Stochastic and Robust Scheduling in the Cloud*

Lin Chen[1], Nicole Megow[2], Roman Rischke[1], and Leen Stougie[3]

1 **Department of Mathematics, Technische Universität Berlin, Germany.** `{lchen,rischke}@math.tu-berlin.de`
2 **Center for Mathematics, Technische Universität München, Germany.** `nicole.megow@tum.de`
3 **Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam & CWI, The Netherlands.** `stougie@cwi.nl`

Abstract

Users of cloud computing services are offered rapid access to computing resources via the Internet. Cloud providers use different pricing options such as (i) time slot reservation in advance at a fixed price and (ii) on-demand service at a (hourly) pay-as-used basis. Choosing the best combination of pricing options is a challenging task for users, in particular, when the instantiation of computing jobs underlies uncertainty.

We propose a natural model for two-stage scheduling under uncertainty that captures such resource provisioning and scheduling problem in the cloud. Reserving a time unit for processing jobs incurs some cost, which depends on when the reservation is made: a priori decisions, based only on distributional information, are much cheaper than on-demand decisions when the actual scenario is known. We consider both stochastic and robust versions of scheduling unrelated machines with objectives of minimizing the sum of weighted completion times $\sum_j w_j C_j$ and the makespan $\max_j C_j$. Our main contribution is an $(8+\epsilon)$-approximation algorithm for the min-sum objective for the stochastic polynomial-scenario model. The same technique gives a $(7.11+\epsilon)$-approximation for minimizing the makespan. The key ingredient is an LP-based separation of jobs and time slots to be considered in either the first or the second stage only, and then approximately solving the separated problems. At the expense of another ϵ our results hold for any arbitrary scenario distribution given by means of a black-box. Our techniques also yield approximation algorithms for robust two-stage scheduling.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.3 Probability and Statistics

Keywords and phrases Approximation Algorithms, Robust Optimization, Stochastic Optimization, Unrelated Machine Scheduling, Cloud Computing

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Users of cloud computing services are offered rapid access to computing resources such as processing power, storage capacity, or network bandwidth via the Internet. Cloud providers, e.g. Amazon EC2, use different pricing options such as *on-demand* and *reserved instances* [1]. In the reservation option, a user pays a priori a fixed amount to reserve resources in advance, whereas on-demand instances are charged on a (e.g. hourly) pay-as-used basis. Users of cloud computing services face the challenging task of choosing the best combination of pricing

* Supported by the German Science Foundation (DFG) under contract ME 3825/1.

options when provisioning resources [4] – in particular, if instances of computing jobs underlie uncertainty.

In this paper, we propose the following general model for *two-stage scheduling with reservation cost under uncertainty* that captures such resource provisioning and scheduling problem in the cloud. In the first stage, we are given distributional information about scheduling scenarios, and in the second stage the actual scenario is revealed. The task is to construct a schedule for the realized scenario. Using a time unit of processing in the schedule incurs some fixed cost, independent of the used capacity (number of machines), but dependent on when the time unit is reserved: low if it is reserved in the first stage, not knowing the actual scenario, and high in the second stage, given full information. Such a cost structure applies, for example, when reserving a time unit on a server gives access to all processors on this server. In the *stochastic setting*, the overall goal is to minimize total expected payment (in both stages) plus scheduling cost. In the *robust setting*, the overall goal is to minimize the maximum, over all scenarios, of payment (in both stages) plus scheduling cost.

This setting opens up a whole new class of scheduling problems with its own particular challenges. As a first problem in this class we focus on scheduling preemptive jobs with release dates on unrelated machines, the most general machine model in scheduling, such as to minimize the total weighted completion time and makespan. The corresponding single-stage, single-scenario versions of these problems are fundamental classical scheduling problems. We give constant approximation algorithms for both objectives in the stochastic and the robust model. Our results for the stochastic setting hold in the most general random model, the so-called black-box model.

Problem Definition. In the underlying deterministic problem we are given a set of jobs $J = \{1, \ldots, n\}$ and a set of machines $M = \{1, \ldots, m\}$. Each job $j \in J$ is specified by a release date $r_j \geq 0$, before which j cannot be processed, a machine-dependent processing time $p_{ij} \in \mathbb{N}$, the processing time when executing j completely on machine $i \in M$, and a weight $w_j \geq 1$. In a feasible schedule each machine runs at most one job at the time and no job runs at more than one machine at the same time. A job may be preempted at any time and may resume processing on the same or any other machine. We assume that time is discretized into unit time slots. For ease of exposition let completion time C_j of a job $j \in J$ be the end of the unit-size time slot in which it actually completes. For every time slot, in which at least one machine is processing, a fixed *reservation cost c* is paid. The objective is to minimize the sum of weighted completion times $\sum_{j\in J} w_j C_j$ or the makespan $C_{\max} := \max_j C_j$ plus total reservation cost.

In the *two-stage* version of this problem, the actual job set to be processed is one of a set $\mathcal{S}$ of possible scenarios. Any time slot can be reserved either in the first stage at cost c, and can be used in every scenario, or in the second stage, for a specific scenario, at cost λc, where $\lambda \geq 1$ is an inflation factor. We assume λ to be defined by the scenario as well. The inflation factor together with the job set, $(\lambda_k \geq 1, J_k \subseteq J)$, make up a scenario $k \in \mathcal{S}$.

In the stochastic setting, we consider two models with respect to randomness. In the *polynomial-scenario model*, the distribution of $\mathcal{S}$ is given explicitly, i.e., each scenario $k \in \mathcal{S}$ is associated with a probability $\pi_k \in [0, 1]$ with $\sum_{k\in\mathcal{S}} \pi_k = 1$. In the *black-box model*, we have efficient access to an oracle that provides samples according to the unknown probability distribution with possibly exponentially many and dependent scenarios. In the robust setting, we restrict to the model with an explicit description of $\mathcal{S}$, called *discrete-scenario model.*

Related Work. Preemptively scheduling unrelated machines to minimize the sum of (weighted) completion times, $R\,|\,pmtn,(r_j)\,|\,\sum(w_j)C_j$, is $\mathcal{APX}$-hard [23] and admits a $(2+\epsilon)$-approximation [18]. The makespan problem $R\,|\,r_j, pmtn\,|\,C_{\max}$ can be solved in polynomial time [15].

Somewhat related to our scheduling problem with reservation cost is the problem of scheduling with variable time-slot cost [6,14,27]. Our second-stage problem involves a special case with time slots having cost either 0 or some fixed amount.

With respect to the stochastic problem, our work is closer to two- or multistage stochastic versions of scheduling problems, see e.g. [2,3], than to traditional *dynamic stochastic scheduling* [17], in which the algorithm's decision on processing a job or not crucially influences the information release. However, the former involve different scheduling problems than ours, and more importantly performance quality is assessed by computational experiments instead of rigorous worst-case analysis. The only work on approximation algorithms for a two-stage scheduling problem we are aware of is by Shmoys and Sozio [21]. They present a $(2+\epsilon)$-approximation for two-stage stochastic throughput maximization on a single machine in which jobs can be deferred in the first stage gaining some profit.

The study of approximation algorithms for two-stage stochastic optimization problems was initiated in [8] with a polynomial-scenario model for a service-provision problem. Subsequently, next to [21] above, various two-stage stochastic versions of combinatorial optimization problems such as set cover, network design, maximum weight matching, etc. were studied, see [25] for a nice overview on the earlier work. General frameworks for solving several two-stage stochastic combinatorial optimization problems approximately have been proposed in [11] and [22]. The cost-sharing based approach in [11] yields a 2-approximation for a two-stage stochastic scheduling problem without release dates on identical parallel machines [16]. It is not clear how to apply it when there are release dates.

In the black-box model, we adopt the *Sample Average Approximation (SAA) method* proposed in [13]. It replaces the distribution on the random parameters by its empirical distribution defined by samples from it. Under certain conditions, good approximate solutions are obtained by drawing only a polynomial number of samples and solving the resulting SAA problem instead [5,26].

In a two-stage setting, robust versions of multiple-scenario combinatorial optimization problems have been studied for covering and network design problems in [7,9,12]. We are not aware of any relevant scheduling work.

Our Contribution. We develop approximation algorithms for the stochastic and robust two-stage variants of classical scheduling problems. Our results rely on a new scheduling-tailored *time slot and job-set separation* procedure, which separates jobs into those processing exclusively on either first-stage reserved slots or second-stage reserved slots. It is inspired by [22] in which the idea of separating clients was introduced in the context of covering and network design problems. The separation in our setting is achieved through solving a generalization of the time-indexed unrelated machine scheduling LP [20] followed by an application of the slow-motion technique, proposed in [19] for min-sum single machine scheduling and extended later, among others, to unrelated machines scheduling in [18]. After separating, our rounding is applied independently to both separated instances. The two (possibly overlapping) solutions are merged to a feasible joint solution. Carefully balancing the change in reservation and scheduling cost by exploiting properties of slow-motion and α-points, the resulting procedure is proven to be an $(8+\epsilon)$-approximation algorithm for the two-stage polynomial-scenario stochastic version of $R\,|\,pmtn, r_j\,|\,\sum_j w_jC_j$ (Sec. 2.3).

Our time slot and job-set separation procedure is based on a general result, which is interesting on its own in the polynomial-scenario model: Given a ρ-approximation for the special case in which slots are reserved only in the first stage, there is an 8ρ-approximation for the two-stage model (Sec. 2.2). For this special case, we give a $\rho = (3+\epsilon)$-approximation (Sec. 2.1).

Our techniques also yield a $64/9 + \epsilon \approx 7.11 + \epsilon$ algorithm for the two-stage stochastic version of the makespan problem $R \mid r_j, pmtn \mid C_{\max}$ (Sec. 2.3).

Adopting the SAA framework, mentioned before, we apply our algorithms to arrive at a sampling-based $(8 + \epsilon)$-approximation algorithm for the min-sum problem and a $(64/9 + \epsilon)$-approximation for minimizing the makespan (Sec. 3). We notice that the work of [5, 26] leads to a first-stage reservation decision. It is not obvious in our model how to construct a good second-stage solution given a set of slots for free from the first-stage solution. In fact, considering this problem independently from the first-stage, it is unclear if a constant approximation exists. But even when considering the two stages jointly, the difficult part is to show how a second-stage solution for some scenario (not necessarily in the sample set) can be found and bounded by the SAA solution for the sample set.

Finally, we argue that our algorithms can be adopted for the min-max robust optimization model with a discrete set of scenarios (Sec. 4). For the min-$\sum w_j C_j$ problem, certain randomized steps of our algorithm must be replaced by deterministic ones losing a factor 2 in the approximation guarantee. For the robust makespan problem we derive a 2-approximation.

In this paper, we consider the most interesting and most general variants of the considered problems. For several special cases we can improve results, omitting details in this paper. E.g., when all jobs in all scheduling scenarios are released at time 0, then obviously the (first-stage) reservation interval will be $[0, t]$ for some t. It is not difficult to see that our considered objective functions (as well as others such as minimizing the ℓ_p-norm of machine loads) of the two-stage problems without release dates are convex in t. Hence, we find the optimal t simply by a combination of binary search for t and known approximation algorithms for the single-stage single-scenario problems to determine the total cost for a given t. Thus, in the absence of release dates, the two-stage problem is not harder than the underlying deterministic problem. This changes drastically when jobs have arbitrary release dates. Further improved results for other special cases, such as less general machine environments or a constant number of scenarios, will be given in the full version of the paper.

2 Polynomial-Scenario Model for Min-Sum Objective

Consider the two-stage stochastic version of $R \mid r_j, pmtn \mid \sum w_j C_j$ in the polynomial-scenario model, in which the set of scenarios $\mathcal{S}$ and its distribution are fully specified. For each scenario $k \in \mathcal{S}$ the triple (π_k, λ_k, J_k) describes its probability of occurring π_k, the inflation factor λ_k, and the set of jobs J_k.

We use a natural LP that generalizes and further relaxes the time-indexed LP for unrelated-machine scheduling [20]. To facilitate the exposition, we will present an LP with exponentially many variables and constraints and derive our algorithms based on its solution, even though we cannot expect to solve it in polynomial time. However, using the standard technique of time-intervals of geometrically increasing size [20] we obtain polynomial-time algorithms loosing only a small factor.

To keep notation amenable, we re-index jobs, such that each job j refers to a unique job-scenario combination, and we let $J := J_1 \cup \ldots \cup J_N$. We choose the time horizon $T = \max_{k \in \mathcal{S}, j \in J_k}\{r_j\} + \max_{k \in \mathcal{S}}\{\sum_{j \in J_k} \max_{i \in M} p_{ij}\}$, an obvious upper bound on any completion time in a reasonable schedule. Variables x_t and x_{kt} represent the first and second stage reservation decisions for time slot $[t, t+1)$. Let y_{ijt} be the amount of time job j is processed in interval $[t, t+1)$ on machine i.

$$\min \sum_{t=0}^{T-1} cx_t + \sum_{k=1}^{N} \pi_k \left(\sum_{j \in J_k} w_j C_j^{LP} + \lambda_k c \sum_{t=0}^{T-1} x_{kt} \right) \tag{1a}$$

$$\text{s.t.} \sum_{j \in J_k} y_{ijt} \leq x_t + x_{kt} \qquad \forall i \in M, k \in \mathcal{S}, 0 \leq t \leq T-1 \tag{1b}$$

$$\sum_{i \in M} y_{ijt} \leq x_t + x_{kt} \qquad \forall j \in J, k \in \mathcal{S}, 0 \leq t \leq T-1 \tag{1c}$$

$$x_t + x_{kt} \leq 1 \qquad \forall k \in \mathcal{S}, 0 \leq t \leq T-1 \tag{1d}$$

$$\sum_{t=r_j}^{T-1} \sum_{i \in M} \frac{y_{ijt}}{p_{ij}} = 1 \qquad \forall j \in J \tag{1e}$$

$$\sum_{t=r_j}^{T-1} \sum_{i \in M} (t+1) \cdot \frac{y_{ijt}}{p_{ij}} = C_j^{LP} \qquad \forall j \in J \tag{1f}$$

$$x_t, x_{kt}, y_{ijt} \in [0,1] \qquad \forall i \in M, j \in J, k \in \mathcal{S}, 0 \leq t \leq T-1 \tag{1g}$$

Constraints (1b),(1d),(1e),(1g) are self-explaining. With (1c) we ensure that no job is processed for more than the total amount reserved in $[t, t+1)$ guaranteeing non-parallelism. For (1f), consider an arbitrary schedule with $t_j = \max_t\{t | y_{ijt} > 0, i \in M\}$, then the *true completion time* of job j in this schedule is $t_j + 1$, while C_j^{LP} offers a lower bound. Thus, even if we enforce all variables to be integral, the LP still gives a relaxation of our problem.

Given an LP solution (x_t, x_{kt}, y_{ijt}), we let $LP^r = \sum_t cx_t + c\sum_{k,t} \pi_k \lambda_k x_{kt}$ denote the reservation cost and we let $LP^s = \sum_{k, j \in J_k} \pi_k w_j C_j^{LP}$ denote the scheduling cost.

2.1 An Algorithm for First-Stage Reservation Only

Consider the special case of the two-stage problem in which all reservation must be done in the first stage; as if all inflation factors λ_k are excessive. We refer to it as *problem with first-stage reservation only*. A lower bound is given by LP_1 obtained from the above LP by setting $x_{kt} = 0$, for all k, t. W.l.o.g. we omit the π_k pre-multiplication in the objective function by assuming it to be incorporated into the weights $w_j, j \in J_k$.

We describe a procedure for rounding a fractional solution (x_t, y_{ijt}) of LP_1 to a feasible integer-value solution. We first round the first-stage decision on reserving slots x_t by maintaining a feasible LP solution, and then we determine the actual schedule. In the first step, it is important to maintain a fractional scheduling solution in which the true completion time of a job j, i.e., $\max\{t+1 \,|\, y_{ijt} > 0, i \in M\}$, does not diverge too much from C_j^{LP}. To that end, we utilize the idea of *slow-motion*, proposed in [19] for single machine scheduling, and extended to unrelated machines scheduling in [18].

For $\alpha \in [0,1]$, let $C_j(\alpha)$ denote the earliest point in time in the LP-solution in which job j has completed an α-fraction of its total processing requirement. We use the following link between $C_j(\alpha)$ an C_j^{LP} adopted from [10], which is used for the analysis of randomized algorithms.

▶ **Lemma 1** ([10]). $\int_0^1 C_j(\alpha) d\alpha = \sum_t \sum_i y_{ijt}/p_{ij} \cdot (t+1/2) = C_j^{LP} - 1/2$.

For deterministic algorithms, however, we use the following relation.

▶ **Lemma 2** ([24]). $C_j^{LP} \geq \alpha + (1-\alpha) \cdot C_j(\alpha)$.

Slow-Motion. Given a fractional solution (x_t, y_{ijt}) for LP_1 and $\beta \geq 1$, we construct a new β-expanded solution $(\beta x_t, \beta y_{ijt})$ that we obtain by mapping every time point t to βt. Then βx_t indicates the amount of reservation for the interval $[\beta t, \beta(t+1))$, and βy_{ijt} the amount of job j scheduled during $[\beta t, \beta(t+1))$.

Obviously $(\beta x_t, \beta y_{ijt})$ is a new feasible solution to LP_1, which over-schedules each job by a fraction of $\beta - 1$. We simply delete the over-scheduled part and apply a lemma given in [19] that upper bounds the completion times of jobs in the expanded solution. We directly adopt their result to our requirement of job completion times being rounded up to integer values.

▶ **Lemma 3** ([19]). *The completion time of job j in the expanded solution is at most $\lceil \beta C_j(1/\beta) \rceil$.*

Rounding time slot reservation. Given any fractional solution (x_t, y_{ijt}), e.g. the expanded LP_1 solution, we show how to round the fractional reservation x_t to 0 or 1 so that the number of slots reserved will not be much higher than $\sum_t x_t$ but sufficiently large to accommodate all workload. We reassign the workload to reserved slots ensuring that the completion times remain relatively small.

We first apply a standard rounding technique, which we call *accumulated reservation:* Let $X_t = \sum_{h=0}^{t} x_h$, for $t \in \{0, 1, \ldots, T-1\}$ and $X_{-1} = 0$. We set $\bar{x}_t = 1$, i.e., we reserve time slot $[t, t+1)$, if $\lfloor X_{t-1} \rfloor \leq \lfloor X_t \rfloor - 1$, and set $\bar{x}_t = 0$ otherwise. In total, we reserve $\lfloor \sum_t x_t \rfloor$ slots this way. To ensure sufficiently reserved time capacity we do an *extra reservation:* if $\bar{x}_t = 1$ and $\bar{x}_{t+1} = 0$ for some t, we reserve additionally the slot $[t+1, t+2)$. The number of extra reserved time slots is no more than the number of accumulatively reserved slots.

▶ **Proposition 4.** *Given a fractional solution x_t, the total cost for rounding to an integral time slot reservation $\bar{x}_t$ is $c \sum_{t=0}^{T-1} \bar{x}_t \leq 2c \lfloor \sum_{t=0}^{T-1} x_t \rfloor \leq 2LP^r$.*

Our reservation policy creates intervals $I_1, I_2, \ldots$ of consecutive reserved time slots, each of them starting with a set of accumulatively reserved time slots and ending with a single extra time slot.

▶ **Lemma 5.** *Every interval $I_h = [\underline{t}_h, \bar{t}_h + 2)$ has enough capacity to accommodate all workload y_{ijt} assigned to time units $[\bar{t}_{h-1} + 1, \bar{t}_{h-1} + 2), \ldots, [\underline{t}_{h+1} - 1, \underline{t}_{h+1})$.*

Proof. Consider interval I_h. Its last time slot $[\bar{t}_h + 1, \bar{t}_h + 2)$ is the extra reserved time slot. The total number (capacity) of the time slots in I_h is $\lfloor X_{\bar{t}_h} \rfloor - \lfloor X_{\bar{t}_{h-1}} \rfloor + 1$. By definition of our accumulative reservation and according to constraints (1b) and (1c), the total workload in terms of y_{ijt} in the interval $[\bar{t}_{h-1} + 1, \underline{t}_{h+1})$ is bounded by

$$\sum_{t=\bar{t}_{h-1}+1}^{\underline{t}_{h+1}-1} x_t = \sum_{t=0}^{\underline{t}_{h+1}-1} x_t - \sum_{t=0}^{\bar{t}_{h-1}} x_t \leq X_{\underline{t}_{h+1}-1} - \lfloor X_{\bar{t}_{h-1}} \rfloor \leq \lfloor X_{\bar{t}_h} \rfloor - \lfloor X_{\bar{t}_{h-1}} \rfloor + 1.$$

◀

The lemma implies that none of the workload y_{ijt}, fractionally assigned to time slots up to time slot $[\bar{t}_h, \bar{t}_h + 1)$, needs to be done later than $\bar{t}_h + 2$ if appropriately reassigned. In particular, this holds for the last reserved interval, i.e., all jobs in all scenarios will have been processed. Even stronger, workload assigned to the time slots $[\bar{t}_h + 1, \bar{t}_h + 2), \ldots, [\underline{t}_{h+1} - 1, \underline{t}_{h+1})$ can be done within interval I_h, unless the release date of some job j is larger than $\bar{t}_h + 1$, preventing it to be scheduled in I_h.

Reassigning workload. Given a (not necessarily feasible) solution with fractional scheduling variables y_{ijt} and integer-valued reserved time slots $\bar{x}_t$, we describe a reassignment procedure to arrive at a feasible solution in which all workload $\bar{y}_{ijt}$ is assigned to reserved slots.

In increasing order of t we claim a total fraction x_t from time slot $[\underline{t}_h, \underline{t}_h + 1)$ if $\bar{t}_{h-1} + 1 \leq t < \underline{t}_h$ for some h. All of the y_{ijt} is moved into this claimed space and added to $\bar{y}_{ij\underline{t}_h}$. Otherwise if $[t, t+1) \in I_h$, and $t \neq \bar{t}_h + 1$, then we claim $\frac{\lfloor X_t \rfloor - X_{t-1}}{X_t - X_{t-1}} x_t$ from $[t, t+1)$ and $\frac{X_t - \lfloor X_t \rfloor}{X_t - X_{t-1}} x_t$ from $[t+1, t+2)$. All of the y_{ijt} is moved in equal proportions $\frac{\lfloor X_t \rfloor - X_{t-1}}{X_t - X_{t-1}} y_{ijt}$ and $\frac{X_t - \lfloor X_t \rfloor}{X_t - X_{t-1}} y_{ijt}$ into the claimed space and added, respectively, to $\bar{y}_{ijt}$ and $\bar{y}_{ij(t+1)}$.

This assignment leaves (unclaimed) capacity $\lceil X_{\bar{t}_h} \rceil - X_{\bar{t}_h}$ of the extra reserved time slot $[\bar{t}_h + 1, \bar{t}_h + 2)$, for each h. As a second reassignment step we remove all y_{ijt} with $\bar{t}_{h-1} + 1 \leq t < \underline{t}_h$ and j with $r_j \leq \bar{t}_{h-1} + 1$ from $\bar{y}_{ij\underline{t}_h}$ and add it to $\bar{y}_{ij\tau}$ where $\tau = \bar{t}_{h-1} + 1$. This is feasible by Lemma 5.

▶ **Lemma 6.** *Applying the reservation and reassignment procedures to a feasible solution (x_t, y_{ijt}) of LP_1 increases the completion time of any job j by at most 1.*

Proof. For every t for which $y_{ijt} > 0$, there are two cases:
Case 1: $[t, t+1) \in I_h$, and $t \neq \bar{t}_h + 1$ for some h. Then by the assignment procedure y_{ijt} is moved into $[t, t+1)$ and $[t+1, t+2)$, whence that part of job j finishes at most 1 time unit later than in the unrounded solution. In particular, this holds for the last t such that $y_{ijt} > 0$.

Case 2: $\bar{t}_{h-1} + 1 \leq t < \underline{t}_h$ for some h. If $r_j \leq \bar{t}_{h-1} + 1$, then y_{ijt} is moved into $[\bar{t}_{h-1} + 1, \bar{t}_{h-1} + 2)$ and finishes earlier, or if $r_j > \bar{t}_{h-1} + 1$, then y_{ijt} is moved into $[\underline{t}_h, \underline{t}_h + 1)$. However, since $p_{ij} \geq 1$ (viz. integer), by the shifted-reservation policy job j cannot have been completed before $\underline{t}_h$, i.e., there must be a $t \geq \underline{t}_h$ and/or another i such that $y_{ijt} > 0$. ◀

The one-stage reservation algorithm. Given an optimal solution to LP$_1$, we apply slow-motion to expand the solution and the time slot reservation to obtain integral reservations to which we reassign the workload. It remains to specify the actual schedule for workload $\bar{y}_{ijt}$ within a time slot $[t, t+1)$ by ensuring that a job is not scheduled in parallel on multiple machines. This is essentially $R|pmtn|C_{max}$ in each time slot, which is polynomial-time solvable [15].

▶ **Theorem 7.** *The one-stage reservation algorithm is a* 3.5*-approximation for two-stage scheduling on unrelated machines with first-stage reservation only.*

Proof (Sketch). Consider an optimal solution to LP$_1$. By slow-motion we derive a β-expanded solution with a reservation cost of βLP^r, and job j completes at time $\lceil \beta C_j(1/\beta) \rceil$ by Lemmas 1 and 3. Applying the rounding of time slot reservation and then reassigning workload, Proposition 4 shows that the reservation cost becomes $2\beta LP^r$, while Lemma 6 ensures that job j completes at $\lceil \beta C_j(1/\beta) \rceil + 1$. Thus, by choosing the expansion parameter β at random according to the density function $f(\alpha) = 3\alpha^2$ where $\alpha = 1/\beta \in [0, 1]$, the total cost for reservation and scheduling can be bounded by:

$$
\begin{aligned}
2LP^r \int_0^1 \frac{1}{\alpha} f(\alpha) d\alpha \quad &+ \quad \sum_j w_j \int_0^1 (\lceil C_j(\alpha)/\alpha \rceil + 1) f(\alpha) d\alpha \\
&\leq \quad 3(LP^r + LP^s) + 1/2 \sum_j w_j.
\end{aligned}
$$

Obviously, $\sum_j w_j \leq \sum_j w_j C_j^{LP}$, since $C_j^{LP} \geq 1$, which implies that the algorithm produces a solution with objective value bounded by $3LP^r + 3.5LP^s$. ◀

A refinement of the algorithm and its analysis give an improved bound.

▶ **Theorem 8.** *There exists a $(3+\epsilon)$-approximation algorithm for the two-stage scheduling problem on unrelated machines with first-stage reservation only.*

2.2 A Generic Algorithm for Two-Stage Scheduling

We first give a simple algorithm that allows a black-box application of the one-stage reservation algorithm above to obtain the following general result.

▶ **Theorem 9.** *Given a ρ-approximation algorithm for the two-stage problem with only first-stage reservation, there exists an 8ρ-approximation algorithm for the two-stage problem.*

The crucial ingredient is to separate the time slots and jobs to be considered for only first-stage or only second-stage reservation.

▶ **Lemma 10.** *Given an optimal solution (x_t, x_{kt}, y_{ijt}) to the LP with objective value $LP^r + LP^s$, there exists a feasible solution $(x'_t, x'_{it}, y'_{hjt})$ satisfying the following* **separation property***:*

- *Any time unit is reserved either in the first stage or in the second stage, or not at all; i.e., for all t, $x'_t > 0 \Rightarrow x'_{kt} = 0\ \forall k$.*
- *A job is scheduled either completely in slots reserved in the first stage, or completely in slots reserved in the second stage, i.e., $J = J_I \cup J_{II}$, s.t. $J_I = \{j \mid x'_t = 0 \Rightarrow y'_{ijt} = 0\ \forall ijt\}$ and $J_{II} = \{j \mid \sum_k x'_{kt} = 0 \Rightarrow y'_{ijt} = 0 \forall ijt\}$.*
- *The objective value is at most $2LP^r + 4LP^s$.*

Proof. We first double the number of time units: for every time unit $[t, t+1)$ we obtain two time units $[2t, 2t+1)$ and $[2t+1, 2t+2)$. We reserve x_t of the even slot $[2t, 2t+1)$, and x_{kt} of the odd slot $[2t+1, 2t+2)$. We split y_{ijt} accordingly, such that for each of the slots $[2t, 2t+1)$ and $[2t+1, 2t+2)$ constraints (1b) and (1c) are satisfied. Notice that in this way we have blown up the scheduling cost by a factor of 2, while the reservation cost remains the same. Furthermore, notice that every job is processed at least half either in odd slots or in even slots. Thus by doubling again each slot and reserving in each of the two the same fraction, we can enforce a job to be either entirely scheduled in slots that are reserved in the first stage, or in slots reserved in the second stage. ◀

Proof (Thm. 9). Let $(x'_t, x'_{it}, y'_{ijt})$ be a feasible LP solution that satisfies the separation property and has objective value $Z' \le 2LP^r + 4LP^s$. We show that the existence of a ρ-approximation algorithm for the problem with first-stage reservation only implies the existence of an algorithm that produces a feasible schedule for the two-stage problem with total cost at most $2\rho Z'$.

Since jobs are divided into J_I and J_{II} in the solution $(x'_t, x'_{it}, y'_{ijt})$, we denote by Z'_I and Z'_{II} their contributions in Z' respectively: $Z' = Z'_I + Z'_{II}$. Consider scheduling J_I with only first-stage reservation and let Z_I be the optimal value of the corresponding LP. Similarly, let Z_{II} be the optimal value of the LP for scheduling J_{II} with only second-stage reservation. Clearly, $Z_I \le Z'_I$ and $Z_{II} \le Z'_{II}$. The ρ-approximation algorithm for the problem with only first-stage reservation for J_I returns a feasible schedule of cost at most ρZ_I. The problem of reserving and scheduling jobs only in the second stage can be separated into N single scenario problems, each of which is like a first-stage reservation problem, we thus also get a feasible schedule of cost at most ρZ_{II} for J_{II}.

Now we need to merge the two schedules for J_I and J_{II}. Notice that the two schedules may overlap in the sense that some slot is reserved in both schedules. To handle this we further double the two schedules. For the schedule of J_I, we double the time units and put whatever is scheduled in $[t, t+1)$ into the even slot $[2t, 2t+1)$, while for the schedule of J_{II}, we put whatever is scheduled in $[t, t+1)$ into the odd slot $[2t+1, 2t+2)$. Now the total cost of the merged solution is bounded by $2\rho(Z_I + Z_{II}) \leq 2\rho Z'$. ◀

Directly applying Theorem 9 gives us a $8 \cdot (3+\epsilon) = 24 + \epsilon'$-approximation.

2.3 A Refined Two-Stage Algorithm

▶ **Theorem 11.** *There is an $(8+\epsilon)$-approximation algorithm for the two-stage stochastic variant of $R \,|\, r_j, pmtn \,|\, \sum w_j C_j$ in the polynomial-scenario model.*

Proof (Idea). Given an optimal LP solution, we first apply slow-motion to get a β-expanded solution. Then we apply time slot and job-set separation (Lemma 10) and obtain jobs and slots to be covered by either first-stage or second-stage reservation only. Then, we apply the technique of accumulative and extra reservation and reassign the workload *separately* on the slots reserved in the first and second stage. Here the last procedure must be carried out with caution so that after we separately round first and second stage reservation, they do not overlap. A careful analysis gives the result. ◀

We conclude this section by remarking that our techniques lead to the following result for the makespan objective.

▶ **Theorem 12.** *There is a $(64/9+\epsilon)$-approximation algorithm for the two-stage stochastic variant of $R \,|\, r_j, pmtn \,|\, C_{\max}$ in the polynomial-scenario model.*

3 The Black-Box Model

We now show that at the expense of another ϵ our results for the two-stage stochastic min-sum and makespan problem hold for any arbitrary scenario distribution given by means of a black-box. Besides that, the problem is as before.

Given a first-stage reservation $\bar{x} \in \{0,1\}^T$, a lower bound on the second-stage cost for a scenario S_k is as follows:

$$q(\bar{x}, S_k) = \min \left\{ \sum_{j \in J_k} w_j C_j^{LP} + \lambda_k c \sum_{t=0}^{T-1} x_{kt} \;\middle|\; \text{(1b) - (1g)} \wedge x_t = \bar{x}_t \;\forall t \right\}.$$

Let $c(x)$ denote the cost of a (possibly fractional) reservation $x \in [0,1]^T$. Then the following gives a lower bound on our two-stage stochastic scheduling problem.

$$\min_{x \in [0,1]^T} f(x) = c(x) + E_{S \in \mathcal{S}}\left[q(x, S)\right]. \tag{2}$$

For an unknown distribution in the black-box model we cannot solve this problem efficiently. However, using the SAA method [13] we can approximate it. We draw a number N of independent samples $S_1, \ldots, S_N$ from the black-box and solve the following sample average problem:

$$\min_{x \in [0,1]^T} \hat{f}(x) = c(x) + \frac{1}{N} \cdot \sum_{k=1}^{N} q(x, S_k). \tag{3}$$

Notice that (3) is exactly the LP of Section 2 with all N scenarios having probability $1/N$, and can thus be solved efficiently. It remains to determine the number of samples N that is needed to guarantee a certain approximation. We can show that our LP in (2) can be cast as a stochastic LP of type required in [26] for obtaining such a result. To that end, we must be given $\lambda := \max_{k \in S} \lambda_k$.

▶ **Lemma 13** ([26]). *There is a polynomially bounded number N such that any optimal solution x^{LP} to the sample average problem* (3) *with N samples satisfies $f(x^{LP}) \leq (1+\epsilon)\min_x f(x)$ with high probability.*

Based on this lemma we can obtain a good integral first-stage solution. We draw N samples and solve problem (3). Let $(x_t^{LP}, x_{kt}^{LP}, y_{ijt}^{LP})$ be an optimal (fractional) solution. Applying our rounding technique (Sec. 2) we derive a solution $(\bar{x}_t, \bar{x}_{kt}, \bar{y}_{ijt})$ with $(\bar{x}_t, \bar{x}_{kt}) \in \{0,1\}$. We fix $\bar{x}_t$ as first-stage reservation.

The difficult part is to find a second-stage solution for some scenario (that is not necessarily in the sample set) and bound it by the LP solution for the sample set. The key is that our rounding procedure for the first stage reservation x only depends on x itself and is independent of the scheduling solution. Given $\bar{x}_t$ and a scenario S, we solve the resulting second-stage problem as follows: we solve the problem $\min_{x \in [0,1]^T} c(x^{LP}) + q(x^{LP}, S)$, which is again exactly the LP of Section 2 with a single scenario S, after fixing first-stage reservation at $x_t = x_t^{LP}$. Let (x'_{kt}, y'_{ijt}) be the optimal solution. Plugging in x_t^{LP} and applying our rounding procedure on $(x_t^{LP}, x'_{kt}, y'_{ijt})$, we get a feasible schedule of total cost at most $(\rho+\epsilon)(c(x^{LP}) + q(x^{LP}, S))$, with $\rho = 8$ for the min-sum objective and $\rho = 64/9$ for the makespan. And, most importantly, the first stage reservation $\bar{x}_t$ is consistent with our first-stage reservation. Using Lemma 13 we have in expectation total cost of at most $(\rho+\epsilon)f(x^{LP}) \leq (\rho + O(\epsilon))\min_x f(x) \leq (\rho+\epsilon')Z^*$.

▶ **Theorem 14.** *In the black-box model, there is a $(\rho+\epsilon)$-approximation algorithm for two-stage stochastic variant of $R\,|\,r_j, pmtn\,|\sum w_j C_j$ ($\rho = 8$) and $R\,|\,r_j, pmtn\,|\,C_{\max}$ ($\rho = 64/9$), respectively.*

4 Two-Stage Robust Scheduling

In the robust setting, we restrict to the model with an explicit description of the scenario set $\mathcal{S}$. The objective is now to minimize the worst-case total cost instead of the expected total cost. Notice that the LP relaxations, that our algorithms in Sec. 2 rely on, can be easily adopted.

Our approximation algorithms for the stochastic model are risk-averse, i.e., the performance guarantee holds for every scenario. Therefore, the techniques used for the stochastic model also apply to the discrete-scenario robust model. For the min-$\sum w_j C_j$ problem, certain randomized steps of our algorithm must be replaced by deterministic ones losing a factor 2 in the approximation guarantee. Such an adaptation is not needed for the robust makespan problem and we directly obtain again a $(7.11+\epsilon)$-approximation algorithm. However, the makespan problem is much easier and we provide a simple 2-approximation algorithm.

▶ **Theorem 15.** *For two-stage discrete-scenario robust scheduling with reservation cost, there is a ρ-approximation algorithm for the scheduling problems $R\,|\,r_j, pmtn\,|\sum w_j C_j$ ($\rho = 16+\epsilon$) and $R\,|\,r_j, pmtn\,|\,C_{\max}$ ($\rho = 2$), respectively.*

5 Conclusion

Inspired by the resource provisioning problem of cloud users, we propose an optimization model that reflects two-stage decision processes in which computing resources must be reserved under uncertainty about the set of computational tasks. It leads to a new class of scheduling problems. We present first results that suggest higher approximation complexity than their single stage, single scenario versions. The quest for better approximations is left for future research.

We also leave open the approximability of the equivalent non-preemptive scheduling problems with release dates. Notice that the second-stage problem would not admit a constant approximation (large inflation factor, 2-partition), unless P=NP, when considering it independently of the first stage problem. However, a two-stage algorithm may yield a constant-factor approximation.

Another interesting variation of the problem arises if a user may reserve machines individually, possibly at machine-dependent rates. We note that, even if reservation costs are uniform over the machines, our proposed LP relaxation has a non-constant integrality gap in this case.

References

1. Amazon EC2 Pricing Options: `https://aws.amazon.com/ec2/pricing/`.
2. Talal Al-Khamis and Rym M'Hallah. A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity. *Computers & OR*, 38(12):1747–1759, 2011.
3. Gerard M. Campbell. A two-stage stochastic program for scheduling and allocating cross-trained workers. *JORS*, 62(6):1038–1047, 2011.
4. Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE Trans. Serv. Comput.*, 5(2):164–177, 2012.
5. Moses Charikar, Chandra Chekuri, and Martin Pál. Sampling bounds for stochastic optimization. In *Proc. of APPROX and RANDOM 2005*, pages 257–269, 2005.
6. Lin Chen, Nicole Megow, Roman Rischke, Leen Stougie, and José Verschae. Optimal algorithms and a PTAS for cost-aware scheduling. *To appear in Proc. of MFCS 2015*, 2015.
7. Kedar Dhamdhere, Vineet Goyal, R. Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *Proc. of FOCS 2005*, pages 367–376, 2005.
8. Shane Dye, Leen Stougie, and Asgeir Tomasgard. The stochastic single resource service-provision problem. *Naval Res. Logist.*, 50(8):869–887, 2003.
9. Uriel Feige, Kamal Jain, Mohammad Mahdian, and Vahab S. Mirrokni. Robust combinatorial optimization with exponential scenarios. In *Proc. of IPCO 2007*, pages 439–453, 2007.
10. Michel X. Goemans. Improved approximation algorthims for scheduling with release dates. In *Proc. of SODA 1997*, pages 591–598, 1997.
11. Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. *SIAM J. Comput.*, 40(5):1361–1401, 2011.
12. Rohit Khandekar, Guy Kortsarz, Vahab S. Mirrokni, and Mohammad R. Salavatipour. Two-stage robust network design with exponential scenarios. *Algorithmica*, 65(2):391–408, 2013.

13 Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.*, 12(2):479–502, 2001.

14 Janardhan Kulkarni and Kamesh Munagala. Algorithms for cost aware scheduling. In *Proc. of WAOA 2012*, pages 201–214, 2013.

15 Eugene L. Lawler and Jacques Labetoulle. On preemptive scheduling of unrelated parallel processors by linear programming. *J. ACM*, 25(4):612–619, 1978.

16 Stefano Leonardi, Nicole Megow, Roman Rischke, Leen Stougie, Chaitanya Swamy, and José Verschae. Scheduling with time-varying cost: Deterministic and stochastic models. Presentation at the 11th Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2013), 2013.

17 Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Springer, 2008.

18 Maurice Queyranne and Maxim Sviridenko. A $(2+\epsilon)$-approximation algorithm for the generalized preemptive open shop problem with minsum objective. *J. Algorithms*, 45(2):202–212, 2002.

19 Andreas S. Schulz and Martin Skutella. Random-based scheduling: New approximations and LP lower bounds. In *Proc. of APPROX and RANDOM 1997*, pages 119–133, 1997.

20 Andreas S. Schulz and Martin Skutella. Scheduling unrelated machines by randomized rounding. *SIAM J. Discrete Math.*, 15(4):450–469, 2002.

21 David B. Shmoys and Mauro Sozio. Approximation algorithms for 2-stage stochastic scheduling problems. In *Proc. of IPCO 2007*, pages 145–157. Springer, 2007.

22 David B. Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *J. ACM*, 53(6):978–1012, 2006.

23 René A. Sitters. Approximability of average completion time scheduling on unrelated machines. In *Proc. of ESA 2008*, pages 768–779, 2008.

24 Martin Skutella. List scheduling in order of α-points on a single machine. In Evripidis Bampis, Klaus Jansen, and Claire Kenyon, editors, *Efficient Approximation and Online Algorithms*, volume 3484 of *LNCS*, pages 250–291. Springer, 2006.

25 Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, 2006.

26 Chaitanya Swamy and David B. Shmoys. Sampling-based approximation algorithms for multistage stochastic optimization. *SIAM J. Comput.*, 41(4):975–1004, 2012.

27 G. Wan and X. Qi. Scheduling with variable time slot costs. *Naval Research Logistics*, 57:159–171, 2010.