Distributionally Robust Scheduling Problems with Release Time

Xun Zhang
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, xunzhang@sjtu.edu.cn

Du Chen*
Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, chendu2017@sjtu.edu.cn

We investigate a single machine scheduling problem under random release time and processing time with unknown distributions. The objective is to minimize the total completion time or makespan. This problem is formulated as a two-stage distributionally robust model based on first two moments. To solve this problem, we employ the linear decision rule and develop an approximated mixed-integer second order cone programming. We conduct computational studies by testing out-sample scheduling instances.

Key words: Single machine scheduling, Release time, Distributionally robust optimization

---

1. Introduction

Scheduling critical resourses to jobs (or activities) is one of the most important problems in the field of production, service, manufacturing and transportation systems. To obtain the best performance in the scheduling problem, the decision-maker has to decide the optimal sequence of the jobs to be processed. In the past several decades, numerous literature study scheduling problems. For a comprehensive and in-depth survey of this field, we refer to the books by Leung (2004) and Pinedo (2008).

In contrast to the deterministic scheduling problem, studies on scheduling problems with uncertain parameters are on the rise in recent years (e.g. see Daniels and Kouvelis (1995), Kasperski (2005), Drwal and Rischke (2016), Chang et al. (2017)). However, most of them only assume that the processing time is stochastic and ignore the stochastic release time. To some extent, solving the scheduling problem with release time is a more challenging task. To the best of our knowledge, only a few papers present their solutions to scheduling problems with uncertain release time. Bachtler et al. (2020) investigate a single machine makespan scheduling problem with uncertain release date, which take values within known intervals. Yue et al. (2018) study a similar problem, but the objective is maximum waiting time (MWT). Liu et al. (2020) apply sample average approximate

* Corresponding author.

method to solve a parallel machine scheduling problem with uncertain release time and processing time. While these papers either do not account for the distribution of release time or assume a konwn distribution, which is hard to obtain in reality.

In this paper, we adopt the distributionally robust optimization (DRO) to solve our problem. DRO assumes that the distribution of uncertain parameters reside in an ambiguity set, which is described by some statistics, such as mean and variance. In addition, these statistics can be easily estimated from data. Via specifying an ambiguity set, DRO optimizes a objective based on the worst-case distribution within an ambiguity set. DRO has been developed in recent years and became one of the most powerful approaches for addressing optimization problem with uncertainty (Ghaoui et al. (2003), Shapiro and Ahmed (2004), Delage and Ye (2010) and Wiesemann et al. (2014)). In practice, many important applications are the multi-stage decision problems. Ben-Tal et al. (2004) were the first to discuss robust multi-stage decision problems. Unfortunately, this kinds of problems are in general computationally intractable. To circumvent the intractability, the linear decision rule has been proposed to provide tractable and efficient solutions (Kuhn et al. (2011), Bertsimas et al. (2019) and Chen et al. (2020)).

This paper deal with the scheduling problem with uncertain job parameters, i.e., uncertain processing time and/or uncertain release time. The objective of the probem is the makespan or total completion time, which are the two of the most important measures to optimize in scheduling theory. Minimising the total completion time tends to minimise the average waiting time of the customers, whereas minimising the makespan tends to minimise the maximum waiting time of the customers. Given the first two moments of release time and processing time, we model this problem as a two-stage distributionally robust optimization problem. Since this problem is an intractable problem, the linear decision rule has been used to provide an approximate solution.

The rest of this paper is organized as follows. After the introduction, we formulate our scheduling problem as a distributionally robust model with a specified ambiguity set. In section 3, we provide a tractable reformulation with linear decision rule to solve the total completion time problem. In section 4, we show that the makespan problem can also be solved in the same way. The conculsion are summarized in section 5.

Notations. Throughout this paper, we use lowercase ***bold*** and UPPERCASE ***bold*** characters to denote vectors and matrices, respectively. Corresponding normal characters denote component-wise elements. Variables with tilde symbols, such as $\tilde{\boldsymbol{r}}$, represent random variables. We use $\boldsymbol{v} \succeq_Q \mathbf{0}$ to indicate that vector $\boldsymbol{v}$ is a second order cone. $\mathbb{R}^n_+$ denotes an n-dimensional nonnegative real number, and $[n]$ represents the set $\{1, 2, 3, \ldots, n\}$.

2. Problem Formulation

In this section, we begin with the deteriministic model, and then provide our two-stage distributionally robust model based on the first two moments.

2.1. Deterministic Single Machine Scheduling Problem

Consider a single machine scheduling problem that a set $\mathscr{J} = \{1,2,\ldots,n\}$ of jobs are processed on one machine. The processing time of job j is denoted as $p_j, j \in \{1,\ldots,n\}$, which is ready to process at its release time r_j. For the simplicity, we denote $\boldsymbol{p} = \{p_1,\ldots,p_n\}$ and $\boldsymbol{r} = \{r_1,\ldots,r_n\}$. In this paper, we mainly concerned with non-preemptive scheduling discipline, which means that a machine can process at most one job at a time. Therefore, the set of the feasible sequence $\mathscr{X}$ can be expressed as follows,

$$\mathscr{X} = \left\{ \boldsymbol{X} \;\middle|\; \begin{array}{l} \sum_{i=1}^n x_{ij} = 1, \forall j \in [n] \\ \sum_{j=1}^n x_{ij} = 1, \forall i \in [n] \\ x_{ij} \in \{0,1\}, \forall i,j \in [n] \end{array} \right\}$$

In this expression, $x_{ij} = 1$ stands for job j is processed at the i-th position of the sequence, and $x_{ij} = 0$ otherwise. The first constraint in above set forces that one job can only be processed at one position, the second constraint requires that one position should be occupied by only one job, the third constraint means that the job cannot be divided.

Given a feasible sequence $\boldsymbol{X}$, we denote the start time of the job at the i-th position by t_i, then the completion time of the job at the i-th position is $c_i = t_i + \sum_{j=1}^n x_{ij} p_j$. Therefore, the makespan (schedule length) can be expressed as $C_{max}(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r}) = t_n + \sum_{j=1}^n x_{nj} p_j$, and the total completion time can be expressed as $T(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r}) = \sum_{i=1}^n \left[t_i + \sum_{j=1}^n x_{ij} p_j \right]$. In this paper, our goal is to find a sequence such that the makespan or total completion time minimum. In general, we denote $f(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r})$ as our objective function, which can be the makespan $C_{max}(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r})$ or total completion time $T(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r})$. Then, our problem can be formulated as a 0-1 integer programming model:

$$\min_{\boldsymbol{X} \in \mathscr{X}} Q(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r})$$

where

$$Q(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r}) = \min_{\boldsymbol{t}} \; f(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r}) \tag{1}$$

$$\text{s.t } t_i \geq \sum_{j=1}^n x_{ij} r_j, \forall i \in [n] \tag{2}$$

$$t_i \geq t_{i-1} + \sum_{j=1}^n x_{i-1,j} p_j, \forall i \in \{2,\ldots,n\} \tag{3}$$

In this formulation, the decision variable is $\boldsymbol{X}$ and $\boldsymbol{t}$ and the objective (1) is the makespan or total completion time. The constraint (2) means that the start time at the i-th the position should greater or equal to the release time of the job at the i-th position , the constraint (3) means that the start time at the i-th position should greater or equal to the completion time of the job at the previous position.

2.2. Distributionally Robust Single Machine Scheduling Problem

In practice, the processing time and release time are usually stochastic, the exact value and distribution are merely known a prior. The only available information we can obtain is some statistical information. In facing this stochastic setting, model $Q(\boldsymbol{X},\boldsymbol{p},\boldsymbol{r})$ is incapable to provide a efficient schedule. To differentiate from the deterministic case, the random processing time and release time are denoted by $\tilde{\boldsymbol{p}}=\{\tilde{p}_1,\dots,\tilde{p}_n\}\in\mathbb{R}_+^n$ and $\tilde{\boldsymbol{r}}=\{\tilde{r}_1,\dots,\tilde{r}_n\}\in\mathbb{R}_+^n$. We assume that random variable $(\tilde{\boldsymbol{p}},\tilde{\boldsymbol{r}})$ follows an distribution $\mathbb{P}$, which is unknown in advance but the the first moment and the second moment are available. Let $\tilde{\boldsymbol{z}}=[\tilde{\boldsymbol{p}},\tilde{\boldsymbol{r}}]^T$, then we denote mean vector and standard deviation vector by $\boldsymbol{\mu}=[\boldsymbol{\mu}^p,\boldsymbol{\mu}^r]^T$ and $\boldsymbol{\sigma}=[\boldsymbol{\sigma}^p,\boldsymbol{\sigma}^r]^T$, respectively. Therefore, an ambiguity set that includes all feasible distributions can be expressed as follows,

$$\mathscr{F}_{\mathrm{MM}}=\left\{\mathbb{P}\in\mathscr{P}_0\left(\mathbb{R}^{2n}\right)\left|\begin{array}{l}\tilde{\mathbf{z}}\sim\mathbb{P}\\ \mathbb{E}_{\mathbb{P}}[\tilde{\mathbf{z}}]=\boldsymbol{\mu}\\ \mathbb{E}_{\mathbb{P}}\left[(\tilde{z}_i-\mu_i)^2\right]\le\sigma_i^2,\forall i\in[2n]\\ \mathbb{P}\left[\tilde{\mathbf{z}}\in\mathbb{R}_+^{2n}\right]=1\end{array}\right.\right\}$$

Now, we are able to provide our distributionally robust model. Given the first stage decision variable $\boldsymbol{x}$, when the random variable $\tilde{\boldsymbol{z}}$ is realized, the second stage problem is to determine the start time such the objective minimum, which can formulated as follows:

$$Q(\boldsymbol{X},\tilde{\boldsymbol{z}})=\min_{\boldsymbol{t}}\ f(\boldsymbol{X},\tilde{\boldsymbol{z}}) \tag{4}$$

$$\text{s.t } t_i\ge\sum_{j=1}^{n}x_{ij}\tilde{z}_{n+j},\forall i\in[n] \tag{5}$$

$$t_i\ge t_{i-1}+\sum_{j=1}^{n}x_{i-1,j}\tilde{z}_j,\forall i\in\{2,\dots,n\} \tag{6}$$

In the first stage, we aim to find a sequence, which immune all distribution in ambiguity set $\mathscr{F}_{\mathrm{MM}}$, to minimize the objective. Hence, we formulate our scheduling problem as the following min-max model,

$$\min_{\boldsymbol{X}\in\mathscr{X}}\sup_{\mathbb{P}\in\mathscr{F}_{\mathrm{MM}}}\mathbb{E}_{\mathbb{P}}\left[Q(\boldsymbol{X},\tilde{\boldsymbol{z}})\right] \tag{7}$$

In this model, we try to find a schedule that minimize the maximal expected makespan or total completion time over the distribution $\mathbb{P}\in\mathscr{F}_{\mathrm{MM}}$.

3. The Total Completion Time Problem

In this section, we mainly focus on solving the total completion time problem. Since the distributionally robust model (7) is hard to solve directly, our methods heavily based on the linear decison rule. The linear decision rule is one of most powerful tools to solve the two stage distributionally robust problem. We will illustrate how the linear decision rule be used in this problem.

3.1. Linear Decision Rule

In this section, we first formulate a new ambiguity set by introducing an auxiliary random variable $\tilde{\boldsymbol{u}}$. Then, we employ a linear decision rule based on the new ambiguity set to approximately tractable mixed-integer second order cone programming.

Consider the original ambiguity set, we introduce an epigraphical random varibale $\tilde{\boldsymbol{u}}$ for the term $\left(\tilde{z}_i-\mu_i\right)^2, i \in \{1,\ldots,2n\}$, and then we present a new ambiguity set $\mathscr{G}$,

$$\mathscr{G}=\left\{\mathbb{P}\in\mathscr{P}_0\left(\mathbb{R}^{2n}\times\mathbb{R}^{2n}\right) \left| \begin{array}{l} (\tilde{\mathbf{z}},\tilde{\mathbf{u}})\sim\mathbb{P} \\ \mathbb{E}_{\mathbb{P}}[\tilde{\mathbf{z}}]=\boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{u}_i\right]\leq\sigma_i^2,\forall i\in[2n] \\ \mathbb{P}\left[(\tilde{\mathbf{z}},\tilde{\mathbf{u}})\in\mathscr{W}_{\mathrm{MM}}\right]=1 \end{array}\right.\right\}$$

and

$$\mathscr{W}_{\mathrm{MM}}=\left\{(\mathbf{z},\mathbf{u})\in\mathbb{R}^{2n}\times\mathbb{R}^{2n} \left| \begin{array}{l} \mathbf{z}\geq\mathbf{0} \\ \mathrm{Case}_{[2n]} \end{array}\right.\right\}$$

with $\mathrm{Case}_{[2n]}: \quad \sqrt{\left(z_j-\mu_j\right)^2+\left(\frac{u_j-1}{2}\right)^2}\leq\frac{u_j+1}{2}, \quad \forall j\in[2n]$

Proposition 1. *The ambiguity set, $\mathscr{F}_{\mathrm{MM}}$, is equivalent to the set of marginal distribution of $\tilde{\boldsymbol{z}}$ under $\mathbb{P}$, for all $\mathbb{P}\in\mathscr{G}$.*

Proof Proposition 1 is a direct result of Proposition 1 in Bertsimas et al. (2019). □

As a consequence of the Proposition 1, we equally reformulate model (7) as below,

$$\min_{\boldsymbol{X}\in\mathscr{X}}\sup_{\mathbb{P}\in\mathscr{G}}\mathbb{E}_{\mathbb{P}}\left[Q(\mathbf{X},(\tilde{\mathbf{z}},\tilde{\mathbf{u}}))\right] \tag{8}$$

However, the model (8) is still hard to solve. In fact, the second stage decision variable $\boldsymbol{t}$ can be seen as a mapping of the random variable $(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{u}})$, hence, model (8) can be reformulated as follows:

$$\begin{aligned}
\min_{\boldsymbol{X}\in\mathscr{X}}\sup_{\mathbb{P}\in\mathscr{G}}\mathbb{E}_{\mathbb{P}} & \left(\sum_{i=1}^{n}\left[t_i(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{u}})+\sum_{j=1}^{n}x_{ij}\tilde{z}_j\right]\right) \\
\text{s.t. } & t_i(\boldsymbol{z},\boldsymbol{u})\geq\sum_{j=1}^{n}x_{ij}z_{n+j},\forall(\boldsymbol{z},\boldsymbol{u})\in\mathscr{W}_{\mathrm{MM}},\forall i\in[n] \\
& t_i(\boldsymbol{z},\boldsymbol{u})\geq t_{i-1}(\boldsymbol{z},\boldsymbol{u})+\sum_{j=1}^{n}x_{i-1,j}z_j,\forall(\boldsymbol{z},\boldsymbol{u})\in\mathscr{W}_{\mathrm{MM}},\forall i\in\{2,\ldots,n\} \\
& \boldsymbol{t}\in\mathscr{R}^{4n,n}
\end{aligned}$$

where $\boldsymbol{t}(\boldsymbol{z},\boldsymbol{u}):\mathbb{R}^{4n}\to\mathbb{R}^n$ is a decision measure function. Note that we do not require $\boldsymbol{t}(\boldsymbol{z},\boldsymbol{u})$ to satisfies any explicity form. In general, above optimization model is an intractable model.

Now, we are ready to present the linear decision rule, which means that the decision measure function has to be a linear function of random variables. By adding this constraint, we could obtain an upper bound of original optimization model. Specifically, the decision variable $\boldsymbol{t}$ is defined as follows,

$$\boldsymbol{t} \in \mathscr{L}^{4n,n} = \left\{ \boldsymbol{t} \in \mathscr{R}^{4n,n} \,\middle|\, \begin{array}{l} \exists \boldsymbol{t}^0, \boldsymbol{t}^{1i}, \boldsymbol{t}^{2i} \in \mathbb{R}^n, \forall i \in [2n] \\ \boldsymbol{t}(\boldsymbol{z}, \boldsymbol{u}) = \boldsymbol{t}^0 + \sum_{i=1}^{2n} (\boldsymbol{t}^{1i} z_i + \boldsymbol{t}^{2i} u_i) \end{array} \right\}$$

Incoporate linear decision rule into model (8), we have the following programming,

$$\begin{aligned} [TFT] \ \min_{x \in \mathscr{X}} \sup_{\mathbb{P} \in \mathscr{G}} & \mathbb{E}_{\mathbb{P}} \left(\sum_{i=1}^n \left[t_i(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) + \sum_{j=1}^n x_{ij} \tilde{z}_j \right] \right) \\ \text{s.t. } & t_i(\boldsymbol{z}, \boldsymbol{u}) \geq \sum_{j=1}^n x_{ij} z_{n+j}, \forall (\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}, \forall i \in [n] \\ & t_i(\boldsymbol{z}, \boldsymbol{u}) \geq t_{i-1}(\boldsymbol{z}, \boldsymbol{u}) + \sum_{j=1}^n x_{i-1,j} z_j, \forall (\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}, \forall i \in \{2, \dots, n\} \\ & \boldsymbol{t} \in \mathscr{L}^{4n \times n} \end{aligned} \tag{9}$$

Next, we aim to provide a tractable reformulation of the problem (9).

3.2. Tractable Reformulation

In this section, we reformulate problem (9) into a mixed integer second order cone programming, which can be solved quickly by the state-of-art commerical solver, such as Gurobi or Mosek. We start our reformulation with considering the inner problem of model (9). Specifically, we first try to reformulate the problem $\sup_{\mathbb{P} \in \mathscr{G}} \mathbb{E}_{\mathbb{P}} \left(\sum_{i=1}^n \left[t_i(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) + \sum_{j=1}^n x_{ij} \tilde{z}_j \right] \right)$, which can be rewritten as follows optimization problem,

$$\begin{aligned} \sup & \int_{(\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}} \sum_{i=1}^n \left(t_i((\boldsymbol{z}, \boldsymbol{u})) + \sum_{j=1}^n x_{ij} z_j \right) dF((\boldsymbol{z}, \boldsymbol{u})) \\ \text{s.t.} & \int_{(\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}} \boldsymbol{z} dF((\boldsymbol{z}, \boldsymbol{u})) = \boldsymbol{\mu} \\ & \int_{(\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}} u_i dF((\boldsymbol{z}, \boldsymbol{u})) \leq \sigma_i^2, \forall i \in [2n] \\ & \int_{(\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}} dF((\boldsymbol{z}, \boldsymbol{u})) = 1 \\ & dF((\boldsymbol{z}, \boldsymbol{u})) \geq 0 \end{aligned} \tag{10}$$

where $F((\boldsymbol{z}, \boldsymbol{u}))$ is the cumulative function. To solve this problem, we provide the following proposition.

PROPOSITION 2. *The problem* (10) *is equivalent to the following model,*

$$\inf \ \boldsymbol{s}^T \boldsymbol{\mu} + \sum_{i=1}^{2n} o_i \sigma_i^2 + v \tag{11}$$

$$s.t.\ v-\sum_{i=1}^{n}t_i^0 \geq \sum_{j=1}^{2n}\left(\frac{1}{2}(\alpha_j-\beta_j)-\mu_j\tau_j\right) \tag{12}$$

$$\tau_j \leq s_j-\sum_{i=1}^{n}t_i^{1j}-\sum_{i=1}^{n}x_{ij}, j\in[n] \tag{13}$$

$$\tau_j \leq s_j-\sum_{i=1}^{n}t_i^{1j}, j\in\{n+1,\ldots,2n\} \tag{14}$$

$$\frac{1}{2}(\alpha_j+\beta_j)=o_j-\sum_{i=1}^{n}t_i^{2j}, j\in[2n] \tag{15}$$

$$\begin{bmatrix}\alpha_j\\ \beta_j\\ \tau_j\end{bmatrix}\succeq_Q 0, \forall j\in[2n] \tag{16}$$

$$\mathbf{o}\geq\mathbf{0}$$

where $\boldsymbol{s},\boldsymbol{o},v,\boldsymbol{t},\alpha_j,\beta_j$ and $\tau_j, j\in[2n]$ are decision variables.

Proof The dual problem of problem (10) can be expressed as follows,

$$\inf\ \boldsymbol{s}^T\boldsymbol{\mu}+\sum_{i=1}^{2n}o_i\sigma_i^2+v \tag{17}$$

$$\text{s.t. } \boldsymbol{s}^T\mathbf{z}+\sum_{i=1}^{2n}o_iu_i+v\geq\sum_{i=1}^{n}\left(t_i(\boldsymbol{z},\boldsymbol{u})+\sum_{j=1}^{n}x_{ij}z_j\right), \forall(\mathbf{z},\mathbf{u})\in\mathscr{W}_{\mathrm{MM}} \tag{18}$$

$$\mathbf{o}\geq\mathbf{0}$$

$$\boldsymbol{t}\in\mathscr{L}^{n\times 4n}$$

where $\boldsymbol{s}, o_i$ and v are the dual variable corresponding to the first constraint, the second constraint and the third constraint, respectively. Note that above problem is still an intractable optimization problem as there is an infinite number of inequality constraints in (18). We now rewrite constraint (18) by considering the linear formulation of the decision variable $\boldsymbol{t}$

$$\begin{aligned}
&\boldsymbol{s}^T\mathbf{z}+\sum_{i=1}^{2n}o_iu_i+v\geq\sum_{i=1}^{n}\left(t_i(\boldsymbol{z},\boldsymbol{u})+\sum_{j=1}^{n}x_{ij}z_j\right), \forall(\mathbf{z},\mathbf{u})\in\mathscr{W}_{\mathrm{MM}}\\
\Leftrightarrow&\boldsymbol{s}^T\mathbf{z}+\sum_{i=1}^{2n}o_iu_i+v\geq\sum_{i=1}^{n}\left(t_i^0+\sum_{j=1}^{2n}(t_i^{1j}z_j+t_i^{2j}u_j)+\sum_{j=1}^{n}x_{ij}z_j\right), \forall(\mathbf{z},\mathbf{u})\in\mathscr{W}_{\mathrm{MM}}\\
\Leftrightarrow&v-\sum_{i=1}^{n}t_i^0\geq\sum_{i=1}^{n}\left(\sum_{j=1}^{2n}(t_i^{1j}z_j+t_i^{2j}u_j)+\sum_{j=1}^{n}x_{ij}z_j\right)-\boldsymbol{s}^T\mathbf{z}-\sum_{i=1}^{2n}o_iu_i, \forall(\mathbf{z},\mathbf{u})\in\mathscr{W}_{\mathrm{MM}}\\
\Leftrightarrow&v-\sum_{i=1}^{n}t_i^0\geq\max_{(\mathbf{z},\mathbf{u})\in\mathscr{W}_{\mathrm{MM}}}\left\{\sum_{i=1}^{n}\left(\sum_{j=1}^{2n}(t_i^{1j}z_j+t_i^{2j}u_j)+\sum_{j=1}^{n}x_{ij}z_j\right)-\boldsymbol{s}^T\mathbf{z}-\sum_{i=1}^{2n}o_iu_i\right\}
\end{aligned} \tag{19}$$

Then, we write the maximum problem in the expression (19),

$$\max_{(\mathbf{z},\mathbf{u})}\ \left\{\sum_{i=1}^{n}\left(\sum_{j=1}^{2n}(t_i^{1j}z_j+t_i^{2j}u_j)+\sum_{j=1}^{n}x_{ij}z_j\right)-\boldsymbol{s}^T\mathbf{z}-\sum_{i=1}^{2n}o_iu_i\right\} \tag{20}$$

$$\text{s.t } \boldsymbol{z} \geq 0$$
$$\sqrt{(z_j-\mu_j)^2+\left(\frac{u_j-1}{2}\right)^2} \leq \frac{u_j+1}{2}, \quad \forall j \in [2n]$$

Take dual of this problem, we could obtain the following problem,

$$\begin{aligned} \inf \ & \sum_{j=1}^{2n}\left(\frac{1}{2}(\alpha_j-\beta_j)-\mu_j\tau_j\right) \\ \text{s.t. } & (13)-(16) \end{aligned} \tag{21}$$

where α_j, β_j and $\tau_j, \forall j \in \{1,\ldots,2n\}$, are the decision variables. Replace (20) with (21) in (19) and incoporate these results in (17), the desire result are obtained immediately. □

As a result of the Proposition 2, the model (9) can be expressed as follows,

$$\inf \ \boldsymbol{s}^T\boldsymbol{\mu}+\sum_{i=1}^{2n} o_i\sigma_i^2+v \tag{22}$$

$$\text{s.t. } t_i(\boldsymbol{z},\boldsymbol{u}) \geq \sum_{j=1}^{n} x_{ij}z_{n+j}, \forall(\boldsymbol{z},\boldsymbol{u}) \in \mathscr{W}_{\mathrm{MM}}, \forall i \in [n] \tag{23}$$

$$t_i(\boldsymbol{z},\boldsymbol{u}) \geq t_{i-1}(\boldsymbol{z},\boldsymbol{u})+\sum_{j=1}^{n} x_{i-1,j}z_j, \forall(\boldsymbol{z},\boldsymbol{u}) \in \mathscr{W}_{\mathrm{MM}}, \forall i \in \{2,\ldots,n\} \tag{24}$$

$$(12)-(16)$$
$$\mathbf{o} \geq \mathbf{0}, \mathbf{X} \in \mathscr{X}$$
$$\boldsymbol{t} \in \mathscr{L}^{4n,n}$$

Note that we still have infinity constraints in (23) and (24). We now prepare to provide their robust counterparts. We first rewrite constraint (23) as follows,

$$\begin{aligned} & t_i(\boldsymbol{z},\boldsymbol{u}) \geq \sum_{j=1}^{n} x_{ij}z_{n+j}, \forall(\boldsymbol{z},\boldsymbol{u}) \in \mathscr{W}_{\mathrm{MM}}, \forall i \in [n] \\ \Leftrightarrow & t_i^0 \geq \max_{(\boldsymbol{z},\boldsymbol{u})\in\mathscr{W}_{\mathrm{MM}}}\left\{\sum_{j=1}^{n} x_{ij}z_{n+j}-\sum_{j=1}^{2n}(t_i^{1j}z_i+t_i^{2j}u_j)\right\}, \forall i \in [n] \end{aligned}$$

For each $i, i \in [n]$, consider the problem

$$\max_{(\boldsymbol{z},\boldsymbol{u})} \left\{\sum_{j=1}^{n} x_{ij}z_{n+j}-\sum_{j=1}^{2n}(t_i^{1j}z_i+t_i^{2j}u_j)\right\}$$
$$\text{s.t } \boldsymbol{z} \geq 0$$
$$\sqrt{(z_j-\mu_j)^2+\left(\frac{u_j-1}{2}\right)^2} \leq \frac{u_j+1}{2}, \quad \forall j \in [2n]$$

Take dual, we obtain the following optimization problem,

$$\min \ \sum_{j=1}^{2n}\left(\frac{1}{2}(a_j^i - b_j^i) - \mu_j c_j^i\right) \tag{25}$$

$$c_j^i \le t_i^{1j}, \forall j \in [n] \tag{26}$$

$$c_j^i \le t_i^{1j} - x_{ij}, \forall j \in \{n+1,\dots,2n\} \tag{27}$$

$$\frac{1}{2}(a_j^i + b_j^i) = t_i^{2j}, j \in [2n] \tag{28}$$

$$\begin{bmatrix} a_j^i \\ b_j^i \\ c_j^i \end{bmatrix} \succeq_Q 0, \forall j \in [2n] \tag{29}$$

Next, we rewrite the (24) as follows,

$$t_i(\boldsymbol{z},\boldsymbol{u}) \ge t_{i-1}(\boldsymbol{z},\boldsymbol{u}) + \sum_{j=1}^{n} x_{i-1,j} z_j, \forall (\boldsymbol{z},\boldsymbol{u}) \in \mathscr{W}_{\mathrm{MM}}, \forall i \in \{2,\dots,n\}$$

$$\Leftrightarrow t_i^0 - t_{i-1}^0 \ge \max_{(\boldsymbol{z},\boldsymbol{u})\in\mathscr{W}_{\mathrm{MM}}} \left\{\sum_{j=1}^{2n}\left[(t_{i-1}^{1j} - t_i^{1j}) z_i + (t_{i-1}^{2j} - t_i^{2j}) u_j)\right] + \sum_{j=1}^{n} x_{i-1,j} z_j\right\}, \forall i \in \{2,\dots,n\}$$

For each $i, \forall i \in \{2,\dots,n\}$, Consider the problem

$$\begin{aligned} \max_{(\boldsymbol{z},\boldsymbol{u})} \ & \left\{\sum_{j=1}^{2n}\left[(t_{i-1}^{1j} - t_i^{1j}) z_i + (t_{i-1}^{2j} - t_i^{2j}) u_j)\right] + \sum_{j=1}^{n} x_{i-1,j} z_j\right\} \\ & \boldsymbol{z} \ge 0 \\ & \sqrt{(z_j - \mu_j)^2 + \left(\frac{u_j - 1}{2}\right)^2} \le \frac{u_j + 1}{2}, \quad \forall j \in [2n] \end{aligned}$$

Take dual, we obtain

$$\min \ \sum_{j=1}^{2n}\left(\frac{1}{2}(d_j^{i-1} - e_j^{i-1}) - \mu_j f_j^{i-1}\right) \tag{30}$$

$$f_j^{i-1} \le t_i^{1j} - t_{i-1}^{1j} - x_{i-1,j}, j \in [n] \tag{31}$$

$$f_j^{i-1} \le t_i^{1j} - t_{i-1}^{1j}, j \in \{n+1,\dots,2n\} \tag{32}$$

$$\frac{1}{2}(d_j^{i-1} + e_j^{i-1}) = t_i^{2j} - t_{i-1}^{2j}, \forall j \in [2n] \tag{33}$$

$$\begin{bmatrix} d_j^{i-1} \\ e_j^{i-1} \\ f_j^{i-1} \end{bmatrix} \succeq_Q 0, \forall j \in [2n] \tag{34}$$

Incoporate these results into problem (22), we obtain one of the main results in this paper. We formally demonstrate it in Theorem 1.

THEOREM 1. *$[TFT]$ is equivalent to the model $[TFT - SOCP]$*

$$
\begin{aligned}
[TFT-SOCP]\ \inf\ & \boldsymbol{s}^T\boldsymbol{\mu}+\sum_{i=1}^{2n} o_i\sigma_i^2+v \\
\text{s.t. } & t_i^0 \geq \sum_{j=1}^{2n}\left(\frac{1}{2}(a_j^i-b_j^i)-\mu_j c_j^i\right), \forall i\in[n] \\
& t_i^0-t_{i-1}^0 \geq \sum_{j=1}^{2n}\left(\frac{1}{2}(d_j^{i-1}-e_j^{i-1})-\mu_j f_j^{i-1}\right), \forall i\in\{2,\dots,n\} \\
& \mathbf{o}\geq \mathbf{0}, \mathbf{X}\in\mathscr{X} \\
& (12)-(16) \\
& (26)-(29), \forall i\in[n] \\
& \text{ and } (31)-(34), \forall i\in\{2,\dots,n\}
\end{aligned}
$$

where the $\boldsymbol{X},\boldsymbol{s},\boldsymbol{o},v,\boldsymbol{t}^0,\boldsymbol{t}^1,\boldsymbol{t}^2,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\tau}$, $\boldsymbol{a}^i,\boldsymbol{b}^i,\boldsymbol{c}^i,\forall i\in\{1,\dots,n\}$, and $\boldsymbol{d}^{i-1},\boldsymbol{e}^{i-1},\boldsymbol{f}^{i-1},\forall i\in\{2,\dots,n\}$ are the decision variables.

3.3. Numerical experiments

In this section, we conduct numerical experiments to validate the effectness of the distributionally robust scheduling models. Our experiments are conducted on a ThinkPad X1 PC, with an Intel (R) Core (TM) i7-10710 U CPU running at 1.10 GigaHertz and a 16.00 Gigabyte of memory. The overall framework was implemented with the programming language Python 3.7, and the optimization models are solved by Gurobi.

3.3.1. Experiment Description For our experiment, we assume that there is $n=8$ jobs to be proceeded. We draw the true mean processing time μ_p from $U[10,50]$. The true mean release time μ_r are generated from the $[0,30n]$. We define $cv=\frac{\sigma}{\mu}$ by the coefficient of variation, which is chosen from the set $\{0.3,0.7,1.0\}$. Therefore, the standard deviation of stochastic processing time and release time are $\boldsymbol{\sigma}_p=cv\boldsymbol{\mu}_p$, $\boldsymbol{\sigma}_r=cv\boldsymbol{\mu}_r$, respectively. Note that, for the simplicity, we do not set different cv for release time and processing time, and the cv is same among all jobs. To ensure fairness, we generate K historical data from a given distribution with true mean and variance. We consider three types of commonly used distribution: truncated normal distribution, log-normal distribution and gamma distribution. Specifically, we generate K records of release time and processing time, i.e., $\{(\hat{\mathbf{p}}^1,\hat{\mathbf{r}}^1),\dots,(\hat{\mathbf{p}}^K,\hat{\mathbf{r}}^K)\}$. Then, we are able to calculate sample mean $\hat{\boldsymbol{\mu}}_r,\hat{\boldsymbol{\mu}}_p$ and sample standard deviation $\hat{\boldsymbol{\sigma}}_r,\hat{\boldsymbol{\sigma}}_p$ for release time and processing time, which are used to solve the distributionally robust model.

We compare the performance of our distributionally robust model and the deterministic model, as well as sample average approximation (SAA). In deterministic model, we solve the model (4) by replacing the random variable with their sample mean value, and obtain the solution Seq^d. SAA has been widely used in stochastic optimization problems. In our problem, SAA can be formulated as a determinsitic model, which is expressed as follows,

$$
\begin{aligned}
[SAA]\ \min_{\boldsymbol{t}^k, \boldsymbol{X}}\ & \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{n}\left[t_i^k + \sum_{j=1}^{n} x_{ij}\hat{p}_j^k\right] \\
\text{s.t } & t_i^k \geq \sum_{j=1}^{n} x_{ij}\hat{r}_j^k, \forall i \in [n], \forall k \in [K] \\
& t_i^k \geq t_{i-1}^k + \sum_{j=1}^{n} x_{i-1,j}\hat{p}_j^k, \forall i \in \{2,\ldots,n\}, \forall k \in [K] \\
& \boldsymbol{X} \in \mathscr{X}
\end{aligned}
$$

We denote Seq^s by the solution of the model $[SAA]$. Then, we solve the DRO, i.e., $[TFT-SOCP]$, based on the sample mean and sample variance and obtain the solution Seq^{dro}.

To evaluate performance of Seq^d, Seq^s, and Seq^{dro}, we compute their corresponding total completion time as follows. We randomly draw 10,000 samples from a given distribution. For each sample, we compute their total completion time corresponding to Seq^d, Seq^s, and Seq^{dro}, respectively. Then, we calculate the following four measures over the 10,000 samples:

- means, denoted by M_d, M_s and M_{dro}
- q-th percentiles, denoted by PT_d^q, PT_s^q and PT_{dro}^q for $q = 75, 95$ and $q = 99$, respectively

Then, we use the values of four measure for determinsitc as our benchmark and compute the gap between SAA and deterministic model and the gap between distributionally robust model and deterministic model. Specifically, they can be compute by following expressions:

$$
\begin{aligned}
Gap_{saa}^m &= \frac{M_d - M_s}{M_d}, Gap_{dro}^m = \frac{M_d - M_{dro}}{M_d} \\
Gap_{saa}^q &= \frac{PT_d^q - PT_s^q}{PT_d^q}, Gap_{dro}^q = \frac{PT_d^q - PT_{dro}^q}{PT_d^q}, q = 75, 95, 99
\end{aligned}
$$

For each types of distribution, we generate 20 instances, and the average results over 20 instances in each cases are reported in Table 1.

Note that the performance of deterministic model is the baseline, we do not provide their result. As shown in Table 1, the deterministic model is dominated by SAA and DRO, because the deterministic model do not account for the variance in the model. Then, we mainly compare performance of SAA and DRO.

The Table 1 shows that the DRO has better performance in terms of q-th percentiles, for $q = 75, 95, 99$. Furthermore, it is worth noting that the benefit of DRO increase as t increase. To some extent, the benefit of DRO increase as cv grows. This finding also proves the nature of DRO, that is DRO tends to provide a more robust solutions.

When we consider the mean value, neither SAA nor DRO shows that it outperforms another. In fact, the performance between these two methods are very close.

Table 1 Total Completion Time Results under Stochastic Release and Processing Time

Dist	cv	SAA				DRO			
		Gap^m_{saa}	Gap^{75}_{saa}	Gap^{95}_{saa}	Gap^{99}_{saa}	Gap^m_{dro}	Gap^{75}_{dro}	Gap^{95}_{dro}	Gap^{99}_{dro}
Log-Normal	0.3	0.76%	1.02%	2.17%	3.38%	0.81%	1.22%	3.15%	5.11%
	0.7	1.65%	2.23%	4.41%	6.59%	1.58%	2.55%	5.19%	7.86%
	1	2.47%	2.82%	6.74%	10.01%	2.14%	3.35%	8.26%	11.73%
Normal	0.3	0.84%	1.22%	1.73%	2.07%	0.83%	1.35%	1.83%	2.22%
	0.7	1.64%	2.39%	4.56%	6.08%	1.64%	2.85%	5.72%	7.71%
	1	2.19%	3.15%	5.01%	6.14%	1.84%	3.37%	5.99%	7.51%
Gamma	0.3	0.92%	1.51%	3.08%	4.05%	0.89%	1.81%	3.78%	4.87%
	0.7	1.06%	1.62%	3.54%	4.62%	1.14%	2.23%	4.76%	6.22%
	1	2.29%	3.49%	7.67%	10.13%	2.31%	4.15%	9.48%	12.43%

4. The Makespan Problem

In this section, we show that linear decision rule can be used to solve makespan problem. In fact, the only difference between total completion time and makespan problem is the objective function.

4.1. Tractable Reformulation

In this section, we provide a tractable reformulation for the makespan problem where both processing time and release time are stochastic. Under the ambiguity set $\mathscr{G}$, the distributionally robust makespan problem can be approximately expressed,

$$
\begin{aligned}
[Cmax]\ \min_{\boldsymbol{X}\in\mathscr{X}} \sup_{\mathbb{P}\in\mathscr{G}} \mathbb{E}_{\mathbb{P}} & \left(t_n(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) + \sum_{j=1}^{n} x_{nj}\tilde{z}_j \right) \\
\text{s.t. } & t_i(\boldsymbol{z}, \boldsymbol{u}) \geq \sum_{j=1}^{n} x_{ij} z_{n+j}, \forall (\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}, \forall i \in [n] \\
& t_i(\boldsymbol{z}, \boldsymbol{u}) \geq t_{i-1}(\boldsymbol{z}, \boldsymbol{u}) + \sum_{j=1}^{n} x_{i-1,j} z_j, \forall (\boldsymbol{z}, \boldsymbol{u}) \in \mathscr{W}_{MM}, \forall i \in \{2, \dots, n\} \\
& \boldsymbol{t} \in \mathscr{L}^{4n\times n}
\end{aligned} \tag{35}
$$

Using the same technical that we have used in the total completion time, we give the equivalent model,

$$
\begin{aligned}
[Cmax - SOCP]\ \inf\ & \boldsymbol{s}^T\boldsymbol{\mu} + \sum_{i=1}^{2n} o_i\sigma_i^2 + v \\
\text{s.t. } & v - t_n^0 \geq \sum_{j=1}^{2n} \left(\frac{1}{2}(\alpha_j - \beta_j) - \mu_j \tau_j \right) \\
& \tau_j \leq s_j - t_n^{1j} - x_{nj}, j \in [n] \\
& \tau_j \leq s_j - t_n^{1j}, j \in \{n+1, \dots, 2n\}
\end{aligned}
$$

$$
\begin{aligned}
&\frac{1}{2}(\alpha_j + \beta_j) = o_j - t_i^{2j}, j \in [2n] \\
&\begin{bmatrix} \alpha_j \\ \beta_j \\ \tau_j \end{bmatrix} \succeq_Q 0, \forall j \in [2n] \\
&t_i^0 \geq \sum_{j=1}^{2n} \left(\frac{1}{2}(a_j^i - b_j^i) - \mu_j c_j^i \right), \forall i \in [n] \\
&t_i^0 - t_{i-1}^0 \geq \sum_{j=1}^{2n} \left(\frac{1}{2}(d_j^{i-1} - e_j^{i-1}) - \mu_j f_j^{i-1} \right), \forall i \in \{2, \ldots, n\} \\
&\mathbf{o} \geq \mathbf{0}, \mathbf{X} \in \mathscr{X} \\
&(26) - (29), \forall i \in [n] \\
&(31) - (34), \forall i \in \{2, \ldots, n\}
\end{aligned}
$$

THEOREM 2. *$[Cmax]$ is equivalent to the model $[Cmax - SOCP]$.*

Proof We omit the proof since it is similar with the proof of Theorem 1. □

Note that linear decision rule can be applied in the setting where either processing time or release time is stochastic. We do not provide the specifical demonstration here.

5. Conclusion

In this paper, we model the single machine scheduling problem with random release time and processing time as a two stage distributionally robust model. Since this problem cannot be solved directly, we provide an approximated mixed-integer SOCP reformulation based on the linear decision rule. For future research, one interesting extension is to consider parallel machine scheduling problem with unknown parameters. Another possible direction is to consider different criterion, such as maximum lateness.

References

Bachtler O, Krumke SO, Le HM (2020) Robust single machine makespan scheduling with release date uncertainty. *Operations Research Letters* .

Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A (2004) Adjustable robust solutions of uncertain linear programs. *Mathematical programming* 99(2):351–376.

Bertsimas D, Sim M, Zhang M (2019) Adaptive distributionally robust optimization. *Management Science* 65(2):604–618.

Chang Z, Song S, Zhang Y, Ding JY, Zhang R, Chiong R (2017) Distributionally robust single machine scheduling with risk aversion. *European Journal of Operational Research* 256(1):261–274.

Chen Z, Sim M, Xiong P (2020) Robust stochastic optimization made easy with rsome. *Management Science* .

Daniels RL, Kouvelis P (1995) Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science* 41(2):363–376.

Delage E, Ye Y (2010) Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research* 58(3):595–612.

Drwal M, Rischke R (2016) Complexity of interval minmax regret scheduling on parallel identical machines with total completion time criterion. *Operations Research Letters* 44(3):354–358.

Ghaoui LE, Oks M, Oustry F (2003) Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations research* 51(4):543–556.

Kasperski A (2005) Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion. *Operations Research Letters* 33(4):431–436.

Kuhn D, Wiesemann W, Georghiou A (2011) Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming* 130(1):177–209.

Leung JY (2004) *Handbook of scheduling: algorithms, models, and performance analysis* (CRC press).

Liu X, Chu F, Zheng F, Chu C, Liu M (2020) Parallel machine scheduling with stochastic release times and processing times. *International Journal of Production Research* 1–20.

Pinedo M (2008) *Scheduling: Theory, Algorithms, And Systems*.

Shapiro A, Ahmed S (2004) On a class of minimax stochastic programs. *SIAM Journal on Optimization* 14(4):1237–1249.

Wiesemann W, Kuhn D, Sim M (2014) Distributionally robust convex optimization. *Operations Research* 62(6):1358–1376.

Yue F, Song S, Zhang Y, Gupta JN, Chiong R (2018) Robust single machine scheduling with uncertain release times for minimising the maximum waiting time. *International Journal of Production Research* 56(16):5576–5592.