j + p_{ij}$. Knowing that $\mathbb{E}[\theta_j \mid R_{ijs}] = s + p_{ij}/2$, we have $\mathbb{E}[C_j \mid R_{ijs}] \leq s + 1.5p_{ij}$. Further, de-conditioning on the choice of j 's rectangle, we have $\mathbb{E}[C_j] \leq \sum_{i,j,s} x_{ijs}(s + 1.5p_{ij})$, which immediately gives a 1.5-approximation due to the linearity of expectation.

Now we sketch how we use strong negative correlations to obtain a better than 1.5-approximation. For the purpose of better intuitions, let's assume that there is *at most* one rectangle R_{ijs} with $x_{ijs} > 0$ for every pair of job j and machine i . In other words, assume that each job has at most one rectangle on machine i . Further, assume all jobs have weight 1. Note that in the upper bound of $\mathbb{E}[C_j \mid R_{ijs}]$, the coefficient of s is just 1. Therefore, if s is considerably big compared to p_{ij} , then job j is an easy job on machine i towards obtaining a better than 1.5-approximation. Another case of job j being easy on machine i is when x_{ijs} is large.

This is because in fact we actually have a better upper bound of $\mathbb{E}[C_j \mid \theta_j, R_{ijs}] \leq \theta_j + p_{ij} - \tau_j x_{ijs}$. The reason why we get the extra negative term is as follows: Recall that the expected delay other jobs cause to job j on machine i is upper bounded by θ_j , which is a clear upper bound on the total area of rectangles of other jobs up to time θ_j . Here, we can take off the area of R_{ijs} of job j before time θ_j , which is exactly $(\theta_j - s)x_{ijs} = \tau_j x_{ijs}$. Roughly speaking, the rectangle R_{ijs} of each bad job j starts near time 0 and has small height x_{ijs} . This was one of the key observations made by Li [31] although the definition of easy jobs is slightly different from ours.

To obtain a better than 1.5-approximation, we need to use Theorem 1.2 (or the corresponding theorem in [8]). To handle bad jobs on machine i using Theorem 1.2, we need to group them carefully. At a high level, we group jobs of similar θ values. That is, we first sample θ value for each rectangle of a bad job, and we group jobs if their θ values fall into the same time interval, which is one of the intervals of exponentially increasing length partitioning the whole time horizon. Using the fact that x_{ijs} is small for bad jobs and the rectangle of bad jobs starts near time 0, with some care, we can bound the total x value of jobs in each group, which is needed to apply Theorem 1.2. Then, thanks to strong negative correlations, when two jobs j and j' have similar θ values, they are less likely to be assigned to the same machine i with a good probability, which is enough to give a better than 1.5-approximation.

While our rounding is different from Li's in many places, the two main differences are as follows. First, we use a random partition of time horizon into the intervals of exponentially increasing lengths whileas Li used a deterministic partition. This is because we found the random partition seemed to give a better grouping of jobs. Further, for analysis, we conceptually group non-overlapping rectangles, so that we have a linear combination of subsets of non-overlapping rectangles, which looks like a solution to the configuration LP [40]. Using this structure also helps to improve the approximation ratio slightly. However, as mentioned before, the improvement of approximation ratio primarily comes from our stronger negative correlations.

1.4 Other Related Work Minimizing total (weighted) completion time is one of the most popular scheduling objectives considered in the literature. For the single machine case, the algorithm highest-density-first, which favors jobs of highest w_j/p_j , is known to be optimal [38]. The problem becomes NP-hard when there are multiple machines [19]. However, when machines are identical ($P \parallel \sum_j w_j C_j$) or uniformly related ($Q \parallel \sum_j w_j C_j$), the problem admits PTASes [1, 37, 12]. Interestingly, even when machines are unrelated, if the objective is to minimize total unweighted completion time ($R \parallel \sum_j C_j$), the problem is

polynomially solvable using a min-cost bipartite matching [23, 10]. Kalaitzis et al. [26] considered an important special case when each job's processing time is proportional to its weight, i.e., $R \parallel w_{ij}/p_{ij} = 1 \parallel \sum_{i,j} w_{ij} C_j$, and gave a 1.21-approximation. Interestingly, their result also achieves a bi-criteria 2-approximation for the makespan objective. A configuration LP, which encodes all possible schedules on each machine, was shown to be solvable optimally within $(1 + \epsilon)$ -factor [40]; see [24] for a discussion on the strength of the configuration LP. The configuration LP was shown to have an integrality gap of at least 1.08 [26]. If jobs have different arrival times, the problem, $1 \parallel r_j \mid \sum_j w_j C_j$, is NP-hard [29] even in the single machine case. When machines are identical ($P \parallel r_j \mid \sum_j w_j C_j$) or related ($Q \parallel r_j \mid \sum_j w_j C_j$), the problem admits PTASes [1, 12]. When machines are unrelated ($R \parallel r_j \mid \sum_j w_j C_j$), 2-approximation [33, 36] had been the best approximation known for long until it was recently improved to 1.8687-approximation [24].

For the makespan objective on unrelated machines, i.e., $R \parallel \max_j C_j$, a classic 2-approximation is known [30]. There have been considerable efforts to improve this ratio for some special cases, e.g. [39, 15]. For the dual objective of maximizing the minimum total load of all machines, see [9, 5, 4, 11, 16]. For $R \parallel (\sum_j (C_j)^k)^{1/k}$, see [6, 27]. For $R \parallel r_j \mid \sum_j (C_j - r_j)$, a poly-logarithmic approximation is known [7]. For the special case of $R \parallel r_j, p_{ij} \in \{p_j, \infty\} \mid \sum_j (C_j - r_j)$, see [20, 21]. For a survey of other approximate scheduling results, see [13].

1.5 Organization We present our iterative fair contention resolution scheme in Section 2. Next, in Section 3, we show that the rounding scheme satisfies all the properties claimed in Theorem 1.2. We present our randomized rounding algorithm for the unrelated machines scheduling problem in Section 4. We give the analysis of the algorithm in Section 5, thereby proving Theorem 1.1.

2 Rounding Procedure Yielding Strong Negative Correlations

2.1 Preliminaries To describe our randomized rounding we need to introduce a probability distribution. Let $\text{Pois}(\lambda)$ denote the Poisson distribution with mean value λ . Recall that $\text{Pois}(\lambda)$ has pmf, $e^{-\lambda} \frac{\lambda^k}{k!}$, over $k \in \mathbb{Z}^+ := \{0, 1, 2, \dots\}$. The following is a well-known property of Poisson distribution.

FACT 2.1. *For any set of independent random variables $\{Z_h \sim \text{Pois}(\lambda_h)\}_{h \in [H]}$, it is the case that $\sum_{h \in [H]} Z_h \sim \text{Pois}(\sum_{h \in [H]} \lambda_h)$.*

For any $\lambda > 0$, let $\widetilde{\text{Pois}}(\lambda)$ denote the probability distribution with the following pmf:

$$\begin{cases} e^{-\lambda} \frac{\lambda^{k-1}}{(k-1)!} & \text{if } k \in \{1, 2, 3, \dots\} \\ 1 - \frac{1-e^{-\lambda}}{\lambda} & \text{otherwise, i.e., } k = 0 \end{cases}$$

This pmf is well-defined since $\sum_{k \geq 1} e^{-\lambda} \frac{\lambda^k}{k!} = \frac{1-e^{-\lambda}}{\lambda} \leq 1$.⁵ Note that $\widetilde{\text{Pois}}(\lambda)$'s probability mass for each value $k > 0$ is $1/\lambda$ times that of $\text{Pois}(\lambda)$.

OBSERVATION 1. Let $\tilde{N} \sim \widetilde{\text{Pois}}(\lambda)$ and B be a Bernoulli (0-1) random variable with mean value λ . Then, $\tilde{N} \cdot B \sim \text{Pois}(\lambda)$.

2.2 Rounding Algorithm We are now ready to describe our randomized rounding. To simplify notation, for every $i \in \mathcal{M}$ and $j \in \mathcal{J}$, if $j \notin G$ for all $G \in \mathcal{G}_i$, then we create a singleton set of job j and add it to $\mathcal{G}_i$. Note that this has no effect on the properties we aim to prove as we do not change the existing groups. So, we can assume wlog that $\bigcup_{G \in \mathcal{G}_i} G = \mathcal{J}$ for all $i \in \mathcal{M}$.

Our randomized rounding is iterative. In the first iteration we perform as described in Figure 2.2. In principle, we do not need to generate potential tickets for job j on machine i unless $B_{ij} = 1$. However, the description where we first generate potential tickets independently for every pair of job j and machine i makes the analysis more intuitive. For brevity, we will interchangeably use B_{ij} and $B_{ij} = 1$.

We now explain how the rounding works in the subsequent iterations, $2, 3, \dots$. Every job assignment is *final*. If job j is assigned to machine i in the ℓ -th iteration (denoted as $i \leftarrow_{\ell} j$) the job is never considered in the subsequent iterations, $\ell + 1, \ell + 2, \dots$. Let $\mathcal{J}_{\leq \ell}$ denote the set of jobs that were assigned in iterations $1, 2, \dots, \ell$, i.e., $\mathcal{J}_{\leq \ell} := \{j \in \mathcal{J} \mid i \leftarrow_{\ell'} j \text{ for some } i \in \mathcal{M}, \ell' \leq \ell\}$. At the beginning of the $\ell + 1$ -th iteration, we update $\mathcal{J}$ to $\mathcal{J} \setminus \mathcal{J}_{\leq \ell}$ and every set $G \in \mathcal{G}_i$ to $G \setminus \mathcal{J}_{\leq \ell}$. Then, we perform the above four steps—all the RVs used in this iteration are different from those used in the previous iterations. The ℓ -th iteration begins only when there exists a job that hasn't been assigned yet, i.e., $\mathcal{J}_{< \ell} \neq \mathcal{J}$. This completes the description of our randomized rounding.

3 Proof of Theorem 1.2

This section is devoted to proving Theorem 1.2. We first make an easy observation which will be useful in the analysis. Since we renew all RVs in each iteration, we add superscript ℓ to RVs if they are of the ℓ -th iteration—for example, N_{ij}^{ℓ} denotes the RV in the ℓ -th iteration corresponding to N_{ij} . For RVs of the first iteration, we omit the superscript.

OBSERVATION 2. Conditioned on no jobs in $\mathcal{J}' \subseteq \mathcal{J}$ having been assigned in the previous iterations, the stochastic process of assigning $\mathcal{J}'$ is identical to the same process starting from the first iteration.

In particular, this observation means: For any event $E(\mathcal{J}', \ell)$ concerning the assignment of some jobs $\mathcal{J}' \subseteq \mathcal{J}$ in the ℓ -th iteration or in the subsequent iterations, we have $\Pr[E(\mathcal{J}', \ell) \mid \mathcal{J}' \cap \mathcal{J}_{\leq \ell-1} = \emptyset] = \Pr[E(\mathcal{J}', 1)]$. For example, consider the fourth property. Then, the observation implies that $\Pr[i \leftarrow_{\geq \ell} j \wedge i \leftarrow_{\geq \ell} j' \mid j, j' \notin \mathcal{J}_{\ell-1}] = \Pr[i \leftarrow j \wedge i \leftarrow j']$. Here $i \leftarrow_{\geq \ell} j$ denotes j being assigned to i in the ℓ -th iteration or later. We illustrate Observation 2 using this as an example. Note that the assignment of j and j' are completely determined by $\{\tilde{N}_{i'j}\}_{i' \in \mathcal{M}}, \{\tilde{N}_{i'j'}\}_{i' \in \mathcal{M}}$, and B_G for all groups G including j or j' . Further, for B_G , what only matters is whether $B_G = j$ or j' , or not. These events, $\tilde{N}_{i'j}, \tilde{N}_{i'j'}$ are not affected by whether other jobs have been assigned or not. Extending this argument to Observation 2 is straightforward.

We now shift to proving the four properties and the running time guarantee.

3.1 First and Second Properties and Running Time

The first property says that the randomized rounding assigns each job to exactly one machine. In each iteration the rounding algorithm attempts to assign jobs that have not been assigned in the previous iterations. Therefore, the first property immediately follows if we show that the algorithm eventually terminates. We will show that the algorithm terminates in $O(\log n)$ iterations with high probability and also in expectation, after showing the second property.

For better readability we first give the analysis for the first iteration and extend it to arbitrary iterations using Observation 2. We first make two easy observations.

OBSERVATION 3. For any $i \in \mathcal{M}$ and $j \in \mathcal{J}$, $N_{ij} \sim \text{Pois}(x_{ij})$.

Proof. The algorithm defines $N_{ij} = B_{ij} \cdot \tilde{N}_{ij}$, where $\tilde{N}_{ij} \sim \widetilde{\text{Pois}}(x_{ij})$ and B_{ij} is a Bernoulli RV with mean value x_{ij} . Thus, this observation follows from Observation 1. $\square$

OBSERVATION 4. For any job $j \in \mathcal{J}$, all RVs $\{N_{ij}\}_{i \in \mathcal{M}}$ are independent.

Proof. This is because all RVs $\{\tilde{N}_{ij}\}_{i \in \mathcal{M}, j \in \mathcal{J}}$ are independent; all RVs $\{B_{ij}\}_{i \in \mathcal{M}}$ are independent; and $N_{ij} = B_{ij} \cdot \tilde{N}_{ij}$. $\square$

Recall that $\mathcal{J}_{< \ell}$ denotes the set of job that are assigned to some machines before the ℓ -th iteration.

LEMMA 3.1. For any job $j \in \mathcal{J}$, machine $i \in \mathcal{M}$ and iteration $\ell \geq 1$, we have $\Pr[i \leftarrow_{\ell} j \mid B_{ij}^{\ell}, j \notin \mathcal{J}_{< \ell}] = 1 - 1/e$.

⁵Here, we used the well-known facts that $e^x = \sum_{k \geq 0} \frac{x^k}{k!}$ and $e^x \geq x + 1$.

1. For each $i \in \mathcal{M}$ and $j \in \mathcal{J}$, let $\tilde{N}_{ij} \sim \widetilde{\text{Pois}}(x_{ij})$ be an independent RV; if $x_{ij} = 0$, then $\tilde{N}_{ij} = 0$. In words, we generate $\tilde{N}_{ij}$ *potential* tickets for each job j on machine i , according to $\widetilde{\text{Pois}}(x_{ij})$, independently.
2. For each $i \in \mathcal{M}$ and $G \in \mathcal{G}_i$, let $B_G \in G \cup \{\text{nil}\}$ be an independent RV such that $\Pr[B_G = j] = x_{ij}$ for all $j \in G$. We use an indicator variable B_{ij} to denote the event $B_G = j$. In words, each group $G \in \mathcal{G}_i$ *recommends* one job j in the group G with probability x_{ij} independently and the event is denoted as B_{ij} ; or it may recommend no jobs.
3. For each $i \in \mathcal{M}$ and $j \in \mathcal{J}$, let $N_{ij} = B_{ij} \cdot \tilde{N}_{ij}$. In words, all $\tilde{N}_{ij}$ potential tickets of job j on machine i become *real* tickets if and only if the event $B_{ij} = 1$ occurs. Job j has real tickets on machine i iff $B_{ij} = 1$ and it has non-zero potential tickets on the machine.
4. Each job $j \in \mathcal{J}$ is *assigned* to machine $i \in \mathcal{M}$, denoted as $i \leftarrow_1 j$, independently with probability $\frac{N_{ij}}{\sum_{i' \in \mathcal{M}} N_{i'j}}$; if $\sum_{i' \in \mathcal{M}} N_{i'j} = 0$, then j is not assigned, denoted as $\text{nil} \leftarrow_1 j$. In words, among all real tickets of each job j across machines, we choose one uniformly at random. If the ticket was generated on machine i , then we *assign* job j to machine i .

Figure 1: The first iteration of our randomized rounding

Proof. Fix a job j and machine i . Say $i = 1$ wlog. By Observation 4 and Fact 2.1, we have $\sum_{i' \in \mathcal{M} \setminus \{1\}} N_{i'j} \sim \text{Pois}(\sum_{i' \in \mathcal{M} \setminus \{1\}} x_{i'j} = 1 - x_{1j})$. For notational convenience, let $\lambda = x_{1j}$, $\tilde{N}_1 = \tilde{N}_{1j}$, and $N_{-1} = \sum_{i' \in \mathcal{M} \setminus \{1\}} N_{i'j}$. Note that $\tilde{N}_1 \sim \widetilde{\text{Pois}}(\lambda)$ and $N_{-1} \sim \text{Pois}(1 - \lambda)$.

$$\begin{aligned}
& \Pr[i \leftarrow_1 j \mid B_{ij}] \\
&= \sum_{k \geq 1, k' \geq 0} \Pr[\tilde{N}_1 = k \wedge N_{-1} = k'] \cdot \frac{k}{k + k'} \\
&= \sum_{k \geq 1, k' \geq 0} \Pr[\tilde{N}_1 = k] \cdot \Pr[N_{-1} = k'] \cdot \frac{k}{k + k'} \\
&= \sum_{k \geq 1, k' \geq 0} e^{-\lambda} \frac{\lambda^k}{k!} \cdot e^{-(1-\lambda)} \frac{(1-\lambda)^{k'}}{k'!} \cdot \frac{k}{k + k'} \\
&= \sum_{k \geq 0, k' \geq 0} e^{-\lambda} \frac{\lambda^k}{k!} \cdot e^{-(1-\lambda)} \frac{(1-\lambda)^{k'}}{k'!} \cdot \frac{1}{k + k' + 1} \\
&= e^{-1} \sum_{k'' \geq 0} \sum_{k, k' \geq 0: k + k' = k''} \frac{\lambda^k}{k!} \cdot \frac{(1-\lambda)^{k'}}{k'!} \cdot k''! \cdot \frac{1}{(k'' + 1)!} \\
&= e^{-1} \sum_{k'' \geq 0} \frac{1}{(k'' + 1)!} (\lambda + (1-\lambda))^{k''} \\
&= e^{-1} \sum_{k'' \geq 1} \frac{1}{(k'')!} = e^{-1}(e - 1) = 1 - 1/e,
\end{aligned}$$

which, combined with Observation 2, yields the lemma. $\square$

COROLLARY 3.1. *For any job $j \in \mathcal{J}$, machine $i \in \mathcal{M}$ and iteration $\ell \geq 1$, we have $\Pr[i \leftarrow_\ell j \mid j \notin \mathcal{J}_{<\ell}] = (1 - 1/e)x_{ij}$.*

Proof. $\Pr[i \leftarrow_1 j] = \Pr[i \leftarrow_1 j \wedge B_{ij}] = \Pr[i \leftarrow_1 j \mid B_{ij}] \cdot \Pr[B_{ij}] = (1 - 1/e)x_{ij}$. Then, we use Observation 2. $\square$

This also implies that job j is assigned in the 1-st iteration with probability exactly $1 - 1/e$.

COROLLARY 3.2. *For any job $j \in \mathcal{J}$ and iteration $\ell \geq 1$, we have $\Pr[\text{nil} \leftarrow_\ell j \mid j \notin \mathcal{J}_{<\ell}] = 1/e$.*

Proof. By the above corollary and the fact that a job can be assigned to at most one machine in each iteration, the probability is $1 - \sum_{i \in \mathcal{M}} (1 - 1/e)x_{ij} = 1/e$. $\square$

We are now ready to complete the proof of the second property.

$$\begin{aligned}
& \Pr[i \leftarrow j] = \sum_{\ell \geq 1} \Pr[i \leftarrow_\ell j] \\
&= \sum_{\ell \geq 1} \Pr[i \leftarrow_\ell j \mid j \notin \mathcal{J}_{<\ell}] \cdot \prod_{\ell' < \ell} \Pr[\text{nil} \leftarrow_{\ell'} j \mid j \notin \mathcal{J}_{<\ell'}] \\
&= \sum_{\ell \geq 1} (1 - 1/e)x_{ij} (1/e)^{\ell-1} \\
&= x_{ij}
\end{aligned}$$

To complete the proof of the first property, we observe that $\Pr[j \notin \mathcal{J}_{\leq \ell}] = \prod_{\ell' \leq \ell} \Pr[\text{nil} \leftarrow_{\ell'} j \mid j \notin \mathcal{J}_{<\ell'}] = (1/e)^\ell$ by Corollary 3.2. Thus, using the linearity of expectation, we know that the expected number of jobs remaining unassigned after $2 \log n$ iterations, i.e., $\mathbb{E}|\mathcal{J} \setminus \mathcal{J}_{\leq 2 \log n}| = n(1/e)^{2 \log n} = 1/n$. Thus, by Markov inequality, the probability that the randomized rounding does not terminate within $2 \log n$ iterations, meaning that there is at least one job unassigned, is at most $1/n$. Also using a simple union bound we know $\Pr[\mathcal{J} \neq \mathcal{J}_{\leq \ell}] \leq \min\{1, n(1/e)^\ell\}$. Thus, the expected number of iterations before the termination is

$\sum_{\ell \geq 1} \min\{1, n(1/e)^\ell\} = O(\log n)$. It is an easy exercise to see each iteration takes $O(mn)$ time assuming that we can determine the value of each RV in $O(1)$ time. As discussed already, it is clear that each job j is assigned to exactly one machine if the algorithm terminates. Thus, we have shown the first property and running time.

3.2 Third Property We restate the third property we aim to prove: Fix a machine $i \in \mathcal{M}$ and two distinct jobs $j, j' \in \mathcal{J}$ from *different* groups of machine i , i.e., $j \in A$ and $j' \in B$ for some $A \neq B \in \mathcal{G}_i$. Then, we have $\Pr[i \leftarrow j \wedge i \leftarrow j'] \leq x_{ij}x_{ij'}$. For notational convenience, let $a = x_{ij}$ and $b = x_{ij'}$, and $\tilde{M}^* = \tilde{N}_{ij}$, $\tilde{N}^* = \tilde{N}_{ij'}$. Assume wlog that $a, b \in (0, 1)$ since otherwise the third property immediately follows from the second.

Here is a very high-level overview of the proof. For the sake of contradiction assume that $\Pr[i \leftarrow j \wedge i \leftarrow j'] > ab$. Then, we will show that the probability remains greater than ab after de-grouping the two jobs j and j' on all machines. This is a contradiction because two events $i \leftarrow j$ and $i \leftarrow j'$ are independent after de-grouping, which would immediately imply $\Pr[i \leftarrow j \wedge i \leftarrow j'] = ab$ due to the second property.

However, the actual analysis is quite involved. To help the reader keep the flow of the analysis, we outline the proof in detail.

3.2.1 Proof Outline For the sake of analysis, we need to define additional notation—then, we can give a more detailed proof overview and explain the technical challenges. Let $M_- := \sum_{i' \in \mathcal{M} \setminus \{i\}} N_{i'j}$ denote the total number of *real* tickets generated for job j on machines other than i . Similarly, let $N_- := \sum_{i' \in \mathcal{M} \setminus \{i\}} N_{i'j'}$ for job j' . Define $P_{\mathcal{G}}(m, n) := \Pr_{\mathcal{G}}[M_- = m, N_- = n]$; here $\mathcal{G}$ in the subscript is to emphasize that this probability is under grouping $\mathcal{G}$. Note that in this section we *override* m and n , which were used to denote the number of machines and jobs respectively in other sections. For comparison, create another grouping $\mathcal{G}'$ by separating the two jobs j and j' in the same group on every machine. That is, for any machine i' where $j, j' \in G$ for some $G \in \mathcal{G}_{i'}$, partition the group G arbitrarily into two groups G_{ij} and $G_{ij'}$, so that $j \in G_{ij}$ and $j' \in G_{ij'}$. Let $P_{\mathcal{G}'}(m, n)$ denote $\Pr_{\mathcal{G}'}[M_- = m, N_- = n]$ under this grouping.

Our goal is to show that

$$(3.6) \quad \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j'] > ab \text{ if } \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j'] > ab$$

Unfortunately, proving this directly seems very challenging. Let's see why. First, our analysis has very little room for loss since we need to show negative correlation for every pair of jobs not grouped together on the fixed machine. For careful analysis, we will have to take a close look at

probabilities fixing some random variables. Then, when we expand $\Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j']$ depending on the iterations in which the jobs are assigned, we run across the recursive structure for the case $j, j' \notin \mathcal{J}_{\leq 1}$. It seems very challenging to compare the above two probabilities in Eqn. (3.6) with this recursive structure combined with certain subtle conditions.

We get around this difficulty by first showing that Eqn. (3.6) *pretending* that after the first iteration, each unassigned job is assigned *independently* in the second iteration. This thought process will ensure that the rounding terminates in two iterations, thus having no recursive structure – let's call this rounding as *shadow rounding*. To relate the shadow rounding to the actual rounding, we will define $\Phi(P, \kappa, \tilde{m}^*, \tilde{n}^*)$. For notational convenience, let $\zeta := \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j']/(ab)$ and $\zeta' := \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j']/(ab)$. Then, Φ will have the following nice properties.

- (a) $\Phi(P_{\mathcal{G}}, \zeta, \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$ for our actual rounding.
- (b) $\Phi(P_{\mathcal{G}'}, \zeta', \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$ for our actual rounding.
- (c) $\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$ for the shadow rounding.
- (d) $\Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$ for the shadow rounding.

Depending on whether $\tilde{m}^* > 0$ or not and whether $\tilde{n}^* > 0$ or not, Φ is slightly different—so we will consider four cases, which are each presented in Sections 3.2.2, 3.2.3, 3.2.4 and 3.2.5. We will show the above (a) and (b) for our actual rounding in Lemmas 3.2, 3.5, 3.7 and 3.9. We do not show (c) and (d) for the shadow rounding because our proof only uses the algebraic difference between $\Phi(P_{\mathcal{G}}, \zeta, \tilde{m}^*, \tilde{n}^*)$ and $\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*)$ and that between $\Phi(P_{\mathcal{G}'}, \zeta', \tilde{m}^*, \tilde{n}^*)$ and $\Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*)$. Still, we mention (c) and (d) above as we believe they could give more intuitions.

Then, we will show that

$$\Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*) \geq \Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*)$$

for all $\tilde{m}^*, \tilde{n}^* \geq 0$ in Lemmas 3.4, 3.6, 3.8 and 3.10. By de-conditioning on $\tilde{M}^*$ and $\tilde{N}^*$ and using the aforementioned algebraic difference, we will be able to show Eqn. (3.6). We will present the details on how to put all the pieces together in Section 3.2.6. This will complete the proof of the third property.

Due to the space constraints, we defer to the full version of this paper the proof of each lemma.

3.2.2 Case $\tilde{M}^*, \tilde{N}^* > 0$ Recall $M_- := \sum_{i' \in \mathcal{M} \setminus \{i\}} N_{i'j}$ denotes the total number of *real* tickets generated for job j on machines other than i and N_- is similarly defined for job j' . Also recall $P(m, n) := \Pr[M_- = m, N_- = n]$.

LEMMA 3.2. For every $\tilde{m}^*, \tilde{n}^* \geq 1$, define:

(3.7)

$$\Phi(P, \kappa, \tilde{m}^*, \tilde{n}^*) := ab \sum_{m \geq 0, n \geq 0} \frac{\tilde{m}^*}{m + \tilde{m}^*} \cdot \frac{\tilde{n}^*}{n + \tilde{n}^*} \cdot P(m, n)$$

(3.8)

$$+ ab \sum_{m \geq 0} \frac{\tilde{m}^*}{m + \tilde{m}^*} \cdot (1 - b) \cdot P(m, 0)$$

(3.9)

$$+ ab \sum_{n \geq 0} \frac{\tilde{n}^*}{n + \tilde{n}^*} \cdot (1 - a) \cdot P(0, n)$$

(3.10)

$$+ ab(1 - a)(1 - b)\kappa \cdot P(0, 0)$$

Then, we have

- $\Phi(P_{\mathcal{G}}, \zeta, \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$; and
- $\Phi(P_{\mathcal{G}'}, \zeta', \tilde{m}^*, \tilde{n}^*) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$.

In the following we re-write Φ , so that we have the coefficient of each $P(m, n)$ explicitly.

LEMMA 3.3. For any $\tilde{m}^*, \tilde{n}^* \geq 1$, we have $\Phi(P, \kappa, \tilde{m}^*, \tilde{n}^*) = \sum_{m, n \geq 0} d(m, n)P(m, n)$, where

$$d(m, n) := \begin{cases} ab \frac{\tilde{m}^*}{m + \tilde{m}^*} \cdot \frac{\tilde{n}^*}{n + \tilde{n}^*} & \text{if } m, n > 0 \\ ab \frac{\tilde{m}^*}{m + \tilde{m}^*} (2 - b) & \text{if } m > 0, n = 0 \\ ab \frac{\tilde{n}^*}{n + \tilde{n}^*} (2 - a) & \text{if } m = 0, n > 0 \\ ab(2 - a)(2 - b) + (\kappa - 1)(1 - a)(1 - b) & \text{if } m = n = 0; \end{cases}$$

LEMMA 3.4. For any $\tilde{m}^*, \tilde{n}^* \geq 1$, $\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) \leq \Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*)$.

3.2.3 Case $\tilde{M}^* > 0, \tilde{N}^* = 0$ Previously, we considered the case $\tilde{M}^*, \tilde{N}^* > 0$. Here, we consider the case $\tilde{M}^* = m^*, \tilde{N}^* = 0$ for any fixed $m^* \geq 1$.

LEMMA 3.5. For any $\tilde{m}^* > 0, \tilde{n}^* = 0$, define,

$$\Phi(P, \kappa, \tilde{m}^*, \tilde{n}^* = 0) := \sum_{m \geq 0} d(m, 0)P(m, 0) \text{ where}$$

$$d(m, n) = \begin{cases} ab \frac{\tilde{m}^*}{m + \tilde{m}^*} & \text{if } m > 0, n = 0 \\ ab(2 - a) + ab(\kappa - 1)(1 - a) & \text{if } m = 0, n = 0 \\ 0 & \text{if } n > 0 \end{cases}$$

Then, we have

- $\Phi(P, \zeta, \tilde{m}^*, 0) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = 0]$; and
- $\Phi(P, \zeta', \tilde{m}^*, 0) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = \tilde{m}^*, \tilde{N}^* = 0]$.

LEMMA 3.6. For all $\tilde{m}^* \geq 1$, we have $\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) \leq \Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*)$.

3.2.4 Case $\tilde{M}^* = 0, \tilde{N}^* > 0$

LEMMA 3.7. For any $\tilde{n}^* > 0$, define,

$$\Phi(P, \kappa, \tilde{m}^* = 0, \tilde{n}^*) := \sum_{n \geq 0} d(0, n)P(0, n) \text{ where}$$

$$d(m, n) = \begin{cases} ab \frac{\tilde{n}^*}{n + \tilde{n}^*} & \text{if } m = 0, n > 0 \\ ab(2 - b) + ab(\kappa - 1)(1 - b) & \text{if } m = 0, n = 0 \\ 0 & \text{if } m > 0 \end{cases}$$

Then, we have

- $\Phi(P, \zeta, 0, \tilde{n}^*) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = 0, \tilde{N}^* = \tilde{n}^*]$; and
- $\Phi(P, \zeta', 0, \tilde{n}^*) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = 0, \tilde{N}^* = \tilde{n}^*]$.

LEMMA 3.8. For all $\tilde{n}^* \geq 1$, $\Phi(P_{\mathcal{G}}, 1, 0, \tilde{n}^*) \leq \Phi(P_{\mathcal{G}'}, 1, 0, \tilde{n}^*)$.

3.2.5 Case $\tilde{M}^* = \tilde{N}^* = 0$ This is the last case we consider.

LEMMA 3.9. Define $\Phi(P, \kappa, 0, 0) := ab\kappa P(0, 0)$.

- $\Phi(P, \zeta, 0, 0) = \Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = 0, \tilde{N}^* = 0]$; and
- $\Phi(P, \zeta', 0, 0) = \Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j' \mid \tilde{M}^* = 0, \tilde{N}^* = 0]$.

LEMMA 3.10. $\Phi(P_{\mathcal{G}}, 1, 0, 0) \leq \Phi(P_{\mathcal{G}'}, 1, 0, 0)$.

3.2.6 Putting All Pieces Together From the definitions stated in Lemmas 3.2, 3.5, 3.7 and 3.9, we have :

(3.11)

$$\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) - \Phi(P_{\mathcal{G}}, \zeta, \tilde{m}^*, \tilde{n}^*) = \begin{cases} -(1 - a)(1 - b)(\zeta - 1)abP_{\mathcal{G}}(0, 0) & \text{if } \tilde{m}^* > 0, \tilde{n}^* > 0 \\ -(1 - a)(\zeta - 1)abP_{\mathcal{G}}(0, 0) & \text{if } \tilde{m}^* > 0, \tilde{n}^* = 0 \\ -(1 - b)(\zeta - 1)abP_{\mathcal{G}}(0, 0) & \text{if } \tilde{m}^* = 0, \tilde{n}^* > 0 \\ -(\zeta - 1)abP_{\mathcal{G}}(0, 0) & \text{if } \tilde{m}^* = 0, \tilde{n}^* = 0 \end{cases}$$

Similarly, we have

(3.12)

$$\Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*) - \Phi(P_{\mathcal{G}'}, \zeta', \tilde{m}^*, \tilde{n}^*) = \begin{cases} -(1 - a)(1 - b)(\zeta' - 1)abP_{\mathcal{G}'}(0, 0) & \text{if } \tilde{m}^* > 0, \tilde{n}^* > 0 \\ -(1 - a)(\zeta' - 1)abP_{\mathcal{G}'}(0, 0) & \text{if } \tilde{m}^* > 0, \tilde{n}^* = 0 \\ -(1 - b)(\zeta' - 1)abP_{\mathcal{G}'}(0, 0) & \text{if } \tilde{m}^* = 0, \tilde{n}^* > 0 \\ -(\zeta' - 1)abP_{\mathcal{G}'}(0, 0) & \text{if } \tilde{m}^* = 0, \tilde{n}^* = 0 \end{cases}$$

In Lemmas 3.4, 3.6, 3.8 and 3.10 we have shown that $\Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) \leq \Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*)$ for all $\tilde{m}^*, \tilde{n}^* \geq 0$. Therefore, we have,

$$\begin{aligned} & \sum_{\tilde{m}^*, \tilde{n}^* \geq 0} q(\tilde{m}^*, \tilde{n}^*) \Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) \\ & \leq \sum_{\tilde{m}^*, \tilde{n}^* \geq 0} q(\tilde{m}^*, \tilde{n}^*) \Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*) \end{aligned}$$

where $q(\tilde{m}^*, \tilde{n}^*) := \Pr[\tilde{M}^* = \tilde{m}^*, \tilde{N}^* = \tilde{n}^*]$.

Using Eqn. (3.11), we have,

$$\begin{aligned} & \sum_{\tilde{m}^*, \tilde{n}^* \geq 0} q(\tilde{m}^*, \tilde{n}^*) \Phi(P_{\mathcal{G}}, 1, \tilde{m}^*, \tilde{n}^*) \\ = & \sum_{\tilde{m}^*, \tilde{n}^* \geq 0} q(\tilde{m}^*, \tilde{n}^*) \Phi(P_{\mathcal{G}}, \zeta, \tilde{m}^*, \tilde{n}^*) \\ & - \sum_{\tilde{m}^*, \tilde{n}^* > 0} q(\tilde{m}^*, \tilde{n}^*) (1-a)(1-b)(\zeta-1)abP_{\mathcal{G}}(0,0) \\ & - \sum_{\tilde{m}^* > 0} q(\tilde{m}^*, 0)(1-a)(\zeta-1)abP_{\mathcal{G}}(0,0) \\ & - \sum_{\tilde{n}^* > 0} q(0, \tilde{n}^*)(1-b)(\zeta-1)abP_{\mathcal{G}}(0,0) \\ & - q(0,0)(\zeta-1)abP_{\mathcal{G}}(0,0) \\ = & \Pr_G[i \leftarrow j, i \leftarrow j'] - abP_{\mathcal{G}}(0,0)(\zeta-1) \cdot \\ & \left((1-a)(1-b)q_{++} + (1-a)q_{+0} + (1-b)q_{0+} + q_{00} \right) \\ = & \zeta ab - abP_{\mathcal{G}}(0,0)(\zeta-1) \cdot \\ & \left((1-a)(1-b)q_{++} + (1-a)q_{+0} + (1-b)q_{0+} + q_{00} \right), \end{aligned}$$

where $q_{++} := \Pr[\tilde{M}^* > 0, \tilde{N}^* > 0]$, $q_{+0} := \Pr[\tilde{M}^* > 0, \tilde{N}^* = 0]$, $q_{0+} := \Pr[\tilde{M}^* = 0, \tilde{N}^* > 0]$, $q_{00} := \Pr[\tilde{M}^* = 0, \tilde{N}^* = 0]$.

Similarly, we obtain

$$\begin{aligned} & \sum_{\tilde{m}^*, \tilde{n}^* \geq 0} q(\tilde{m}^*, \tilde{n}^*) \Phi(P_{\mathcal{G}'}, 1, \tilde{m}^*, \tilde{n}^*) \\ = & \zeta' ab - abP_{\mathcal{G}'}(0,0)(\zeta'-1) \cdot \\ & \left((1-a)(1-b)q_{++} + (1-a)q_{+0} + (1-b)q_{0+} + q_{00} \right) \end{aligned}$$

Thus, we have,

$$\begin{aligned} (3.13) \quad & \zeta ab - abP_{\mathcal{G}}(0,0)(\zeta-1) \cdot \\ & \left((1-a)(1-b)q_{++} + (1-a)q_{+0} + (1-b)q_{0+} + q_{00} \right) \\ (3.14) \quad & \leq \zeta' ab - abP_{\mathcal{G}'}(0,0)(\zeta'-1) \cdot \\ & \left((1-a)(1-b)q_{++} + (1-a)q_{+0} + (1-b)q_{0+} + q_{00} \right) \end{aligned}$$

Note that Eqn. (3.13) and (3.14) are linear ζ and ζ' , respectively. Further, $q_{++} + q_{+0} + q_{0+} + q_{00} = 1$ and $q_{++} > 0$ since $a, b \in (0,1)$. Therefore, both ζ and ζ' have strictly positive coefficients in Eqn. (3.13) and (3.14), respectively. Further, Eqn. (3.13) has value ab when $\zeta = 1$ and Eqn. (3.14) has value ab when $\zeta' = 1$. Thus, if $\zeta > 1$, then it must be the case that $\zeta' > 1$.

To summarize, we have shown that if $\Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j'] > x_{ij}x_{ij'}$, then it must be the case that $\Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j'] > x_{ij}x_{ij'}$. Since $\mathcal{G}'$ was obtained from $\mathcal{G}$ by de-grouping the two jobs j and j' on one machine, by repeatedly refining $\mathcal{G}'$ further, we know that what we have shown still holds true when $\mathcal{G}'$ doesn't group j and j' together on any machines. As mentioned before, this implies that $i \leftarrow j$ and $i \leftarrow j'$ are independent under $\mathcal{G}'$. Thus, we have $\Pr_{\mathcal{G}'}[i \leftarrow j \wedge i \leftarrow j'] = x_{ij}x_{ij'}$. Thus, if we assume $\Pr_{\mathcal{G}}[i \leftarrow j \wedge i \leftarrow j'] > x_{ij}x_{ij'}$, then we obtain a contradiction. This completes the proof of the third property.

3.3 Fourth Property For ease of reference, we re-state the fourth property we aim to prove: For every $i \in \mathcal{M}$ and $j \neq j' \in \mathcal{J}$ such that $j, j' \in G$ for some $G \in \mathcal{G}_i$, we have $\Pr[i \leftarrow j \wedge i \leftarrow j'] \leq \frac{1}{1+e}(e^{x_{ij}} + e^{x_{ij'}})x_{ij}x_{ij'}$.

We first upper bound the probability that neither j nor j' are assigned in the first iteration. Towards this end, we need the following proposition.

PROPOSITION 3.1. *For any job $j \in \mathcal{J}$ and machine $i \in \mathcal{M}$, we have,*

- $\Pr[N_{ij} = 0 \mid \neg B_{ij}] = 1$.
- $\Pr[N_{ij} = 0 \mid B_{ij}] = 1 - \frac{1 - \exp(-x_{ij})}{x_{ij}}$.

Proof. The first claim is immediate from the algorithm definition, that is, $N_{ij} = \tilde{N}_{ij}B_{ij}$. The second claim follows since $\Pr[N_{ij} > 0 \mid B_{ij}] = \Pr[\tilde{N}_{ij} > 0] = \sum_{k>0} e^{-\lambda} \frac{\lambda^k}{k!} = (1 - \exp(-\lambda))/\lambda$ where $\lambda = x_{ij}$. $\square$

LEMMA 3.11. *For any two jobs $j \neq j' \in \mathcal{J}$, $\Pr[nil \leftarrow_1 j \wedge nil \leftarrow_1 j'] \leq 1/e^2$.*

Proof. The event occurs if and only if no tickets are generated for j and j' , i.e., $\sum_{i \in \mathcal{M}} (N_{ij} + N_{ij'}) = 0$. We will show that for any machine i ,

$$(3.15) \quad \Pr[N_{ij} + N_{ij'} = 0] \leq e^{-x_{ij} - x_{ij'}}.$$

Since the events $\{N_{ij} + N_{ij'} = 0\}_{i \in \mathcal{M}}$ are independent, Eqn. (3.15) would imply $\Pr[\sum_{i \in \mathcal{M}} (N_{ij} + N_{ij'}) = 0] = \prod_{i \in \mathcal{M}} \Pr[N_{ij} + N_{ij'} = 0] \leq \prod_{i \in \mathcal{M}} \exp(-x_{ij} - x_{ij'}) = \exp(-\sum_{i \in \mathcal{M}} x_{ij} - \sum_{i \in \mathcal{M}} x_{ij'}) = 1/e^2$, as desired.

It now remains to show Eq. (3.15). Towards this end, fix a machine $i \in \mathcal{M}$. For notational convenience, let $a := x_{ij}$ and $b := x_{ij'}$. There are two cases we need to consider.

If j and j' are not grouped together on machine i , the two RVs N_{ij} and $N_{ij'}$ are independent. By Observation 3, we know that $N_{ij} \sim \text{Pois}(x_{ij} = a)$ and $N_{ij'} \sim \text{Pois}(x_{ij'} = b)$. Therefore, $\Pr[N_{ij} = 0 \wedge N_{ij'} = 0] = \Pr[N_{ij} = 0] \cdot \Pr[N_{ij'} = 0] = e^{-a} \cdot e^{-b}$.

We now consider the other case where $j, j' \in G$ for some $G \in \mathcal{G}_i$. Depending on which job (or none) is recommended by the group G , we consider three disjoint events: $\neg B_{ij} \wedge \neg B_{ij'}, B_{ij}, B_{ij'}$.

$$\begin{aligned}
& \Pr[N_{ij} + N_{ij'} = 0] \\
&= \Pr[N_{ij} + N_{ij'} = 0 \wedge \neg B_{ij} \wedge \neg B_{ij'}] \\
&\quad + \Pr[N_{ij} + N_{ij'} = 0 \wedge B_{ij}] + \Pr[N_{ij} + N_{ij'} = 0 \wedge B_{ij'}] \\
&= \Pr[\neg B_{ij} \wedge \neg B_{ij'}] + \Pr[N_{ij} = 0 \wedge B_{ij}] \\
&\quad + \Pr[N_{ij'} = 0 \wedge B_{ij'}] \\
&= (1 - a - b) + \Pr[N_{ij} = 0 \mid B_{ij}] \cdot \Pr[B_{ij}] \\
&\quad + \Pr[N_{ij'} = 0 \mid B_{ij'}] \cdot \Pr[B_{ij'}] \\
(3.16) \quad &= (1 - a - b) + a \cdot \left(1 - \frac{1 - e^{-a}}{a}\right) + b \cdot \left(1 - \frac{1 - e^{-b}}{b}\right) \\
&\quad [\text{Proposition 3.1}] \\
&= e^{-a} + e^{-b} - 1 \\
&= e^{-a-b} - (e^{-a} - 1)(e^{-b} - 1) \\
&\leq e^{-a-b} = e^{-x_{ij} - x_{ij'}},
\end{aligned}$$

where the second equality follows since $\neg B_{ij}$ and $\neg B_{ij'}$ imply $N_{ij} = B_{ij} \tilde{N}_{ij} = 0$ and $N_{ij'} = B_{ij'} \tilde{N}_{ij'} = 0$, respectively. $\square$

By Observation 2, we obtain the following corollary.

COROLLARY 3.3. *For any two jobs, $j \neq j' \in \mathcal{J}$ and machine $i \in \mathcal{M}$, we have $\Pr[i \leftarrow_{\geq 2} j \wedge i \leftarrow_{\geq 2} j'] \leq (1/e^2) \Pr[i \leftarrow j \wedge i \leftarrow j']$.*

We now consider the other case when $i \leftarrow_1 j \wedge i \leftarrow_{\geq 2} j'$ or $i \leftarrow_{\geq 2} j \wedge i \leftarrow_1 j'$. This means that one of the two jobs is assigned to machine i and the other is not assigned to any machines in the first iteration.

LEMMA 3.12. *For any machine $i \in \mathcal{M}$ and for any two jobs $j \neq j' \in G$ for some $G \in \mathcal{G}_i$, we have $\Pr[i \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j'] \leq (1 - 1/e)(1/e)e^{x_{ij} + x_{ij'}}$.*

Proof. Fix any i, j, j', G satisfying the lemma precondition. Say $i = 1$ wlog. We will show that $\Pr[1 \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j']$ is maximized when j and j' are not grouped together on any machines except machine 1. Towards this end, if there is a machine $i' \neq i$, say $i' = 2$, where $j, j' \in G'$ for some $G' \in \mathcal{G}_2$, we create another grouping by splitting G' into A and B arbitrarily such that $j \in A$ and $j' \in B$. Then we

will show that the probability is no smaller under grouping G' than it is under G . Repeating this argument will prove the claim.

Let P_G denote $\Pr[i \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j']$ under grouping G . Define $P_{G'}$ analogously. Our goal is to show $P_G \leq P_{G'}$. We will compare P_G and $P_{G'}$ fixing $N_{1j}, N_{1j'}, \tilde{N}_{2j}, \tilde{N}_{2j'}, N_{-2j} := \sum_{i'' \in \mathcal{M}, i'' \neq 2} N_{i''j}$, and $N_{-2j'} := \sum_{i'' \in \mathcal{M}, i'' \neq 2} N_{i''j'}$. Note that $\text{nil} \leftarrow_1 j'$ only if $N_{1j'} = N_{-2j'} = 0$. Also, $1 \leftarrow_1 j$ only if $N_{1j} > 0$. So, assume $N_{1j} > 0$; then we also have $N_{-2j} \geq N_{1j} > 0$.

Given that the aforementioned RVs are fixed, the only factor that potentially makes P_G and $P_{G'}$ different is how B_{2j} and $B_{2j'}$ are related—the two are disjoint under G but are independent under G' . We first observe that $P_G = P_{G'}$ if $\tilde{N}_{2j'} = 0$. This is because if $\tilde{N}_{2j'} = 0$, then $N_{2j'} = 0$ no matter what, meaning that $B_{2j'}$ has no effect on P_G or $P_{G'}$; further, $\Pr[B_{2j}]$ is the same under both G and G' . So, we assume that the fixed $\tilde{N}_{2j'} > 0$.

To calculate P_G and $P_{G'}$, we consider three disjoint events depending on the recommendation made by G' on machine 2, namely $B_{2j}, B_{2j'}$, and $\neg B_{2j} \wedge \neg B_{2j'}$. If $B_{2j} = 1$, then $N_{2j} = \tilde{N}_{2j}$. Note that per the above discussion, we can safely assume that $N_{1j'} = N_{-2j'} = 0, N_{1j} > 0, \tilde{N}_{2j'} > 0$. So, we have $\Pr[1 \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j' \wedge B_{2j}] = x_{2j} \cdot \frac{N_{1j}}{\tilde{N}_{2j} + N_{-2j}}$. If $B_{2j'} = 1$, then $N_{2j'} = \tilde{N}_{2j'} > 0$, therefore $j' \in \mathcal{J}_{\leq 1}$. Finally, if $\neg B_{2j} \wedge \neg B_{2j'}$, then $N_{2j} = N_{2j'} = 0$ —so we have $\Pr[1 \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j' \wedge \neg B_{2j} \wedge \neg B_{2j'}] = (1 - x_{2j} - x_{2j'}) \frac{N_{1j}}{N_{-2j}}$. Therefore, we have,

$$P_G = \frac{N_{1j}}{N_{-2j} + \tilde{N}_{2j}} a + \frac{N_{1j}}{N_{-2j}} (1 - a - b),$$

where we let $a := x_{2j}, b := x_{2j'}$ for notational convenience.

We now focus on calculating $P_{G'}$. Note that under grouping G' , B_{2j} and $B_{2j'}$ are independent. For $j' \notin \mathcal{J}_{\leq 1}$ to happen, it must be the case that $B_{2j'} = 0$ since the fixed $\tilde{N}_{2j'} > 0$. Then, by considering whether B_{ij} or not, we have,

$$P_{G'} = (1 - b) \left(\frac{N_{1j}}{N_{-2j} + \tilde{N}_{2j}} a + \frac{N_{1j}}{N_{-2j}} (1 - a) \right)$$

Then, we have,

$$P_{G'} - P_G = ab \left(-\frac{N_{1j}}{N_{-2j} + \tilde{N}_{2j}} + \frac{N_{1j}}{N_{-2j}} \right) \geq 0,$$

since $N_{-2j} \geq N_{1j}$, as desired.

Therefore, to upper bound $\Pr[i \leftarrow_1 j \wedge \text{nil} \leftarrow_1 j']$, we can safely assume that no group, except G on machine 1, has both jobs j and j' simultaneously. This implies that $N_{-1j} := \sum_{i'' \in \mathcal{M}, i'' \neq 1} N_{i''j}$ and $N_{-1j'} := \sum_{i'' \in \mathcal{M}, i'' \neq 1} N_{i''j'}$ are independent. Note that $N_{-1j} \sim \text{Pois}(1 - x_{1j})$ and $N_{-1j'} \sim$

$\text{Pois}(1 - x_{1j'})$ by Observations 3 and 4. We now derive,

$$\begin{aligned}
& \Pr[1 \leftarrow j \wedge \text{nil} \leftarrow j'] \\
&= \Pr[1 \leftarrow j \wedge B_{1j} \wedge N_{1j'} = N_{-1j'} = 0] \\
& \quad [1 \leftarrow j \text{ only if } B_{1j}] \\
&= \Pr[1 \leftarrow j \wedge B_{1j} \wedge N_{-1j'} = 0] \\
& \quad [N_{1j'} = 0 \text{ if } B_{1j}, \text{ since } j, j' \in G \text{ on machine 1}] \\
&= \Pr[1 \leftarrow j \wedge B_{1j}] \cdot \Pr[N_{-1j'} = 0] \\
&= \Pr[1 \leftarrow j \mid B_{1j}] \cdot \Pr[B_{1j}] \cdot e^{-(1-x_{1j'})} \\
& \quad [N_{-1j'} \sim \text{Pois}(1 - x_{1j'})] \\
&= (1 - 1/e)e^{-(1-x_{1j'})} x_{1j} \quad [\text{Lemma 3.1}]
\end{aligned}$$

The third equation follows since $N_{-1j'}$ is determined by the recommendation of groups on machines other than 1, which don't include j , and $\{\tilde{N}_{i''j'}\}_{i'' \in \mathcal{M}, i'' \neq 1}$ — all these are independent of B_{1j} and $1 \leftarrow j$ since j and j' are not grouped together on any machines except machine 1. $\square$

COROLLARY 3.4. *For any machine $i \in \mathcal{M}$ and for any two jobs $j \neq j' \in G$ for some $G \in \mathcal{G}_i$, we have*

$$\begin{aligned}
& \Pr[i \leftarrow j \wedge i \leftarrow_{\geq 2} j'] \leq (1 - 1/e)(1/e)e^{x_{ij'} x_{ij} x_{ij'}}; \text{ and} \\
& \Pr[i \leftarrow_{\geq 2} j \wedge i \leftarrow j'] \leq (1 - 1/e)(1/e)e^{x_{ij} x_{ij'} x_{ij}}.
\end{aligned}$$

Proof. We show the first inequality as follows.

$$\begin{aligned}
& \Pr[i \leftarrow j \wedge i \leftarrow_{\geq 2} j'] \\
&= \Pr[i \leftarrow j \wedge i \leftarrow_{\geq 2} j' \wedge \text{nil} \leftarrow j'] \\
&= \Pr[i \leftarrow_{\geq 2} j' \mid i \leftarrow j \wedge \text{nil} \leftarrow j'] \cdot \Pr[i \leftarrow j \wedge \text{nil} \leftarrow j'] \\
&= \Pr[i \leftarrow_{\geq 1} j'] \cdot \Pr[i \leftarrow j \wedge \text{nil} \leftarrow j'] \\
&\leq x_{ij'}(1 - 1/e)e^{-(1-x_{1j'})} x_{ij},
\end{aligned}$$

where the penultimate equality is due to Observation 2; and the last equality is due to the second property and Lemma 3.12. The second inequality can be analogously shown using its symmetricity to the first. $\square$

We are now ready to complete the proof of the fourth property. For any two jobs $j \neq j' \in G$ for some $G \in \mathcal{G}_i$, we know $i \leftarrow j$ and $i \leftarrow j'$ cannot happen simultaneously since B_{ij} and $B_{ij'}$ are disjoint. Therefore, by Corollaries 3.3 and 3.4, we have

$$\begin{aligned}
& \Pr[i \leftarrow j \wedge i \leftarrow j'] \\
&\leq \Pr[i \leftarrow j \wedge i \leftarrow_{\geq 2} j'] \\
& \quad + \Pr[i \leftarrow_{\geq 2} j \wedge i \leftarrow_{\geq 1} j'] + \Pr[i \leftarrow_{\geq 2} j \wedge i \leftarrow_{\geq 2} j'] \\
&\leq (1 - 1/e)(1/e)(e^{x_{ij} + e^{x_{ij'}}} x_{ij} x_{ij'}) \\
& \quad + (1/e^2) \Pr[i \leftarrow j \wedge i \leftarrow j']
\end{aligned}$$

By rearranging terms, we have

$$\begin{aligned}
& \Pr[i \leftarrow j \wedge i \leftarrow j'] \\
&\leq \frac{(1 - 1/e)(1/e)}{1 - 1/e^2} (e^{x_{ij} + e^{x_{ij'}}} x_{ij} x_{ij'}) \\
&= \frac{1}{e + 1} (e^{x_{ij} + e^{x_{ij'}}} x_{ij} x_{ij'})
\end{aligned}$$

This completes the proof of the fourth property.

4 Unrelated Machines Scheduling: Rounding Algorithm

In this section we describe how we round the optimal solution to the LP described in Section 1.3. As mentioned, we will view the LP solution $\{x_{ijs}\}_{i \in \mathcal{M}, j \in \mathcal{J}, s}$ as a collection of rectangles.

DEFINITION 4.1. *For every $x_{ijs} > 0$, where $i \in \mathcal{M}, j \in \mathcal{J}, s \geq 0$, there is a rectangle R_{ijs} that starts at time s and ends at time $s + p_{ij}$ of height x_{ijs} .*

DEFINITION 4.2. *The height of job $j \in \mathcal{J}$ on machine $i \in \mathcal{M}$ is defined as $x_{ij} := \sum_s x_{ijs}$.*

To apply Theorem 1.2, we need to group jobs on each machine, which will be done stochastically. For every $j \in \mathcal{J}$ and $i \in \mathcal{M}$ such that $x_{ij} > 0$, we choose τ_{ij} uniformly at random from $(0, p_{ij}]$; and select one rectangle R_{ijs} with probability $\frac{x_{ijs}}{x_{ij}}$ as the *representative* rectangle of job j on machine i , which is denoted as R_{ij} .

We now define good jobs and bad jobs on each machine. Intuitively, a job is good on a machine if it is scheduled a lot on the machine in the LP solution or it starts considerably late compared its size (the starting point of the job's representative rectangle R_{ij} is large compared to its size p_{ij})—we say a job is good because if all jobs are good, then we can obtain a better than 1.5-approximation by independent rounding.

DEFINITION 4.3. *Given $R_{ij} = R_{ijs}$, job $j \in \mathcal{J}$ is good on machine $i \in \mathcal{M}$ if $s \geq \frac{1}{10}p_{ij}$ or $x_{ij} \geq \frac{9}{100}$; otherwise, job j is bad on machine i .*

DEFINITION 4.4. *A rectangle R_{ijs} is good if $s \geq \frac{1}{10}p_{ij}$ or $x_{ij} \geq \frac{9}{100}$, otherwise bad.*

Note that a job j is good on machine i if its representative rectangle R_{ij} on machine i is good. So, job j being good or bad is stochastic.

Before grouping jobs, we shift every rectangle to the right—how much a rectangle $R_{ij} = R_{ijs}$ is shifted depends on its start time and how much j is scheduled on the machine: Rectangle $R_{ij} = R_{ijs}$ is shifted to the right by

- $0.34(s + x_{ij}p_{ij})$ if $x_{ij} \geq \frac{9}{100}$

- 0.34s otherwise.

We denote the starting point of R_{ijs} after shifting as $\hat{s}$. For each job j on machine i with $R_{ij} = R_{ijs}$ and random offset τ_{ij} , we set $\theta_{ij} = \hat{s} + \tau_{ij}$. We will schedule the jobs assigned to each machine i in increasing order of their θ values on machine i .

Now to decide where to assign each job j , we use Theorem 1.2. To apply the theorem we need to group jobs on each machine. To define the grouping $\mathcal{G}_i$ on each machine i , fix a machine $i \in \mathcal{M}$. We only group bad jobs on machine i —or equivalently, we create a singleton group for each job that is not grouped together with any other jobs on the machine. To group bad jobs on machine i , we use a set of random grid points that are exponentially increasing. Choose a number ρ from $(1/10, 1)$ uniformly at random. A point in time is called a grid time if it is of the form $\rho 10^l$ for some integer l . Define the k -th grid interval, $I_k := (\rho 10^k, \rho 10^{k+1})$. Note that grid times partition the time horizon $(0, \infty)$ into $\{I_k\}_{k \in \mathbb{Z}}$.

We associate job j with interval I_k on machine i if $\theta_{ij} \in I_k$ and an independent coin gives a head—the coin gives a head with probability $u = 1/2$ —and we denote this event as $j \rightsquigarrow_i I_k$. Now consider all jobs associated with an interval I_k on machine i . If their total height is less than 1, i.e., $\sum_{j: j \rightsquigarrow_i I_k} x_{ij} \leq 1$, we group them together and add this group to $\mathcal{G}_i$. If two jobs j and j' are grouped together on machine i , we will denote the event as $j \stackrel{i}{\sim} j'$; otherwise $j \not\stackrel{i}{\sim} j'$. We now use Theorem 1.2 to decide where to assign each job.

As mentioned above, given that every job is assigned to a machine, we order the jobs assigned to the same machine i , in increasing order of their θ_{ij} value. This completes the description of our randomized rounding.

5 Unrelated Machines Scheduling: Analysis

This section is devoted to the analysis of the randomized algorithm in Section 4 with the goal of proving Theorem 1.1.

We first remind the reader of the following fact.

OBSERVATION 5. *All properties of our randomized rounding method (stated in Theorem 1.2) hold true for any fixed value of the random variables $\{\theta_{ij}\}_{i,j}$ and ρ . In particular, $\Pr[i \leftarrow j] = x_{ij}$ for any θ values of jobs and ρ value and is independent of how jobs are grouped together on each machine and which rectangles are chosen as jobs' representative rectangles.*

Let $\tilde{C}_{j^*}$ be the completion time of a fixed job j^* in the schedule returned by the rounding algorithm. Using conditional expectation and the law of total expectation, we

have

$$\begin{aligned} \mathbb{E}[\tilde{C}_{j^*}] &= \sum_{i \in \mathcal{M}} \sum_{s^*} \Pr[i \leftarrow j^*] \Pr[R_{ij^*} = R_{ij^*s^*} \mid i \leftarrow j^*] \\ &\quad \cdot \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \\ &= \sum_{i \in \mathcal{M}} \sum_{s^*} x_{ij^*s^*} \Pr[R_{ij^*} = R_{ij^*s^*} \mid i \leftarrow j^*] \\ &\quad \cdot \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \end{aligned}$$

As stated in Observation 5, the event $i \leftarrow j^*$ is independent of $R_{ij^*} = R_{ij^*s^*}$, which happens with probability $\frac{x_{ij^*s^*}}{x_{ij^*}}$. Thus, we can simplify the above as follows.

$$(5.17) \quad \mathbb{E}[\tilde{C}_{j^*}] = \sum_{i \in \mathcal{M}} \sum_{s^*} x_{ij^*s^*} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}]$$

From now on, we fix machine i . Our key lemma in the analysis is the following.

LEMMA 5.1. *For each job j^* , we have*

$$\begin{aligned} &\sum_{s^*} x_{ij^*s^*} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \\ &\leq 1.488 \sum_{s^*} x_{ij^*s^*} (s^* + p_{ij^*}) \end{aligned}$$

Plugging this inequality into Eqn.(5.17) would immediately imply Theorem 1.1:

$$\mathbb{E}[\tilde{C}_{j^*}] \leq 1.488 \sum_i \sum_{s^*} x_{ij^*s^*} (s^* + p_{ij^*}),$$

as summing $\mathbb{E}[\tilde{C}_{j^*}]$ over all jobs multiplied by their weight gives an upper bound of our algorithm's expected objective by 1.488 times the LP optimum objective.

The rest of this section is devoted to proving Lemma 5.1. Following observations will be useful for our analysis.

OBSERVATION 6. *For any job j^* and any rectangle $R_{ij^*s^*}$ we have*

(5.18)

$$\begin{aligned} &\mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \\ &= \int_0^{p_{ij^*}} \frac{1}{p_{ij^*}} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}, \tau_{ij^*} = \tau] d\tau \end{aligned}$$

Proof. Recall that τ_{ij^*} is chosen from $(0, p_{ij^*}]$ uniformly at random. Using conditional expectation and the law of total expectation, we have,

$$\begin{aligned} &\mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \\ &= \int_0^{p_{ij^*}} \Pr[\tau_{ij^*} = \tau \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] \\ &\quad \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}, \tau_{ij^*} = \tau] d\tau \end{aligned}$$

Note that τ_{ij^*} is independent of $R_{ij^*} = R_{ij^*s^*}$, and $i \leftarrow j^*$ is independent of these two events as observed in Observation 5. Thus, we have $\Pr[\tau_{ij^*} \in [\tau, \tau + d\tau) \mid i \leftarrow j^*, R_{ij^*} = R_{ij^*s^*}] = \Pr[\tau_{ij^*} \in [\tau, \tau + d\tau)] = \frac{d\tau}{p_{ij^*}}$. This completes the proof. $\square$

For the sake of analysis we define notation $\hat{L}_{ijs}(\theta)$ for any rectangle R_{ijs} and ant time point θ to be the length of part of the rectangle R_{ijs} that appears before time θ after shifting. Note that $\hat{L}_{ijs}(\theta) = \min\{\theta, \hat{s} + p_{ij}\} - \hat{s}$ if $\hat{s} < \theta$, otherwise $\hat{L}_{ijs}(\theta) = 0$. The next observation measures the probability that a job j will have a smaller θ value than the fixed job j^* on machine i . If this event occurs, and further, both jobs are assigned to the machine i , job j will delay job j^* .

OBSERVATION 7. For any two jobs j^* and j on machine i , any fixed value θ_{ij^*} and any R_{ijs} , we have $\Pr[\theta_{ij} \leq \theta_{ij^*} \mid R_{ij} = R_{ijs}] = \frac{\hat{L}_{ijs}(\theta_{ij^*})}{p_{ij}}$.

Proof. Since the representative rectangle of job j on machine i is fixed, the event $\theta_{ij} = \hat{s} + \tau_{ij} \leq \theta_{ij^*}$ happens iff $\tau_{ij} \in (0, \max\{0, \min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - \hat{s}\}] = (0, \hat{L}_{ijs}(\theta_{ij^*})]$. Since τ_{ij} is chosen from $(0, p_{ij}]$ uniformly at random, the event $\tau_{ij} \in (0, \hat{L}_{ijs}(\theta_{ij^*})]$ occurs with probability $\frac{\hat{L}_{ijs}(\theta_{ij^*})}{p_{ij}}$. $\square$

5.1 Proof of the Lemma 5.1 We now get back to proving our main lemma, Lemma 5.1. Towards that end, we first express j^* 's expected completion times in terms of the quantities we defined. The following lemma breaks down j^* 's expected completion time conditioned on the fixed job j^* being assigned to machine i , the rectangle $R_{ij^*s^*}$ being selected as j^* 's representative on machine i and the value of θ_{ij^*} being fixed. Note that $x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*})$ is exactly the volume of rectangle R_{ijs} appearing before time θ_{ij^*} after shifting and $\sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*}) - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*})$ is total volume of all rectangles excluding those of job j^* appearing before time θ_{ij^*} after shifting—this is exactly how much other jobs would delay job j^* if we used the standard independent rounding. The gain comes from representative rectangles that are grouped together with the fixed $R_{ij^*s^*}$, in which jobs of such rectangles are less likely to be assigned to the same machine i due to the strong negative correlation property.

For brevity, we may shorten $R_{ij} = R_{ijs}$ simply as R_{ijs} particularly when it is stated in the condition of a probability or expectation. Throughout this paper, we set $\eta := \frac{2\exp(0.09)}{e+1} < 0.589$, which comes from the following: Recall the fourth property of Theorem 1.2. For any two distinct bad jobs j and j' , as $x_{ij}, x_{ij'} < 9/100$, we have $\Pr[i \leftarrow j \wedge i \leftarrow j' \mid j \stackrel{i}{\sim} j'] \leq \frac{2\exp(0.09)}{e+1} x_{ij} x_{ij'} < 0.589 x_{ij} x_{ij'}$.

LEMMA 5.2. For any job j^* , representative rectangle $R_{ij^*s^*}$ and any fixed $\tau_{ij^*} \in (0, p_{ij^*}]$, we have,

$$\begin{aligned} & \mathbb{E} [\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*}] - p_{ij^*} \\ &= \sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*}) - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*}) \\ & \quad - (1 - \eta) \sum_{j \neq j^*, s} x_{ijs} \cdot p_{ij} \\ & \quad \cdot \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}] \end{aligned}$$

where e_{ij} denotes the event that $\theta_{ij} \leq \theta_{ij^*}$.

Proof. For brevity, we omit the fixed τ_{ij^*} and representative rectangle $R_{ij^*s^*}$ from the condition. Then, we have,

$$\begin{aligned} & \mathbb{E} [\tilde{C}_{j^*} \mid i \leftarrow j^*] - p_{ij^*} \\ &= \sum_{j \neq j^*, e_{ij}} \Pr[i \leftarrow j \mid i \leftarrow j^*] \cdot p_{ij} \\ &= \sum_{j \neq j^*, e_{ij}, j \stackrel{i}{\sim} j^*} \Pr[i \leftarrow j \mid i \leftarrow j^*] \cdot p_{ij} \\ & \quad + \sum_{j \neq j^*, e_{ij}, j \not\sim j^*} \Pr[i \leftarrow j \mid i \leftarrow j^*] \cdot p_{ij} \\ &\leq \sum_{j \neq j^*, e_{ij}, j \stackrel{i}{\sim} j^*} x_{ij} \cdot p_{ij} + \eta \sum_{j \neq j^*, e_{ij}, j \not\sim j^*} x_{ij} \cdot p_{ij} \\ &= \sum_{j \neq j^*, e_{ij}} x_{ij} \cdot p_{ij} - (1 - \eta) \sum_{j \neq j^*, e_{ij}, j \stackrel{i}{\sim} j^*} x_{ij} \cdot p_{ij} \end{aligned}$$

where the inequality follows from negative and strong negative correlation properties of our randomized rounding algorithm stated in Theorem 1.2.

Note that fixing the representative rectangle $R_{ij^*s^*}$ and the value of τ_{ij^*} fixes the value of θ_{ij^*} . Then, thanks to Observation 7, the first term becomes

$$\begin{aligned} & \sum_{j \neq j^*, s} x_{ij} \cdot p_{ij} \cdot \Pr[R_{ij} = R_{ijs}, e_{ij}] \\ &= \sum_{j \neq j^*, s} x_{ij} \cdot p_{ij} \cdot \Pr[e_{ij} \mid R_{ijs}] \cdot \Pr[R_{ij} = R_{ijs}] \\ &= \sum_{j \neq j^*, s} x_{ij} \cdot \frac{x_{ijs}}{x_{ij}} \cdot p_{ij} \cdot \frac{\hat{L}_{ijs}(\theta_{ij^*})}{p_{ij}} \quad [\text{Observation 7}] \\ &= \sum_{j \neq j^*, s} x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*}) \\ &= \sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*}) - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*}) \end{aligned}$$

The second term becomes

$$\begin{aligned}
& (1 - \eta) \sum_{j \neq j^*, s} x_{ij} \cdot p_{ij} \cdot \Pr[R_{ijs}, e_{ij}, j \stackrel{i}{\sim} j^*] \\
&= (1 - \eta) \sum_{j \neq j^*, s} x_{ij} \cdot p_{ij} \cdot \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ijs}] \cdot \Pr[R_{ijs}] \\
&= (1 - \eta) \sum_{j \neq j^*, s} x_{ijs} \cdot p_{ij} \cdot \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ijs}]
\end{aligned}$$

Thus, bringing back the omitted conditions τ_{ij^*} and $R_{ij^*s^*}$ to the equation, we have the lemma. $\square$

LEMMA 5.3. For each job j^* , and any representative rectangle $R_{ij^*s^*}$, we have

$$\begin{aligned}
(5.19) \quad & \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \\
& \leq \hat{s}^* + 1.5p_{ij^*} - \int_0^{p_{ij^*}} \frac{1}{p_{ij^*}} \cdot \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\hat{s}^* + \tau) d\tau
\end{aligned}$$

Proof. The trivial lower bound for the second negative term of the lemma 5.2 is zero. Thus

$$\begin{aligned}
& \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*}] \\
& \leq p_{ij^*} + \sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta_{ij^*}) - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*})
\end{aligned}$$

Note that the total volume of all rectangles appearing before time θ is at most θ . Further, $\sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta)$ is total volume of all rectangles appearing before time θ after shifting. Since we shift rectangles to the right side, we have $\sum_{j,s} x_{ijs} \cdot \hat{L}_{ijs}(\theta) \leq \theta$. Thus we have

$$\begin{aligned}
& \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*}] \\
&= p_{ij^*} + \theta_{ij^*} - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*}) \\
&= p_{ij^*} + \hat{s}^* + \tau_{ij^*} - \sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\theta_{ij^*})
\end{aligned}$$

Thus, taking the integral on the value of τ_{ij^*} , we have,

$$\begin{aligned}
& \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \\
& \leq p_{ij^*} + \hat{s}^* \\
& \quad + \int_0^{p_{ij^*}} \tau \cdot \frac{1}{p_{ij^*}} d\tau \\
& \quad - \int_0^{p_{ij^*}} \left(\sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\hat{s}^* + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau \\
& \leq 1.5 \cdot p_{ij^*} + \hat{s}^* \\
& \quad - \int_0^{p_{ij^*}} \left(\sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\hat{s}^* + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau
\end{aligned}$$

as desired. $\square$

LEMMA 5.4. For each job j we have

$$\begin{aligned}
& \sum_{s^*} x_{ij^*s^*} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \\
& \leq (1.5 - \frac{x_{ij^*}}{2}) x_{ij^*} p_{ij^*} + \sum_{s^*} x_{ij^*s^*} \hat{s}^*
\end{aligned}$$

Proof. Applying Lemma 5.3 to all rectangles $\{R_{ij^*s^*}\}_{s^*}$ of job j^* on machine i , we can say

$$\begin{aligned}
(5.20) \quad & \sum_{s^*} x_{ij^*s^*} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \\
& \leq \sum_{s^*} x_{ij^*s^*} \left(\hat{s}^* + 1.5p_{ij^*} \right. \\
& \quad \left. - \int_0^{p_{ij^*}} \left(\sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\hat{s}^* + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau \right)
\end{aligned}$$

Consider negative term here.

$$\begin{aligned}
& \sum_{s^*} x_{ij^*s^*} \int_0^{p_{ij^*}} \left(\sum_s x_{ij^*s} \cdot \hat{L}_{ij^*s}(\hat{s}^* + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau \\
&= \sum_{\{s^*, s\}: s \neq s^*} x_{ij^*s^*} \cdot x_{ij^*s} \cdot \int_0^{p_{ij^*}} \left(\hat{L}_{ij^*s}(\hat{s}^* + \tau) \right. \\
& \quad \left. + \hat{L}_{ij^*s^*}(\hat{s} + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau \\
(5.21) \quad & + \sum_{s^*} x_{ij^*s^*}^2 \int_0^{p_{ij^*}} \hat{L}_{ij^*s^*}(\hat{s}^* + \tau) \cdot \frac{1}{p_{ij^*}} d\tau
\end{aligned}$$

We can show that for any two different s^* and s we have

$$(5.22) \quad \int_0^{p_{ij^*}} \left(\hat{L}_{ij^*s}(\hat{s}^* + \tau) + \hat{L}_{ij^*s^*}(\hat{s} + \tau) \right) \cdot \frac{1}{p_{ij^*}} d\tau = p_{ij^*}$$

W.l.o.g we can assume $\hat{s}^* < \hat{s}$. Consider the case that two rectangles $R_{ij^*s^*}$ and R_{ij^*s} do not overlap after shifting. In this case, for any τ , $\hat{L}_{ij^*s^*}(\hat{s} + \tau) = p_{ij^*}$. Thus the equality (5.22) holds in this case. Now consider the case that two rectangles $R_{ij^*s^*}$ and R_{ij^*s} overlap after shifting. In this case when $\tau \in (0, \hat{s} - \hat{s}^*]$, $\hat{L}_{ij^*s}(\hat{s}^* + \tau) = 0$. When $\tau \in (\hat{s} - \hat{s}^*, p_{ij^*}]$, $\hat{L}_{ij^*s}(\hat{s}^* + \tau) = \hat{s}^* + \tau - \hat{s}$. For any

$\tau, \hat{L}_{ij^*s^*}(\hat{s} + \tau) = \min\{\hat{s} + \tau - \hat{s}^*, p_{ij^*}\}$. Thus we have

$$\begin{aligned} & \int_0^{p_{ij^*}} (\hat{L}_{ij^*s^*}(\hat{s}^* + \tau) + \hat{L}_{ij^*s^*}(\hat{s} + \tau)) \cdot \frac{1}{p_{ij^*}} d\tau \\ &= \int_{\hat{s}-\hat{s}^*}^{p_{ij^*}} (\hat{s}^* + \tau - \hat{s}) \cdot \frac{1}{p_{ij^*}} d\tau \\ &+ \int_0^{s^*+p_{ij^*}-s} (s + \tau - s^*) \cdot \frac{1}{p_{ij^*}} d\tau \\ &+ \int_{s^*+p_{ij^*}-s}^{p_{ij^*}} (p_{ij^*}) \cdot \frac{1}{p_{ij^*}} d\tau \\ &= p_{ij^*} \end{aligned}$$

Since $\hat{L}_{ij^*s^*}(\hat{s}^* + \tau) = \tau$ we have

$$(5.23) \quad \int_0^{p_{ij^*}} \hat{L}_{ij^*s^*}(\hat{s}^* + \tau) \cdot \frac{1}{p_{ij^*}} d\tau = p_{ij^*}/2$$

Applying Eqn.(5.22) and Eqn.(5.23) in Eqn.(5.21), the negative term of Eqn.(5.20) becomes:

$$\begin{aligned} & \sum_{\{s^*, s\}: s \neq s^*} x_{ij^*s^*} \cdot x_{ij^*s} \cdot p_{ij^*} + \sum_{s^*} x_{ij^*s^*}^2 \cdot \frac{p_{ij^*}}{2} \\ &= \frac{p_{ij^*}}{2} \left(\sum_{s^*} x_{ij^*s^*} \right)^2 \\ &= \frac{p_{ij^*} \cdot x_{ij^*}^2}{2} \end{aligned}$$

Plugging this equality in Eqn.(5.20) gives the lemma. $\square$

We now consider three cases in the following to prove Lemma 5.1.

5.1.1 When job j^* is good on machine i because $x_{ij^*} > \frac{9}{100}$ The following corollary follows from Lemma 5.4; the proof is deferred to the full version of this paper.

COROLLARY 5.1. *When job j^* is good on i because $x_{ij^*} > \frac{9}{100}$ we have*

$$\sum_{s^*} x_{ij^*s^*} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \leq 1.486 \sum_{s^*} x_{ij^*s^*} (s^* + p_{ij^*})$$

This gives Lemma 5.1 for each good job j^* with $x_{ij^*} \geq \frac{9}{100}$. Now we focus on proving Lemma 5.1 for each job j^* with $x_{ij^*} < \frac{9}{100}$ in the following.

5.1.2 When job j^* with $x_{ij^*} < \frac{9}{100}$ is good on machine i because its representative rectangle $R_{ij^*s^*}$ is good on i The goal of this subsection is to show Lemma 5.1 for each job j^* with $x_{ij^*} < \frac{9}{100}$ and any good representative rectangle

$R_{ij^*s^*}$. Towards this end, it suffices to show the following.

$$\begin{aligned} \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] &\leq \hat{s}^* + 1.5p_{ij^*} = 1.34s^* + 1.5p_{ij^*} \\ &= 1.486s^* - 0.146s^* + 1.5p_{ij^*} \\ &\leq 1.486s^* - 0.0146p_{ij^*} + 1.5p_{ij^*} \\ &\leq 1.486(s^* + p_{ij^*}) \end{aligned}$$

The first inequality is due to Lemma 5.3. The first equality comes from this fact as $x_{ij^*} < \frac{9}{100}$, we shift rectangle $R_{ij^*s^*}$ to the right side by $0.34s^*$, and the inequality follows from $10s^* \geq p_{ij^*}$ for any good rectangle $R_{ij^*s^*}$.

Thus, we have shown Lemma 5.1 for this case.

5.1.3 When job j^* is bad on machine i In this subsection we consider the case that $x_{ij^*} < \frac{9}{100}$ and the representative rectangle of job j^* is bad on i . To show Lemma 5.1 for this case, we will show,

$$(5.24) \quad \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*}] \leq p_{ij^*} + \hat{s}^* + 0.976\tau_{ij^*}$$

Then, by taking the integral on the value of τ_{ij^*} , we have,

$$\begin{aligned} & \mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \\ &\leq p_{ij^*} + \hat{s}^* + \int_0^{p_{ij^*}} 0.976\tau \cdot \frac{1}{p_{ij^*}} d\tau = \hat{s}^* + 1.488p_{ij^*} \end{aligned}$$

Since $\hat{s}^* = 1.34s^*$, this means we have $\mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}] \leq 1.488(s^* + p_{ij^*})$, as desired.

To prove Eqn.(5.24), we will reformat the time-indexed LP solution $\{x_{ijs}\}_{ijs}$, so that we have a linear combination of non-overlapping rectangles on each machine. This view will make our analysis easier. Formally, we will define a collection $\mathcal{F}_i$ of subsets of rectangles on each machine i that satisfies the following properties.

1. $\sum_{f \in \mathcal{F}_i} z_{if} \leq 1$
2. A configuration $f \in \mathcal{F}_i$ is a set of disjoint rectangles; that is, for any two distinct rectangles $R_{ijs}, R_{ijs'} \in f$, $(s, s + p_{ij})$ and $(s', s' + p_{ij})$ are disjoint.
3. For each rectangle R_{ijs} , we have $\sum_{f \ni R_{ijs}} z_{if} = x_{ijs}$.

An easy way to obtain this linear combination decomposition to replace each rectangle with sufficiently many copies of the same height preserving its total height. Then, it is not difficult to see that we can group rectangles as desired. We note that we can directly obtain this type of solution by solving a configuration LP [40].⁶

⁶The configuration LP solution is more structured than the above decomposition in that each configuration has at most one rectangle of each job. However, we do not need such a strong property for our rounding.

Now, we restate Lemma 5.2 using the above configuration view. As we will focus on a fixed machine i , we may omit $\mathcal{F}_i$. Due to the space constraints, we defer the proof of the following corollary to the full version of this paper.

COROLLARY 5.2. *For any job j^* , representative rectangle $R_{ij^*s^*}$ and any fixed $\tau_{ij^*} \in (0, p_{ij^*}]$, we have,*

$$\begin{aligned} & \mathbb{E} \left[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*} \right] - p_{ij^*} \\ &= \sum_f z_{if} \left(\hat{L}_f(\theta_{ij^*}) - \sum_{R_{ij^*s} \in f} \hat{L}_{ij^*s}(\theta_{ij^*}) \right) \\ & \quad - (1 - \eta) \sum_f \sum_{R_{ij^*s} \in f: j \neq j^*} z_{if} \cdot p_{ij} \\ & \quad \cdot \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ij^*s}, R_{ij^*s^*}, \tau_{ij^*}] \end{aligned}$$

where $\hat{L}_f(\theta_{ij^*})$ is the total length of all rectangles in f appearing up to time θ_{ij^*} after shifting.

We define D_f as the contribution of each configuration f to $\mathbb{E}[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*}]$ as following.

$$\begin{aligned} D_f &= z_{if} \hat{L}_f(\theta_{ij^*}) - z_{if} \sum_{R_{ij^*s} \in f} \hat{L}_{ij^*s}(\theta_{ij^*}) \\ & \quad - z_{if}(1 - \eta) \\ & \quad \cdot \sum_{R_{ij^*s} \in f: j \neq j^*} p_{ij} \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ij^*s}, R_{ij^*s^*}, \tau_{ij^*}] \end{aligned}$$

With this definition, we can say,

$$(5.25) \quad \mathbb{E} \left[\tilde{C}_{j^*} \mid i \leftarrow j^*, R_{ij^*s^*}, \tau_{ij^*} \right] = p_{ij^*} + \sum_f D_f$$

Since $\sum_f z_{if} = 1$ and $0.976\theta_{ij^*} \leq \hat{s}^* + 0.976\tau_{ij^*}$, if we show that

$$(5.26) \quad D_f \leq z_{if}(0.976\theta_{ij^*}) \text{ for all configurations } f,$$

then we can prove Eqn. (5.24), and consequently we can get Lemma 5.1 for a bad job j^* .

Henceforth, the goal of our analysis is to show Eqn. (5.26). As mentioned there is a random variable ρ in our algorithm that affects grid points, and therefore, affects the grouping of bad jobs on each machine. Thus, for each f , $R_{ij^*s}, R_{ij^*s^*} \in f$, where $j \neq j^*$, the probability $\Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ij^*s}, R_{ij^*s^*}, \tau_{ij^*}]$ in D_f depends on the value of ρ . As ρ is sampled from $(\frac{1}{10}, 1]$ uniformly at random, applying marginal probability and then Bayes' rule, we

have

$$(5.27) \quad \begin{aligned} D_f &= z_{if} \left(\hat{L}_f(\theta_{ij^*}) - \sum_{R_{ij^*s} \in f} \hat{L}_{ij^*s}(\theta_{ij^*}) \right. \\ & \quad - (1 - \eta) \sum_{R_{ij^*s} \in f: j \neq j^*} p_{ij} \cdot \frac{10}{9} \\ & \quad \cdot \left. \int_{\frac{1}{10}}^1 \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ij^*s}, R_{ij^*s^*}, \tau_{ij^*}, \rho] d\rho \right) \end{aligned}$$

Let $I \in \{I_k\}$ be the interval containing θ_{ij^*} and g be the starting point of this interval. Note that the rectangle $R_{ij^*s^*}$ is associated with I with probability $u = 1/2$. It is worth mentioning that, when θ_{ij^*} is fixed, fixing ρ determines I , and therefore, g as well. We first show that g is considerably smaller than θ_{ij^*} in expectation. Note that for any ρ value, $g > 0.1\theta_{ij^*}$.

LEMMA 5.5. *For any value of θ_{ij^*} , $\mathbb{E}[g] \leq \frac{11}{20}\theta_{ij^*}$.*

Proof. Let k be an integer such that $10^k \leq \theta_{ij^*} < 10^{k+1}$. Let α be such that $\theta_{ij^*} = \alpha 10^{k+1}$. Note that by definition of θ_{ij^*} and α , we have $\frac{1}{10} \leq \alpha < 1$. Since ρ is sampled from $(\frac{1}{10}, 1)$ uniformly at random, the interval including θ_{ij^*} must be either $I_k = (\rho 10^k, \rho 10^{k+1})$ or $I_{k+1} = (\rho 10^{k+1}, \rho 10^{k+2})$. Precisely, we have $g = \rho 10^{k+1}$ when $\rho \leq \alpha$, otherwise $g = \rho 10^k$. Therefore, we derive,

$$\begin{aligned} \mathbb{E}[g] &= \frac{10}{9} \int_{\frac{1}{10}}^{\alpha} \rho \cdot 10^{k+1} d\rho + \int_{\alpha}^1 \rho \cdot 10^k d\rho \\ &= 10^{k+1} \left(\frac{\alpha^2}{2} + \frac{1}{20} \right) \leq 10^{k+1} \frac{11}{20} \alpha = \frac{11}{20} \theta_{ij^*} \end{aligned}$$

The inequality follows from the fact that $\frac{\alpha^2}{2} + \frac{1}{20} \leq \frac{11}{20}\alpha$ when $\frac{1}{10} \leq \alpha < 1$. The last equality is immediate from the definition of θ_{ij^*} . $\square$

The following observation will be useful in our analysis.

OBSERVATION 8. *For any configuration f and value $\theta' \leq \theta_{ij^*}$, such that no rectangle $R_{ij^*s'}$ in f includes θ' , i.e. $\theta' \notin (s', s' + p_{ij'})$ we have $\hat{L}_f(\theta_{ij^*}) \leq \theta_{ij^*} - \min(0.34\theta', \theta_{ij^*} - \theta')$.*

Proof. Since θ' is not contained in the interior of any rectangle in f , a rectangle in f is either to the left or to the right of θ' . The rectangles in f after θ' will be shifted to the right side by at least $0.34\theta'$. When $0.34\theta' > \theta_{ij^*} - \theta'$, all rectangles after θ' in f will be shifted to the right side of θ_{ij^*} . Thus $\hat{L}_f(\theta_{ij^*}) \leq \theta'$. When $0.34\theta' < \theta_{ij^*} - \theta'$, the rectangles in f from $\theta_{ij^*} - 0.34\theta'$ to θ_{ij^*} will be shifted to the right side of θ_{ij^*} . Thus $\hat{L}_f(\theta_{ij^*}) \leq \theta_{ij^*} - 0.34\theta'$. Combining two cases gives the observation. $\square$

Consider the the interval $H = (0.1\theta_{ij^*}, 0.97\theta_{ij^*}]$. We upper bound D_f by considering three cases.

Case 1: The interval H is not a sub-interval of any rectangle (more precisely, the interval defined by the rectangle's starting and ending times) in f . In this case there is $\theta' \in H$ that is not interior of any rectangle in f . By observation 8, $\hat{L}_f(\theta_{ij^*}) \leq \theta_{ij^*} - \min\{0.03\theta_{ij^*}, 0.34\theta'\} \leq \theta_{ij^*} - \min\{0.03\theta_{ij^*}, 0.34 \times 0.1\theta_{ij^*}\} \leq 0.97\theta_{ij^*}$. Thus $D_f \leq z_{if}0.97\theta_{ij^*}$.

Case 2: The interval H is a sub-interval of a good rectangle $R_{ijs} \in f$. We first observe that the rectangle R_{ijs} is shifted to the right side by at least $0.03\theta_{ij^*}$. If $x_{ij} \geq \frac{9}{100}$, the rectangle R_{ijs} is shifted to the right side by at least $0.34(s + 0.09p_{ij}) \geq 0.34(0.09(s + p_{ij})) > 0.03\theta_{ij^*}$. When $x_{ij} < \frac{9}{100}$, R_{ijs} is good because $s > \frac{1}{10}p_{ij}$. In this case R_{ijs} is shifted to the right side by $0.34s \geq 0.34(0.09s + 0.91s) \geq 0.34 \times 0.09(s + p_{ij}) > 0.03\theta_{ij^*}$. As $s + 0.03\theta_{ij^*} < \theta_{ij^*}$, this means the total length of rectangles in f up to time θ_{ij^*} decreases by at least $0.03\theta_{ij^*}$ due to the shifting. Thus, we have $D_f \leq z_{if}0.97\theta_{ij^*}$.

Case 3: The interval H is a sub-interval of a bad rectangle $R_{ijs} \in f$. This is the case where we utilize strong negative correlations.

$$\begin{aligned} & \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ = & \Pr[e_{ij}, \theta_{ij} \in I, j \rightsquigarrow_i I, j^* \rightsquigarrow_i I, \\ & \sum_{j' \rightsquigarrow_i I} x_{ij'} \leq 1 \text{ for } I \text{ s.t. } \theta_{ij^*} \in I \mid R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ = & \Pr[g < \theta_{ij} < \min\{\theta_{ij^*}, \hat{s} + p_{ij}\} \mid R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ & \cdot \Pr[j \rightsquigarrow_i I \mid e_{ij}, \theta_{ij} \in I, R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ & \cdot \Pr[j^* \rightsquigarrow_i I \mid j \rightsquigarrow_i I, e_{ij}, \theta_{ij} \in I, R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ & \cdot \Pr[\sum_{j' \rightsquigarrow_i I} x_{ij'} \leq 1 \mid j^* \rightsquigarrow_i I, j \rightsquigarrow_i I, e_{ij}, \\ & \quad \theta_{ij} \in I, R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ \geq & \frac{\max\{\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - g, 0\}}{p_{ij}} \cdot u \cdot u \\ & \cdot \Pr[\sum_{j' \neq j, j^* \rightsquigarrow_i I} x_{ij'} \leq 0.82] \end{aligned}$$

Note that in the above equations, I is a grid interval including θ_{ij^*} . Thus, fixing $R_{ij^*s^*}$ and τ_{ij^*} means fixing I and its starting point g . In the third equation, the first probability is for the event e_{ij} and $\theta_{ij} \in I$. The second and third probabilities are for the events that j and j^* are associated with I respectively, conditioned on their θ values being in I – they are both $u = 1/2$. The last probability is lower bounded by the probability that the total height of jobs, other than j, j^* , assigned to I is at most 0.82, as only bad jobs are associated with I , meaning $x_{ij}, x_{ij^*} \leq 9/100$.

To keep the flow of the analysis, we defer the proof of the following lemma to the subsequent section.

LEMMA 5.6. $\Pr[\sum_{j' \neq j, j^* \rightsquigarrow_i I} x_{ij'} \leq 0.82] \geq 0.5317$.

Assuming Lemma 5.6 holds true, we can complete our analysis. We have

$$\begin{aligned} & \Pr[e_{ij}, j \stackrel{i}{\sim} j^* \mid R_{ijs}, R_{ij^*s^*}, \tau_{ij^*}, \rho] \\ & > 0.1323 \cdot \frac{\max\{\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - g, 0\}}{p_{ij}} \end{aligned}$$

Applying the above inequality to Eqn. (5.27) we have.

$$\begin{aligned} D_f & \leq z_{if} \hat{L}_f(\theta_{ij^*}) \\ & \quad - z_{if} \sum_{R_{ij^*s^*} \in f} \hat{L}_{ij^*s^*}(\theta_{ij^*}) \\ & \quad - z_{if}(1 - \eta) \int_{\frac{1}{10}}^1 \frac{10}{9} \cdot 0.1323 \\ & \quad \quad \cdot (\max\{\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - g, 0\}) d\rho \\ & \leq z_{if} \hat{L}_f(\theta_{ij^*}) \\ & \quad - z_{if} \sum_{R_{ij^*s^*} \in f} \hat{L}_{ij^*s^*}(\theta_{ij^*}) \\ & \quad - z_{if}(1 - \eta) \int_{\frac{1}{10}}^1 \frac{10}{9} \cdot 0.1323 \cdot (\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - g) d\rho \\ & \leq z_{if} \hat{L}_f(\theta_{ij^*}) \\ & \quad - z_{if} \cdot 0.1323 \cdot (1 - \eta) \cdot (\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} - E[g]) \end{aligned}$$

The last inequalities come from the fact $\min\{\theta_{ij^*}, \hat{s} + p_{ij}\} \geq \min\{\theta_{ij^*}, s + p_{ij}\}$ and Lemma 5.5. We observe that $\hat{L}_f(\theta_{ij^*})$ is also upper bounded by $\min\{\theta_{ij^*}, s + p_{ij}\}$. To see this, assume $s + p_{ij} < \theta_{ij^*}$ since otherwise this claim is immediate from the definition of $\hat{L}_f(\theta_{ij^*})$. Note that all rectangles in f starting at time $s + p_{ij}$ or later are shifted to the right by at least $0.34(s + p_{ij}) \geq 0.34 \cdot 0.97\theta_{ij^*}$. Since $s + p_{ij} \geq 0.97\theta_{ij^*}$, this means all those rectangles are shifted to the right side of θ_{ij^*} . Thus, we have the claim. Thus, we have,

$$\begin{aligned} D_f & \leq z_{if}(1 - 0.1323 \cdot (1 - \eta)) \cdot \min\{\theta_{ij^*}, s + p_{ij}\} \\ & \quad + z_{if}0.1323 \cdot (1 - \eta) \cdot \frac{11}{20} \cdot \theta_{ij^*} \\ & \leq z_{if} \left((1 - 0.1323 \cdot (1 - \eta)) \cdot \frac{9}{20} \theta_{ij^*} \right) \end{aligned}$$

Having $\eta < 0.589$, we can say $D_f \leq z_{if} \cdot 0.976 \theta_{ij^*}$.

5.1.4 Proof of Lemma 5.6 It now remains to prove Lemma 5.6.

LEMMA 5.7. For any configuration f on machine i and any grid interval I , we have

$$\sum_{R_{ijs} \in f: R_{ijs} \text{ is bad on } i} \frac{|I \cap (\hat{s}, \hat{s} + p_{ij})|}{p_{ij}} \leq 1$$

Proof. Fix a machine i . We first show that at most two bad rectangles from the same configuration f , after shifting, can overlap with a fixed grid interval I on a fixed machine i . To streamline our analysis, by scaling all time points uniformly, we assume wlog that $I = (1/10, 1)$. For the sake of contradiction, say there are more than two bad rectangles from f overlapping with I after shifting. Suppose $\hat{R}_{ij's'}$ is the last bad rectangle overlapping with I and $\hat{R}_{ijs}$ is the second to the last bad rectangle overlapping with I . Here we use $\hat{R}$ to denote the rectangle R after shifting.

The proof idea is to show that even if $\hat{R}_{ij's'}$ barely overlaps with I , the second to the last rectangle $\hat{R}_{ijs}$ must start before I . So, we can assume $\hat{s}' = 1$. For notational convenience, let $p := p_{ij}$ and $p' := p_{ij'}$. As $R_{ij's'}$ is bad, we know $\hat{s}' = 1.34s'$. From the observation that the two rectangles do not overlap and their relative order doesn't change by the shifting, we have $s' \geq s + p$. Since R_{ijs} is bad, we have $s \leq (1/10)p$. Therefore, we have $1 = 1.34s' \geq 1.34(s + p) \geq 1.34(11s) = 14.75s$. Thus, we have $\hat{s} = 1.34s \leq 1/11$. This means $\hat{R}_{ijs}$ starts before I does. This, $\hat{R}_{ijs}$ is the first bad rectangle from f overlapping with I , a contradiction.

From the above argument, it is straightforward to see that if the first bad rectangle overlapping with I is fully contained in I , then it is the only bad rectangle from f overlapping with I . In this case, the summation is exactly one. To see this is in fact the maximum of the summation, fix two adjacent bad rectangles $\hat{R}_{ijs}$ and $\hat{R}_{ij's'}$ from the same configuration, move I to the left or to the right. Since $p' > p$ from the above argument, it is easy to see that when we increase $\hat{R}_{ij's'}$'s overlap with I by one unit and decrease $\hat{R}_{ijs}$'s overlap with I by one unit, the summation decreases. This implies the summation is at most one, as desired. $\square$

LEMMA 5.8. For any machine, the expected total height of jobs associated with a grid interval I on the machine is at most u .

Proof. We have

$$\begin{aligned} \mathbb{E}\left[\sum_{j \rightsquigarrow_i I} x_{ij}\right] &= \sum_j x_{ij} \Pr[j \rightsquigarrow_i I] \\ &= \sum_j x_{ij} \sum_{s: R_{ijs} \text{ is bad on } i} \frac{x_{ijs}}{x_{ij}} \Pr[j \rightsquigarrow_i I \mid R_{ijs}] \\ &= \sum_j x_{ij} \sum_{s: R_{ijs} \text{ is bad on } i} \frac{x_{ijs}}{x_{ij}} \cdot u \cdot \frac{|I \cap (\hat{s}, \hat{s} + p_{ij})|}{p_{ij}} \\ &= u \sum_f \sum_{R_{ijs} \in f: R_{ijs} \text{ is bad on } i} z_{if} \cdot \frac{|I \cap (\hat{s}, \hat{s} + p_{ij})|}{p_{ij}} \\ &\leq u \sum_f z_{if} \quad (\text{Lemma 5.7}) \\ &\leq u \end{aligned}$$

The third equality follows since j is associated with I when $\theta_{ij} \in I$, which occurs with probability $\frac{|I \cap (\hat{s}, \hat{s} + p_{ij})|}{p_{ij}}$ conditioned on $R_{ij} = R_{ijs}$, and an independent fair coin ($u = 1/2$) gives a head. $\square$

To prove Lemma 5.6, we use the following well-known concentration inequality.

THEOREM 5.1. (THEOREM 2.3 OF [32]) Let Z be the sum of n independent random variables where each random variable takes value in $[0, K]$. Let $\mu = \mathbb{E}[Z]$. Then for any $\lambda \in [0, 1]$, we have

$$\Pr[Z \geq (1 + \lambda)\mu] \leq e^{-\lambda^2 \mu / 3K}.$$

Let $Z_{j'}$ denote the indicator variable for the event $j' \rightsquigarrow_i I$. Then, we know $\mathbb{E}[Z := \sum_{j' \neq j, j^*} x_{ij'} Z_{j'}] \leq u = 1/2$. Note that $x_{ij'} Z_{j'} \leq 9/100$, as j' is associated with interval I only when it is bad on machine i ; thus, we can set $K = 9/100$. Further, $\{Z_{j'}\}_{j' \neq j, j^*}$ are independent from one another. Therefore, by setting $\lambda = 0.64$, we obtain,

$$\begin{aligned} \Pr\left[\sum_{j' \neq j, j^* \rightsquigarrow_i I} x_{ij'} \leq 0.82\right] &= 1 - \Pr\left[\sum_{j' \neq j, j^* \rightsquigarrow_i I} x_{ij'} \geq 0.82\right] \\ &\geq 1 - \exp(-0.64^2 \cdot 0.5 / (3 \cdot 0.09)) > 0.5317. \end{aligned}$$

This completes the proof of Lemma 5.6.

References

- [1] F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein, and M. Sviridenko. Approximation schemes for minimizing average weighted completion time with release dates. In *FOCS*, 1999.
- [2] A. A. Ageev and M. I. Sviridenko. Approximation algorithms for maximum coverage and max cut with given sizes of parts. In *IPCO*, 1999.

- [3] S. Arora, A. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *FOCS*, 1996.
- [4] A. Asadpour, U. Feige, and A. Saberi. Santa claus meets hypergraph matchings. In *APPROX*. Springer, 2008.
- [5] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010.
- [6] Y. Azar and A. Epstein. Convex programming for scheduling unrelated parallel machines. In *STOC*, 2005.
- [7] N. Bansal and J. Kulkarni. Minimizing flow-time on unrelated machines. In *STOC 2015*, pages 851–860, 2015.
- [8] N. Bansal, A. Srinivasan, and O. Svensson. Lift-and-round to improve weighted completion time on unrelated machines. In *STOC 2016*, pages 156–167, 2016.
- [9] N. Bansal and M. Sviridenko. The santa claus problem. In *STOC*, 2006.
- [10] J. Bruno, E. G. Coffman Jr, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17(7):382–387, 1974.
- [11] D. Chakrabarty, J. Chuzhoy, and S. Khanna. On allocating goods to maximize fairness. In *FOCS*, 2009.
- [12] C. Chekuri and S. Khanna. A ptas for minimizing weighted completion time on uniformly related machines. In *ICALP 2001*, pages 848–861. Springer, 2001.
- [13] C. Chekuri and S. Khanna. Approximation algorithms for minimizing average weighted completion time. 2004.
- [14] C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *SODA 2011*, pages 1080–1097. SIAM, 2011.
- [15] T. Ebenlendr, M. Krčál, and J. Sgall. Graph balancing: a special case of scheduling unrelated parallel machines. In *SODA*, 2008.
- [16] U. Feige. On allocations that maximize fairness. In *SODA*, 2008.
- [17] U. Feige and J. Vondrák. Approximation algorithms for allocation problems: Improving the factor of $1 - 1/e$. In *FOCS 2006*, pages 667–676, 2006.
- [18] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *JACM*, 53(3):324–360, 2006.
- [19] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.
- [20] N. Garg and A. Kumar. Minimizing average flow-time : Upper and lower bounds. In *FOCS*, 2007.
- [21] N. Garg, A. Kumar, and V. N. Muralidhara. Minimizing total flow-time: The unrelated case. In *ISAAC*, 2008.
- [22] H. Hoogeveen, P. Schuurman, and G. J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. *INFORMS Journal on Computing*, 13(2):157–168, 2001.
- [23] W. Horn. Minimizing average flow time with parallel machines. *Operations Research*, 21(3):846–847, 1973.
- [24] S. Im and S. Li. Better unrelated machine scheduling for weighted completion time via random offsets from non-uniform distributions. In *FOCS*, 2016.
- [25] J. Kahn and P. M. Kayll. On the stochastic independence properties of hard-core distributions. *Combinatorica*, 17(3):369–391, 1997.
- [26] C. Kalaitzis, O. Svensson, and J. Tarnawski. Unrelated machine scheduling of jobs with uniform smith ratios. In *SODA*, 2017.
- [27] V. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. A unified approach to scheduling on unrelated parallel machines. *Journal of the ACM (JACM)*, 56(5):28, 2009.
- [28] V. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Minimum weighted completion time. In *Encyclopedia of Algorithms*, pages 544–546. Springer, 2008.
- [29] J. K. Lenstra, A. R. Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of discrete mathematics*, 1:343–362, 1977.
- [30] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3):259–271, 1990.
- [31] S. Li. Scheduling to minimize total weighted completion time via time-indexed linear programming relaxations. In *FOCS 2017*, pages 283–294, 2017.
- [32] C. McDiarmid. Concentration, probabilistic methods for algorithmic discrete mathematics, 195–248. *Algorithms Combin*, 16, 1998.
- [33] A. S. Schulz and M. Skutella. Scheduling unrelated machines by randomized rounding. *SIAM Journal on Discrete Mathematics*, 15(4):450–469, 2002.
- [34] P. Schuurman and G. J. Woeginger. Polynomial time approximation algorithms for machine scheduling: Ten open problems. *Journal of Scheduling*, 2(5):203–213, 1999.
- [35] J. Sethuraman and M. S. Squillante. Optimal scheduling of multiclass parallel machines. In *SODA*, pages 963–964, 1999.
- [36] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *J. ACM*, 48(2):206–242, 2001.
- [37] M. Skutella and G. J. Woeginger. A ptas for minimizing the total weighted completion time on identical parallel machines. *Mathematics of Operations Research*, 25(1):63–75, 2000.
- [38] W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- [39] O. Svensson. Santa claus schedules jobs on unrelated machines. *SIAM Journal on Computing*, 41(5):1318–1341, 2012.
- [40] M. Sviridenko and A. Wiese. Approximating the configuration-lp for minimizing weighted sum of completion times on unrelated machines. In *IPCO*. 2013.