European Journal of Operational Research 256 (2017) 261–274

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Distributionally robust single machine scheduling with risk aversion

Zhiqi Chang[a], Shiji Song[a,*], Yuli Zhang[b], Jian-Ya Ding[a], Rui Zhang[c], Raymond Chiong[d]

[a] Department of Automation, Tsinghua University, Beijing 100084, PR China
[b] Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China
[c] School of Economics and Management, Xiamen University of Technology, Xiamen 361024, PR China
[d] School of Design, Communication and Information Technology, The University of Newcastle, Callaghan, NSW 2308, Australia

ARTICLE INFO

Article history:
Received 25 November 2015
Accepted 11 June 2016
Available online 16 June 2016

Keywords:
Robustness and sensitivity analysis
Distributionally robust optimization
Single machine scheduling
CVaR
Total flow time

ABSTRACT

This paper presents a distributionally robust (DR) optimization model for the single machine scheduling problem (SMSP) with random job processing time (JPT). To the best of our knowledge, it is the first time a DR optimization approach is applied to production scheduling problems in the literature. Unlike traditional stochastic programming models, which require an exact distribution, the presented DR-SMSP model needs only the mean-covariance information of JPT. Its aim is to find an optimal job sequence by minimizing the worst-case Conditional Value-at-Risk (Robust CVaR) of the job sequence's total flow time. We give an explicit expression of Robust CVaR, and decompose the DR-SMSP into an assignment problem and an integer second-order cone programming (I-SOCP) problem. To efficiently solve the I-SOCP problem with uncorrelated JPT, we propose three novel Cauchy-relaxation algorithms. The effectiveness and efficiency of these algorithms are evaluated by comparing them to a CPLEX solver, and robustness of the optimal job sequence is verified via comprehensive simulation experiments. In addition, the impact of confidence levels of CVaR on the tradeoff between optimality and robustness is investigated from both theoretical and practical perspectives. Our results convincingly show that the DR-SMSP model is able to enhance the robustness of the optimal job sequence and achieve risk reduction with a small sacrifice on the optimality of the mean value. Through the simulation experiments, we have also been able to identify the strength of each of the proposed algorithms.

1. Introduction

Scheduling is a decision-making process that deals with the allocation of one or more resources to activities over given time periods. It plays an important role in most production and manufacturing systems as well as in most information processing units (Pinedo, 2012). As a classical type of production scheduling models, deterministic versions of the single machine scheduling problem (SMSP) have been extensively studied over the past decades (Köksalan & Kondakci, 1997; Koulamas, 2010; Quan & Xu, 2013). However, deterministic models may not characterize real manufacturing systems operating in highly uncertain environments well. As pointed out by McKay, Safayeni, and Buzacott (1988), ignoring uncertainty in this context can lead to original optimal solutions becoming infeasible. To address this, various stochastic programming (SP) and robust optimization (RO) models have been developed to deal with uncertainty issues in scheduling problems.

SP models regard parameters related to uncertainty (e.g., processing time, due dates, and setup time) as random variables with known distributions, and aim at optimizing performance measures such as expectation or variance under uncertain circumstances. To this end, many evaluating indicators have been studied for the SMSP, such as flow time (Agrawala, Coffman, Garey, & Tripathi, 1984), maximum lateness (Wu & Zhou, 2008), weighted tardy time (Jia, 2001), the number of tardy jobs (Jang, 2002; Jang & Klein, 2002; Seo, Klein, & Jang, 2005) and weighted number of early and tardy jobs (Soroush, 2007), among others. However, such SP models are usually NP-hard (Soroush, 2007; Trietsch & Baker, 2008; van den Akker & Hoogeveen, 2008), which can only be tackled by heuristic approaches or time consuming dynamic programming methods. Moreover, knowledge about the exact probabilistic distribution of uncertain parameters required in SP techniques is usually hard to obtain.

RO models (Ben-Tal, El Ghaoui, & Nemirovski, 2009) aim to optimize scheduling performance in the worst case scenario. The uncertain parameters in RO are regarded as varying variables in given

* Corresponding author. Tel.: +8613810202800.
E-mail addresses: changzq13@mails.tsinghua.edu.cn (Z. Chang), shijis@mail.tsinghua.edu.cn, gaohuang.thu@gmail.com (S. Song), zhangyuli@mail.tsinghua.edu.cn (Y. Zhang), ding-jy12@mails.tsinghua.edu.cn (J.-Y. Ding), r.zhang@ymail.com (R. Zhang), Raymond.Chiong@newcastle.edu.au (R. Chiong).

http://dx.doi.org/10.1016/j.ejor.2016.06.025
0377-2217/

uncertainty sets instead of random variables with known distributions (see Gabrel, Murat, and Thiele (2014) for a comprehensive review on RO). For production scheduling problems, the uncertainty set is usually selected as a continuous interval or a finite set of different values. In this setting, the problem of interest is how the worst case scenario (robust cost) is defined and how the robust cost is minimized. Three criteria for robust cost calculation have been introduced by Kouvelis and Yu (2013): absolute robustness, robust deviation and relative robust deviation. Absolute robustness considers minimizing the objective value of the worst case directly. Robust deviation (or absolute regret) minimizes the largest possible difference between the observed objective value and the optimal one, while relative robust deviation (or relative regret) deals with the ratio of the largest possible observed value to the optimal value.

RO approaches were first introduced by Daniels and Kouvelis (1995) to tackle the SMSP. They formalized the robust scheduling concept for scheduling situations with uncertain or variable processing time periods. The performance criterion of interest was the total flow time (TFT), and absolute as well as relative robust deviations were chosen for measuring robustness. They proved that the discrete-scenario version of this problem is NP-hard, and showed some dominance relations between jobs, which were used in the solution schemes. Yang and Yu (2002) studied the same problem under all three robustness measures, and proved that it is NP-complete even for very restricted cases. Aloulou and Della Croce (2008) presented some algorithmic and computational complexity results for several SMSPs with different uncertain parameters and performance measures, with absolute robustness as the robustness criterion. de Farias, Zhao, and Zhao (2010) considered an SMSP with uncertain processing time, aiming to minimize the total weighted completion time under absolute robustness. They solved the problem at hand by implementing a cutting-plane algorithm.

Compared to the discrete-scenario case, the continuous case, where uncertainty sets are selected as continuous intervals, have received much more research attention. Kasperski (2005) minimized the maximal lateness of an SMSP with precedence constraints. Kasperski and Zieliński (2006) considered the SMSP with uncertain processing time to minimize the maximal regret of TFT under a job precedence constraint. Lebedev and Averbakh (2006) studied a similar model but without the job precedence constraint. Montemanni (2007) proposed a mixed integer linear programming formulation for the resulting optimization problem and explained how some known preprocessing rules can be translated into valid inequalities for the formulation. Tadayon and Smith (2015) studied robust SMSPs under four alternative optimization criteria, and three different uncertainty sets were designed to describe the uncertainty. Furthermore, Lu, Lin, and Ying (2012, 2014a), Lu, Ying, and Lin (2014b) studied the robust SMSP with different robustness measures and uncertain parameters. By analyzing the worst case scenario, they transformed the problems into a mixed integer linear program (Lu et al., 2014b), a robust constrained shortest path problem (Lu et al., 2012) and a robust traveling salesman problem (Lu, Lin, & Ying, 2014a), and solved them with a simple iterative improvement heuristic, a simulated annealing-based algorithm and a local search-based heuristic, respectively.

In this paper, we extend this line of research by developing a novel distributionally robust (DR) optimization model for the SMSP with uncertain job processing time (JPT). Typical RO models as discussed above utilize only the variation range of their uncertain parameters. In our DR-SMSP model, the JPT parameter is regarded as a random variable. We use not only the support information but also the mean and variance of it, and a specific form of distribution is not required. Due to the randomness of JPT, the TFT of job sequences is also a random variable. To evaluate the random objective values of different job sequences, some performance measures should be considered. One option is the popular criterion of expected value (Skutella & Uetz, 2005). However, the expectation criterion does not consider the deviation of random variables. Another option is the variance, which is regarded as a risk measure (De, Ghosh, & Wells, 1992). In this case, the risk averse attitude of a decision maker is taken into account. Nevertheless, in the minimization process of variance, positive and negative deviations have no distinction, which may mislead the decision maker. We therefore adopt the conditional value-at-risk (CVaR) criterion for our DR optimization model. The concept of CVaR was introduced by Rockafellar and Uryasev (2000), under which the minimization of expectation and variance tends to be simultaneous.

The original DR optimization model was introduced by Scarf, Arrow, and Karlin (1958) for an inventory problem. In this model, a distributional set characterized by mean and variance is defined to cover the true distribution. Depending on the context, different forms of distributional sets can be considered, such as a set containing unimodal distributions (Shapiro, 2006), a set based on the moments of the distribution (Doan, Kruk, & Wolkowicz, 2012; Dupačová, 1987; Popescu, 2007), or a set described by uncertain moments (Delage & Ye, 2010). More recent development of DR optimization can be found in the review by Gabrel et al. (2014). In addition to generic DR stochastic programming problems and inventory problems, DR optimization has been extended to many other fields, such as portfolio optimization problems (Doan, Li, & Natarajan, 2015; Nguyen & Lo, 2012), dynamic network design problems (Sun, Gao, Szeto, Long, & Zhao, 2014), workforce scheduling problems (Liao, Van Delft, & Vial, 2013) and appointment scheduling problems (Jiang, Shen, & Zhang, 2015). To the best of our knowledge, however, this work is the first to introduce a DR optimization approach in production scheduling problems.

To solve the DR-SMSP, we first propose an explicit expression of robust CVaR in a semi-infinite support case. Based on this expression, the problem is decomposed into two subproblems: an assignment problem (AP) and an integer second-order cone programming (I-SOCP) problem. The AP can be exactly optimized by an ordering strategy, and in the special case when the means and variances of processing times are consistent, the I-SOCP problem can also be solved by the same strategy. In the general case when the means and variances of processing times are not consistent, we propose to efficiently solve the I-SOCP problem with specially developed Cauchy-relaxation algorithms. The design of these algorithms was inspired by the popular iterative procedures from the field of machine learning (Huang, Zhang, Song, & Chen, 2015; Lanckriet, Ghaoui, Bhattacharyya, & Jordan, 2003), where there are always some relaxation variables introduced to obtain the upper or lower bound of the original objective function, and the bound is alternately optimized with fixed relaxation variables or fixed original decision variables in each iteration. In our Cauchy-relaxation algorithms, as implied by the name, the relaxations are achieved via Cauchy inequalities.

The rest of this paper is organized as follows. In Section 2, we first provide a brief introduction of production scheduling and describe the deterministic SMSP model as well as our DR-SMSP model. After that, we present the explicit expression of Robust CVaR, based on which the DR-SMSP is decomposed into the AP and I-SOCP problem. Section 3 outlines the solution methods for solving the AP and three novel Cauchy-relaxation algorithms for the I-SOCP problem. Experimental design and results are then discussed in Section 4, followed by the conclusion in Section 5.

2. Problem statement and formulation

2.1. Deterministic single machine scheduling

Production scheduling is the specific and concrete implementation of production planning decisions. When production plans for each work shift have been made according to upper-level plans, only jobs that need to be performed in a shift and their latest completion times are specified, while the allocation of these jobs to the machines and the order of processing the jobs on each machine are to be determined in the production scheduling process. Generally, such dispatching decisions should be made for each work shift, which means the time frame for production scheduling is mainly days or hours. However, when emergencies happen and result in changes in job information or machine status, a previously optimal schedule may become invalid and rescheduling should be performed within a short period of time.

The mathematical model of production scheduling contains m machines used to process n jobs. A schedule specifies the sequence in which the jobs are to be processed on each machine. It is a fundamental assumption that a machine can process only one job at a time,[1] and a job can be processed by only one machine at the same time. In addition to these basic constraints, a feasible schedule should also satisfy some other requirements related to specific production environments. The aim of production scheduling is to find the optimal schedule, which optimizes one or more performance criteria, among all feasible ones. Although production scheduling problems originally arise from a manufacturing context, many other interpretations are possible: machines and jobs can stand for hospital equipment and patients, runways and landings at an airport, or processing units and programs in a computing environment. Any situation fitting into the framework described above can fall within the scope of scheduling (Kan, 2012; Pinedo, 2012).

Based on the type of facilities and the technological requirement of jobs, the machine environment of a production scheduling model can be categorized into several basic types, such as single machines, parallel machines, flow shop, job shop and open shop. The model considered in this paper is the fundamental SMSP, which involves only one machine. The SMSP usually possesses some particular properties, and analyzing it can provide inspiration for other complicated scheduling models. In practice, complicated scheduling problems can often be decomposed into a series of subproblems of the SMSP, and scheduling problems with one bottleneck machine can also be formulated as an SMSP. Therefore, studying the SMSP has great significance from both theoretical and practical points of view.

The deterministic version of SMSP contains a set $\mathbb{J} = \{1, 2, \ldots, n\}$ of n jobs and one machine. The processing time of job j on the machine is $p_j, j = 1, \ldots, n$, and all the jobs are released at the beginning of the scheduling period, i.e., $r_j = 0, j = 1, \ldots, n$. The aim of the problem is to find the optimal job sequence with the minimum TFT ($\Sigma_j F_j$), where the flow time F_j of job j means the time elapsed from its release to its completion. Since the TFT includes the waiting time of each job, it is widely regarded as a good measure of overall production efficiency. Employing the notations used by Lu et al. (2014b), the processing sequence of these n jobs can be represented by a matrix $\boldsymbol{X} = \{x_{ij}, i, j = 1, \ldots, n\}$, where $x_{ij} = 1$ indicates that job j is assigned to the ith position in the sequence, and $x_{ij} = 0$ otherwise. Because the release times are all zero, F_j can be obtained by the sum of p_j and processing times of all the jobs before job j in a given sequence. By the definition of $\boldsymbol{X}$, the TFT under $\boldsymbol{X}$ can be expressed as

$$\text{(TFT)} \quad f(\boldsymbol{X}, \boldsymbol{p}) = \sum_{j=1}^{n} \sum_{i=1}^{n} (n - i + 1) p_j x_{ij}. \tag{1}$$

Therefore, the deterministic SMSP can be formulated as a 0–1 integer programming model:

$$\text{(SMSP)} \quad \min_{\boldsymbol{X} \in \mathbb{X}} f(\boldsymbol{X}, \boldsymbol{p}), \tag{2}$$

where $\mathbb{X}$ is the feasible set of $\boldsymbol{X}$, i.e.,

$$\mathbb{X} = \left\{ \boldsymbol{X} \in \mathbb{R}^{n \times n} \,|\, \sum_{i=1}^{n} x_{ij} = 1, j = 1, \ldots, n; \ \sum_{j=1}^{n} x_{ij} = 1, \right. \\ \left. i = 1, \ldots, n; \ x_{ij} \in \{0, 1\}, i = 1, \ldots, n, \ j = 1, \ldots, n \right\}. \tag{3}$$

The constraints in $\mathbb{X}$ require that each job should take only one position and each position should be occupied by only one job in a feasible job sequence.

The above deterministic SMSP can be exactly solved by the classical shortest processing time (SPT) rule (Smith, 1956). In an uncertain case, however, the SPT rule may become invalid since sometimes the processing times of different jobs cannot be compared. Thus, robust scheduling approaches should be developed to deal with the uncertainty and the formulation of our proposed DR-SMSP model is discussed in the following subsections.

2.2. Distributionally robust single machine scheduling

In most of the research studies on robust machine scheduling, the uncertain JPT is represented via an uncertainty set (Lu et al., 2012; Lu et al., 2014b; Tadayon & Smith, 2015), and usually a budget parameter is considered to control the degree of solution conservatism. In the work of Lu et al. (2014b), for example, the uncertain JPT is represented using the following uncertainty set:

$$\mathbb{U} = \left\{ \boldsymbol{p} \in \mathbb{R}_+^n | p_j \in [\bar{p}_j, \bar{p}_j + \hat{p}_j], \ \forall j \in \mathbb{J}; \sum_{j \in \mathbb{J}} \frac{p_j - \bar{p}_j}{\hat{p}_j} \le \Gamma \right\}, \tag{4}$$

where the uncertain processing time p_j has a nominal value $\bar{p}_j (\bar{p}_j \ge 0)$ and a maximum variation $\hat{p}_j (\hat{p}_j \ge 0)$. In each scenario, the total variation in JPT is less than or equal to the budget parameter Γ (i.e.,$\sum_{j \in \mathbb{J}} \frac{p_j - \bar{p}_j}{\hat{p}_j} \le \Gamma$).

In this paper, we attempt to take into account more information of uncertain JPT. Processing time $\boldsymbol{p} \in \mathbb{R}_+^n$ is regarded as a random vector subject to an ambiguous distribution P^p, which is unknown but belongs to a set of probability distributions $\mathbb{P}^p$ described by mean and covariance. The distributional set $\mathbb{P}^p$ can be described as

$$\mathbb{P}^p = \left\{ P^p | \ \mathrm{Sup}(p_j) = [0, \infty), \forall j \in \mathbb{J}; \ \mathrm{E}(\boldsymbol{p}) = \boldsymbol{\mu} \ ; \ \mathrm{Cov}(\boldsymbol{p}) = \boldsymbol{\Sigma} \right\}, \tag{5}$$

where $\mathrm{Sup}(p_j)$ represents the support set of each JPT, and $\mathrm{E}(\boldsymbol{p})$ and $\mathrm{Cov}(\boldsymbol{p})$ indicate the mean vector and covariance matrix of JPT, respectively.

Besides the DR framework, we also consider the risk aversion attitude of a decision maker. To integrate the effect of expectation and variance, the risk measure criterion CVaR is used. The CVaR_α of random loss $\boldsymbol{Z} \in \mathbb{R}$ can be calculated as

$$\mathrm{CVaR}_\alpha(\boldsymbol{Z}) = \mathbb{E}[\boldsymbol{Z} | \boldsymbol{Z} \ge \inf\{z : \mathrm{Prob}(\boldsymbol{Z} > z) \le 1 - \alpha\}], \tag{6}$$

where $\alpha \in (0, 1)$ is the confidence level of CVaR. Under each level α, $\mathrm{CVaR}_\alpha(\boldsymbol{Z})$ represents the expectation of the $(1 - \alpha) \times 100$ percent largest outcomes of $\boldsymbol{Z}$, which are the worst $(1 - \alpha) \times 100$ percent of random loss.

[1] This assumption holds except in the case of batch-processing machines.

When the probability distribution P^z of $\boldsymbol{Z}$ is ambiguous but specified by a distributional set $\mathbb{P}^z$ described as

$$\mathbb{P}^z = \left\{ P^z | \ \mathrm{Sup}(\boldsymbol{Z}) = [0, \infty), \ \mathrm{E}(\boldsymbol{Z}) = \mu_z, \ \mathrm{Var}(\boldsymbol{Z}) = \sigma_z^2 \right\}, \tag{7}$$

the worst-case $\mathrm{CVaR}_\alpha(\boldsymbol{Z})$ in $\mathbb{P}^z$ is referred to as the robust $\mathrm{CVaR}_\alpha(\boldsymbol{Z})$, i.e., $\mathrm{RCVaR}_\alpha(\boldsymbol{Z})$, which is formally defined as

$$\mathrm{RCVaR}_\alpha \ (\boldsymbol{Z}) = \sup_{P^z \in \mathbb{P}^z} \mathrm{CVaR}_\alpha \ (\boldsymbol{Z}). \tag{8}$$

Based on the above, the DR-SMSP with CVaR aversion can be formulated as

$$(\text{DR-SMSP1}) \ \min_{\boldsymbol{X} \in \mathbb{X}} \sup_{P^p \in \mathbb{P}^p} \ \mathrm{CVaR}_\alpha \ (f(\boldsymbol{X}, \boldsymbol{p})), \tag{9}$$

where $\mathbb{X}$ is the feasible set of $\boldsymbol{X}$ defined in Eq. (3), and $\mathbb{P}^p$ is the distributional set of $\boldsymbol{p}$ defined in Eq. (5). By the definition of RCVaR_α in Eq. (8), DR-SMSP1 can be rewritten as

$$(\text{DR-SMSP2}) \ \boldsymbol{X}^* = \ \arg\min_{\boldsymbol{X} \in \mathbb{X}} \ \mathrm{RCVaR}_\alpha^p \ (f(\boldsymbol{X}, \boldsymbol{p})), \tag{10}$$

where the superscript p in RCVaR_α^p indicates that the distributional set for calculating RCVaR_α is $\mathbb{P}^p$.

2.3. The property of RCVaR

In this subsection, we propose an explicit expression for RCVaR_α under the distributional set with a semi-infinite support set.

When $\boldsymbol{Z}$ is a random variable subject to an ambiguous distribution P^z in the distributional set $\mathbb{P}^z$ as described in Eq. (7), $\mathrm{RCVaR}_\alpha(\boldsymbol{Z})$ can be calculated according to the expression shown in Theorem 1.

Theorem 1. *For any random variable $\boldsymbol{Z} \in \mathbb{R}_+$ subject to a distribution in the distributional set $\mathbb{P}^z$, its RCVaR_α value can be calculated as follows:*

$$\mathrm{RCVaR}_\alpha \ (\boldsymbol{Z}) = \begin{cases} \dfrac{\mu_z}{1-\alpha}, & \text{if} \ \ 0 \le \alpha \le \dfrac{\sigma_z^2}{\sigma_z^2 + \mu_z^2} \\ \mu_z + \sqrt{\dfrac{\alpha}{1-\alpha}} \cdot \sqrt{\sigma_z^2}, & \text{if} \ \ \dfrac{\sigma_z^2}{\sigma_z^2 + \mu_z^2} \le \alpha \le 1, \end{cases} \tag{11}$$

where $\mathbb{P}^z = \left\{ P^z | \ \mathrm{Sup} \ (\boldsymbol{Z}) = [0, \infty), \ \mathrm{E} \ (\boldsymbol{Z}) = \mu_z, \ \mathrm{Var} \ (\boldsymbol{Z}) = \sigma_z^2 \right\}$.

Proof. See the Appendix. □

Let $\boldsymbol{\pi} = \sum_{i=1}^n (n-i+1)\boldsymbol{x}_i^T$, where $\boldsymbol{x}_i$ denotes the ith row vector of $\boldsymbol{X}$. By this definition, $\boldsymbol{\pi}$ is a permutation of integers $(1, \cdots, n)$, whose order is contrary to the sequence of jobs. That is, given $\boldsymbol{\pi} = (\pi(1), \pi(2), \ldots, \pi(n))$, $\pi(i) = j$ means that job j is operated in the $(n-i+1)$th position. Furthermore, the feasible set of $\boldsymbol{\pi}$ is:

$$\Pi = \left\{ \boldsymbol{\pi} \ | \boldsymbol{\pi} = \sum_{i=1}^n (n-i+1)\boldsymbol{x}_i^T, \ \boldsymbol{X} \in \mathbb{X} \right\}, \tag{12}$$

and the corresponding TFT can be rewritten as

$$f(\boldsymbol{\pi}, \boldsymbol{p}) = f(\boldsymbol{X}, \boldsymbol{p}) = \boldsymbol{\pi}^T \boldsymbol{p}. \tag{13}$$

Due to the randomness of $\boldsymbol{p}$, $f(\boldsymbol{\pi}, \boldsymbol{p})$ is a random variable for each $\boldsymbol{\pi} \in \Pi$, which can be denoted as $\boldsymbol{F}_\pi \in \mathbb{R}_+$. Based on the mean vector and covariance matrix of $\boldsymbol{p}$, the mean and variance of $\boldsymbol{F}_\pi$ are given respectively by

$$\mu_f(\boldsymbol{\pi}) = \boldsymbol{\pi}^T \boldsymbol{\mu} \ ; \quad \sigma_f^2(\boldsymbol{\pi}) = \boldsymbol{\pi}^T \boldsymbol{\Sigma} \boldsymbol{\pi}. \tag{14}$$

Let the distributional set of $\boldsymbol{F}_\pi$ be

$$\mathbb{P}^f = \left\{ P^f | \ \mathrm{Sup}(\boldsymbol{F}_\pi) = [0, \infty), \ \mathrm{E}(\boldsymbol{F}_\pi) = \mu_f(\boldsymbol{\pi}), \ \mathrm{Var}(\boldsymbol{F}_\pi) = \sigma_f^2(\boldsymbol{\pi}) \right\}, \tag{15}$$

an upper bound of $\mathrm{RCVaR}_\alpha^p(f(\boldsymbol{\pi}, \boldsymbol{p}))$ can be obtained by optimizing over the distributional set $\mathbb{P}^f$, that is,

$$\mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi) = \sup_{P^f \in \mathbb{P}^f} \mathrm{CVaR}_\alpha(\boldsymbol{F}_\pi) \ \ge \sup_{P^p \in \mathbb{P}^p} \mathrm{CVaR}_\alpha(f(\boldsymbol{\pi}, \boldsymbol{p})). \tag{16}$$

This is because for any feasible random vector $\boldsymbol{p} \sim P^p$ on the right-hand side, the corresponding projected random variable $\boldsymbol{F}_\pi^p = \boldsymbol{\pi}^T \boldsymbol{p}$ has support set $[0, \infty)$, mean $\boldsymbol{\pi}^T \boldsymbol{\mu} = \mu_f(\boldsymbol{\pi})$ and variance $\boldsymbol{\pi}^T \boldsymbol{\Sigma} \boldsymbol{\pi} = \sigma_f^2(\boldsymbol{\pi})$; hence, $\boldsymbol{F}_\pi^p$ is feasible on the left-hand side.

DR-SMSP2 can now be converted into the following DR-SMSP3 by replacing $\mathrm{RCVaR}_\alpha^p(f(\boldsymbol{\pi}, \boldsymbol{p}))$ with its upper bound:

$$(\text{DR-SMSP3}) \ \ \boldsymbol{\pi}^* = \ \arg\min_{\boldsymbol{\pi} \in \Pi} \ \mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi). \tag{17}$$

In view of Theorem 1, $\mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi)$ can be calculated as

$$\mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi) = \begin{cases} \dfrac{\mu_f(\boldsymbol{\pi})}{1-\alpha}, & \text{if} \ \ 0 \le \alpha \le \dfrac{\sigma_f^2(\boldsymbol{\pi})}{\sigma_f^2(\boldsymbol{\pi}) + \mu_f^2(\boldsymbol{\pi})} \\ \mu_f(\boldsymbol{\pi}) + k\sqrt{\sigma_f^2(\boldsymbol{\pi})}, & \text{if} \ \ \dfrac{\sigma_f^2(\boldsymbol{\pi})}{\sigma_f^2(\boldsymbol{\pi}) + \mu_f^2(\boldsymbol{\pi})} \le \alpha \le 1, \end{cases} \tag{18}$$

where $k = \sqrt{\frac{\alpha}{1-\alpha}}$.

2.4. Decompose the DR-SMSP

According to the expression for $\mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi)$ in Eq. (18), DR-SMSP3 contains a complicated segment constraint, which makes the problem hard to solve. In this subsection, we propose a theorem to illustrate that the RCVaR optimization model can be decomposed into two independent and relatively simple cone programming problems, thereby avoiding the difficulty of dealing with the segment constraint.

Theorem 2. *For any $\boldsymbol{\pi} \in \Pi$ with its random loss function $\boldsymbol{F}_\pi$ subject to a distribution in*

$$\mathbb{P}^f = \{ P^f | \ \mathrm{Sup} \ (\boldsymbol{F}_\pi) = [0, \infty), \ \mathrm{E} \ (\boldsymbol{F}_\pi) = \mu_f(\boldsymbol{\pi}), \mathrm{Var} \ (\boldsymbol{F}_\pi) = \sigma_f^2(\boldsymbol{\pi}) \}, \tag{19}$$

the following result should hold:

$$\min_{\boldsymbol{\pi} \in \Pi} \ \mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi) = \min \ \{R_1^*, R_2^*\}, \tag{20}$$

where
$$R_1^* = \ \min_{\boldsymbol{\pi} \in \Pi} \frac{\mu_f(\boldsymbol{\pi})}{1-\alpha}, \tag{21}$$

$$R_2^* = \ \min_{\boldsymbol{\pi} \in \Pi} \mu_f(\boldsymbol{\pi}) + k\sqrt{\sigma_f^2(\boldsymbol{\pi})} \ . \tag{22}$$

Proof. Let $R^* = \min_{\boldsymbol{\pi} \in \Pi} \ \mathrm{RCVaR}_\alpha(\boldsymbol{F}_\pi)$ and $\bar{R} = \min\{R_1^*, R_2^*\}$, it is easy to find that $\bar{R} \le R^*$. Next, $\bar{R} \ge R^*$ will be shown to prove the equivalence between $\bar{R}$ and R^*.

For any $\boldsymbol{\pi} \in \Pi$, if $0 \le \alpha \le \frac{\sigma_f^2(\boldsymbol{\pi})}{\sigma_f^2(\boldsymbol{\pi}) + \mu_f^2(\boldsymbol{\pi})}$, i.e., $\sigma_f^2(\boldsymbol{\pi}) \ge k^2 \mu_f^2(\boldsymbol{\pi})$, then

$$\mu_f(\boldsymbol{\pi}) + k\sqrt{\sigma_f^2(\boldsymbol{\pi})} \ \ge \frac{\mu_f(\boldsymbol{\pi})}{1-\alpha}. \tag{23}$$

On the contrary, if $\frac{\sigma_f^2(\boldsymbol{\pi})}{\sigma_f^2(\boldsymbol{\pi}) + \mu_f^2(\boldsymbol{\pi})} \le \alpha \le 1$, i.e., $\sigma_f^2(\boldsymbol{\pi}) \le k^2 \mu_f^2(\boldsymbol{\pi})$, then

$$\mu_f(\boldsymbol{\pi}) + k\sqrt{\sigma_f^2(\boldsymbol{\pi})} \ \le \frac{\mu_f(\boldsymbol{\pi})}{1-\alpha}. \tag{24}$$

We first assume that $\bar{R} = R_1^*$, if the $\boldsymbol{\pi}_1^*$ corresponding to R_1^* satisfies the inequality $\alpha \le \frac{\sigma_f^2(\boldsymbol{\pi}_1^*)}{\sigma_f^2(\boldsymbol{\pi}_1^*) + \mu_f^2(\boldsymbol{\pi}_1^*)}$, then $\boldsymbol{\pi}_1^*$ is a feasible solution of $\min_{\boldsymbol{\pi} \in \Pi} \mathrm{RCVaR}_\alpha(\boldsymbol{F}_\pi)$. In this case, the objective value of $\boldsymbol{\pi}_1^*$

is greater than the optimal value, i.e., $\bar{R} = \mathrm{RCVaR}_\alpha\left(\boldsymbol{F}_\pi(\boldsymbol{\pi}_1^*)\right) \geq R^*$. On the other hand, if $\boldsymbol{\pi}_1^*$ makes $\alpha \geq \frac{\sigma_f^2(\boldsymbol{\pi}_1^*)}{\sigma_f^2(\boldsymbol{\pi}_1^*)+\mu_f^2(\boldsymbol{\pi}_1^*)}$, then according to Eq. (24) the following relationships should hold:

$$\bar{R} = \frac{\mu_f(\boldsymbol{\pi}_1^*)}{1-\alpha} \geq \mu_f(\boldsymbol{\pi}_1^*) + k\sqrt{\sigma_f^2(\boldsymbol{\pi}_1^*)} = \mathrm{RCVaR}_\alpha(\boldsymbol{F}_\pi(\boldsymbol{\pi}_1^*)) \geq R^*. \tag{25}$$

Hence, when $\bar{R} = R_1^*$, there holds the inequality $\bar{R} \geq R^*$. When $\bar{R} = R_2^*$, we can get this result by the same way. Putting the two situations together, we know that $\bar{R} \geq R^*$, which verifies the equivalence between $\bar{R}$ and R^* along with the truth that $\bar{R} \leq R^*$. □

Theorem 2 suggests that DR-SMSP3 can be optimized by solving an AP and an I-SOCP problem. Along with the mean and variance of $\boldsymbol{F}_\pi$ shown in Eq. (14), the decomposed model DR-SMSP4 is described as follows:

$$\text{(DR-SMSP4)} \min_{\boldsymbol{\pi}\in\Pi} \ \mathrm{RCVaR}_\alpha^f(\boldsymbol{F}_\pi) = \min\ \{R_1^*, R_2^*\}, \tag{26}$$

$$\textit{where} \quad R_1^* = \min_{\boldsymbol{\pi}\in\Pi} \frac{\boldsymbol{\pi}^T\boldsymbol{\mu}}{1-\alpha}, \quad (\text{AP } R_1) \tag{27}$$

$$R_2^* = \min_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}^T\boldsymbol{\mu} + k\sqrt{\boldsymbol{\pi}^T\boldsymbol{\Sigma}\boldsymbol{\pi}}. \quad (\text{I-SOCP } R_2) \tag{28}$$

3. Solution algorithms

In the general case, DR-SMSP4 can be solved by some commercial solvers, such as IBM®ILOG®CPLEX. However, when the size of problem instances is large, solving the I-SOCP R_2 is computationally challenging. By analyzing the characteristics of this problem, we propose more effective solution strategies and algorithms for the special case when the processing times of different jobs are uncorrelated. Under this circumstance, the mean and variance of $\boldsymbol{F}_\pi$ are rewritten respectively as

$$\mu_f(\boldsymbol{\pi}) = \boldsymbol{\pi}^T\boldsymbol{\mu} = \sum_{j=1}^n \pi_j\mu_j; \quad \sigma_f^2(\boldsymbol{\pi}) = \boldsymbol{\pi}^T\boldsymbol{\Sigma}\boldsymbol{\pi} = \sum_{j=1}^n \pi_j^2\sigma_j^2, \tag{29}$$

and R_1^* and R_2^* in DR-SMSP4 are converted into

$$(\text{AP } R_1) \ \ R_1^* = \min_{\boldsymbol{\pi}\in\Pi} \ (\sum_{j=1}^n \pi_j\mu_j)/(1-\alpha), \tag{30}$$

$$(\text{I-SOCP } R_2) \ \ R_2^* = \min_{\boldsymbol{\pi}\in\Pi} \sum_{j=1}^n \pi_j\mu_j + k\sqrt{\sum_{j=1}^n \pi_j^2\sigma_j^2}. \tag{31}$$

To solve DR-SMSP4, AP R_1 and I-SOCP R_2 need to be solved separately. Subproblem R_1 is similar to the deterministic version of SMSP. Considering $\mu_j/(1-\alpha)$ to be p_j in the SMSP, R_1 can be optimized by the SPT rule (Smith, 1956). That is, the uncertain version of subproblem R_1 is solvable by sequencing the jobs in non-decreasing order of their processing times' means. This is denoted as the shortest average processing time (SAPT) first rule in the rest of this paper.

As for subproblem R_2, the optimal sequence is affected by both means and variances of the processing times. In terms of mean, $\boldsymbol{\pi}$ should be arrayed by non-increasing order of the means. In terms of variance, the non-increasing order of variances shall be used. Only in the situation when the orders of means and variances are consistent, i.e., jobs with smaller means possess smaller variances, the two subproblems R_1 and R_2 share the same optimal solution, leading to the following theorem.

Theorem 3. *Any SAPT schedule is optimal for DR-SMSP4, when the means and variances of uncertain processing times are consistent.*

Theorem 3 describes the solution strategy of DR-SMSP4 in the special case when the means and variances are consistent, which can be satisfied by distributions such as Poisson and Exponential. However, this condition does not hold in most cases. When consistency is violated, the SAPT rule will no longer be valid for subproblem R_2. Therefore, more pervasive solution algorithms should be proposed to deal with R_2 in the general case.

3.1. Complete Cauchy-relaxation algorithms

Subproblem R_2 cannot be optimized by direct sorting in the general case due to the conflict of means and variances' orders. From the expression for the objective function, we know that this conflict comes from the different parameters of the two parts related to $\boldsymbol{\pi}$. Thus, if the two parts of $\boldsymbol{\pi}$ can be integrated into one, the optimal sequence can be obtained by non-increasing order of this integrated coefficient. Based on this idea, two Cauchy inequality relaxations are considered to achieve this integration.

We first introduce a nonnegative vector $\boldsymbol{\eta} \in \mathbb{R}_+^n$ to relax the mean part in R_2. For each component $j = 1, \cdots, n$, the following Cauchy inequality should hold:

$$\pi_j \times \mu_j = \left(\pi_j \times \frac{\mu_j}{\eta_j}\right) \times \eta_j \ \leq \ \frac{1}{2}\left(\pi_j^2 + (\frac{\mu_j}{\eta_j})^2\right) \times \eta_j, \quad j = 1, \cdots, n, \tag{32}$$

and the equality holds if and only if $\eta_j = \mu_j/\pi_j$.

As for the variance part in R_2, a nonnegative variable $t \in \mathbb{R}_+$ is introduced to remove the radical in the expression. The Cauchy inequality relaxation is given by

$$\sqrt{\sum_{j=1}^n \pi_j^2\sigma_j^2} \ \leq \ \frac{1}{2t}\sum_{j=1}^n \pi_j^2\sigma_j^2 + \frac{t}{2}, \tag{33}$$

where the equality holds when $t = \sqrt{\sum_{j=1}^n \pi_j^2\sigma_j^2}$.

Let $f_A(\boldsymbol{\pi})$ denote the original objective function of R_2, that is,

$$f_A(\boldsymbol{\pi}) = \sum_{j=1}^n \pi_j\mu_j + k\sqrt{\sum_{j=1}^n \pi_j^2\sigma_j^2}. \tag{34}$$

Taking advantage of the above two Cauchy inequality relaxations, $f_A(\boldsymbol{\pi})$ can be relaxed to $f_B(\boldsymbol{\pi}, \boldsymbol{\eta}, t)$ detailed as follows:

$$f_B(\boldsymbol{\pi}, \boldsymbol{\eta}, t) = \sum_{j=1}^n \pi_j^2\left[\frac{1}{2}\eta_j + \frac{k}{2t}\sigma_j^2\right] + \frac{1}{2}\sum_{j=1}^n \frac{\mu_j^2}{\eta_j} + \frac{kt}{2}. \tag{35}$$

Then, R_2 is relaxed to the following problem:

$$(R_2^B) \ \ \min_{\boldsymbol{\pi},\boldsymbol{\eta},t} f_B(\boldsymbol{\pi}, \boldsymbol{\eta}, t), \qquad s.t.\ \boldsymbol{\pi} \in \Pi, \boldsymbol{\eta} \in \mathbb{R}_+^n, t \in \mathbb{R}_+. \tag{36}$$

This relaxation achieves the integration of the two parts related to $\boldsymbol{\pi}$, making it possible for the sorting strategy to obtain the optimal solution when $\boldsymbol{\eta}$ and t are fixed. Next, we show the equivalence between R_2^B and R_2 from the perspective of optimization.

Property 1. *Let* $(\boldsymbol{\pi}^*, \boldsymbol{\eta}^*, t^*)$ *be an optimal solution for problem* $\min_{\boldsymbol{\pi},\boldsymbol{\eta},t} f_B(\boldsymbol{\pi}, \boldsymbol{\eta}, t)$, *then* $\boldsymbol{\pi}^*$ *is also an optimal solution for problem* $\min_{\boldsymbol{\pi}} f_A(\boldsymbol{\pi})$.

Proof. $f_B(\boldsymbol{\pi}, \boldsymbol{\eta}, t) \geq f_A(\boldsymbol{\pi})$ takes equality if and only if:

$$\eta_j = \frac{\mu_j}{\pi_j}, \ \ j = 1, \cdots, n; \quad t = \sqrt{\sum_{j=1}^n \pi_j^2\sigma_j^2}. \tag{37}$$

Therefore, $\min_{\eta,t}\ f_B(\pi,\eta,t)=f_A(\pi)$, and the optimal solution is just the equivalent condition of Eq. (37). It implies that[2]

$$f_B(\pi^*,\ \eta^*,\ t^*)=f_B(\pi^*,\ \mu\ /\pi^*,\ \sqrt{\pi^{*2}\cdot\sigma^2})=f_A(\pi^*),\tag{38}$$

where $(\pi^*,\ \eta^*,\ t^*)$ is the optimal solution for $\min_{\pi,\eta,t}\ f_B(\pi,\eta,t)$.

For any $\pi\in\Pi$, let $\eta=\mu\ /\pi,\ \ t=\sqrt{\pi^2\cdot\sigma^2}$, then

$$f_A(\pi^*)=f_B(\pi^*,\eta^*,t^*)\le f_B(\pi,\eta,t)=f_A(\pi),\tag{39}$$

which illustrates that π^* is also an optimal solution for problem $\min_\pi\ f_A(\pi)$. □

In view of Property 1, we can concentrate on solving the relaxed version R_2^B instead of the original problem R_2. According to the optimality condition, the optimal η and t can be obtained by Eq. (37) when π is fixed. On the other hand, when η and t are fixed, the optimal π can be obtained by sorting $\xi(\eta,\ t)$ in non-increasing order, where $\xi(\eta,\ t)$ is given by:

$$\xi(\eta,t)=\left(\frac{1}{2}\eta+\frac{k}{2t}\sigma^2\right).\tag{40}$$

Based on these features, an iterated descent algorithm, the Complete Cauchy-Relaxation Algorithm (CCRA), is designed to find the optimal solution for R_2^B. The pseudocode of CCRA is given in Algorithm 1.

Algorithm 1 Complete Cauchy-Relaxation Algorithm (CCRA).

Input: μ, σ, α and $\text{Iter}_{\max}$ (the maximum iteration).
1: Initialize π by non-increasing order of μ, and denote the sorted sequence as π_0.
2: **for** $m=1$ to $\text{Iter}_{\max}$ **do**
3: Solve $\min\limits_{\eta,t}\ f_B(\pi_{m-1},\eta,t)$ with fixed π_{m-1}. The optimal solution is $\eta_m=\mu\ /\pi_{m-1}\ ,\ \ t_m=\sqrt{\pi_{m-1}^2\cdot\sigma^2}$.
4: Solve $\min\limits_{\pi}\ f_B(\pi,\eta_m,t_m)$ with η_m and t_m fixed. The optimal solution π_m is obtained by non-increasing sorting of $\xi(\eta_m,t_m)$.
5: **if** $\pi_m=\pi_{m-1}$ **then**
6: Take $f_B^*(\alpha)=f_B(\pi_m,\eta_m,t_m)$ as the estimated optimal value for R_2^B and $\pi^*(\alpha)=\pi_m$ as the corresponding optimal solution.
7: Break the loop.
8: **end if**
9: **end for**
Output: $\pi^*(\alpha)$ and $f_B^*(\alpha)$.

As can be seen from the pseudocode, the objective function $f_B(\pi,\ \eta,\ t)$ strictly decreases in each iteration. This is illustrated in Property 2.

Property 2. *The CCRA (Algorithm 1) is a strict descent algorithm, that is,*

$$f_B(\pi_m,\eta_m,t_m)<f_B(\pi_{m-1},\eta_{m-1},t_{m-1}),\tag{41}$$

where m represents the iteration in Algorithm 1.

Proof. At the beginning of mth iteration, the current objective value is $f_B(\pi_{m-1},\eta_{m-1},t_{m-1})$. After step 3, the current value is given by $f_B(\pi_{m-1},\eta_m,t_m)=\min_{\eta,t}\ f_B(\pi_{m-1},\eta,t)$, which implies that

$$f_B(\pi_{m-1},\eta_m,t_m)\le f_B(\pi_{m-1},\eta_{m-1},t_{m-1}).\tag{42}$$

[2] For two vectors $x\in\mathbb{R}^n, y\in\mathbb{R}^n$, we define $x/y=(x_1/y_1,\cdots,x_n/y_n)$, where $y_i\neq 0,\ i=1,\cdots,n$.

After step 4, the current value becomes

$$f_B(\pi_m,\eta_m,t_m)=\min_{\pi}\ f_B(\pi,\eta_m,t_m)\le f_B(\pi_{m-1},\eta_m,t_m).\tag{43}$$

Thus, we have $f_B(\pi_m,\eta_m,t_m)\le f_B(\pi_{m-1},\eta_{m-1},t_{m-1})$, which means that the objective value becomes smaller or at least stays the same after each iteration.

Furthermore, let $f_B(\pi_m,\eta_m,t_m)=f_B(\pi_{m-1},\eta_{m-1},t_{m-1})$, due to the uniqueness of the optimal solution in step 3, this equation requires $\eta_{m-1}=\eta_m$ and $t_{m-1}=t_m$, which further indicates that $\pi_{m-2}=\pi_{m-1}$. However, when $\pi_{m-2}=\pi_{m-1}$, the algorithm has already stopped in the $(m-1)$th iteration. That is to say, the equivalent condition of Eq. (42) can never be satisfied in the iteration process, which proves the strict descent feature of Algorithm 1. □

Although the CCRA is a descending algorithm, it may get stuck at local optimal points because of the integer feature of π. As shown in the pseudocode, the CCRA terminates when $\pi_m=\pi_{m-1}$ or the maximum iteration is reached. Note, however, that different pairs, say $(\eta_m,\ t_m)$ and (η_{m-1},t_{m-1}), can lead to the same sorting result of $\xi(\eta_m,\ t_m)$ and $\xi(\eta_{m-1},t_{m-1})$ and make $\pi_m=\pi_{m-1}$, thus resulting in early convergence of the algorithm.

To address this issue, a modified CCRA (M-CCRA) is proposed. Considering the close relationship between local optimal solutions and initial values, more initial values are chosen in this modified algorithm M-CCRA. In the CCRA, initial value π_0 is obtained by non-increasing sorting of μ. However, in the modified algorithm, there is a total of L initial values $(\pi_0^1,\dots,\pi_0^L)$ arrayed by

$$\theta_l\mu+(1-\theta_l)\boldsymbol{\sigma^2},\qquad l=1,\dots,L\tag{44}$$

where θ_l varies from 0 to 1 with a step length $\theta_s=1/(L-1)$, that is,

$$\theta_l=(l-1)\times\theta_s,\qquad l=1,\dots,L.\tag{45}$$

Under this setting, the initial value is determined by both the mean and variance of input data. The value of θ represents the weight of mean and the step length θ_s controls the number of initial values. For each initial value, the process of CCRA leads to an estimated solution. Among these solutions, the one that gives the minimum objective value is selected as the final output solution of M-CCRA. The pseudocode of M-CCRA is shown in Algorithm 2.

It is worth noting that the M-CCRA still cannot guarantee global optimality for solving R_2^B. When step length θ_s is small enough, however, the estimated solution can be very close to or even just be the global optimal solution.

In both the CCRA (Algorithm 1) and M-CCRA (Algorithm 2), the main computational burden lies in sorting of components $\xi(\eta_m,\ t_m)$. Therefore, the computational complexity in each iteration is $O(n\log n)$ for the CCRA and $O(nL\log n)$ for the M-CCRA, where L is the number of initial values in the M-CCRA.

3.2. Partial Cauchy-relaxation algorithm

In the CCRA, the two parts (i.e., mean and variance) related to π are both relaxed, making $\min_\pi\ f_B(\pi,\eta_m,t_m)$ part in the algorithm tractable by simple sorting. However, this relaxation introduces $(n+1)$ additional variables (i.e., $\eta\in\mathbb{R}_+^n$ and $t\in\mathbb{R}_+$) to the problem, and results in a simple sorting subproblem in each iteration, which can make the CCRA converge prematurely or get stuck at local optimal solutions. To overcome this shortcoming, a Partial Cauchy-Relaxation Algorithm (PCRA) is proposed, within which only one variable $t\in\mathbb{R}_+$ is introduced, and an assignment subproblem instead of the simple sorting one is obtained.

Algorithm 2 Modified Complete Cauchy-Relaxation Algorithm (M-CCRA).

Input: $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, α, L and $\mathrm{Iter}_{\max}$.
Let $\theta_s = 1/(L-1)$.
for $l = 1$ to L **do**
  Let $\theta_l = (l-1)\times\theta_s$.
  Initialize $\boldsymbol{\pi}$ by non-increasing order of $\theta_l\boldsymbol{\mu} + (1-\theta_l)\boldsymbol{\sigma}^2$, and denote the sorted sequence as $\boldsymbol{\pi}_0^l$.
  for $m = 1$ to $\mathrm{Iter}_{\max}$ **do**
    Solve $\min\limits_{\boldsymbol{\eta},t} f_B(\boldsymbol{\pi}_{m-1}, \boldsymbol{\eta}, t)$ with fixed $\boldsymbol{\pi}_{m-1}$. The optimal solution is $\boldsymbol{\eta}_m = \boldsymbol{\mu}\ /\boldsymbol{\pi}_{m-1}$, $t_m = \sqrt{\boldsymbol{\pi}_{m-1}^2 \cdot \boldsymbol{\sigma}^2}$.
    Solve $\min\limits_{\boldsymbol{\pi}} f_B(\boldsymbol{\pi}, \boldsymbol{\eta}_m, t_m)$ with $\boldsymbol{\eta}_m$ and t_m fixed. The optimal solution $\boldsymbol{\pi}_m$ is obtained by non-increasing sorting of $\boldsymbol{\xi}(\boldsymbol{\eta}_m, t_m)$.
    if $\boldsymbol{\pi}_m = \boldsymbol{\pi}_{m-1}$ **then**
      Calculate $f_B^l = f_B(\boldsymbol{\pi}_m, \boldsymbol{\eta}_m, t_m)$ and let $\boldsymbol{\pi}^l = \boldsymbol{\pi}_m$.
      Break this inner loop.
    end if
  end for
end for
Find $f_B^*(\alpha) = \min\limits_l f_B^l$ as the estimated optimal value of R_2^B and $\boldsymbol{\pi}^*(\alpha) = \boldsymbol{\pi}^l$ is the corresponding optimal solution.
Output: $\boldsymbol{\pi}^*(\alpha)$ and $f_B^*(\alpha)$.

In the PCRA, the original objective function $f_A(\boldsymbol{\pi})$ is relaxed to $f_C(\boldsymbol{\pi}, t)$ as:

$$f_C(\boldsymbol{\pi}, t) = \sum_{j=1}^{n} \pi_j\mu_j + \frac{k}{2t}\sum_{j=1}^{n}\pi_j^2\sigma_j^2 + \frac{kt}{2}, \tag{46}$$

and the equality of $f_A(\boldsymbol{\pi}) \le f_C(\boldsymbol{\pi}, t)$ holds if and only if

$$t = \sqrt{\sum_{j=1}^{n}\pi_j^2\sigma_j^2}. \tag{47}$$

Then, R_2 is relaxed to problem R_2^C as follows:

$$(R_2^C)\quad \min_{\boldsymbol{\pi},t} f_C(\boldsymbol{\pi}, t), \qquad s.t.\ \boldsymbol{\pi} \in \Pi, t \in \mathbb{R}_+. \tag{48}$$

Similar to the CCRA, this partial relaxation maintains the equivalence between R_2^C and original R_2 from the perspective of optimization, which is illustrated in Property 3.

Property 3. *Let $(\boldsymbol{\pi}^*,\ t^*)$ be an optimal solution for problem $\min_{\boldsymbol{\pi},t}\ f_C(\boldsymbol{\pi}, t)$, then $\boldsymbol{\pi}^*$ is also an optimal solution for problem $\min_{\boldsymbol{\pi}}\ f_A(\boldsymbol{\pi})$.*

Proof. Similar to the proof of Property 1. □

Because $x_{ij} \in \{0, 1\}$ and $\sum_{i=1}^n x_{ij} = 1$ as constrained in the definition of $\mathbb{X}$ (Eq. (3)), π_j^2 can be calculated as

$$\pi_j^2 = \sum_{i=1}^{n}(n-i+1)^2 x_{ij}^2 = \sum_{i=1}^{n}(n-i+1)^2 x_{ij}. \tag{49}$$

Therefore, $f_C(\boldsymbol{\pi}, t)$ can be written as a function of $\boldsymbol{X}$ and t:

$$f_C(\boldsymbol{X}, t) = \sum_{j=1}^{n}\sum_{i=1}^{n} x_{ij}\left[(n-i+1)\mu_j + \frac{k}{2t}(n-i+1)^2\sigma_j^2\right] + \frac{kt}{2}. \tag{50}$$

Let $c_{ij} = \left[(n-i+1)\mu_j + \frac{k}{2t}(n-i+1)^2\sigma_j^2\right]$, $f_C(\boldsymbol{X},\ t)$ is simplified as:

$$f_C(\boldsymbol{X}, t) = \sum_{j=1}^{n}\sum_{i=1}^{n} x_{ij}c_{ij} + \frac{kt}{2}. \tag{51}$$

When t is fixed, the expression in Eq. (51) is a standard form of AP, which can be efficiently solved by the Hungarian Method (Kuhn, 1955).

According to the analyses above, the optimal t is given by the equivalent condition Eq. (47) when $\boldsymbol{\pi}$ is fixed. On the other hand, the subproblem becomes an AP when t is fixed. Based on these features, the PCRA is used to find the optimal solution for R_2^C. The pseudocode of PCRA is presented in Algorithm 3.

Algorithm 3 Partial Cauchy-Relaxation Algorithm (PCRA).

Input: $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, α and $\mathrm{Iter}_{\max}$ (the maximum iteration).
Initialize $\boldsymbol{\pi}$ by non-increasing order of $\boldsymbol{\mu}$, and denote the sorted sequence as $\boldsymbol{\pi}_0$.
Get $\boldsymbol{X}_0$ from $\boldsymbol{\pi}_0$ according to the definition of $\boldsymbol{\pi}$.
for $m = 1$ to $\mathrm{Iter}_{\max}$ **do**
  Solve $\min\limits_{t} f_C(\boldsymbol{\pi}, t)$ with fixed $\boldsymbol{\pi}_{m-1}$. The optimal solution is $t_m = \sqrt{\boldsymbol{\pi}_{m-1}^2 \cdot \boldsymbol{\sigma}^2}$.
  Solve $\min\limits_{\boldsymbol{X}} f_C(\boldsymbol{X}, t)$ with t_m fixed. The optimal solution $\boldsymbol{X}_m$ is obtained by the Hungarian Method.
  Calculate $\boldsymbol{\pi}_m = \sum\limits_{i=1}^{n}(n-i+1)\boldsymbol{x}_{mi}^T$.
  if $\boldsymbol{\pi}_m = \boldsymbol{\pi}_{m-1}$ **then**
    Take $f_C^*(\alpha) = f_C(\boldsymbol{X}_m, t_m)$ as the estimated optimal value for R_2^C and $\boldsymbol{X}^*(\alpha) = \boldsymbol{X}_m, \boldsymbol{\pi}^*(\alpha) = \boldsymbol{\pi}_m$ as the corresponding optimal solution.
    Break the loop.
  end if
end for
Output: $\boldsymbol{\pi}^*(\alpha)$, $\boldsymbol{X}^*(\alpha)$ and $f_C^*(\alpha)$.

This PCRA introduces only one variable t to relax the variance part, while the mean part is kept the same. So the relaxation here is tighter than that in the CCRA discussed in Section 3.1. Moreover, an AP instead of a simple sorting problem is obtained in each iteration, which can effectively reduce the probability of getting trapped at local optimal solutions. These changes substantially reduce the gap between the estimated solution and the global optimal solution, which will be further illustrated in the experiment section.

The main computational burden of the PCRA lies in the implementation of Hungarian Method, whose complexity is given by $O(n^3)$ (Kuhn, 1955). This is higher than the CCRA's $O(n\log n)$ complexity. In other words, the PCRA may gain better estimation accuracy at the cost of computation time.

4. Numerical experiments and results

We conducted comprehensive computational experiments to evaluate the robustness of the proposed DR-SMSP model and the performance of corresponding solution algorithms. All the algorithms were coded in Matlab®(version R2013a(8.1.0.604) for Windows x64) and experiments executed on a personal computer with an Intel Core i7-4770K 3.50 gigahertz CPU and 32 gigabyte RAM. The experimental design and computational results are discussed in the following subsections.

4.1. Design and setup

For our experiments, the size of the problem instances (i.e., number of jobs n) was selected from the set $n \in \{10, 15, 20, 30, 50, 100, 200\}$. The processing time for each job was a random variable with unknown distribution specified by a known mean μ_j and variance σ_j^2. These means were generated as random integers from a uniform distribution in interval [10, 50], and the standard deviations were also random integers generated from a uniform distribution in interval [1, 30]. Furthermore, the confidence level α of CVaR was set at 0.95 by default unless otherwise specified in the text. The parameter values used in the three Cauchy-relaxation algorithms were set at $\text{Iter}_{\max} = n \times 30$ and $\theta_s = 0.05$.

To evaluate the effectiveness and efficiency of our Cauchy-relaxation algorithms, the proposed CCRA, M-CCRA and PCRA were compared to a commercial solver IBM®ILOG®CPLEX (version V12.6.2 for Windows x64) in terms of objective values and CPU time. CPLEX®for Matlab, which is an extension to IBM®ILOG®CPLEX Optimizers in the Matlab framework, was used. This ensures that the three proposed algorithms and the CPLEX solver were compared on the same ground, i.e., operated in the same environment and shared the same computational resource.

To adapt subproblem R_2 to the CPLEX solver, we reformulated it as a standard 0–1 SOCP problem. The objective function $f_A(\boldsymbol{\pi})$ for R_2 can be rewritten as

$$f_A(\boldsymbol{X}) = \sum_{j=1}^{n}\sum_{i=1}^{n}(n-i+1)\mu_j x_{ij} + k\sqrt{\sum_{j=1}^{n}\sum_{i=1}^{n}(n-i+1)^2\sigma_j^2 x_{ij}^2}. \tag{52}$$

Then, by introducing a new decision variable ν, R_2 was formulated as a standard 0–1 SOCP problem as follows:

$$\begin{aligned} &(\text{CPLEX } R_2) && \min \sum_{j=1}^{n}\sum_{i=1}^{n}(n-i+1)\mu_j x_{ij} + k\nu, \\ &\text{s.t.} && \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{n}(n-i+1)^2\sigma_j^2 x_{ij}^2} \le \nu, \\ &&& \boldsymbol{X} \in \mathbb{X}, \end{aligned} \tag{53}$$

where $\mathbb{X}$ is the feasible set of $\boldsymbol{X}$ given in Eq. (3).

The algorithm used by the CPLEX solver is branch and cut, which is an exact algorithm. It cannot find the optimal solution of 0–1 SOCP with large-scale instances ($n \ge 20$) within a few hours. Also, due to the uncertainty of production environments, there are always some jobs that might be added to or removed from the set of jobs waiting for processing. Whenever the job information changes, a previous optimal sequence would become invalid and rescheduling is needed. A production system is supposed to respond quickly to these changes, and the response time of many hours or longer is not acceptable in practice. Therefore, the maximum computing time for the CPLEX solver was set at 10,800 seconds (3 hours), and the 'best-so-far' solutions obtained within the time limit (i.e., 3 hours) were used for the comparison.

The proposed DR-SMSP model aims to find a robust sequence that minimizes RCVaR_α of the TFT. When a robust sequence $\boldsymbol{X}^*(\alpha)$ is obtained, the corresponding theoretical mean μ_f and standard deviation σ_f as well as RCVaR_α of the TFT can be calculated by Eq. (29) and Eq. (18), respectively. To evaluate the robustness of solution $\boldsymbol{X}^*(\alpha)$, $\boldsymbol{X}^*(\alpha)$ was compared to the nominal solution $\overline{\boldsymbol{X}}$ obtained from the deterministic model. The processing time of each job in the corresponding deterministic version of SMSP was set to be the mean of processing time in the DR-SMSP. Therefore, $\overline{\boldsymbol{X}}$ can be directly obtained by the SAPT rule in the DR-SMSP model. In this experiment, 5,000 independent runs with different mean vectors $\boldsymbol{\mu}$ and standard deviation vectors $\boldsymbol{\sigma}$ were generated to obtain the average μ_f, σ_f, and RCVaR_α of $\boldsymbol{X}^*(\alpha)$ and $\overline{\boldsymbol{X}}$ for different confidence levels. In each run, the mean μ_j and standard deviation σ_j of each job j were randomly selected from interval [10, 50] and [1, 30], respectively.

We compared the robust sequence $\boldsymbol{X}^*(\alpha)$ and nominal solution $\overline{\boldsymbol{X}}$ in three aspects, i.e., theoretical mean μ_f, standard deviation σ_f and RCVaR_α of the TFT. To describe the difference between robust and nominal solutions in these aspects, three corresponding evaluation variables were used, denoted as: mean price $\text{MP}(\alpha)$, deviation reduction $\text{DR}(\alpha)$ and risk reduction $\text{RR}(\alpha)$. They are explicitly defined as follows:

$$\text{MP}(\alpha) = \mu_f(\boldsymbol{X}^*(\alpha)) - \mu_f(\overline{\boldsymbol{X}}), \tag{54}$$

$$\text{DR}(\alpha) = \sigma_f(\overline{\boldsymbol{X}}) - \sigma_f(\boldsymbol{X}^*(\alpha)), \tag{55}$$

$$\text{RR}(\alpha) = \text{RCVaR}_\alpha(\overline{\boldsymbol{X}}) - \text{RCVaR}_\alpha(\boldsymbol{X}^*(\alpha)). \tag{56}$$

Here, $\text{MP}(\alpha)$ describes the price rise caused by employing the robust sequence $\boldsymbol{X}^*(\alpha)$ instead of the nominal one $\overline{\boldsymbol{X}}$; $\text{DR}(\alpha)$ refers to the standard deviation reduction gained by using the robust sequence $\boldsymbol{X}^*(\alpha)$; and $\text{RR}(\alpha)$ represents the RCVaR_α difference between two solutions, which shows the risk reduction caused by using the robust sequence $\boldsymbol{X}^*(\alpha)$. The relative counterparts of these three evaluation variables are given by:

$$\text{R-MP}(\alpha) = [\mu_f(\boldsymbol{X}^*(\alpha)) - \mu_f(\overline{\boldsymbol{X}})]/\mu_f(\boldsymbol{X}^*(\alpha)), \tag{57}$$

$$\text{R-DR}(\alpha) = [\sigma_f(\overline{\boldsymbol{X}}) - \sigma_f(\boldsymbol{X}^*(\alpha))]/\sigma_f(\boldsymbol{X}^*(\alpha)), \tag{58}$$

$$\text{R-RR}(\alpha) = [\text{RCVaR}_\alpha(\overline{\boldsymbol{X}}) - \text{RCVaR}_\alpha(\boldsymbol{X}^*(\alpha))]/\text{RCVaR}_\alpha(\boldsymbol{X}^*(\alpha)). \tag{59}$$

The above evaluation metrics focus on the theoretical robustness of robust sequence $\boldsymbol{X}^*(\alpha)$. In the following part, practical robustness will be considered. Given a certain mean vector $\boldsymbol{\mu}$ and standard deviation vector $\boldsymbol{\sigma}$ in a run, the two solutions, i.e., $\boldsymbol{X}^*(\alpha)$ and $\overline{\boldsymbol{X}}$ can be obtained. Then, plenty of processing time instances can be generated to compare the TFT of them in different aspects. In our experiment, a total of 500,000 processing time instances were generated from a normal distribution[3] with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. For each instance, $\text{TFT}(\boldsymbol{X}^*(\alpha))$s and $\text{TFT}(\overline{\boldsymbol{X}})$s were obtained for different confidence levels. Then, the means, standard deviations, and CVaR_αs of TFTs obtained by the two solutions were compared. Here, the uncertainty of processing time is reflected in $\boldsymbol{\sigma}$, and the uncertainty of TFT is reflected in its standard deviations and CVaR_αs. Therefore, the sensitivity of uncertainty in the DR-SMSP model, i.e., how the uncertainty of TFT is influenced by the uncertainty of processing time, was investigated in the process of practical robustness analysis. In the following comparison, we denote the difference of means as robust price $\text{RP}(\alpha)$, the difference of standard deviations as robust benefit $\text{RB}(\alpha)$, and the difference of CVaR_αs as hedge value $\text{HV}(\alpha)$. Their definitions are as follows:

$$\text{RP}(\alpha) = \mu\text{TFT}(\boldsymbol{X}^*(\alpha)) - \mu\text{TFT}(\overline{\boldsymbol{X}}), \tag{60}$$

$$\text{RB}(\alpha) = \sigma\text{TFT}(\overline{\boldsymbol{X}}) - \sigma\text{TFT}(\boldsymbol{X}^*(\alpha)), \tag{61}$$

$$\text{HV}(\alpha) = \text{CVaR}_\alpha(\overline{\boldsymbol{X}}) - \text{CVaR}_\alpha(\boldsymbol{X}^*(\alpha)), \tag{62}$$

where μTFT and σTFT represent the mean and standard deviation of the TFT, respectively. $\text{RP}(\alpha)$ and $\text{RB}(\alpha)$ are equivalent to the price and benefit in terms of TFT when adopting the DR model instead of a deterministic one. $\text{HV}(\alpha)$ shows the risk that the decision maker needs to take if a nominal solution $\overline{\boldsymbol{X}}$ is employed. The corresponding relative counterparts are given as

$$\text{R-RP}(\alpha) = [\mu\text{TFT}(\boldsymbol{X}^*(\alpha)) - \mu\text{TFT}(\overline{\boldsymbol{X}})]/\mu\text{TFT}(\boldsymbol{X}^*(\alpha)), \tag{63}$$

$$\text{R-RB}(\alpha) = [\sigma\text{TFT}(\overline{\boldsymbol{X}}) - \sigma\text{TFT}(\boldsymbol{X}^*(\alpha))]/\sigma\text{TFT}(\boldsymbol{X}^*(\alpha)), \tag{64}$$

$$\text{R-HV}(\alpha) = [\text{CVaR}_\alpha(\overline{\boldsymbol{X}}) - \text{CVaR}_\alpha(\boldsymbol{X}^*(\alpha))]/\text{CVaR}_\alpha(\boldsymbol{X}^*(\alpha)). \tag{65}$$

Note that the proposed DR model does not require any information about the distributional type of processing time, as solution robustness can also be evaluated under some other processing time distribution assumptions. To extend on the above design,

[3] The distribution is not necessarily a normal distribution, and the choice here is just for convenience.

Table 1
Results obtained by the CCRA, M-CCRA, PCRA and CPLEX.

	Average objective value for R_2				Average computation time (s)			
n	CCRA	M-CCRA	PCRA	CPLEX	CCRA	M-CCRA	PCRA	CPLEX
10	2645.44	2593.13	**2591.28**	**2591.28**	0.0002	0.0018	0.0057	1.8580
15	5127.11	5067.49	**5064.30**	**5064.30**	0.0003	0.0033	0.0122	3981.53
20	8697.59	8608.25	**8600.45**	8609.06	0.0004	0.0044	0.0267	10800.00
30	17668.52	17584.63	**17573.22**	17659.60	0.0006	0.0067	0.0892	10800.00
50	44141.95	44069.25	**44046.85**	44466.68	0.0057	0.0189	0.4739	10800.00
100	161429.87	161373.81	**161338.21**	162445.25	0.0034	0.0499	4.6516	10800.00
200	583562.81	583499.11	**583440.49**	591614.14	0.0098	0.1486	72.2052	10800.00

Table 2
Comparison of relative average objective values of R_2(percent).

	Relative average objective values of R_2				
n	CCRA (percent)	M-CCRA (percent)	PCRA (percent)	CPLEX (percent)	Best
10	2.09	0.071	**0**	**0**	PCRA&CPLEX
15	1.24	0.063	**0**	**0**	PCRA&CPLEX
20	1.13	0.090	**0**	0.10	PCRA
30	0.54	0.064	**0**	0.49	PCRA
50	0.22	0.051	**0**	0.95	PCRA
100	0.06	0.022	**0**	0.69	PCRA
200	0.02	0.010	**0**	1.40	PCRA

the normal distribution used to generate processing time instances was replaced by a uniform distribution, Gamma distribution and Laplace distribution with the same $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. For each distribution, the TFTs of both the robust model and the deterministic one were calculated to illustrate the robustness under different distribution assumptions.

4.2. Experimental results and analyses

4.2.1. Evaluation of the Cauchy-relaxation algorithms

To evaluate the performances of the proposed Cauchy-relaxation algorithms, we compared the CCRA, M-CCRA and PCRA to an IBM®ILOG®CPLEX solver under the Matlab®framework in terms of objective values and CPU time consumption. As mentioned before, the maximum computing time for the CPLEX solver was set to 10,800 seconds (3 hours). It is worth noting that all of the computation time discussed in this section is CPU time used in running the algorithms without reading or writing. For problem instances of each size, 30 trials with independently generated means and variances were conducted to obtain the average results.

Table 1 shows the results obtained by the proposed algorithms and CPLEX solver. The optimal (or best) solutions found among the four methods are highlighted in bold. We can see from the table that CPLEX is able to find optimal solutions for R_2 on small-scale instances ($n = 10, 15$) within reasonable time. It is noticeable that these solutions are identical to those obtained by the PCRA. As for problem instances of larger scale ($n \geq 20$), CPLEX fails to obtain optimal solutions within the acceptable time (3 hours). Thus, the 'best-so-far' solutions obtained within the set time frame (i.e., 3 hours) are presented. As revealed in Table 1, solutions given by the PCRA are better than those obtained by CPLEX on problem instances with n above 20.

For a clearer comparison, the relative objective values pitted against the best solutions are presented in Table 2. The best solutions, which are the benchmarks of these relative objective values, are highlighted in bold. From the table, we observe that the PCRA always performs better than the M-CCRA. Note, however, that the relative optimization gaps between the CCRA, M-CCRA and PCRA are getting smaller as the problem scale increases. When compared to CPLEX, the PCRA outperforms it to a larger extent as the problem scale increases. For problem instances with $n \geq 50$, even the CCRA can perform better than CPLEX. These results suggest that although global optimality is not guaranteed, an acceptable suboptimal solution, which is superior to the one obtained by CPLEX, can be generated by the PCRA within polynomial time. For large-scale problem instances, the CCRA and M-CCRA can also reach near optimal solutions. These results confirm the effectiveness of the proposed Cauchy-relaxation algorithms.

In terms of computational efficiency, it is observed in Table 1 that the CCRA and M-CCRA require less computational efforts, while the PCRA needs a little more (but still acceptable) computation time. For CPLEX, its computation time increases drastically as the problem size increases. The running time of CPLEX would reach 3 hours even when the problem size (i.e., n) is as small as 20. These results indicate that the proposed Cauchy-relaxation algorithms can solve problem instances of practical size efficiently. As discussed in Section 4.1, the production system is supposed to have a quick response to changes in the environment, and rescheduling should be done within a short time. Although in theory the CPLEX solver can obtain the optimal solution when enough computation time is given, the required time might be many days or even months for problems of large scale, which is not acceptable in practice. Under this circumstance, the three proposed Cauchy-relaxation algorithms, which can find solutions better than those obtained by the CPLEX solver within a few minutes, are of great significance in practical production systems.

4.2.2. Solution robustness and sensitivity analysis

In addition to evaluating the performances of the algorithms, we also investigated the robustness of the proposed DR-SMSP model under different confidence levels. Specifically, we looked at the results between the robust sequence $\boldsymbol{X}^*(\alpha)$ obtained by the DR-SMSP and the nominal solution $\overline{\boldsymbol{X}}$ by the deterministic SMSP. We first analyzed the average difference between $\boldsymbol{X}^*(\alpha)$ and $\overline{\boldsymbol{X}}$ in terms of theoretical mean μ_f, standard deviation σ_f and RCVaR$_\alpha$ of the TFT. Then, numerous processing time instances were generated to compare the practical mean μTFT, standard deviation σTFT and CVaR$_\alpha$ for a specific $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ case. At last, different processing times were generated from different distributions to help evaluate the robustness of distribution.

Table 3 shows the average theoretical means and standard deviations of the TFT under different confidence level settings. A total

Table 3
Average theoretical means and standard deviations of the TFT (n=10).

	Theoretical mean of TFT				Theoretical standard deviation of TFT			
α	$\bar{\mu}_f(\boldsymbol{X}^*(\alpha))$	$\bar{\mu}_f(\overline{\boldsymbol{X}})$	MP(α)	R-MP(α) (percent)	$\bar{\sigma}_f(\boldsymbol{X}^*(\alpha))$	$\bar{\sigma}_f(\overline{\boldsymbol{X}})$	DR(α)	R-DR(α) (percent)
0.99	1580.97	1346.78	234.19	14.81	223.94	349.94	126.00	56.26
0.98	1563.33	1345.56	217.77	13.93	226.55	351.48	124.93	55.14
0.97	1549.33	1351.46	197.87	12.77	229.19	350.47	121.28	52.92
0.96	1532.32	1349.58	182.74	11.93	232.99	351.34	118.35	50.79
0.95	1519.90	1349.07	170.83	11.02	235.26	351.38	116.12	49.36
0.94	1511.03	1352.88	158.15	10.47	239.37	351.56	112.19	46.87
0.93	1497.51	1351.50	146.01	9.75	240.72	349.69	108.97	45.27
0.92	1482.48	1345.49	136.99	9.24	244.44	350.80	106.36	43.51
0.91	1481.35	1351.43	129.92	8.77	246.78	351.71	104.93	42.52
0.90	1472.19	1350.90	121.29	8.24	249.91	351.86	101.95	40.80
0.8	1421.22	1351.35	69.87	4.92	271.06	351.82	80.76	29.79
0.7	1395.46	1350.98	44.48	3.19	284.72	349.67	64.95	22.81
0.6	1383.56	1352.74	30.82	2.23	297.43	352.50	55.07	18.52
0.5	1370.57	1349.19	21.38	1.56	305.25	351.48	46.23	15.15
0.4	1366.72	1351.84	14.88	1.09	313.72	352.68	38.96	12.42
0.3	1362.24	1352.41	9.83	0.72	319.56	351.50	31.94	10.00
0.2	1353.50	1347.66	5.84	0.43	327.33	352.22	24.89	7.60
0.1	1353.85	1351.57	2.28	0.17	336.35	351.15	14.80	4.40
0	1353.21	1353.21	0.00	0.00	350.07	350.10	0.03	0.008

Table 4
Average RCVaR_α of the TFT (n=10).

	RCVaR_α of TFT			
α	$\text{RCVaR}_\alpha(\boldsymbol{X}^*(\alpha))$	$\text{RCVaR}_\alpha(\overline{\boldsymbol{X}})$	RR(α)	R-RR(α) (percent)
0.99	3825.63	4857.49	1031.86	26.97
0.98	3147.58	3804.03	656.45	20.86
0.97	2863.02	3353.52	490.50	17.13
0.96	2670.42	3067.04	396.62	14.85
0.95	2547.27	2880.70	333.43	13.09
0.94	2450.52	2739.01	288.49	11.77
0.93	2376.01	2626.87	250.86	10.56
0.92	2318.09	2542.81	224.72	9.69
0.91	2262.84	2463.41	200.57	8.86
0.90	2227.22	2408.42	181.20	8.14
0.8	1965.12	2056.27	91.15	4.64
0.7	1830.44	1886.13	55.69	3.04
0.6	1741.42	1776.90	35.48	2.04
0.5	1673.76	1698.64	24.88	1.49
0.4	1615.98	1632.74	16.76	1.04
0.3	1568.65	1579.69	11.04	0.70
0.2	1520.74	1527.26	6.52	0.43
0.1	1461.14	1463.58	2.44	0.17
0	1354.98	1354.98	0.00	0

of 5,000 scenarios with different $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ were independently generated to calculate the average results. The average RCVaR_α results in Table 4 were obtained through the same settings. It is revealed from Table 3 that for each α, $\bar{\mu}_f(\boldsymbol{X}^*(\alpha))$ is greater than $\bar{\mu}_f(\overline{\boldsymbol{X}})$ while $\bar{\sigma}_f(\boldsymbol{X}^*(\alpha))$ is less than $\bar{\sigma}_f(\overline{\boldsymbol{X}})$. As for the average RCVaR_α in Table 4, robust sequence $\boldsymbol{X}^*(\alpha)$, which is theoretically the DR-SMSP's optimal solution, certainly possesses the 'smaller' results. These results show that the proposed DR-SMSP model can achieve risk reduction with a sacrifice on the optimality of the mean value.

From the results of nominal solution $\overline{\boldsymbol{X}}$ shown in Tables 3 and 4, we see that the confidence level α has no effect on $\bar{\mu}_f$ and $\bar{\sigma}_f$, while the RCVaR_α of $\overline{\boldsymbol{X}}$ reduces with α. The latter can be explained by the proportional relation between α and RCVaR_α in the second case of Eq. (11) in Theorem 1.

Next, we examined the results obtained by the robust sequence $\boldsymbol{X}^*(\alpha)$. Along with the reduction of α, μ_f and RCVaR_α decrease while σ_f increases. In particular, the values of all the three different evaluation variables (i.e., MP(α), DR(α) and RR(α)) and their corresponding relative counterparts (i.e., R-MP(α), R-DR(α) and R-RR(α)) decrease when α decreases, which indicates that the results obtained by $\boldsymbol{X}^*(\alpha)$ and $\overline{\boldsymbol{X}}$ are getting closer in the decreasing process of α. When $\alpha = 0$, the results obtained by $\boldsymbol{X}^*(\alpha)$ and $\overline{\boldsymbol{X}}$ are almost the same. This phenomenon can be explained as follows. The definition of CVaR_α is the expectation over the worst $(1-\alpha)\times 100$ percent of random loss. As α decreases, the proportion contributing to CVaR_α increases (i.e., $(1-\alpha)$ gets close to 1, so that the worst $(1-\alpha)\times 100$ percent is approaching the whole support set), which reduces the difference between the objective functions of the DR-SMSP and the deterministic SMSP. When α equals 0, the difference disappears.

To illustrate this phenomenon more clearly, the normal distributions[4] of different $\bar{\mu}_f$ and $\bar{\sigma}_f$ obtained in selected cases $\alpha = \{0.99, 0.8, 0.6, 0.4, 0.2, 1\}$ are depicted in Fig. 1. The blue curves represent the TFT distributions with $\bar{\mu}_f(\overline{\boldsymbol{X}})$ and $\bar{\sigma}_f(\overline{\boldsymbol{X}})$ listed in Table 3, while other colors represent the results given by $\boldsymbol{X}^*(\alpha)$ under different α settings. As shown in this figure, all the blue curves are almost the same. Curves of other colors have greater means and smaller deviations than the blue ones, and these differences decrease with α.

Tables 5 and 6 present the practical mean, standard deviation and CVaR_α of TFTs under a certain $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ case (n=10). The parameters were randomly chosen as $\boldsymbol{\mu} = [45; 33; 48; 25; 27; 35; 41; 28; 37; 37]$ and $\boldsymbol{\sigma} = [19; 16; 26; 25; 13; 20; 7; 5; 18; 26]$. A total of 500,000 instances were generated from a normal distribution to obtain the practical results. It is revealed from the tables that these practical results are homologous with the theoretical ones presented in Tables 3 and 4. Under the same setting of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, the results of TFT given by $\boldsymbol{X}^*(\alpha)$ have greater means, but smaller deviations and CVaR_αs. The comparison of empirical distributions between TFT($\overline{\boldsymbol{X}}$)s and TFT($\boldsymbol{X}^*(\alpha)$)s with $\alpha = 0.99$ is shown in Fig. 2. This not only demonstrates the practical robustness of $\boldsymbol{X}^*(\alpha)$, but also verifies that the proposed DR-SMSP model possesses lower sensitivity on the uncertainty of processing time.

As α decreases, μTFT($\overline{\boldsymbol{X}}$) and σTFT($\overline{\boldsymbol{X}}$) fluctuate within a tiny range, while μTFT($\boldsymbol{X}^*(\alpha)$) decreases and σTFT($\boldsymbol{X}^*(\alpha)$) increases sig-

[4] The distribution considered here does not have to be normal. We give a specific probability distribution on this occasion to help better illustrate the effect of α on the computational results.

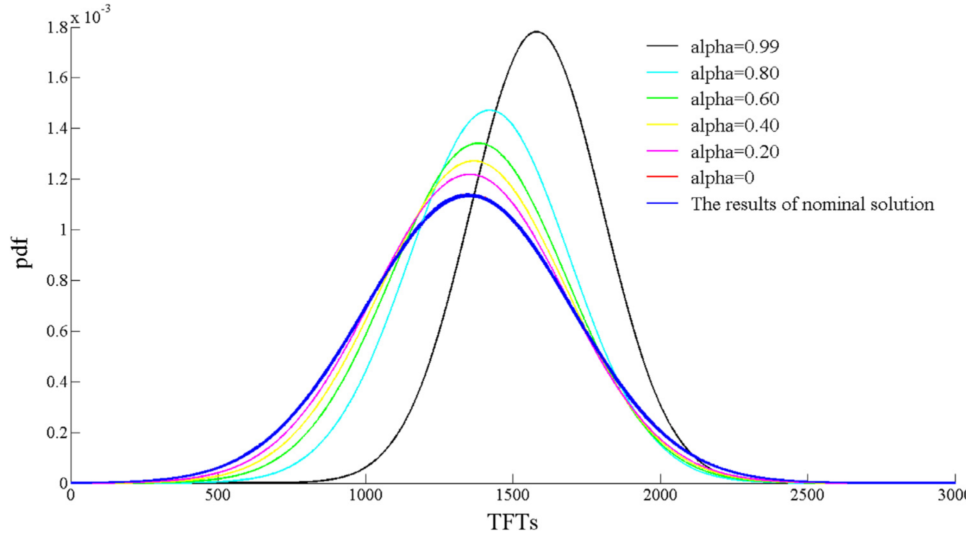

Fig. 1. Distributions of TFTs under different α settings. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

Table 5
Practical means and standard deviations of the TFT (n=10, normal distribution).

	Practical mean of TFT				Practical standard deviation of TFT			
α	μTFT($\boldsymbol{X}^*(\alpha)$)	μTFT($\bar{\boldsymbol{X}}$)	RP(α)	R-RP(α) (percent)	σTFT($\boldsymbol{X}^*(\alpha)$)	σTFT($\bar{\boldsymbol{X}}$)	RB(α)	R-RB(α) (percent)
0.99	1873.17	1757.85	115.32	6.16	258.62	359.92	101.30	39.17
0.9	1836.43	1752.50	83.93	4.57	262.54	356.60	94.06	35.83
0.8	1807.90	1753.69	54.21	3.00	276.30	358.94	82.64	29.91
0.7	1801.84	1749.16	52.68	2.92	275.31	356.10	80.79	29.35
0.6	1781.24	1754.16	27.08	1.52	298.26	359.48	61.22	20.53
0.5	1778.49	1749.79	28.70	1.61	295.85	357.82	61.97	20.95
0.4	1761.16	1750.55	10.61	0.60	319.11	362.61	43.50	13.63
0.3	1762.06	1752.96	9.10	0.52	314.01	357.98	43.97	14.00
0.2	1756.94	1750.96	5.98	0.34	326.48	363.56	37.08	11.36
0.1	1760.09	1754.65	5.44	0.31	319.63	355.68	36.05	11.28
0	1755.54	1755.54	0.00	0.00	355.48	359.78	4.30	1.21

Table 6
CVaR_α of TFT (n=10, normal distribution).

	CVaR_α of TFT			
α	$\mathrm{CVaR}_\alpha(\boldsymbol{X}^*(\alpha))$	$\mathrm{CVaR}_\alpha(\bar{\boldsymbol{X}})$	HV(α)	R-HV(α) (percent)
0.99	2563.39	2722.84	159.45	6.22
0.9	2297.21	2381.06	83.85	3.65
0.8	2193.57	2254.50	60.93	2.78
0.7	2120.08	2161.50	41.42	1.95
0.6	2068.68	2101.05	32.37	1.57
0.5	2013.84	2034.93	21.09	1.05
0.4	1967.41	1984.96	17.55	0.89
0.3	1917.96	1930.87	12.91	0.67
0.2	1871.05	1877.72	6.67	0.36
0.1	1822.61	1824.47	1.86	0.10
0	1755.54	1755.54	0.00	0.00

nificantly. These lead to the reduction of RP(α), RB(α), R-RP(α) and R-RB(α). As for the perspective of risk aversion shown in Table 6, the CVaR_αs of $\bar{\boldsymbol{X}}$ and $\boldsymbol{X}^*(\alpha)$ as well as the difference between them both decrease with α. It is clear that these simulation results are consistent with the experimental results based on theoretical scenarios.

Table 7 shows the solution robustness under different probability distribution settings. 500,000 instances were generated according to the uniform, Laplace, Gamma and normal probability distribution functions with mean values set at $\boldsymbol{\mu} = [45; 33; 48; 25; 27; 35; 41; 28; 37; 37]$ and deviation values as $\boldsymbol{\sigma} = [19; 16; 26; 25; 13; 20; 7; 5; 18; 26]$ (which are identical to that considered for the normal distribution case). The theoretical mean, standard deviation and RCVaR_α are referenced in the last line of the table. The best solutions, which are the benchmarks of these relative objective values, are highlighted in bold. We can see from

Table 7
Simulation results for different distributions (n=10, $\alpha = 0.95$).

	Mean of TFT			Standard deviation of TFT			CVaR_α of TFT		
Distribution	μTFT($\boldsymbol{X}^*$)	μTFT($\bar{\boldsymbol{X}}$)	RP	σTFT($\boldsymbol{X}^*$)	σTFT($\bar{\boldsymbol{X}}$)	RB	$\mathrm{CVaR}_\alpha(\boldsymbol{X}^*)$	$\mathrm{CVaR}_\alpha(\bar{\boldsymbol{X}})$	HV
Uniform	1848.29	1748.63	99.66	259.84	359.69	99.85	2375.88	2466.12	90.24
Normal	1851.77	1754.93	96.84	261.36	358.91	97.55	2385.39	2497.78	112.39
Laplace	1851.15	1751.63	99.52	260.28	360.31	100.03	2399.94	2540.95	141.01
Gamma	1847.01	1749.04	97.97	259.61	359.09	99.48	2434.43	2655.89	221.46
Theoretical	**1850.00**	**1752.00**	**98.00**	**261.00**	**359.69**	**98.69**	**2987.68**	**3319.86**	**332.18**

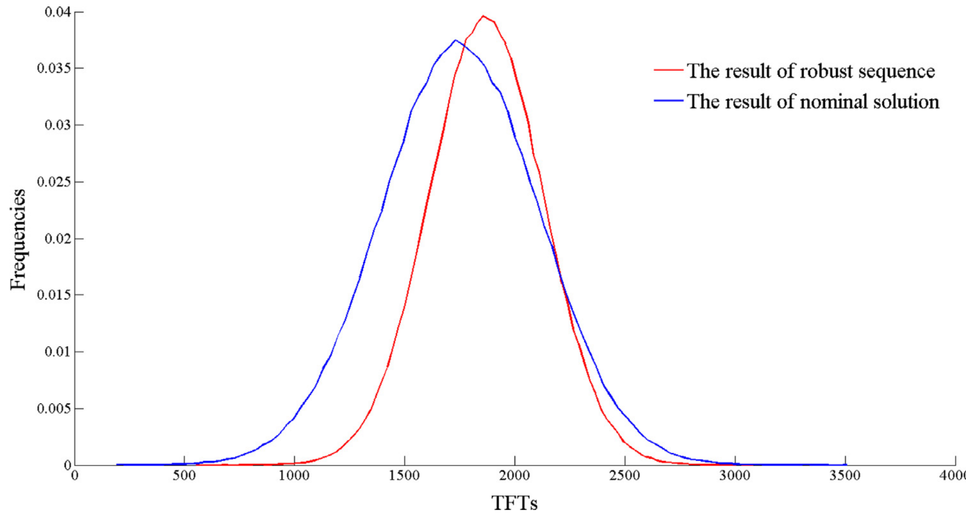

Fig. 2. Practical distributions of simulated TFTs (n=10,α = 0.99).

the table that previously discussed results, which were mainly concerned with the comparison of mean, deviation and CVaR values between nominal solutions and robust sequences, also hold under different parameter distribution assumptions. Furthermore, the means and standard deviations of different distributions are gathering around the theoretical values with minimal swings. Although the CVaR$_\alpha$s of different distributions are distinct due to the definition, all of them are less than the theoretical RCVaR$_\alpha$. These results have clearly demonstrated the distributional robustness of the proposed DR-SMSP model.

5. Conclusions

In this paper, we have studied the DR-SMSP aiming at minimizing robust CVaR of the TFT. The JPT is assumed to be a random vector subject to an ambiguous distribution with only the mean vector and covariance matrix specified. By giving an explicit expression of robust CVaR$_\alpha$, the DR-SMSP is decomposed into an AP and an I-SOCP problem. The AP is easily optimized via the SAPT rule, while I-SOCP is much more complicated.

To efficiently solve the I-SOCP problem, three novel Cauchy-relaxation algorithms (i.e., the CCRA, M-CCRA and PCRA) have been proposed to obtain a robust sequence with minimum RCVaR$_\alpha$. Our experimental results convincingly demonstrated the effectiveness and efficiency of these algorithms compared to a standard CPLEX solver. Through the experiments, the strength of each algorithm has also been identified: the PCRA possesses the best solutions, while the CCRA and M-CCRA provide relatively good solutions with less computational effort. Furthermore, we compared the generated robust sequence to the nominal solution of the deterministic model under different confidence level settings. It is shown that the proposed DR-SMSP model can achieve risk reduction with a sacrifice on the optimality of the mean value.

There are several issues that deserve further investigation. First, the proposed explicit expression of RCVaR$_\alpha$ can be extended into a finite interval support set case and applied to some other problems related to robust CVaR. Second, the design strategy for Cauchy-relaxation algorithms can be adapted to solve other optimization problems with similar characteristics, such as mixed integer second-order cone programming and mixed integer quadratic programming. Moreover, other performance measures, e.g., the weighted completion time, total tardiness and makespan, can be considered in the DR-SMSP model. Finally, the DR framework can be extended to other scheduling problems, such as parallel machine (Ding, Song, Zhang, & Chiong, 2016), job shop (Zhang & Chiong, 2016) and flow shop (Ding et al., 2015) scheduling problems.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under grants 61273233, 61473141 and 61503211.

Appendix: Proof of Theorem 1

Proof. Based on the equivalent calculated property of CVaR given by Rockafellar and Uryasev (2002), i.e.,

$$\mathrm{CVaR}_\alpha(\boldsymbol{Z}) = \inf\left\{k + (1-\alpha)^{-1}E[\boldsymbol{Z}-k]_+ :\ k \in \mathbb{R}\right\}, \tag{66}$$

we can rewrite RCVaR$_\alpha$ as

$$\mathrm{RCVaR}_\alpha\ (\boldsymbol{Z}) = \sup_{P^z\in\mathbb{P}^z} \inf\left\{k + (1-\alpha)^{-1}E[\boldsymbol{Z}-k]_+ :\ k \in \mathbb{R}\right\}. \tag{67}$$

Since $\mathrm{Sup}(\boldsymbol{Z}) = [0, \infty)$, the infimum above is taken at $k = 0$ when $k \le 0$. Thus, the range of $k \in \mathbb{R}$ can be reduced to $k \ge 0$ and RCVaR$_\alpha(\boldsymbol{Z})$ can be further rewritten as

$$\mathrm{RCVaR}_\alpha\ (\boldsymbol{Z}) = \sup_{P^z\in\mathbb{P}^z} \min_{k\ge 0}\left\{k + (1-\alpha)^{-1}E[\boldsymbol{Z}-k]_+\right\}. \tag{68}$$

Based on the min-max theorem proposed by Zhu and Fukushima (2009), the minimization and maximization can be exchanged as

$$\mathrm{RCVaR}_\alpha\ (\boldsymbol{Z}) = \min_{k\ge 0}\left\{k + (1-\alpha)^{-1} \sup_{P^z\in\mathbb{P}^z} E[\boldsymbol{Z}-k]_+\right\}. \tag{69}$$

Lo (1987) has obtained the upper bound of $E[\boldsymbol{Z}-k]_+$ in $\mathbb{P}^z$, that is,

$$\sup_{\boldsymbol{Z}\in\mathbb{P}^z} E[\boldsymbol{Z}-k]_+ = \begin{cases} \mu_z - k + \dfrac{k\sigma_z^2}{\sigma_z^2+\mu_z^2}, & \text{if } k \le \dfrac{\sigma_z^2+\mu_z^2}{2\mu_z} \\ \dfrac{\sqrt{\sigma_z^2+(\mu_z-k)^2}+\mu_z-k}{2}, & \text{if } k \ge \dfrac{\sigma_z^2+\mu_z^2}{2\mu_z}. \end{cases} \tag{70}$$

Let $C(k) = k + (1-\alpha)^{-1}\sup_{\mathbb{P}^z\in\mathbb{P}^z} E[\boldsymbol{Z}-k]_+$, according to this upper bound, $C(k)$ can be expressed as

$$C(k) = \begin{cases} k + (1-\alpha)^{-1}\left(\mu_z - k + \dfrac{k\sigma_z^2}{\sigma_z^2+\mu_z^2}\right), \\ \qquad\qquad \text{if } 0 \le k \le \dfrac{\sigma_z^2+\mu_z^2}{2\mu_z} \\ k + (1-\alpha)^{-1}\left(\dfrac{\sqrt{\sigma_z^2+(\mu_z-k)^2}+\mu_z-k}{2}\right), \\ \qquad\qquad \text{if } k \ge \dfrac{\sigma_z^2+\mu_z^2}{2\mu_z}. \end{cases} \tag{71}$$

We next solve $\min_{k\ge 0} C(k)$ in different cases, and then integrate the solutions to get the final representation of $\mathrm{RCVaR}_\alpha(\boldsymbol{Z})$.

Case 1 When $0 \le k \le \frac{\sigma_z^2+\mu_z^2}{2\mu_z}$, $C_1(k)$ is a monotonic linear function, i.e.,

$$C_1(k) = k + (1-\alpha)^{-1}\left(\mu_z - k + \frac{k\sigma_z^2}{\sigma_z^2+\mu_z^2}\right) = \left(1 - \frac{1}{1-\alpha}\cdot\frac{\mu_z^2}{\sigma_z^2+\mu_z^2}\right)k + \frac{\mu_z}{1-\alpha}. \tag{72}$$

1.1 When $C_1(k)$ is non-decreasing, i.e., $1 - \frac{1}{1-\alpha}\cdot\frac{\mu_z^2}{\sigma_z^2+\mu_z^2} \ge 0 \Leftrightarrow 0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2}$, the minimum of $C_1(k)$ is $C_1^{*1} = \frac{\mu_z}{1-\alpha}$.

1.2 When $C_1(k)$ is non-increasing, i.e., $1 - \frac{1}{1-\alpha}\cdot\frac{\mu_z^2}{\sigma_z^2+\mu_z^2} \le 0 \Leftrightarrow \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1$, the minimum of $C_1(k)$ is $C_1^{*2} = \frac{\sigma_z^2}{2\mu_z} + \frac{2-\alpha}{2(1-\alpha)}\mu_z$.

Case 2 When $k \ge \frac{\sigma_z^2+\mu_z^2}{2\mu_z}$, $C_2(k)$ is a convex function, i.e.,

$$C_2(k) = k + (1-\alpha)^{-1}\left(\frac{\sqrt{\sigma_z^2+(\mu_z-k)^2}+\mu_z-k}{2}\right). \tag{73}$$

Based on the first order optimal condition, the stationary point of $C_2(k)$ is $\bar{k} = \mu_z + \frac{(2\alpha-1)\sigma_z}{2\sqrt{\alpha-\alpha^2}}$.

2.1 If $\bar{k} \le \frac{\sigma_z^2+\mu_z^2}{2\mu_z}$, i.e., $0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2}$, the minimum of $C_2(k)$ is taken at $k = \frac{\sigma_z^2+\mu_z^2}{2\mu_z}$, that is, $C_2^{*1} = \frac{\sigma_z^2+\mu_z^2}{2\mu_z} + \frac{\mu_z}{2(1-\alpha)}$.

2.2 If $\bar{k} \ge \frac{\sigma_z^2+\mu_z^2}{2\mu_z}$, i.e., $\frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1$, the minimum of $C_2(k)$ is taken at $\bar{k}$, that is, $C_2^{*2} = \mu_z + \sqrt{\frac{\alpha}{1-\alpha}}\cdot\sqrt{\sigma_z^2}$.

Integrating the above cases, the optimum value of $\min_{k\ge 0} C(k)$ can be represented as follows:

$$C^* = \begin{cases} \min\left\{C_1^{*1}, C_2^{*1}\right\}, & \text{if } 0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \\ \min\left\{C_1^{*2}, C_2^{*2}\right\}, & \text{if } \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1. \end{cases} \tag{74}$$

By comparing the four values, we can get $C_1^{*1} \le C_2^{*1}$ for $0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2}$, and $C_1^{*2} \ge C_2^{*2}$ for $\frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1$. Then C^* can be rewritten as

$$C^* = \begin{cases} C_1^{*1} = \frac{\mu_z}{1-\alpha}, & \text{if } 0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \\ C_2^{*2} = \mu_z + \sqrt{\frac{\alpha}{1-\alpha}}\cdot\sqrt{\sigma_z^2}, & \text{if } \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1. \end{cases} \tag{75}$$

Therefore, the final representation of $\mathrm{RCVaR}_\alpha(\boldsymbol{Z})$ is

$$\mathrm{RCVaR}_\alpha(\boldsymbol{Z}) = \begin{cases} \frac{\mu_z}{1-\alpha}, & \text{if } 0 \le \alpha \le \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \\ \mu_z + \sqrt{\frac{\alpha}{1-\alpha}}\cdot\sqrt{\sigma_z^2}, & \text{if } \frac{\sigma_z^2}{\sigma_z^2+\mu_z^2} \le \alpha \le 1. \end{cases} \tag{76}$$

The proof is completed. □

References

Agrawala, A., Coffman, E., Garey, M., & Tripathi, S. (1984). A stochastic optimization algorithm minimizing expected flow times on uniform processors. *IEEE Transactions on Computers, 33*(4), 351–356.

Aloulou, M. A., & Della Croce, F. (2008). Complexity of single machine scheduling problems under scenario-based uncertainty. *Operations Research Letters, 36*(3), 338–342.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

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

De, P., Ghosh, J. B., & Wells, C. E. (1992). Expectation-variance analysis of job sequences under processing time uncertainty. *International Journal of Production Economics, 28*(3), 289–297.

de Farias, I. R., Zhao, H., & Zhao, M. (2010). A family of inequalities valid for the robust single machine scheduling polyhedron. *Computers & Operations Research, 37*(9), 1610–1614.

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

Ding, J.-Y., Song, S., Gupta, J. N., Zhang, R., Chiong, R., & Wu, C. (2015). An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing, 30*, 604–613.

Ding, J. Y., Song, S., Zhang, R., & Chiong, R. (2016). Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Transactions on Automation Science & Engineering, 13*(2), 1138–1154.

Doan, X. V., Kruk, S., & Wolkowicz, H. (2012). A robust algorithm for semidefinite programming. *Optimization Methods and Software, 27*(4–5), 667–693.

Doan, X. V., Li, X., & Natarajan, K. (2015). Robustness to dependency in portfolio optimization using overlapping marginals. *Operations Research, 63*(6), 1468–1488.

Dupačová, J. (1987). The minimax approach to stochastic programming and an illustrative application. *Stochastics: An International Journal of Probability and Stochastic Processes, 20*(1), 73–88.

Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research, 235*(3), 471–483.

Huang, G., Zhang, J., Song, S., & Chen, Z. (2015). Maximin separation probability clustering. In *AAAI conference on artificial intelligence*.

Jang, W. (2002). Dynamic scheduling of stochastic jobs on a single machine. *European Journal of Operational Research, 138*(3), 518–530.

Jang, W., & Klein, C. M. (2002). Minimizing the expected number of tardy jobs when processing times are normally distributed. *Operations Research Letters, 30*(2), 100–106.

Jia, C. (2001). Stochastic single machine scheduling with an exponentially distributed due date. *Operations Research Letters, 28*(5), 199–203.

Jiang, R., Shen, S., & Zhang, Y. (2015). *Integer programming approaches for appointment scheduling with random no-shows and service durations*. Social Science Electronic Publishing.

Kan, A. R. (2012). *Machine scheduling problems: Classification, complexity and computations*. Springer Science & Business Media.

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

Kasperski, A., & Zieliński, P. (2006). An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Information Processing Letters, 97*(5), 177–180.

Köksalan, M., & Kondakci, S. K. (1997). Multiple criteria scheduling on a single machine: A review and a general approach. In *Essays in decision making* (pp. 331–341). Springer.

Koulamas, C. (2010). The single-machine total tardiness scheduling problem: Review and extensions. *European Journal of Operational Research, 202*(1), 1–7.

Kouvelis, P., & Yu, G. (2013). *Robust discrete optimization and its applications*: vol.14. Springer Science & Business Media.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*(1–2), 83–97.

Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., & Jordan, M. I. (2003). A robust minimax approach to classification. *The Journal of Machine Learning Research, 3*, 555–582.

Lebedev, V., & Averbakh, I. (2006). Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discrete Applied Mathematics, 154*(15), 2167–2177.

Liao, S., Van Delft, C., & Vial, J.-P. (2013). Distributionally robust workforce scheduling in call centres with uncertain arrival rates. *Optimization Methods and Software, 28*(3), 501–522.

Lo, A. W. (1987). Semi-parametric upper bounds for option prices and expected payoffs. *Journal of Financial Economics, 19*(2), 373–387.

Lu, C.-C., Lin, S.-W., & Ying, K.-C. (2012). Robust scheduling on a single machine to minimize total flow time. *Computers & Operations Research, 39*(7), 1682–1691.

Lu, C.-C., Lin, S.-W., & Ying, K.-C. (2014). Minimizing worst-case regret of makespan on a single machine with uncertain processing and setup times. *Applied Soft Computing, 23*, 144–151.

Lu, C.-C., Ying, K.-C., & Lin, S.-W. (2014). Robust single machine scheduling for minimizing total flow time in the presence of uncertain processing times. *Computers & Industrial Engineering, 74*, 102–110.

McKay, K. N., Safayeni, F. R., & Buzacott, J. A. (1988). Job-shop scheduling theory: What is relevant? *Interfaces, 18*(4), 84–90.

Montemanni, R. (2007). A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data. *Journal of Mathematical Modelling and Algorithms, 6*(2), 287–296.

Nguyen, T.-D., & Lo, A. W. (2012). Robust ranking and portfolio optimization. *European Journal of Operational Research, 221*(2), 407–416.

Pinedo, M. L. (2012). *Scheduling: Theory, algorithms, and systems*. Springer Science & Business Media.

Popescu, I. (2007). Robust mean-covariance solutions for stochastic optimization. *Operations Research, 55*(1), 98–112.

Quan, O. Y., & Xu, H. Y. (2013). The review of the single machine scheduling problem and its solving methods. In *Applied mechanics and materials: 411* (pp. 2081–2084). Trans Tech Publications.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk, 2*, 21–42.

Rockafellar, R. T., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance, 26*(7), 1443–1471.

Scarf, H., Arrow, K., & Karlin, S. (1958). A min–max solution of an inventory problem. *Studies in the Mathematical Theory of Inventory and Production, 10*, 201–209.

Seo, D. K., Klein, C. M., & Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers & Industrial Engineering, 48*(2), 153–161.

Shapiro, A. (2006). Worst-case distribution analysis of stochastic programs. *Mathematical Programming, 107*(1–2), 91–96.

Skutella, M., & Uetz, M. (2005). Stochastic machine scheduling with precedence constraints. *SIAM Journal on Computing, 34*(4), 788–802.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly, 3*(1–2), 59–66.

Soroush, H. (2007). Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem. *European Journal of Operational Research, 181*(1), 266–287.

Sun, H., Gao, Z., Szeto, W., Long, J., & Zhao, F. (2014). A distributionally robust joint chance constrained optimization model for the dynamic network design problem under demand uncertainty. *Networks and Spatial Economics, 14*(3–4), 409–433.

Tadayon, B., & Smith, J. C. (2015). Algorithms and complexity analysis for robust single-machine scheduling problems. *Journal of Scheduling, 18*(6), 575–592.

Trietsch, D., & Baker, K. R. (2008). Minimizing the number of tardy jobs with stochastically-ordered processing times. *Journal of Scheduling, 11*(1), 71–73.

van den Akker, M., & Hoogeveen, H. (2008). Minimizing the number of late jobs in a stochastic setting using a chance constraint. *Journal of Scheduling, 11*(1), 59–69.

Wu, X., & Zhou, X. (2008). Stochastic scheduling to minimize expected maximum lateness. *European Journal of Operational Research, 190*(1), 103–115.

Yang, J., & Yu, G. (2002). On the robust single machine scheduling problem. *Journal of Combinatorial Optimization, 6*(1), 17–33.

Zhang, R., & Chiong, R. (2016). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production, 112*, 3361–3375.

Zhu, S., & Fukushima, M. (2009). Worst-case conditional value-at-risk with application to robust portfolio management. *Operations Research, 57*(5), 1155–1168.