Tightening CVaR Approximations via Scenario-Wise Scaling for Chance-Constrained Programming

Nan Jiang[1], Rui Chen[2]

[1] The Hong Kong University of Science and Technology (nanjiang@ust.hk)

[2] The Chinese University of Hong Kong, Shenzhen (rchen@cuhk.edu.cn)

March 30, 2026

Abstract

Chance-constrained programs (CCPs) provide a powerful modeling framework for decision-making under uncertainty, but their nonconvex feasible regions make them computationally challenging. A widely used convex inner approximation replaces chance constraints with Conditional Value-at-Risk (CVaR) constraints; however, the resulting solutions can be overly conservative and suboptimal. We propose a scenario-wise scaling approach that strengthens CVaR approximations for CCPs with finitely supported uncertainty. The method introduces scaling factors that reweight individual scenarios within the CVaR constraint, yielding a family of potentially tighter inner approximations. We establish sufficient conditions under which, for a suitable choice of scaling factors, the scaled CVaR approximation attains the same optimal value as the original CCP and admits a (near-)optimal solution of the CCP. We show that these conditions are tight and further relax them in the convex setting. We also show that optimizing over scenario-wise scaling factors is NP-hard. To address this computational challenge, we develop efficient heuristic and sequential convex approximation algorithms that iteratively update the scaling factors and generate improved feasible solutions. Numerical experiments demonstrate that the proposed methods consistently improve upon standard CVaR and state-of-the-art convex approximations, often reducing conservativeness while maintaining tractability.

1 Introduction

Chance-constrained programs (CCPs) provide a principled way to optimize decisions under uncertainty by explicitly controlling the probability of constraint violation. Specifically, a CCP chooses a decision $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to minimize (without loss of generality) a linear cost subject to the requirement that some uncertain constraint holds with high probability:

$$v^* = \min_{\boldsymbol{x} \in \mathcal{X}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \mathbb{P}\left\{ \tilde{\boldsymbol{\xi}} \colon g(\boldsymbol{x}, \tilde{\boldsymbol{\xi}}) \le 0 \right\} \ge 1 - \varepsilon \right\}, \tag{1}$$

where $\tilde{\boldsymbol{\xi}}$ is a random vector, $g(\cdot,\cdot)$ models the uncertain feasibility constraint, and $\varepsilon \in (0,1)$ is a risk tolerance. Throughout this paper, we assume that CCP (1) is feasible, and the uncertainty is finitely supported, as specified below.

Assumption 1. *The underlying probability distribution of $\tilde{\boldsymbol{\xi}}$ is finite, i.e. $\tilde{\boldsymbol{\xi}}$ has N possible realizations (i.e., scenarios) $\{\boldsymbol{\xi}^1, \boldsymbol{\xi}^2, \cdots, \boldsymbol{\xi}^N\}$ with $p_i = \mathbb{P}\{\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}^i\}$ for $i \in [N]$, and $\sum_{i \in [N]} p_i = 1$.*

This assumption is common in the CCP literature (see, e.g., [28, 24, 33, 3]), as one can often approximate a general distribution using a finite sample. With the finitely supported distribution $\mathbb{P}$, we can equivalently rewrite CCP (1) as

$$v^* = \min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \sum_{i\in[N]} p_i \mathbb{I}\left[g(\boldsymbol{x}, \boldsymbol{\xi}^i) \le 0\right] \ge 1-\varepsilon \right\}, \tag{2}$$

where $\mathbb{I}[\cdot]$ is the zero-one indicator function.

Without loss of generality, we assume that the function $g(\boldsymbol{x}, \boldsymbol{\xi})$ is defined as $g(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{j\in[J]} g_j(\boldsymbol{x}, \boldsymbol{\xi})$, where $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \colon \mathbb{R}^n \times \Xi \to \mathbb{R}$ for all $j \in [J] := \{1, \ldots, J\}$. Therefore, formulation (2) can model both single and joint chance-constrained programs. When $J = 1$, CCP (2) includes a single chance constraint; otherwise, it contains a joint chance constraint where $g(\boldsymbol{x}, \boldsymbol{\xi}^i) \le 0$ if and only if $g_j(\boldsymbol{x}, \boldsymbol{\xi}^i) \le 0$ for each $j \in [J]$.

1.1 Motivation

Despite its modeling power, CCP (2) is computationally challenging: even with linear $g(\cdot, \boldsymbol{\xi})$ and $\mathcal{X} = \mathbb{R}^n_+$, the problem is NP-hard [28]. As a result, a large body of work develops convex approximations of (2). A convenient way to express the chance constraint is through the violation probability. Indeed, CCP (2) is equivalent to

$$v^* = \min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} : \sum_{i\in[N]} p_i \, \mathbb{I}\big[g(\boldsymbol{x}, \boldsymbol{\xi}^i) > 0\big] \le \varepsilon \right\}, \tag{3}$$

where we define the (nonconvex) violation probability function

$$\bar{p}(\boldsymbol{x}) := \sum_{i\in[N]} p_i \, \mathbb{I}\big[g(\boldsymbol{x}, \boldsymbol{\xi}^i) > 0\big] \,.$$

One particular approach to derive conservative inner approximations of (3) is to replace $\bar{p}(\boldsymbol{x})$ with a convex approximation $\widehat{p}(\boldsymbol{x})$ satisfying $\widehat{p}(\boldsymbol{x}) \ge \bar{p}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. Among convex approximations obtained by upper-bounding the indicator function via a convex function, the Conditional Value-at-Risk (CVaR) approximation (i.e., $\widehat{p}(\boldsymbol{x}) = \inf_{\beta<0} (-\beta)^{-1}\{\sum_{i\in[N]} p_i[g(\boldsymbol{x}, \boldsymbol{\xi}^i) - \beta]_+\}$) is known to be the tightest [30]. Despite its tightness, the CVaR approximation differs significantly from the original CCP in many aspects. One particular property of the original CCP that is lost in the CVaR approximation is the invariance with respect to scaling of each scenario constraint.

Specifically, for any fixed $\boldsymbol{\alpha} \in \mathbb{R}^N_{++}$, define $h_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{\xi}^i) := \alpha_i g(\boldsymbol{x}, \boldsymbol{\xi}^i)$ for $i \in [N]$. Then constraints

$$g(\boldsymbol{x}, \boldsymbol{\xi}^i) \le 0 \Leftrightarrow h_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{\xi}^i) \le 0$$

are equivalent scenario by scenario. Therefore, the chance constraint itself is unchanged by positive scaling of each scenario constraint. In contrast, convex approximations that aggregate scenario functions, such as the CVaR approximation, depend on the magnitudes of scenario constraints. Therefore, changing the scale of individual scenarios can change the strength of the approximation even though the underlying CCP is identical. This issue is particularly visible in CCPs with a covering structure, which appear in numerous applications where certain demands have to be met with high probability (see, e.g., [13, 14, 37, 40, 41]). A common special case has $\mathcal{X} = \mathbb{R}^n_+$, $\boldsymbol{c} \in \mathbb{R}^n_+$, and $g(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{j\in[J]} g_j(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{j\in[J]} b^i_j - (\boldsymbol{A}^i_j)^\top \boldsymbol{x}$ with $\boldsymbol{A}^i_j \in \mathbb{R}^n_+$, $b^i_j > 0$ for

$i \in [N]$ and $j \in [J]$. A standard preprocessing step [33, 45] is row-wise normalization, i.e., dividing each row $b_j^i - (\boldsymbol{A}_j^i)^\top \boldsymbol{x}$ by the demand b_j^i so that each normalized constraint has a unit constant term. This normalization preserves the feasible set of the original CCP, but it can change the optimal solutions of conservative inner approximations like CVaR. More generally, different choices of the scaling vector $\boldsymbol{\alpha}$ in the CVaR approximation may lead to different optimal solutions and optimal objective values. A suitable choice of $\boldsymbol{\alpha}$ potentially reduces conservativeness and improves the optimal objective value of the approximation (see Example 2 in Appendix A for a concrete example).

1.2 Literature Review

Initially introduced by [7, 6], CCPs have been widely used to handle uncertainty in optimization problems. For example, in the energy sector, CCPs can be employed to address the challenges posed by various sources of uncertainty. Uncertainty may come from the intermittent nature of renewable energy sources, fluctuations in energy demand, potential transmission line failures, and other stochastic factors that can impact the overall performance and reliability of the energy systems (see, e.g., [11, 32]). In finance, CCPs are widely used in risk management, specifically through the optimization of investment portfolios under uncertain market conditions. By incorporating market volatility and other stochastic factors into financial models, CCPs enable financial analysts and portfolio managers to devise strategies that balance potential returns with acceptable levels of risk (see, e.g., [31, 12]). In logistics, CCPs facilitate the effective planning and operation of supply chains by accounting for the variability in demand and uncertainties in transportation. These models allow for optimizing inventory levels, routing, and scheduling, ensuring supply chain operations remain efficient and resilient against disruptions (see, e.g., [15, 16]). Interested readers are referred to [2, 25] for a comprehensive review.

Despite its significance, the feasible region of CCP (1) is generally nonconvex, making the problem challenging to solve directly. To address this, several approaches have been proposed in the literature to solve CCP (1). One approach is the sample average approximation (SAA) method proposed by [35, 27], which reformulates CCP (1) as a mixed-integer program. Standard optimization solvers can then handle this reformulation; however, solving it to optimality may still be computationally challenging in practice. Another approach is to develop convex inner approximations of the nonconvex chance constraint (see, e.g., [29, 5, 30, 1]). The best-known convex approximation is to replace the chance constraint in CCP (1) with the conditional value-at-risk (CVaR) approximation proposed by [30]. The CVaR approximation usually returns a feasible yet sub-optimal solution. Recently, bisection-based methods [1, 21, 22] like ALSO-X and ALSO-X# are proposed to improve the CVaR approximation. This improvement occurs because ALSO-X refines the solution iteratively, allowing for a more effective optimization process compared to the one-shot CVaR approximation.

In this paper, we propose to use scenario-wise scaling to improve the CVaR approximation. Scaling the constraints is crucial for ensuring the effectiveness and efficiency of the optimization process (see, e.g., [26]) and is important for improving the performance of LP/MIP solvers (see, e.g., [4] and the references therein). Scaling within the CVaR approximation of chance-constrained programs has appeared in the literature, particularly for achieving a better approximation for joint chance constraints (see, e.g., [8, 47, 46, 9]). [8] use scaling to approximate joint chance-constrained problems, improving upon the standard approach using Bonferroni's inequality. [47] show that CVaR approximation can be exact with scaling for distributional robust joint chance-constrained problems with moment ambiguity sets. However, existing works focus on scaling of individual constraints in joint chance-constrained settings and do not consider scaling applied to individual scenarios. As a result, the conditions under which the scenario-wise scaling within the CVaR approximation of a CCP is exact, and how it relates to the optimal objective value of a CCP, have remained unclear. This work fills this gap. In particular, we identify sufficient conditions under which the scaled CVaR approximation preserves an optimal solution of a CCP, and we further establish NP-hardness results for the scaled CVaR approximation. We also develop efficient algorithms to improve CVaR approximation using scenario-wise scaling, and our numerical study demonstrates the effectiveness of the proposed algorithms.

Organization. The remainder of the paper is organized as follows: Section 2 details the scaling procedure and illustrates the advantages of scaling. Section 3 introduces an efficient heuristic. Section 4 numerically validates the proposed methods. Section 5 concludes the paper.

Notation. The following notation is used throughout the paper. We use bold letters (e.g., $\boldsymbol{x}, \boldsymbol{A}$) to denote vectors and matrices and use corresponding non-bold letters to denote their components. We use $\mathbb{R}_+(/\mathbb{Z}_+)$ to denote the set of nonnegative real(/integer) numbers, and $\mathbb{R}_{++}$ to denote the set of positive numbers, i.e., $\mathbb{R}_+ := [0,\infty)$ and $\mathbb{R}_{++} := (0,\infty)$. We let $\boldsymbol{e}$ be the vector of all ones. Given an integer n, we let $[n] := \{1,2,\ldots,n\}$. Given a real number t, we let $(t)_+ := \max\{t, 0\}$. Given a finite set I, we let $|I|$ denote its cardinality. We let $\tilde{\boldsymbol{\xi}}$ denote a random vector and denote its realizations by $\boldsymbol{\xi}$.

2 Scenario-Wise Scaled CVaR Approximation

For a given random variable $\tilde{\boldsymbol{X}}$ with probability distribution $\mathbb{P}$ and cumulative distribution function $F_{\tilde{\boldsymbol{X}}}(s) = \mathbb{P}\{\tilde{\boldsymbol{X}} \le s\}$, and a given risk parameter $\varepsilon \in (0,1)$, $(1-\varepsilon)$ Value-at-Risk (VaR) of $\tilde{\boldsymbol{X}}$ is defined as $\mathrm{VaR}_{1-\varepsilon}(\tilde{\boldsymbol{X}}) := \min_s\{s : F_{\tilde{\boldsymbol{X}}}(s) \ge 1-\varepsilon\}$. Following the definition of VaR,

$$\mathbb{P}\left\{\tilde{\boldsymbol{X}} \le 0\right\} \ge 1-\varepsilon \Leftrightarrow \mathrm{VaR}_{1-\varepsilon}(\tilde{\boldsymbol{X}}) \le 0.$$

The corresponding Conditional Value-at-Risk (CVaR) is defined by

$$\mathrm{CVaR}_{1-\varepsilon}(\tilde{\boldsymbol{X}}) := \min_{\beta}\{\beta + \varepsilon^{-1}\mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{X}} - \beta]_+\},$$

which serves as an upper bound of $\mathrm{VaR}_{1-\epsilon}(\tilde{\boldsymbol{X}})$. Note that

$$\min_{\beta}\{\beta + \varepsilon^{-1}\mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{X}} - \beta]_+\} \le 0 \Leftrightarrow \min_{\beta \le 0}\{\beta + \varepsilon^{-1}\mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{X}} - \beta]_+\} \le 0.$$

Applying the above arguments to CCP (2), one can then obtain a CVaR (inner) approximation [30, 34] of CCP (2):

$$v^{\mathrm{CVaR}} = \min_{\boldsymbol{x}\in\mathcal{X}}\left\{\boldsymbol{c}^\top\boldsymbol{x} \colon \min_{\beta\le 0}\left\{\beta + \frac{1}{\varepsilon}\sum_{i\in[N]} p_i\left[\{g(\boldsymbol{x},\boldsymbol{\xi}^i) - \beta\}_+\right]\right\} \le 0\right\}. \tag{4}$$

Introducing nonnegative auxiliary variables $\boldsymbol{s}$, the CVaR approximation (4) can be reformulated as

$$v^{\mathrm{CVaR}} = \min_{\boldsymbol{x}\in\mathcal{X},\beta\le 0,\boldsymbol{s}\ge\mathbf{0}}\left\{\boldsymbol{c}^\top\boldsymbol{x} \colon \varepsilon\beta + \sum_{i\in[N]} p_i s_i \le 0, s_i + \beta \ge g(\boldsymbol{x},\boldsymbol{\xi}^i),\ i\in[N]\right\}. \tag{5}$$

As discussed in Section 1.1, for each constraint $g(\boldsymbol{x}, \boldsymbol{\xi}^i)$ with $i \in [N]$, we introduce a corresponding scaling variable α_i for scenario-wise scaling. For fixed scaling factors $\boldsymbol{\alpha} \geq \boldsymbol{e}$, we define an $\boldsymbol{\alpha}$-*scaled* CVaR approximation:

$$v^{\mathrm{CVaR}}(\boldsymbol{\alpha}) := \min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N] \right\}, \tag{6}$$

and we denote the *optimally scaled* CVaR approximation by

$$\begin{aligned} v_S^{\mathrm{CVaR}} &:= \inf_{\boldsymbol{\alpha} \geq \boldsymbol{e}} v^{\mathrm{CVaR}}(\boldsymbol{\alpha}) \\ &= \inf_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{\alpha} \geq \boldsymbol{e}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N] \right\}. \end{aligned} \tag{7}$$

To the best of our knowledge, the scaled CVaR approximation formulation of the form (6) is introduced in the literature for the first time. While [8] refers to a notion of "scaling" within the CVaR approximation, it focuses on scaling each individual constraint in the joint chance constrained setting, rather than individual scenarios. In our setting, if we set $\alpha_i = 1$ for each $i \in [N]$ in (6), we recover the original CVaR approximation (4), i.e., $v^{\mathrm{CVaR}}(\boldsymbol{e}) = v^{\mathrm{CVaR}}$. We note that (6) is precisely the CVaR approximation of a scaled version of CCP (2), i.e., the CVaR approximation of $v^* = \min_{\boldsymbol{x} \in \mathcal{X}} \{\boldsymbol{c}^\top \boldsymbol{x} \colon \sum_{i \in [N]} p_i \mathbb{I}\left[\alpha_i g(\boldsymbol{x}, \boldsymbol{\xi}^i) \leq 0\right] \geq 1 - \varepsilon\}$. It then follows that (7) is a conservative approximation of the CCP (2), which implies the following result.

Proposition 1. *We have $v^* \leq v^{\mathrm{CVaR}}(\boldsymbol{\alpha})$ for all $\boldsymbol{\alpha} \geq \boldsymbol{e}$, and therefore $v^* \leq v_S^{\mathrm{CVaR}} \leq v^{\mathrm{CVaR}}$.*

Note that if $v^* = v_S^{\mathrm{CVaR}}$ and there exists $\widehat{\boldsymbol{\alpha}} \geq \boldsymbol{e}$ such that $v_S^{\mathrm{CVaR}} = v^{\mathrm{CVaR}}(\widehat{\boldsymbol{\alpha}})$, then any solution of (6) with $\boldsymbol{\alpha} = \widehat{\boldsymbol{\alpha}}$ is an optimal solution of the original CCP (2) due to the conservativeness of scaled CVaR approximations. We next identify sufficient conditions such that the optimally scaled CVaR approximation (7) provides an exact solution to CCP (2).

Theorem 1 (Exact case)**.** *Let $\boldsymbol{x}^*$ denote an optimal solution of CCP (2) and define $I^* := \{i \in [N] : g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0\}$. Suppose that $\sum_{i \in I^*} p_i > 1 - \varepsilon$. Then, we have $v^* = v_S^{\mathrm{CVaR}}$, and there exists $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta}, \widehat{\boldsymbol{s}})$ such that $v_S^{\mathrm{CVaR}} = v^{\mathrm{CVaR}}(\widehat{\boldsymbol{\alpha}})$ and $(\boldsymbol{x}^*, \widehat{\beta}, \widehat{\boldsymbol{s}})$ minimizes the scaled CVaR approximation (7).*

Proof. Due to Proposition 1, we only need to show that $v_S^{\mathrm{CVaR}} \leq v^*$. Under the conditions that $g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0$ for all $i \in I^*$ and $\sum_{i \in I^*} p_i > 1 - \varepsilon$, we use $\boldsymbol{x}^*$ to construct a feasible solution to the scaled CVaR approximation (7). Define $\tau := \sum_{i \in [N] \setminus I^*} p_i \in [0, \varepsilon)$ and $\bar{\alpha} := \sum_{i \in [N] \setminus I^*} p_i g(\boldsymbol{x}^*, \boldsymbol{\xi}^i)/(\varepsilon - \tau)$. The construction is as follows:

$$\begin{aligned} &\widehat{\alpha}_i = \max\left\{-\frac{\bar{\alpha}}{g(\boldsymbol{x}^*, \boldsymbol{\xi}^i)}, 1\right\},\ i \in I^*;\ \widehat{\alpha}_i = 1,\ i \in [N] \setminus I^*; \\ &\widehat{\beta} = -\bar{\alpha}; \\ &\widehat{s}_i = 0,\ i \in I^*;\ \widehat{s}_i = g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) + \bar{\alpha},\ i \in [N] \setminus I^*. \end{aligned}$$

Then it is easy to verify that $\widehat{s}_i + \widehat{\beta} \geq \widehat{\alpha}_i g(\boldsymbol{x}^*, \boldsymbol{\xi}^i)$ for each $i \in [N]$ and

$$\varepsilon\widehat{\beta} + \sum_{i \in [N]} p_i \widehat{s}_i = -\varepsilon\bar{\alpha} + \sum_{i \in [N] \setminus I^*} p_i [g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) + \bar{\alpha}] = -(\varepsilon - \tau)\bar{\alpha} + \sum_{i \in [N] \setminus I^*} p_i g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) \leq 0.$$

Therefore, $(\boldsymbol{x}^*, \widehat{\beta}, \widehat{\boldsymbol{s}}, \widehat{\boldsymbol{\alpha}})$ satisfies all constraints in the scaled CVaR approximation (7), which implies $v^{\mathrm{CVaR}}(\widehat{\boldsymbol{\alpha}}) \leq \boldsymbol{c}^\top \boldsymbol{x}^* = v^*$. This completes the proof. □

We make the following remarks about Theorem 1:

(i) Example 3 in Appendix A illustrates the correctness of Theorem 1. In fact, the two conditions of Theorem 1 (i.e., $g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0$ for all $i \in I^*$ and $\sum_{i\in I^*} p_i > 1 - \varepsilon$) are essentially tight for guaranteeing the equivalence between the optimal objective values of the scaled CVaR approximation and its exact CCP counterpart. If either of these two conditions in Theorem 1 is not satisfied, Example 4 and Example 5 in Appendix A illustrate that the scaled CVaR approximation (7) may not provide an optimal solution;

(ii) Theorem 1 extends to distributionally robust chance constrained programs under the type-∞ Wasserstein ambiguity set, since the corresponding formulations are equivalent to CCPs of the form (2) (see, e.g., Proposition 8 in [21]). The extension of Theorem 1 to type-∞ Wasserstein distributionally robust chance constrained programs also suggests a conceptual connection to adversarial robustness. In particular, type-∞ Wasserstein distributionally robust optimization has been shown to be closely related to adversarial training formulations in machine learning (see, e.g., 38). Exploring whether the scaling ideas developed in this paper can be adapted to adversarial training remains an interesting direction for future research;

(iii) The condition $g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0$ for all $i \in I^*$ can be viewed as a special case of the setting in [17], where the authors studied a strict chance constraint that requires strict satisfaction of the constraint $g(\boldsymbol{x}, \boldsymbol{\xi}^i) < 0$ for all $i \in [N]$. Nevertheless, in the convex case, we will relax this condition. In the purely discrete case, we can always slightly perturb the constraint such that, for all $x \in \mathcal{X}$, $g(\boldsymbol{x}, \boldsymbol{\xi}^i) \leq 0$ if and only if $g(\boldsymbol{x}, \boldsymbol{\xi}^i) < 0$;

(iv) We remark that the second condition is not restrictive at all. For example, in chance constraints with equiprobable scenarios (i.e., $p_i = 1/N$ for each $i \in [N]$), the condition $\sum_{i\in I^*} p_i > 1 - \varepsilon$ can accommodate cases where $N\varepsilon$ is not an integer. Otherwise, since the distribution is finite, one can always perturb ε to satisfy the condition $\sum_{i\in I^*} p_i > 1 - \varepsilon$ if $g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0$ for all $i \in I^*$.

To provide further insight into Theorem 1, we consider the following example.

Example 1. *Let us consider the following example with $N = 6$, $p_i = 1/6$ for each $i \in [N]$, and $\varepsilon = 5/12$. For a given optimal solution $\boldsymbol{x}^*$, suppose that the scenarios are ordered such that $g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_1}) \leq g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_2}) \leq g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_3}) \leq g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_4}) < 0$ and $0 < g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_5}) \leq g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_6})$. By Theorem 1, it follows that $I^* = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ and $\sum_{i\in I^*} p_i = 2/3 > 1 - \varepsilon$. This example satisfies the conditions in Theorem 1, which implies that $v^* = v_S^{\mathrm{CVaR}}$. As illustrated in Figure* **??***, in this example,* $\mathrm{CVaR}_{1-\varepsilon}[g(\boldsymbol{x}^*, \boldsymbol{\xi})]$ *can be expressed as*

$$\mathrm{CVaR}_{1-\varepsilon}[g(\boldsymbol{x}^*, \boldsymbol{\xi})] = \frac{1}{5/12}\left[\frac{1}{12} g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_4}) + \frac{1}{6} g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_5}) + \frac{1}{6} g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_6})\right].$$

Clearly, $\sum_{i\in I^*} p_i > 1 - \varepsilon$ with $I^* = \{i \in [N] : g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0\}$ guarantees the scenario-wise scaled CVaR approximation to recover the optimal objective value (by choosing α_{σ_4} large enough). If we instead perturb the risk parameter to $\varepsilon = 1/3$, then in this example we obtain $\sum_{i\in I^*} p_i = 1 - \varepsilon$. In this case, the corresponding CVaR reduces to $\mathrm{CVaR}_{1-\varepsilon}[g(\boldsymbol{x}^*, \boldsymbol{\xi})] = 1/2[g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_5}) + g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_6})] > 0$. *Even if the scenarios $\boldsymbol{\xi}^{\sigma_5}, \boldsymbol{\xi}^{\sigma_6}$ are scaled with factors $\alpha_{\sigma_5} \geq 1$ and $\alpha_{\sigma_6} \geq 1$, respectively, it is impossible to satisfy*

$$\mathrm{CVaR}_{1-\varepsilon}[h_{\boldsymbol{\alpha}}(\boldsymbol{x}^*, \boldsymbol{\xi})] = 1/2[\alpha_{\sigma_5} g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_5}) + \alpha_{\sigma_6} g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_6})] \leq 0.$$

Consequently, the scenario-wise CVaR approximation cannot recover the optimal objective value in this setting with $\sum_{i\in I^*} p_i = 1 - \varepsilon$. ◇

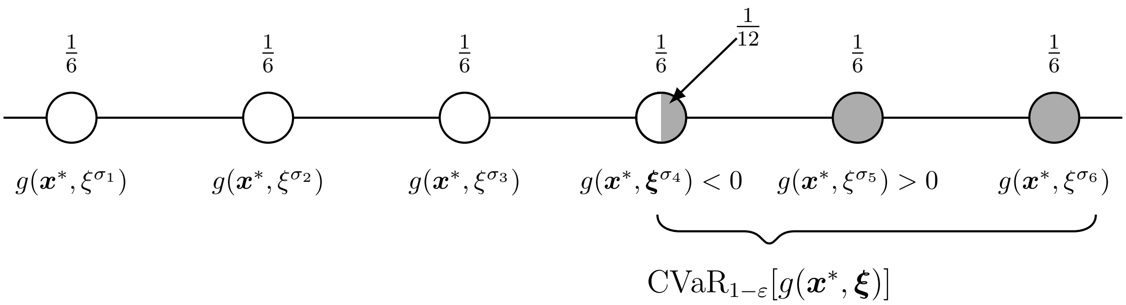
$\frac{1}{6}$
$\frac{1}{6}$
$\frac{1}{6}$
$\frac{1}{6}$
$\frac{1}{12}$
$\frac{1}{6}$
$\frac{1}{6}$
$g(\boldsymbol{x}^*, \xi^{\sigma_1})$
$g(\boldsymbol{x}^*, \xi^{\sigma_2})$
$g(\boldsymbol{x}^*, \xi^{\sigma_3})$
$g(\boldsymbol{x}^*, \boldsymbol{\xi}^{\sigma_4}) < 0$
$g(\boldsymbol{x}^*, \xi^{\sigma_5}) > 0$
$g(\boldsymbol{x}^*, \xi^{\sigma_6})$
$\text{CVaR}_{1-\varepsilon}[g(\boldsymbol{x}^*, \boldsymbol{\xi})]$

Figure 1: An Illustration of Conditions in Theorem 1 with $N = 6$ and $\varepsilon = 5/12$, adapted from Figure 2 of [36].

The condition $\sum_{i \in I^*} p_i > 1 - \varepsilon$ in Theorem 1 may appear restrictive. However, it can be significantly relaxed when all constraints, including the deterministic ones (i.e., constraints in $\boldsymbol{x} \in \mathcal{X}$) and the realization of the uncertain constraint in each scenario (i.e., the function $g(\boldsymbol{x}, \cdot)$), are convex.

Theorem 2 (Convex case)**.** *Let $\boldsymbol{x}^*$ denote an optimal solution of CCP* (2)*, and define $\bar{I}^* := \{i \in [N] : g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) \leq 0\}$ and $\bar{\mathcal{X}}^* = \{\boldsymbol{x} \in \mathcal{X} : g(\boldsymbol{x}, \boldsymbol{\xi}^i) \leq 0,\ i \in \bar{I}^*\}$. Then in the convex case (i.e., $g(\cdot, \boldsymbol{\xi}^i)$ is convex for all $i \in \bar{I}^*$, and set $\mathcal{X}$ is convex), $v^* = v_S^{\text{CVaR}}$ if there exists $\bar{I} \subseteq \bar{I}^*$ such that (i) for each $i \in \bar{I}$, there exists $\boldsymbol{x}^i \in \bar{\mathcal{X}}^*$ such that $g(\boldsymbol{x}^i, \boldsymbol{\xi}^i) < 0$; and (ii) $\sum_{i \in \bar{I}} p_i > 1 - \varepsilon$.*

Proof. Similar to the proof of Theorem 1, we only need to show that $v_S^{\text{CVaR}} \leq v^*$. For $\epsilon \in (0, 1)$, define $\boldsymbol{x}(\epsilon) := (1 - \epsilon)\boldsymbol{x}^* + (\epsilon/|\bar{I}|) \sum_{i \in \bar{I}} \boldsymbol{x}^i$. Due to convexity of set $\mathcal{X}$ and $\{g(\cdot, \boldsymbol{\xi}^i)\}_{i \in \bar{I}}$, we have $\boldsymbol{x}(\epsilon) \in \bar{\mathcal{X}}^*$ and $g(\boldsymbol{x}(\epsilon), \boldsymbol{\xi}^i) < 0,\ i \in \bar{I}$. Then, following the proof of Theorem 1 (replacing $\boldsymbol{x}^*$ by $\boldsymbol{x}(\epsilon)$), we have $v_S^{\text{CVaR}} \leq \boldsymbol{c}^\top \boldsymbol{x}(\epsilon)$ for all $\epsilon \in (0, 1)$. Therefore, $v_S^{\text{CVaR}} \leq \lim_{\epsilon \to 0} \boldsymbol{c}^\top \boldsymbol{x}(\epsilon) = \boldsymbol{c}^\top \boldsymbol{x}^*$. □

We make the following remarks about Theorem 2:

(i) Even though the infimum of (7) may not be attainable (such as Example 2 of Appendix A), the scaled CVaR approximation (7) admits solutions arbitrarily close to its infimum, i.e., any ϵ-optimal solution of the scaled CVaR approximation (7) implies an ϵ-optimal solution of CCP (2);

(ii) Interested readers are referred to Example 6 of Appendix A for an illustration of the correctness of Theorem 2. Similarly, when the assumption $\sum_{i \in \bar{I}} p_i > 1 - \varepsilon$ of Theorem 2 is not met, the scaled CVaR approximation (7) may fail to find the optimal solution (see, e.g., Example 7 in Appendix A).

Motivated by Theorem 2, we establish a simple yet practically relevant sufficient condition under which $v^* = v_S^{\text{CVaR}}$ holds without requiring knowledge of the optimal solution. Specifically, this sufficient condition consists of two parts, namely, (i) the existence of a point $\bar{\boldsymbol{x}} \in \mathcal{X}$ such that $g(\bar{\boldsymbol{x}}, \boldsymbol{\xi}^i) < 0$ for all $i \in [N]$, and (ii) $\sum_{i \in I} p_i \neq \varepsilon$ for all $I \subseteq [N]$. Note that part (ii) of the sufficient condition is not restrictive following remark (iv) of Theorem 1. Next, we further propose a practical procedure for identifying such a certificate $\bar{\boldsymbol{x}}$. Given a tolerance $\bar{\delta} < 0$ (e.g., $\bar{\delta} = -10^{-5}$), we may compute $\bar{\boldsymbol{x}}$ by solving the following feasibility problem:

$$\bar{\boldsymbol{x}} \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} \left\{0 \colon g(\boldsymbol{x}, \boldsymbol{\xi}^i) \leq \bar{\delta},\ i \in [N]\right\}. \tag{8}$$

In a wide range of chance constrained programs, including stochastic lot-sizing problems [35], portfolio optimization [45, 10], and transportation applications [10], it is easy to construct a feasible solution $\bar{\boldsymbol{x}}$ to problem (8).

According to the equivalence of CCP (2) and the scaled CVaR approximation (7) in Theorem 2, and given that solving CCP (2) is in general NP-hard (see, e.g., [28]), solving the scaled CVaR approximation (7) is also NP-hard. We formally present the NP-hardness result below.

Lemma 1 (Theorem 1 in [28])**.** *The optimal objective value of a chance constrained linear program is NP-hard to compute, even when the problem takes the special form*

$$\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{n} x_i : \mathbb{P}\{\max_{i\in[n]}(\xi_i - x_i) \leq 0\} \geq 1-\varepsilon, \boldsymbol{x} \in \mathbb{R}^n_+ \right\},$$

and ξ has a finite support with $p_1 = p_2 = \ldots = p_N = 1/N$.

Combining Theorem 2 and Lemma 1, we can show the NP-hardness of the scaled CVaR approximation (7).

Theorem 3. *The scaled CVaR approximation* (7) *is NP-hard to solve, even when $\boldsymbol{c} = \boldsymbol{e}$, $\mathcal{X} = \mathbb{R}^n_+$, $\Xi \subset \mathbb{R}^n$ and $g(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{i\in[n]}(\xi_i - x_i)$.*

Proof. We first show that the two assumptions in Theorem 2 are satisfied by the CCP

$$\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{n} x_i : \mathbb{P}\{\max_{i\in[n]}(\xi_i - x_i) \leq 0\} \geq 1-\varepsilon, \boldsymbol{x} \geq \boldsymbol{0} \right\}. \tag{9}$$

Without loss of generality, assumption (ii) in Theorem 2 is satisfied by (9) because one can always add a small enough quantity to ε so that the resulting CCP is equivalent to the original CCP (9) while assumption (ii) in Theorem 2 is satisfied. Regarding assumption (i) in Theorem 2, let $\boldsymbol{\xi}^{\max}_{+} := \left\{(\max_{i\in[N]} \xi^i_j)_+\right\}_{j=1}^{n} \in \mathbb{R}^n_+$. Note that $g(\boldsymbol{\xi}^{\max}_{+} + \boldsymbol{e}, \boldsymbol{\xi}^i) < 0$ for all $i \in [N]$. Therefore, assumption (i) in Theorem 2 is satisfied. Consequently, Theorem 2 implies that the scaled CVaR approximation of the CCP yields the optimal objective value as (9). Moreover, Lemma 1 implies that the corresponding scaled CVaR approximation is NP-hard to solve. □

3 Solution Approaches

As we have shown in the last section, solving the scaled CVaR approximation (7) is, in general, NP-hard. In this section, we introduce some heuristics for solving the scaled CVaR approximation (7). We first develop an efficient heuristic approach to update the scaling factors $\boldsymbol{\alpha}$. We then introduce a sequential convex approximation approach.

3.1 An Efficient Heuristic Based on Theorem 1

In contrast to the algorithms proposed in [8, 47], our first solution approach does not optimize over $\boldsymbol{\alpha}$ within an optimization problem. Instead, we update $\boldsymbol{\alpha}$ based on the solution construction described in Theorem 1. To illustrate, we assume in this section that $\sum_{i\in I} p_i \neq \varepsilon$ for all $I \subseteq [N]$. In Algorithm 1, we iteratively update $\boldsymbol{\alpha}$ in the scaled CVaR approximation (7).

Proposition 2. *Let $\delta_2 = 0$ and set the initial solution of Algorithm 1 as the solution from CVaR approximation* (4)*. The sequence of objective values $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k\in\mathbb{Z}_+}$ generated by Algorithm 1 is monotonically nonincreasing, bounded from below, and hence convergent.*

Algorithm 1 A Heuristic to Solve the Scaled CVaR Approximation (7)

```
1: Input: Let $k \leftarrow 0$ with initial solution $\boldsymbol{x}^{(0)}$, let $\delta_1$ denote the stopping criterion parameter, let $\delta_2$ denote a violation threshold, and initialize $\Delta \leftarrow \infty$
2: while $\Delta \geq \delta_1$ do
3:     Denote $I_k = \{i \in [N] : g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) < \delta_2\}$ and $\bar{\alpha} = \frac{1}{\varepsilon - \sum_{i \in [N] \setminus I_k} p_i} \sum_{i \in [N] \setminus I_k} p_i g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)$. Update $\boldsymbol{\alpha}^{(k+1)}$ as
       $\alpha_i^{(k+1)} = \max\left\{ -\frac{\bar{\alpha}}{g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)}, 1, \alpha_i^{(k)} \right\},\ i \in I_k; \alpha_i^{(k+1)} = 1,\ i \in [N] \setminus I_k$
4:     Solve the scaled CVaR approximation (7) with $\boldsymbol{\alpha}^{(k+1)}$, i.e.,
       $v^{\mathrm{CVaR}}(\boldsymbol{\alpha}^{(k+1)}) = \min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i^{(k+1)} g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N] \right\},$
       $(\boldsymbol{x}^{(k+1)}, \beta^{(k+1)}, \boldsymbol{s}^{(k+1)}) \in \arg\min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i^{(k+1)} g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N] \right\}$
5:     Let $\Delta \leftarrow \left| \boldsymbol{c}^\top \boldsymbol{x}^{(k)} - \boldsymbol{c}^\top \boldsymbol{x}^{(k+1)} \right|$ and $k \leftarrow k + 1$
6: end while
7: Output: A feasible solution $\boldsymbol{x}^{(k)}$ and its objective $\bar{v}_S^{\mathrm{CVaR}} = \boldsymbol{c}^\top \boldsymbol{x}^{(k)}$
```

Proof. At iteration $k+1$ of Step 4 in Algorithm 1, the scaled CVaR approximation (7) we are solving is

$$v^{\mathrm{CVaR}}(\boldsymbol{\alpha}^{(k+1)}) = \min_{\substack{\boldsymbol{x} \in \mathcal{X}, \\ \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i^{(k+1)} g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N] \right\}. \tag{10}$$

We only need to show that $\boldsymbol{x}^{(k)}$ is feasible to the scaled CVaR approximation (10). Let $(\boldsymbol{x}^{(k)}, \beta^{(k)}, \boldsymbol{s}^{(k)})$ be an optimal solution at the k-th iteration after solving the scaled CVaR approximation with $\boldsymbol{\alpha}^{(k)}$. Then we have

$$\begin{aligned} s_i^{(k)} + \beta^{(k)} &\geq \alpha_i^{(k)} g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) \geq g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) = \alpha_i^{(k+1)} g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i),\ i \in [N] \setminus I_k, \\ s_i^{(k)} + \beta^{(k)} &\geq \alpha_i^{(k)} g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) \geq \alpha_i^{(k+1)} g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i),\ i \in I_k. \end{aligned}$$

Hence, $\boldsymbol{x}^{(k)}$ is feasible to the scaled CVaR approximation (10). Thus, the sequence of $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k \in \mathbb{Z}_+}$ is monotonically nonincreasing. Given that the objective function $\boldsymbol{c}^\top \boldsymbol{x}$ is bounded below by v^*, the monotonicity of the sequence of $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k \in \mathbb{Z}_+}$ implies its convergence. □

We make the following remarks about Proposition 2:

(i) From Proposition 2, we can demonstrate that, using the CVaR solution as an initial solution, the output of Algorithm 1 is no worse than the CVaR approximation (4), i.e., $\bar{v}_S^{\mathrm{CVaR}} \leq v^{\mathrm{CVaR}}$. When the CVaR approximation (4) is infeasible (see, e.g., Example 3 of Appendix A), other solutions, such as ones generated by the ALSO-X# approach proposed by [22], can be used as an initial solution;

(ii) The scaling coefficient updating procedure proposed in Algorithm 1 can be easily incorporated into other CVaR-based approximation algorithms. For example, the scaling coefficient updating procedure can be integrated into ALSO-X# (see, e.g., [22]) to enhance its performance further. For detailed discussions, we refer interested readers to Appendix B;

(iii) At each iteration, we can warm-start the process with the solution found in the previous iteration;

(iv) In the implementation of our numerical experiments, to avoid an excessively large $\bar{\boldsymbol{\alpha}}$ (which may lead to numerical issues), in Step 3 of Algorithm 1, we select $\delta_2 = -0.005$. Note that, in such cases, the sequence of objective values $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k\in\mathbb{Z}_+}$ generated may not converge, and we may set a maximum iteration limit. If the maximum iteration limit is reached, the minimum value from the sequence $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k\in\mathbb{Z}_+}$ is reported as the final output.

In the subsequent subsection, we discuss how to use sequential convex approximations to obtain a stronger initial solution for Algorithm 1.

3.2 A Sequential Convex Approximation Approach

Recall that throughout the paper we define $g(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{j\in[J]} g_j(\boldsymbol{x}, \boldsymbol{\xi})$, which allows formulation (2) to cover both single ($J = 1$) and joint ($J > 1$) chance constrained programs. Using this definition, we can equivalently write the optimally scaled CVaR approximation (7) as

$$v_S^{\text{CVaR}} = \inf_{\substack{\boldsymbol{x}\in\mathcal{X},\beta\leq 0,\\ \boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}}} \left\{ \boldsymbol{c}^\top\boldsymbol{x} \colon \varepsilon\beta + \sum_{i\in[N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i g_j(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N], j \in [J] \right\}.$$

The product terms $\{\alpha_i g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)\}_{i\in[N],j\in[J]}$ bring difficulty to the solution of the scaled CVaR approximation. One way to address it is to use the well-known difference-of-convex (DC) approach (see, e.g., Section 2 of [42]). Using the fact that $xy = 1/4[(x+y)^2 - (x-y)^2]$, we can reformulate the scaled CVaR approximation (7) as

$$v_S^{\text{CVaR}} = \inf_{\substack{\boldsymbol{x}\in\mathcal{X},\beta\leq 0,\\ \boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}}} \left\{ \boldsymbol{c}^\top\boldsymbol{x} \colon \begin{array}{l} \varepsilon\beta + \sum_{i\in[N]} p_i s_i \leq 0, \\ 4s_i + 4\beta \geq [\alpha_i + g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2 - [\alpha_i - g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2,\ i \in [N], j \in [J] \end{array} \right\}.$$

This reformulation allows us to apply sequential convex conservative approximations. To ensure the validity of the sequential convex approximations, throughout this subsection, we assume that the function $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ is affine in $\boldsymbol{x}$ for each $j \in [J]$, i.e., $g_j(\boldsymbol{x}, \boldsymbol{\xi}^i) = \boldsymbol{a}_j(\xi^i)^\top\boldsymbol{x} - b_j(\xi^i)$ with $\boldsymbol{a}_j : \Xi \to \mathbb{R}^n$, and $b_j : \Xi \to \mathbb{R}$. This affine structure arises in a broad class of CCP models and has been widely used in the CCP literature (see, e.g., [33, 3, 45, 44, 18, 19, 10]).

In the sequential convex approximation, at iteration $k+1$, for each $i \in [N], j \in [J]$, we replace $[\alpha_i - g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2$ by its first-order Taylor approximation using the solution from the previous iteration k, that is,

$$\begin{aligned}[\alpha_i - g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2 \geq {} & [\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]^2 + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)](\alpha_i - \alpha_i^{(k)}) \\ & + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]\nabla_{\boldsymbol{x}^{(k)}} g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)^\top[\boldsymbol{x} - \boldsymbol{x}^{(k)}],\end{aligned}$$

Algorithm 2 A Sequential Convex Approximation Algorithm to Solve the Scaled CVaR Approximation (7)

```
Input: Let $k \leftarrow 0$ with initial solution $\boldsymbol{x}^{(0)}$, let $\delta_1$ denote the stopping tolerance parameter, and initialize $\Delta \leftarrow \infty$
while $\Delta \geq \delta_1$ do
    Solve Problem (11) to obtain $\boldsymbol{x}^{(k+1)}$
    Let $\Delta \leftarrow \left|\boldsymbol{c}^\top \boldsymbol{x}^{(k)} - \boldsymbol{c}^\top \boldsymbol{x}^{(k+1)}\right|$ and $k \leftarrow k+1$
end while
Output: A feasible solution $\boldsymbol{x}^{(k)}$ and its objective value $\bar{v}_S^{\mathrm{CVaR}}(DC) = \boldsymbol{c}^\top \boldsymbol{x}^{(k)}$
```

where $\nabla_{\boldsymbol{x}} g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)$ denotes the derivative of $g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)$ for $\boldsymbol{x}^{(k)}$. Then, we solve the following program:

$$
\begin{aligned}
(\boldsymbol{x}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}) \in \operatorname{proj}_{(\boldsymbol{x},\boldsymbol{\alpha})} \arg \min_{\substack{\boldsymbol{x}\in\mathcal{X},\, \beta\leq 0,\\ \boldsymbol{s}\geq\boldsymbol{0},\, \boldsymbol{\alpha}\geq\boldsymbol{e}}} \quad & \boldsymbol{c}^\top \boldsymbol{x}, \\
\text{s.t.} \quad & \varepsilon\beta + \sum_{i\in[N]} p_i s_i \leq 0, \\
& [\alpha_i + g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2 \leq 4[s_i + \beta] + [\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]^2 \\
& \quad + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)](\alpha_i - \alpha_i^{(k)}) \\
& \quad + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]\nabla_{\boldsymbol{x}^{(k)}} g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)^\top [\boldsymbol{x} - \boldsymbol{x}^{(k)}], \ i \in [N], j \in [J].
\end{aligned}
\tag{11}
$$

Problem (11) serves as a convex approximation of scaled CVaR approximation (7), which is iteratively solved in a sequential convex approximation algorithm as summarized in Algorithm 2.

Algorithm 2 is similar to the sequential convex approximation (SCA) algorithm proposed by [20]. Based on the property 1 and property 2 in [20], by initializing Algorithm 2 with the solution obtained from the CVaR approximation (4), the sequence of objective values $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k\in\mathbb{Z}_+}$ generated by Algorithm 2 is monotonically nonincreasing, bounded from below, and therefore convergent. Hence, we can show that the output of Algorithm 2 improves upon the CVaR approximation (4), i.e., $\bar{v}_S^{\mathrm{CVaR}}(DC) \leq v^{\mathrm{CVaR}}$, if Algorithm 2 is initialized using the CVaR solution. These properties of Algorithm 2 are formally summarized below.

Corollary 1. *Suppose we set the initial solution of Algorithm 2 as the solution from CVaR approximation* (4), *then the sequence of objective values* $\{\boldsymbol{c}^\top \boldsymbol{x}^{(k)}\}_{k\in\mathbb{Z}_+}$ *generated by Algorithm 2 is monotonically nonincreasing, bounded from below, and hence convergent, and the output of Algorithm 2 is better than CVaR approximation* (4), *i.e.,* $\bar{v}_S^{\mathrm{CVaR}}(DC) \leq v^{\mathrm{CVaR}}$.

We remark that even though Algorithm 2 remains a difference-of-convex algorithm (DCA), it is different from the DCA proposed in [20], where the latter directly approximates the chance constraint using difference-of-convex functions.

Different from Algorithm 1, which updates $\boldsymbol{\alpha}$ and $\boldsymbol{x}$ separately, Algorithm 2 updates them simultaneously. However, note that Algorithm 2 may encounter numerical issues when using commercial solvers because Problem (11) involves many quadratic constraints. To reduce the number of quadratic constraints in Problem (11), we take a hybrid strategy inspired by Algorithm 1. Specifically, let $I_k = \{i \in [N] : g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) < 0\}$. Rather than relaxing all product terms $\{\alpha_i g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)\}_{i\in[N], j\in[J]}$, the key idea is to relax only those terms where $i \in I_k$. For $i \in [N] \setminus I_k$, we simply set $\alpha_i = 1$. The details of this hybrid approach are presented in Algorithm 3.

Algorithm 3 A Hybrid Algorithm to Solve the Scaled CVaR Approximation (7)

```
Input: Let $k \leftarrow 0$ with initial solution $\boldsymbol{x}^{(0)}$, let $\delta_1$ denote the stopping tolerance parameter,
  let $\delta_2$ denote a violation threshold, and initialize $\Delta \leftarrow \infty$
while $\Delta \geq \delta_1$ do
    Denote $I_k = \{i \in [N] : g(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i) < \delta_2\}$ and $\bar{I}_k = [N] \setminus I_k$
    Obtain $(\boldsymbol{x}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})$ by solving
    $$\begin{aligned}
    (\boldsymbol{x}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}) \in \operatorname{proj}_{(\boldsymbol{x},\boldsymbol{\alpha})} & \operatorname*{arginf}_{\substack{\boldsymbol{x}\in\mathcal{X},\beta\leq 0,\\ \boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}}} \ \boldsymbol{c}^\top \boldsymbol{x}, \\
    \text{s.t.} \quad & \varepsilon\beta + \sum_{i\in[N]} p_i s_i \leq 0, \\
    & s_i + \beta \geq g_j(\boldsymbol{x}, \boldsymbol{\xi}^i), \alpha_i = 1, \ i \in \bar{I}_k, j \in [J] \\
    & [\alpha_i + g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)]^2 \leq 4[s_i + \beta] + [\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]^2 \\
    & \qquad + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)](\alpha_i - \alpha_i^{(k)}) \\
    & \qquad + 2[\alpha_i^{(k)} - g_j(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^i)]\nabla_{\boldsymbol{x}^{(k)}} g_j(\boldsymbol{x}, \boldsymbol{\xi}^i)^\top [\boldsymbol{x} - \boldsymbol{x}^{(k)}], \ i \in I_k, j \in [J]
    \end{aligned}$$
    Let $\Delta \leftarrow \left|\boldsymbol{c}^\top \boldsymbol{x}^{(k)} - \boldsymbol{c}^\top \boldsymbol{x}^{(k+1)}\right|$ and $k \leftarrow k + 1$
end while
Let $(\boldsymbol{x}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})$ be the input of Algorithm 1, run Algorithm 1 until it terminates, and let
  $\boldsymbol{x}^H$ and its objective value $\bar{v}_S^{\mathrm{CVaR}}(H)$ be the output of Algorithm 1
Output: A feasible solution $\boldsymbol{x}^H$ and its objective value $\bar{v}_S^{\mathrm{CVaR}}(H) = \boldsymbol{c}^\top \boldsymbol{x}^H$
```

Moreover, combining elements of Algorithm 2 and Algorithm 1 can further enhance the solution quality. The key observation is that, when Algorithm 2 terminates with the prespecified tolerance or encounters numerical difficulties, its output provides a feasible solution $\boldsymbol{x}^{(k)}$ to the scaled CVaR approximation with the scaling vector fixed at its final value $\boldsymbol{\alpha}^{(k)}$. Therefore, after Algorithm 2 terminates, we can fix the scaling vector at the final iterate $\boldsymbol{\alpha}^{(k)}$ and solve Step 4 of Algorithm 1 once more. This step cannot worsen the objective value returned by Algorithm 2.

Corollary 2. *Let $(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)})$ be the output of Algorithm 2. Then solving Step 4 of Algorithm 1 with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)}$ cannot produce an objective value worse than that returned by Algorithm 2. In particular, the optimal value of the scaled CVaR approximation with alpha fixed at $\boldsymbol{\alpha}^{(k)}$ is no greater than the objective value produced by Algorithm 2.*

Motivated by Corollary 2, we can first apply Algorithm 2 to generate an initial solution, followed by Algorithm 1 to refine subsequent iterations. Because both algorithms preserve monotonicity in the objective sequence, the final output sequence is consistently nonincreasing. A summary of this hybrid approach is provided in Algorithm 3, and its detailed implementation is discussed in Section 4.

3.3 Incorporating Auxiliary Optimality Information

Based on the solution construction in Theorem 2 and Algorithm 1, we observe that if a solution violates a scenario, the corresponding constraint will not be scaled up. This observation motivates us to incorporate auxiliary optimality information to fix the value of the corresponding α in the scaled CVaR approximation (7). We employ a straightforward and effective method to incorporate the optimality information and fix $\boldsymbol{\alpha}$.

Corollary 3. *Let $\boldsymbol{x}^*$ denote an optimal solution of CCP* (2) *and define $I^* := \{i \in [N] : g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) \leq 0\}$. Under the assumptions of Theorem 1 or Theorem 2, we have*

$$v^* = \inf\{v_S^{\mathrm{CVaR}}(\boldsymbol{\alpha}) : \alpha_i = 1,\ i \in [N] \setminus I^*, \alpha_i \geq 1,\ i \in I^*\}.$$

In the next section, we demonstrate how Corollary 3 can be integrated into our algorithmic implementations.

4 Numerical Study

In this section, we numerically demonstrate the effectiveness of the proposed methods. All the instances in this section are executed in Python 3.9 with calls to solver Gurobi (version 11.0.3 with default settings) on a personal PC with an Apple M1 Pro processor and 16GB of memory. To evaluate the effectiveness of the proposed algorithms, we use "Improvement" to denote the percentage of differences between the value of a proposed algorithm and CVaR approximation, i.e.,

$$\text{Improvement (\%)} = \frac{\text{CVaR approximation value} - \text{value of a proposed algorithm}}{|\text{CVaR approximation value}|} \times 100\%.$$

4.1 Joint Chance Constraint: Instances from [39]

The joint CCP that we test admits the following form:

$$v^* = \min_{\boldsymbol{x} \in [0,1]^n} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \frac{1}{N} \sum_{i \in [N]} \mathbb{I} \left[\max_{j \in [J]} \left\{ \sum_{k \in [n]} \xi_{j,k}^i x_k - b_j^i \right\} \leq 0 \right] \geq 1 - \varepsilon \right\}.$$

We evaluate the proposed method on three sets of joint CCP instances with $N = 3000$, *1-4-multi-3000*, *1-6-multi-3000*, and *1-7-multi-3000* from [39]. To approximate the scaled CVaR approximation (7), we employ the following two steps:

Step 1. We use the solution of CVaR approximation (4) as the initial starting point for Algorithm 3 and run Algorithm 3 for one iteration to determine the scaling factor $\boldsymbol{\alpha}$.

Step 2. We use the scaling factor $\boldsymbol{\alpha}$ from Step 1 as input to Algorithm 1 and implement Algorithm 1 with $\delta_2 = -0.005$ for 25 iterations to obtain the best objective value.

For both Step 1 and Step 2, we consider incorporating the auxiliary optimality information discussed in Corollary 3. We use v_U^* to denote the best upper bound of the scaled CVaR approximation (7) and solve $\eta_i = \min_{\boldsymbol{x} \in [0,1]^n} \{\boldsymbol{c}^\top \boldsymbol{x} : \sum_{k \in [n]} \xi_{j,k}^i x_k - b_j^i \leq 0,\ j \in [J]\}$ for each $i \in [N]$. Following the approach recently employed in [23], if $\eta_i > v_U^*$ for some $i \in [N]$, we set $\alpha_i = 1$.

We record the total running time of the steps above as the running time of approximating the scaled CVaR approximation (7). For comparison, we also implement the ALSO-X# algorithm from [22]. In ALSO-X# (see the detailed algorithm in Algorithm 4 of Appendix B), we choose t_L as the quantile bound from [39], t_U as the CVaR approximation (4), and $\delta_A = 0.05$. To ensure that $N\varepsilon$ is not an integer, we consider the risk parameters $\varepsilon \in \{0.050333, 0.100333, 0.200333, 0.300333\}$. Table 1 summarizes the average numerical performance across instances, while the detailed results are reported in Table 3 in Appendix C.

We observe that our approach to solving the scaled CVaR approximation (7) consistently improves the objective value compared to the CVaR approximation (4). In most cases, our method

Table 1: Numerical Results of a Joint CCP with Different ε

ε	n	J	Instance	CVaR Approximation	ALSO-X#			Scaled CVaR Approximation		
					Value	Time (s)	Improvement	Value	Time (s)	Improvement
0.050333	20	10	1-4-multi-3000	−5778.50	−5845.66	83.83	1.16%	−5851.09	525.05	1.26%
	39	5	1-6-multi-3000	−9864.42	−9976.61	83.86	1.14%	−9979.76	530.35	1.17%
	50	5	1-7-multi-3000	−15766.16	−15879.72	108.17	0.72%	−15885.74	690.25	0.76%
0.100333	20	10	1-4-multi-3000	−5826.18	−5901.36	84.28	1.29%	−5905.43	531.48	1.36%
	39	5	1-6-multi-3000	−9947.12	−10081.34	84.15	1.35%	−10086.58	538.40	1.40%
	50	5	1-7-multi-3000	−15852.28	−15995.49	108.00	0.90%	−16004.91	691.10	0.96%
0.200333	20	10	1-4-multi-3000	−5880.51	−5970.65	99.69	1.53%	−5977.62	660.87	1.65%
	39	5	1-6-multi-3000	−10047.76	−10219.45	85.62	1.71%	−10226.30	543.84	1.78%
	50	5	1-7-multi-3000	−15962.55	−16145.73	107.37	1.15%	−16153.17	688.10	1.19%
0.300333	20	10	1-4-multi-3000	−5917.73	−6022.59	83.89	1.77%	−6031.27	563.86	1.92%
	39	5	1-6-multi-3000	−10120.96	−10325.93	85.49	2.03%	−10332.91	541.71	2.09%
	50	5	1-7-multi-3000	−16041.49	−16260.96	108.01	1.37%	−16268.84	686.50	1.42%
						Average	**1.34%**			**1.41%**

achieves greater improvement than the ALSO-X# method and demonstrates superior average performance. Note that our approach is different from ALSO-X# and does not rely on the bisection procedure that the latter requires. Although our method for approximating the scaled CVaR approximation (7) requires a longer computation time than the ALSO-X# method, due to the quadratic constraints in Algorithm 3 and the increased number of iterations in Algorithm 1, it yields better objective value improvements, highlighting its effectiveness despite the added computational cost. We also remark that we compare our method with the approach that uses an interior point local solver IPOPT [43] to directly solve the scaled CVaR approximation (7). The fact that our method outperformed IPOPT further suggests that the nonconvex optimization problem (7) is challenging to solve in practice. Detailed numerical comparisons can be found in Table 5 of Appendix D.

4.2 Portfolio Optimization

In this subsection, we study a portfolio optimization problem adapted from [45], with relevant formulations appearing in [33, 10, 19]. We consider the following formulation, where the decision vector is restricted to $[0,1]^n$ and the total allocation is controlled by an investment budget constraint $\sum_{k\in[n]} x_k \leq 0.2n$:

$$v^* = \min_{\boldsymbol{x}\in[0,1]^n} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \frac{1}{N}\sum_{i\in[N]} \mathbb{I}\left[\sum_{k\in[n]} \xi_k^i x_k \geq 1\right] \geq 1-\varepsilon, \sum_{k\in[n]} x_k \leq 0.2n \right\}.$$

For numerical experiments, we follow the data generation scheme in [45], where n is fixed to 50, $N \in \{500, 1000\}$, $\{\boldsymbol{\xi}^i\}_{i\in[N]}$ are i.i.d. uniform random variables in the range from 0.8 to 1.2 and the risk parameter $\varepsilon \in \{0.050333, 0.100333, 0.200333, 0.300333\}$. For each combination of (n, N, ϵ), we generate 5 instances. The cost vector $\boldsymbol{c}$ is generated at random, where each component takes an integer value uniformly sampled from $\{1, \cdots, 100\}$. We employ the following two steps to approximate the scaled CVaR approximation (7):

Step 1. We use the solution of CVaR approximation (4) as the initial starting point for Algorithm 3.

Step 2. We implement Algorithm 1 with $\delta_2 = -0.005$ for 20 iterations to obtain the best objective value.

Similar to the joint chance constraint case discussed in Section 4.1, we incorporate the auxiliary optimality information outlined in Corollary 3 in both Step 1 and Step 2. Let v_U^* de-

Table 2: Numerical Results of Portfolio Optimization

ε	n	N	CVaR Approximation	ALSO-X#			Scaled CVaR Approximation		
				Value	Time (s)	Improvement	Value	Time (s)	Improvement
0.050333	50	500	2.86	2.80	0.38	1.76%	2.79	9.62	2.37%
		1000	2.84	2.79	0.70	1.63%	2.77	19.03	2.48%
0.100333	50	500	2.78	2.68	0.48	3.24%	2.68	9.91	3.66%
		1000	2.77	2.69	0.94	2.36%	2.67	20.55	3.54%
0.200333	50	500	2.68	2.53	0.64	5.08%	2.52	10.20	5.89%
		1000	2.67	2.52	1.23	5.00%	2.50	20.94	6.00%
0.300333	50	500	2.60	2.37	0.68	7.76%	2.36	10.41	8.92%
		1000	2.58	2.35	1.34	7.84%	2.33	21.79	9.12%
					Average	**4.33%**			**5.25%**

*Each row is the average over five independent instances.

note the best upper bound of the scaled CVaR approximation (7). For each $i \in [N]$, we solve $\eta_i = \min_{\boldsymbol{x}\in[0,1]^n}\{\boldsymbol{c}^\top \boldsymbol{x} : \sum_{k\in[n]} \xi_k^i x_k \geq 1, \sum_{k\in[n]} x_k \leq 0.2n\}$. If $\eta_i > v_U^*$ for any $i \in [N]$, then we enforce $\alpha_i = 1$. All other settings are consistent with those described in Section 4.1. Table 2 summarizes the average numerical performance across instances, whereas the detailed results under different choices of ε and N are provided in Table 4 in Appendix C. Overall, our approach for solving the scaled CVaR approximation (7) consistently yields higher-quality solutions than the ALSO-X# method. When averaged across all tested instances, the improvement achieved by the scaled CVaR approach increases with the violation level ε, ranging from 2.43% versus 1.70% at $\varepsilon = 0.050333$ to 9.02% versus 7.80% at $\varepsilon = 0.300333$. These results indicate that the proposed method attains improved objective values, particularly for moderate-to-high risk parameters, while requiring only a modest increase in computational time.

5 Conclusion

In this paper, we investigated the scaled CVaR approximation for solving CCPs. We provided sufficient conditions under which the scaled CVaR approximation preserves an optimal solution of a CCP. We also developed efficient algorithms to solve the scaled CVaR approximation. Our numerical study confirmed the effectiveness of the proposed algorithms. An important direction for future research is the extension of the proposed scaling framework to broader distributionally robust settings, including general Wasserstein ambiguity sets, which may require new definitions of scaling.

References

[1] Shabbir Ahmed, James Luedtke, Yongjia Song, and Weijun Xie. Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs. *Mathematical Programming*, 162(1):51–81, 2017.

[2] Shabbir Ahmed and Alexander Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *Tutorials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 261–269. 2008.

[3] Shabbir Ahmed and Weijun Xie. Relaxations and approximations of chance constraints under finite distributions. *Mathematical Programming*, 170:43–65, 2018.

[4] Timo Berthold and Gregor Hendel. Learning to scale mixed-integer programs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3661–3668, 2021.

[5] Giuseppe Carlo Calafiore and Marco C Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[6] Abraham Charnes and William W Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39, 1963.

[7] Abraham Charnes, William W Cooper, and Gifford H Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.

[8] Wenqing Chen, Melvyn Sim, Jie Sun, and Chung-Piaw Teo. From CVaR to uncertainty set: Implications in joint chance-constrained optimization. *Operations Research*, 58(2):470–485, 2010.

[9] Zhi Chen, Daniel Kuhn, and Wolfram Wiesemann. On approximations of data-driven chance constrained programs over Wasserstein balls. *Operations Research Letters*, 51(3):226–233, 2023.

[10] Zhi Chen, Daniel Kuhn, and Wolfram Wiesemann. Data-driven chance constrained programs over Wasserstein balls. *Operations Research*, 72(1):410–424, 2024.

[11] Jehum Cho and Anthony Papavasiliou. Exact mixed-integer programming approach for chance-constrained multi-area reserve sizing. *IEEE Transactions on Power Systems*, 39(2):3310–3323, 2023.

[12] Yan Deng, Huiwen Jia, Shabbir Ahmed, Jon Lee, and Siqian Shen. Scenario grouping and decomposition algorithms for chance-constrained programs. *INFORMS Journal on Computing*, 33(2):757–773, 2021.

[13] Yan Deng and Siqian Shen. Decomposition algorithms for optimizing multi-server appointment scheduling with chance constraints. *Mathematical Programming*, 157:245–276, 2016.

[14] Darinka Dentcheva, András Prékopa, and Andrzej Ruszczynski. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1):55–77, 2000.

[15] Thai Dinh, Ricardo Fukasawa, and James Luedtke. Exact algorithms for the chance-constrained vehicle routing problem. *Mathematical Programming*, 172(1):105–138, 2018.

[16] Shubhechyya Ghosal and Wolfram Wiesemann. The distributionally robust chance-constrained vehicle routing problem. *Operations Research*, 68(3):716–732, 2020.

[17] Grani A Hanasusanto, Vladimir Roitch, Daniel Kuhn, and Wolfram Wiesemann. Ambiguous joint chance constraints under mean and dispersion information. *Operations Research*, 65(3):751–767, 2017.

[18] Nam Ho-Nguyen, Fatma Kılınç-Karzan, Simge Küçükyavuz, and Dabeen Lee. Distributionally robust chance-constrained programs with right-hand side uncertainty under Wasserstein ambiguity. *Mathematical Programming*, pages 1–32, 2022.

[19] Nam Ho-Nguyen, Fatma Kilinç-Karzan, Simge Küçükyavuz, and Dabeen Lee. Strong formulations for distributionally robust chance-constrained programs with left-hand side uncertainty under Wasserstein ambiguity. *INFORMS Journal on Optimization*, 5(2):211–232, 2023.

[20] L Jeff Hong, Yi Yang, and Liwei Zhang. Sequential convex approximations to joint chance constrained programs: A monte carlo approach. *Operations Research*, 59(3):617–630, 2011.

[21] Nan Jiang and Weijun Xie. ALSO-X and ALSO-X+: Better convex approximations for chance constrained programs. *Operations Research*, 70(6):3581–3600, 2022.

[22] Nan Jiang and Weijun Xie. ALSO-X#: Better convex approximations for distributionally robust chance constrained programs. *Mathematical Programming*, 213:575–638, 2025.

[23] Nan Jiang and Weijun Xie. The terminator: An integration of inner and outer approximations for solving Wasserstein distributionally robust chance constrained programs via variable fixing. *INFORMS Journal on Computing*, 37(2):381–412, 2025.

[24] Simge Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1):31–56, 2012.

[25] Simge Küçükyavuz and Ruiwei Jiang. Chance-constrained optimization under limited distributional information: A review of reformulations based on sampling and distributional robustness. *EURO Journal on Computational Optimization*, 10:100030, 2022.

[26] Walter E Lillo, Mei Heng Loh, Stefen Hui, and Stanislaw H Zak. On solving constrained optimization problems with neural networks: A penalty method approach. *IEEE Transactions on Neural Networks*, 4(6):931–940, 1993.

[27] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[28] James Luedtke, Shabbir Ahmed, and George L Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[29] Arkadi Nemirovski and Alexander Shapiro. Scenario approximations of chance constraints. In *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 3–47. Springer, 2006.

[30] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

[31] Bernardo K Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. Computational study of a chance constrained portfolio selection problem. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009.

[32] Álvaro Porras, Line Roald, Juan Miguel Morales, and Salvador Pineda. Unifying chance-constrained and robust optimal power flow for resilient network operations. *IEEE Transactions on Control of Network Systems*, 12(1):1052–1061, 2025.

[33] Feng Qiu, Shabbir Ahmed, Santanu S Dey, and Laurence A Wolsey. Covering linear programming with violations. *INFORMS Journal on Computing*, 26(3):531–546, 2014.

[34] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[35] Andrzej Ruszczyński. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming*, 93:195–215, 2002.

[36] Sergey Sarykalin, Gaia Serraino, and Stan Uryasev. Value-at-risk vs. conditional value-at-risk in risk management and optimization. In *Tutorials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 270–294. 2008.

[37] Takayuki Shiina. Numerical solution technique for joint chance-constrained programming problem: An application to electric power capacity expansion. *Journal of the Operations Research Society of Japan*, 42(2):128–140, 1999.

[38] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[39] Yongjia Song, James R Luedtke, and Simge Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.

[40] Andrews K Takyi and Barbara J Lence. Surface water quality management using a multiple-realization chance constraint method. *Water Resources Research*, 35(5):1657–1670, 1999.

[41] Srinivas Talluri, Ram Narasimhan, and Anand Nair. Vendor performance with supply risk: A chance-constrained dea approach. *International Journal of Production Economics*, 100(2):212–222, 2006.

[42] Pham Dinh Tao and LT Hoai An. Convex analysis approach to dc programming: theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

[43] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[44] Weijun Xie. On distributionally robust chance constrained programs with Wasserstein distance. *Mathematical Programming*, 186(1):115–155, 2021.

[45] Weijun Xie and Shabbir Ahmed. Bicriteria approximation of chance-constrained covering problems. *Operations Research*, 68(2):516–533, 2020.

[46] Wenzhuo Yang and Huan Xu. Distributionally robust chance constraints for non-linear uncertainties. *Mathematical Programming*, 155:231–265, 2016.

[47] Steve Zymler, Daniel Kuhn, and Berç Rustem. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137:167–198, 2013.

A Examples

Example 2. Consider a single CCP with 2 equiprobable scenarios (i.e., $p_1 = p_2 = 1/2$), risk parameter $\varepsilon = 2/3$, set $\mathcal{X} = \mathbb{R}^2_+$, constraint $g(\boldsymbol{x}, \boldsymbol{\xi}) = 1 - \boldsymbol{\xi}^\top \boldsymbol{x}$, and $\boldsymbol{\xi}^1 = (1,0)^\top$, and $\boldsymbol{\xi}^2 = (1,1)^\top$. In this case, the CCP (2) is equivalent to the following mixed-integer linear program

$$v^* = \min_{\boldsymbol{x} \in \mathbb{R}^2_+, \boldsymbol{z} \in \{0,1\}^2} \left\{ 2x_1 + x_2 \colon x_1 \geq z_1, x_1 + x_2 \geq z_2, z_1 + z_2 \geq 1 \right\}. \tag{12}$$

Note that here we have $v^* = 1$. The CVaR approximation of the CCP (12) is

$$v^{\text{CVaR}} = \min_{x_1 \geq 0, x_2 \geq 0, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ 2x_1 + x_2 : \begin{array}{l} x_1 \geq 1 - s_1 - \beta, x_1 + x_2 \geq 1 - s_2 - \beta, \\ (s_1 + s_2)/2 + 2\beta/3 \leq 0 \end{array} \right\},$$

with $v^{\text{CVaR}} = 2$. To improve the CVaR approximation, we can implement a scaling procedure. To illustrate, note that, given $\alpha \geq 1$, $g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ is equivalent to $\bar{g}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ with $\bar{g}(\boldsymbol{x}, \boldsymbol{\xi}^1) = g(\boldsymbol{x}, \boldsymbol{\xi}^1)$ and $\bar{g}(\boldsymbol{x}, \boldsymbol{\xi}^2) = \alpha g(\boldsymbol{x}, \boldsymbol{\xi}^2)$. In this case, one can replace $g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ by $\bar{g}(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0$ in the original CCP (12), resulting in the following CVaR approximation:

$$v^{\text{CVaR}}(\alpha) = \min_{x_1 \geq 0, x_2 \geq 0, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ 2x_1 + x_2 : \begin{array}{l} x_1 \geq 1 - s_1 - \beta, \alpha(x_1 + x_2) \geq \alpha - s_2 - \beta, \\ (s_1 + s_2)/2 + 2\beta/3 \leq 0 \end{array} \right\}.$$

Note that the optimal objective value v^* is independent of the scaling factor α while $v^{\text{CVaR}}(\alpha)$ varies with changes in α. When $\alpha \in [1, 3)$, $v^{\text{CVaR}}(\alpha) = 2$ with an optimal solution $x_1^* = 1, x_2^* = 0, s_1^* = s_2^* = 0, \beta^* = 0$. When $\alpha \in (3, \infty)$, $v^{\text{CVaR}}(\alpha) = 1 + 3/\alpha$ with an optimal solution $x_1^* = 0, x_2^* = 1 + \frac{3}{\alpha}, s_1^* = 4, s_2^* = 0, \beta^* = -3$. The optimal objective values $v^{\text{CVaR}}(\alpha)$ of the scaled CVaR approximation are plotted in Figure 2. As we may observe in this example, $v^* = \lim_{\alpha \to \infty} v^{\text{CVaR}}(\alpha)$. ◇

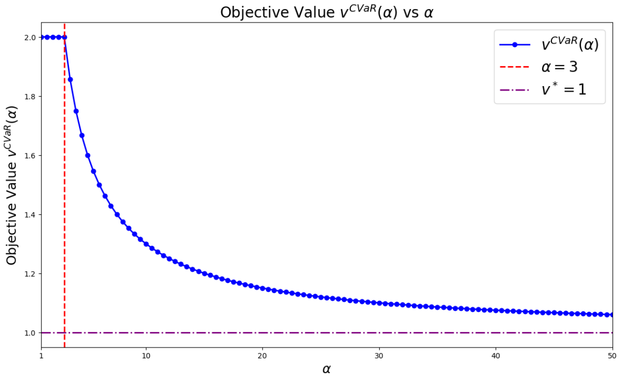

Figure 2: The Scaled CVaR Approximation Objective $v^{\text{CVaR}}(\alpha)$ for Scaling Factor $\alpha \in [1, 50]$.

Example 3. Consider a single CCP with 4 equiprobable scenarios (i.e., $N = 4$, $p_1 = p_2 = p_3 = p_4 = 1/4$), risk parameter $\varepsilon = 1/2$, deterministic set $\mathcal{X} = \mathbb{Z}_+$, constraint $g(\boldsymbol{x}, \boldsymbol{\xi}) = \xi_1 x - \xi_2$, $\xi_1^1 = -9$, $\xi_1^2 = \xi_1^3 = \xi_1^4 = 4$, $\xi_2^1 = -10$, and $\xi_2^2 = \xi_2^3 = \xi_2^4 = 2$. In this example, CCP (2) resorts to

$$v^* = \min_{x \in \mathbb{Z}_+} \left\{ -x \colon \mathbb{I}(9x \geq 10) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 2) \geq 2 \right\}.$$

By a simple calculation, the optimal value is $v^* = 0$ with the optimal solution $x^* = 0$. Notice that with the solution $x^* = 0$, we have (i) $4x^* = 0 < 2$; and (ii) $1/4[\mathbb{I}(9x^* \geq 10) + \mathbb{I}(4x^* \leq 2) + \mathbb{I}(4x^* \leq 2) + \mathbb{I}(4x^* \leq 2)] > 1 - 1/2$. Hence, the two conditions in Theorem 1 are satisfied. The scaled CVaR approximation (7) is expected to provide the optimal solution in this example. We first check the corresponding CVaR approximation, that is,

$$v^{\mathrm{CVaR}} = \min_{x \in \mathbb{Z}_+, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}} \left\{ -x : \begin{array}{c} -9x + 10 \leq s_1, 4x - 2 \leq s_2 + \beta, 4x - 2 \geq s_3 + \beta, \\ 4x - 2 \leq s_4 + \beta, \sum_{i \in [4]} s_i/4 + \beta/2 \leq 0 \end{array} \right\}.$$

It turns out this CVaR approximation is infeasible.

We then consider the scaled CVaR approximation (7), which can be recast as,

$$v_S^{\mathrm{CVaR}} = \inf_{\substack{x \in \mathbb{Z}_+, \beta \leq 0, \\ \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{\alpha} \geq \boldsymbol{e}}} \left\{ -x : \begin{array}{c} -9x\alpha_1 + 10\alpha_1 \leq s_1 + \beta, 4x\alpha_2 - 2\alpha_2 \leq s_2 + \beta, 4x\alpha_3 - 2\alpha_3 \leq s_3 + \beta, \\ 4x\alpha_4 - 2\alpha_4 \leq s_4 + \beta, \sum_{i \in [4]} s_i/4 + \beta/2 \leq 0 \end{array} \right\}.$$

We obtain $v_S^{\mathrm{CVaR}} = 0$ with $x^* = 0, \beta^* = -10, \alpha_1^* = 1, \alpha_2^* = \alpha_3^* = \alpha_4^* = 5, s_1^* = 20, s_2^* = s_3^* = s_4^* = 0$. Therefore, in this example, we can use the scaled CVaR approximation (7) to find the optimal solution. ◇

Example 4. Consider a single CCP with 4 equiprobable scenarios (i.e., $N = 4$, $p_1 = p_2 = p_3 = p_4 = 1/4$), risk parameter $\varepsilon = 1/2$, deterministic set $\mathcal{X} = \mathbb{Z}_+$, constraint $g(\boldsymbol{x}, \boldsymbol{\xi}) = \xi_1 x - \xi_2$, $\xi_1^1 = -9$, $\xi_1^2 = \xi_1^3 = \xi_1^4 = 4$, $\xi_2^1 = -10$, and $\xi_2^2 = \xi_2^3 = 2, \xi_2^4 = 0$. In this example, CCP (2) resorts to

$$v^* = \min_{x \in \mathbb{Z}_+} \left\{ -x \colon \mathbb{I}(9x \geq 10) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 0) \geq 2 \right\}.$$

By a simple calculation, the optimal value is $v^* = 0$ with the optimal solution $x^* = 0$. We also have $4x^* = 0 \not< 0$, which violates the first assumption of Theorem 1. In this case, the scaled CVaR approximation (7) can be written as

$$v_S^{\mathrm{CVaR}} = \inf_{x \in \mathbb{Z}_+, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{\alpha} \geq \boldsymbol{e}} \left\{ -x : \begin{array}{c} -9x\alpha_1 + 10\alpha_1 \leq s_1 + \beta, 4x\alpha_2 - 2\alpha_2 \leq s_2 + \beta, \\ 4x\alpha_3 - 2\alpha_3 \leq s_3 + \beta, 4x\alpha_4 \leq s_4 + \beta, \sum_{i \in [4]} s_i/4 + \beta/2 \leq 0 \end{array} \right\}.$$

However, in this case, the scaled CVaR approximation is infeasible. This implies that when the condition $g(\boldsymbol{x}^*, \boldsymbol{\xi}^i) < 0$ for each $i \in I^*$ in Theorem 1 is violated, the scaled CVaR approximation may not yield the optimal solution. ◇

Example 5. Consider a single CCP with 4 equiprobable scenarios (i.e., $N = 4$, $p_1 = p_2 = p_3 = p_4 = 1/4$), risk parameter $\varepsilon = 1/4$, deterministic set $\mathcal{X} = \mathbb{Z}_+$, constraint $g(\boldsymbol{x}, \boldsymbol{\xi}) = \xi_1 x - \xi_2$, $\xi_1^1 = -9$, $\xi_1^2 = \xi_1^3 = \xi_1^4 = 4$, $\xi_2^1 = -10$, and $\xi_2^2 = \xi_2^3 = \xi_2^4 = 2$. In this example, CCP (2) resorts to

$$v^* = \min_{x \in \mathbb{Z}_+} \left\{ -x \colon \mathbb{I}(9x \geq 10) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 2) + \mathbb{I}(4x \leq 2) \geq 3 \right\}.$$

By a simple calculation, the optimal value is $v^* = 0$ with the optimal solution $x^* = 0$. We also have $1/4[\mathbb{I}(9x^* \geq 10) + \mathbb{I}(4x^* \leq 2) + \mathbb{I}(4x^* \leq 2) + \mathbb{I}(4x^* \leq 2)] = 3/4 \not> 1 - 1/4$, which violates the second assumption of Theorem 1. In this case, the scaled CVaR approximation (7) can be written as

$$v_S^{\text{CVaR}} = \inf_{x\in\mathbb{Z}_+,\beta\leq 0,\boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}} \left\{ -x : \begin{array}{c} -9x\alpha_1 + 10\alpha_1 \leq s_1 + \beta, 4x\alpha_2 - 2\alpha_2 \leq s_2 + \beta, \\ 4x\alpha_3 - 2\alpha_3 \leq s_3 + \beta, 4x\alpha_4 - 2\alpha_4 \leq s_4 + \beta, \sum_{i\in[4]} s_i/4 + \beta/4 \leq 0 \end{array} \right\}.$$

However, this scaled CVaR approximation is infeasible, which implies that we cannot use the scaled CVaR approximation to find the optimal solution when we violate the condition $\sum_{i\in I^*} p_i > 1-\varepsilon$ in Theorem 1. ◇

Example 6. Consider a single CCP with 3 equiprobable scenarios (i.e., $N = 3$, $p_1 = p_2 = p_3 = 1/3$), risk parameter $\varepsilon = 0.4$, set $\mathcal{X} = \mathbb{R}_+^2$, constraint $g(\boldsymbol{x}, \boldsymbol{\xi}) = 1 - \boldsymbol{\xi}^\top \boldsymbol{x}$, and $\boldsymbol{\xi}^1 = (1,0)^\top$, $\boldsymbol{\xi}^2 = (1,1)^\top$, and $\boldsymbol{\xi}^3 = (1,1)^\top$. In this case, the CCP is equivalent to the following mixed-integer linear program

$$v^* = \min_{\boldsymbol{x}\in\mathbb{R}_+^2,\boldsymbol{z}\in\{0,1\}^3} \{3x_1 + 2x_2 \colon x_1 \geq z_1, x_1 + x_2 \geq z_2, x_1 + x_2 \geq z_3, z_1 + z_2 + z_3 \geq 2\} = 2.$$

CVaR approximation is

$$v^{\text{CVaR}} = \min_{\substack{x_1\geq 0,x_2\geq 0,\\ \beta\leq 0,\boldsymbol{s}\geq\boldsymbol{0}}} \left\{ 3x_1 + 2x_2 : \begin{array}{c} x_1 \geq 1 - s_1 - \beta, x_1 + x_2 \geq 1 - s_2 - \beta, \\ x_1 + x_2 \geq 1 - s_3 - \beta, \sum_{i\in[3]} s_i/3 + 0.4\beta \leq 0 \end{array} \right\}.$$

We have $v^{\text{CVaR}} = 3$. We then consider the scaled CVaR approximation (7), that is,

$$v_S^{\text{CVaR}} = \inf_{\substack{x_1\geq 0,x_2\geq 0,\\ \beta\leq 0,\boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}}} \left\{ 3x_1 + 2x_2 : \begin{array}{c} x_1\alpha_1 \geq \alpha_1 - s_1 - \beta, x_1\alpha_2 + x_2\alpha_2 \geq \alpha_2 - s_2 - \beta, \\ x_1\alpha_3 + x_2\alpha_3 \geq \alpha_3 - s_3 - \beta, \sum_{i\in[3]} s_i/3 + 0.4\beta \leq 0 \end{array} \right\}.$$

We obtain $v_S^{\text{CVaR}} = 2$. For $\alpha > 10$, an optimal solution of the scaled CVaR approximation is $x_1^* = 0, x_2^* = 1 + 5/\alpha_2^*, \beta^* = -5, \alpha_1 = 1, \alpha_2^* = \alpha_3^*, s_1^* = 6, s_2^* = s_3^* = 0$. Hence, $\lim_{\alpha_2^*\to\infty} x_2^* = 1$. Therefore, in this example, we can use the scaled CVaR approximation (7) to find the optimal solution. ◇

Example 7. Revisit Example 6 with the risk parameter $\varepsilon = 1/3$. The scaled CVaR approximation (7) can be written as

$$v_S^{\text{CVaR}} = \inf_{\substack{x_1\geq 0,x_2\geq 0,\\ \beta\leq 0,\boldsymbol{s}\geq\boldsymbol{0},\boldsymbol{\alpha}\geq\boldsymbol{e}}} \left\{ 3x_1 + 2x_2 : \begin{array}{c} x_1\alpha_1 \geq \alpha_1 - s_1 - \beta, x_1\alpha_2 + x_2\alpha_2 \geq \alpha_2 - s_2 - \beta, \\ x_1\alpha_3 + x_2\alpha_3 \geq \alpha_3 - s_3 - \beta, \sum_{i\in[3]} s_i/3 + \beta/3 \leq 0 \end{array} \right\}.$$

We obtain $v_S^{\text{CVaR}} = 3$ with $x_1^* = 1, x_2^* = 0, \beta^* = 0, \alpha_1^* = \alpha_2^* = \alpha_3^* = 1, s_1^* = s_2^* = s_3^* = 0$, which is the same solution from CVaR approximation. Hence, in this example, we cannot use the scaled CVaR approximation (7) to find the optimal solution. ◇

B Scaling in ALSO-X#

The ALSO-X# method with a bilevel structure, recently proposed by [22], can be used to solve CCP (2). In the lower-level ALSO-X#, we solve the CVaR loss function with a given upper bound of the upper-level objective value. We then check whether its optimal solution $\boldsymbol{x}^*$ satisfies the chance constraint. The upper-level ALSO-X# searches for the best upper bound of the objective value. Formally, the output of ALSO-X# admits the form:

Algorithm 4 ALSO-X# [22]

```
1: Input: Let $\delta_A$ denote the stopping tolerance parameter, and $t_L$ and $t_U$ be the known lower and upper bounds of the optimal value of CCP (2), respectively
2: while $t_U - t_L > \delta_A$ do
3:     Let $t = (t_L + t_U)/2$ and $(\boldsymbol{x}^{A\#}, \beta^{A\#})$ be an optimal solution of the lower-level ALSO-X# (13a):
           $$(\boldsymbol{x}^{A\#}, \beta^{A\#}) \in \arg \min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0} \left\{ \varepsilon\beta + \sum_{i \in [N]} p_i \left[ g(\boldsymbol{x}, \boldsymbol{\xi}^i) - \beta \right]_+ : \boldsymbol{c}^\top \boldsymbol{x} \leq t \right\} \quad (13a)$$
4:     Let $t_L = t$ if $\boldsymbol{x}^{A\#}$ satisfies the checking condition of the upper-level ALSO-X# (13b):
           $$\sum_{i \in [N]} p_i \mathbb{I} \left[ g(\boldsymbol{x}^{A\#}, \boldsymbol{\xi}^i) \leq 0 \right] \geq 1 - \varepsilon; \quad (13b)$$
       otherwise, $t_U = t$
5: end while
6: Output: A feasible solution $\boldsymbol{x}^{A\#}$ and its objective value $t_U$ to CCP (2)
```

To enhance the solution quality of Algorithm 4, we can integrate ALSO-X# with the scaling procedure proposed in Algorithm 1. In the scaled ALSO-X#, we first execute ALSO-X# Algorithm 4, and when Step 4 of ALSO-X# Algorithm 4 encounters an infeasible solution (i.e., the optimal solution $\boldsymbol{x}^{A\#}$ of the lower-level ALSO-X# (13a) violates the chance constraint $\sum_{i \in [N]} p_i \mathbb{I} \left[g(\boldsymbol{x}^{A\#}, \boldsymbol{\xi}^i) \leq 0 \right] < 1 - \varepsilon$), then we run the scaling procedure in Algorithm 1 with the same t to update the coefficient of the constraint $g(\boldsymbol{x}, \cdot)$ and see if we can find a feasible solution. If YES, we further decrease $t_U = t$; otherwise, we increase $t_L = t$. The detailed procedure for the scaled ALSO-X# algorithm is shown in Algorithm 5.

We make the following remarks about the scaled ALSO-X# algorithm Algorithm 5:

(i) Formally, the scaled ALSO-X# Algorithm 5 enhances the solution quality of ALSO-X# Algorithm 4;

(ii) We can use the solutions of the the lower-level ALSO-X# (14a) as warm-starts for the scaling procedure in Algorithm 5. However, the detailed implementation of Algorithm 5 is beyond the scope of this paper and is left for future research.

C Detailed Numerical Results in Section 4

In this section, we present the detailed numerical results from Section 4, reported in Tables 3 and 4.

Table 3: Numerical Results of a Joint CCP with Different ε

ε	n	J	Instance	CVaR Approximation	ALSO-X#			Scaled CVaR Approximation		
					Value	Time (s)	Improvement	Value	Time (s)	Improvement
0.050333	20	10	1-4-multi-3000-1	−5789.51	−5843.79	82.49	0.94%	−5855.93	514.46	1.15%
			1-4-multi-3000-2	−5779.51	−5846.49	82.29	1.16%	−5847.95	516.39	1.18%
			1-4-multi-3000-3	−5769.41	−5843.38	82.27	1.28%	−5851.34	514.92	1.42%
			1-4-multi-3000-4	−5783.57	−5851.97	86.89	1.18%	−5852.95	544.46	1.20%
			1-4-multi-3000-5	−5770.51	−5842.69	85.23	1.25%	−5847.29	535.01	1.33%
	39	5	1-6-multi-3000-1	−9847.60	−9959.17	84.92	1.13%	−9955.26	519.66	1.09%
			1-6-multi-3000-2	−9866.40	−9974.94	83.07	1.10%	−9981.91	527.56	1.17%
			1-6-multi-3000-3	−9871.75	−9977.33	82.69	1.07%	−9984.87	510.72	1.15%
			1-6-multi-3000-4	−9883.64	−10003.54	82.87	1.21%	−10000.99	537.30	1.19%
			1-6-multi-3000-5	−9852.74	−9968.08	85.73	1.17%	−9975.77	556.52	1.25%
	50	5	1-7-multi-3000-1	−15778.48	−15889.29	112.51	0.70%	−15887.76	702.28	0.69%
			1-7-multi-3000-2	−15750.66	−15865.71	105.56	0.73%	−15875.14	676.66	0.79%
			1-7-multi-3000-3	−15782.39	−15890.83	104.74	0.69%	−15898.61	665.38	0.74%
			1-7-multi-3000-4	−15774.55	−15892.31	109.25	0.75%	−15904.19	709.18	0.82%
			1-7-multi-3000-5	−15744.71	−15860.44	108.82	0.74%	−15863.01	697.77	0.75%
						Average	**1.01%**			**1.06%**
0.100333	20	10	1-4-multi-3000-1	−5831.68	−5896.96	83.84	1.12%	−5905.09	528.52	1.26%
			1-4-multi-3000-2	−5828.17	−5905.42	84.64	1.33%	−5911.56	531.10	1.43%
			1-4-multi-3000-3	−5821.63	−5904.40	82.41	1.42%	−5905.65	523.53	1.44%
			1-4-multi-3000-4	−5829.83	−5904.91	85.21	1.29%	−5910.76	541.50	1.39%
			1-4-multi-3000-5	−5819.61	−5895.10	85.31	1.30%	−5894.11	532.75	1.28%
	39	5	1-6-multi-3000-1	−9925.97	−10051.00	82.90	1.26%	−10058.75	526.54	1.34%
			1-6-multi-3000-2	−9945.20	−10074.19	83.89	1.30%	−10083.20	540.89	1.39%
			1-6-multi-3000-3	−9956.57	−10094.98	84.01	1.39%	−10093.75	536.86	1.38%
			1-6-multi-3000-4	−9970.29	−10116.59	85.54	1.47%	−10119.14	558.59	1.49%
			1-6-multi-3000-5	−9937.59	−10069.96	84.41	1.33%	−10078.05	529.11	1.41%
	50	5	1-7-multi-3000-1	−15863.65	−16007.58	104.56	0.91%	−16011.72	673.40	0.93%
			1-7-multi-3000-2	−15838.68	−15984.78	108.66	0.92%	−15999.82	691.10	1.02%
			1-7-multi-3000-3	−15864.52	−15995.88	104.93	0.83%	−16006.65	679.16	0.90%
			1-7-multi-3000-4	−15860.93	−16006.49	112.01	0.92%	−16019.04	711.60	1.00%
			1-7-multi-3000-5	−15833.63	−15982.70	109.83	0.94%	−15987.32	700.24	0.97%
						Average	**1.18%**			**1.24%**
0.200333	20	10	1-4-multi-3000-1	−5882.23	−5970.88	86.58	1.51%	−5980.71	577.25	1.67%
			1-4-multi-3000-2	−5882.54	−5971.24	86.85	1.51%	−5973.87	640.85	1.55%
			1-4-multi-3000-3	−5878.65	−5972.94	143.45	1.60%	−5978.93	906.55	1.71%
			1-4-multi-3000-4	−5885.45	−5975.44	86.07	1.53%	−5986.93	569.93	1.72%
			1-4-multi-3000-5	−5873.65	−5962.76	95.51	1.52%	−5967.68	609.77	1.60%
	39	5	1-6-multi-3000-1	−10021.39	−10186.63	86.27	1.65%	−10194.21	545.36	1.72%
			1-6-multi-3000-2	−10041.39	−10202.49	86.25	1.60%	−10213.07	545.86	1.71%
			1-6-multi-3000-3	−10060.69	−10239.33	84.70	1.78%	−10241.54	540.22	1.80%
			1-6-multi-3000-4	−10077.44	−10258.85	85.58	1.80%	−10261.55	551.13	1.83%
			1-6-multi-3000-5	−10037.88	−10209.94	85.31	1.71%	−10221.13	536.61	1.83%
	50	5	1-7-multi-3000-1	−15971.22	−16145.13	107.16	1.09%	−16158.38	698.58	1.17%
			1-7-multi-3000-2	−15953.23	−16146.98	108.54	1.21%	−16153.00	688.87	1.25%
			1-7-multi-3000-3	−15966.21	−16137.15	105.53	1.07%	−16152.76	665.84	1.17%
			1-7-multi-3000-4	−15973.34	−16161.04	108.43	1.18%	−16152.64	693.69	1.12%
			1-7-multi-3000-5	−15948.75	−16138.35	107.20	1.19%	−16149.07	693.50	1.26%
						Average	**1.46%**			**1.54%**
0.300333	20	10	1-4-multi-3000-1	−5918.46	−6021.46	83.16	1.74%	−6035.26	551.97	1.97%
			1-4-multi-3000-2	−5918.62	−6021.50	82.34	1.74%	−6031.71	565.63	1.91%
			1-4-multi-3000-3	−5917.11	−6026.06	84.21	1.84%	−6030.98	558.55	1.92%
			1-4-multi-3000-4	−5923.33	−6026.11	85.33	1.74%	−6034.92	576.17	1.88%
			1-4-multi-3000-5	−5911.10	−6017.82	84.42	1.81%	−6023.47	566.99	1.90%
	39	5	1-6-multi-3000-1	−10091.29	−10287.13	84.76	1.94%	−10291.76	545.71	1.99%
			1-6-multi-3000-2	−10110.44	−10303.21	87.72	1.91%	−10312.85	548.39	2.00%
			1-6-multi-3000-3	−10136.15	−10344.58	84.32	2.06%	−10342.07	527.24	2.03%
			1-6-multi-3000-4	−10154.42	−10375.44	85.69	2.18%	−10384.77	543.57	2.27%
			1-6-multi-3000-5	−10112.49	−10319.31	84.94	2.05%	−10333.10	543.62	2.18%
	50	5	1-7-multi-3000-1	−16046.86	−16258.91	107.78	1.32%	−16266.73	689.61	1.37%
			1-7-multi-3000-2	−16035.80	−16267.09	112.93	1.44%	−16277.78	715.99	1.51%
			1-7-multi-3000-3	−16043.21	−16256.97	104.97	1.33%	−16270.95	666.46	1.42%
			1-7-multi-3000-4	−16050.82	−16265.87	105.29	1.34%	−16268.84	681.97	1.36%
			1-7-multi-3000-5	−16030.74	−16255.95	109.07	1.40%	−16259.88	678.48	1.43%
						Average	**1.72%**			**1.81%**

Table 4: Numerical Results of Portfolio Optimization

ε	n	N	Instance with Seeds	CVaR Approximation	ALSO-X#			Scaled CVaR Approximation		
					Value	Time (s)	Improvement	Value	Time (s)	Improvement
0.050333	50	500	1231	4.18	4.08	0.53	2.33%	4.10	9.89	1.91%
			1232	1.19	1.19	0.28	0.00%	1.17	9.30	1.78%
			1233	2.61	2.58	0.29	1.26%	2.56	9.54	1.88%
			1234	1.63	1.60	0.28	2.22%	1.58	9.45	3.36%
			1235	4.69	4.55	0.52	3.01%	4.55	9.94	2.95%
		1000	1231	4.18	4.09	1.00	2.16%	4.10	19.60	1.90%
			1232	1.19	1.19	0.50	0.00%	1.16	18.47	2.01%
			1233	2.60	2.57	0.51	1.12%	2.55	19.16	1.89%
			1234	1.62	1.59	0.50	1.84%	1.57	18.34	3.09%
			1235	4.63	4.49	1.00	3.02%	4.47	19.57	3.53%
						Average	**1.70**%			**2.43**%
0.100333	50	500	1231	4.08	3.92	0.77	3.97%	3.93	10.34	3.65%
			1232	1.17	1.13	0.27	3.47%	1.13	9.41	3.00%
			1233	2.56	2.53	0.30	1.37%	2.50	9.96	2.27%
			1234	1.57	1.53	0.28	2.60%	1.50	9.41	4.51%
			1235	4.54	4.32	0.79	4.79%	4.32	10.44	4.88%
		1000	1231	4.09	3.97	1.51	3.10%	3.97	20.92	2.96%
			1232	1.16	1.16	0.55	0.00%	1.13	20.28	2.90%
			1233	2.55	2.52	0.51	1.39%	2.49	18.91	2.34%
			1234	1.56	1.51	0.53	2.67%	1.49	20.75	4.53%
			1235	4.47	4.26	1.60	4.65%	4.25	21.89	4.94%
						Average	**2.80**%			**3.60**%
0.200333	50	500	1231	3.94	3.70	0.82	6.15%	3.71	10.58	5.77%
			1232	1.14	1.11	0.53	2.42%	1.09	10.04	4.38%
			1233	2.50	2.41	0.53	3.69%	2.39	9.96	4.56%
			1234	1.50	1.41	0.52	6.15%	1.39	9.89	7.39%
			1235	4.34	4.04	0.78	6.96%	4.02	10.52	7.33%
		1000	1231	3.97	3.76	1.57	5.27%	3.74	21.74	5.84%
			1232	1.13	1.11	1.01	2.35%	1.09	20.58	4.30%
			1233	2.49	2.40	1.01	3.71%	2.38	20.16	4.45%
			1234	1.48	1.38	1.06	6.82%	1.36	20.98	8.27%
			1235	4.27	3.97	1.52	6.85%	3.96	21.26	7.12%
						Average	**5.04**%			**5.94**%
0.300333	50	500	1231	3.82	3.48	0.78	9.13%	3.47	10.90	9.15%
			1232	1.11	1.08	0.53	2.95%	1.05	10.06	5.51%
			1233	2.44	2.30	0.53	5.69%	2.27	10.01	7.00%
			1234	1.43	1.29	0.52	10.16%	1.25	10.01	12.51%
			1235	4.18	3.73	1.04	10.87%	3.74	11.08	10.45%
		1000	1231	3.85	3.51	1.59	8.70%	3.50	22.32	8.98%
			1232	1.11	1.08	1.00	2.92%	1.05	20.76	5.31%
			1233	2.43	2.29	1.01	5.62%	2.26	20.80	7.09%
			1234	1.41	1.27	1.05	10.38%	1.23	21.22	12.82%
			1235	4.10	3.62	2.06	11.56%	3.63	23.89	11.38%
						Average	**7.80**%			**9.02**%

Algorithm 5 Scaled ALSO-X#

```
**Input:** Let $\delta_A$ denote the stopping tolerance parameter, and $t_L$ and $t_U$ be the known lower and upper bounds of the optimal value of CCP (2), respectively
**while** $t_U - t_L > \delta_A$ **do**
    Let $t = (t_L + t_U)/2$ and $(\boldsymbol{x}^{A\#}, \beta^{A\#})$ be an optimal solution of the lower-level ALSO-X# (14a):
        $(\boldsymbol{x}^{A\#}, \beta^{A\#}) \in \arg \min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0} \left\{ \varepsilon\beta + \sum_{i \in [N]} p_i \left[ g(\boldsymbol{x}, \boldsymbol{\xi}^i) - \beta \right]_+ : \boldsymbol{c}^\top \boldsymbol{x} \leq t \right\}$   (14a)
    Let $t_L = t$ if $\boldsymbol{x}^{A\#}$ satisfies the checking condition of the upper-level ALSO-X# (13b), i.e., $\sum_{i \in [N]} p_i \mathbb{I} \left[ g(\boldsymbol{x}^{A\#}, \boldsymbol{\xi}^i) \leq 0 \right] \geq 1 - \varepsilon$; otherwise, run the scaling procedure in Algorithm 1 to update the coefficient of the constraint $g(\boldsymbol{x}, \cdot)$. If the solution output from the scaling procedure in Algorithm 1 is feasible to the CCP, let $t_L = t$; otherwise, $t_U = t$
**end while**
**Output:** A feasible solution $\bar{\boldsymbol{x}}^{A\#}$ and its objective value $t_U$ to CCP (2)
```

D Numerical Results Using IPOPT Solver ([43])

To further evaluate the performance of our method discussed in Section 4, we compare it with the IPOPT solver (see, e.g., [43]) for directly solving the scaled CVaR approximation (7). We consider solving the scaled CVaR approximation (7) under different upper-bound settings, that is, we consider $\alpha_U \in \{10000, 20000, 50000\}$ in the scaled CVaR approximation (7) as

$$v_S^{\text{CVaR}} = \min_{\boldsymbol{x} \in \mathcal{X}, \beta \leq 0, \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{\alpha}} \left\{ \boldsymbol{c}^\top \boldsymbol{x} \colon \varepsilon\beta + \sum_{i \in [N]} p_i s_i \leq 0, s_i + \beta \geq \alpha_i g(\boldsymbol{x}, \boldsymbol{\xi}^i),\ i \in [N], 1 \leq \alpha_i \leq \alpha_U,\ i \in [N] \right\}.$$

For all numerical results presented in this section, we use the solution of CVaR approximation (4) as the initial starting point for the IPOPT solver. Each instance is solved with a time limit of 3600 seconds, using the default settings described in [43]. If we cannot solve the instance within the time limit, we denote the result as "—." We use "Improvement" to denote the percentage of differences between the value obtained from the IPOPT solver and CVaR approximation, i.e.,

$$\text{Improvement (\%)} = \frac{\text{CVaR approximation value} - \text{value of the IPOPT solver}}{|\text{CVaR approximation value}|} \times 100\%.$$

Since the solution of the IPOPT solver may converge to a local optimum that is worse than the initial solution, the Improvement reported may be negative. If IPOPT fails to find feasible solutions within the time limit, the average (marked with parentheses) will be calculated over instances for which IPOPT does find a feasible solution.

Following the same settings as those described in Section 4.1, the results obtained using the IPOPT solver are presented in Table 5. We find that the IPOPT solver is unstable when solving the scaled CVaR approximation (7) and the performance of IPOPT is sensitive to the choice of the upper bound of $\boldsymbol{\alpha}$. Regardless of the upper bound on $\boldsymbol{\alpha}$, the IPOPT solver fails to solve all instances to local optima.

Table 5: Numerical Results of a Joint CCP with Different ε Using the IPOPT Solver

ε	n	J	Instance	CVaR Approximation	IPOPT with $\alpha_U = 10000$			IPOPT with $\alpha_U = 20000$			IPOPT with $\alpha_U = 50000$		
					Value	Improvement	Time (s)	Value	Improvement	Time (s)	Value	Improvement	Time (s)
0.050333	20	10	1-4-multi-3000-1	−5789.51	−5856.36	1.15%	42.98	−3907.21	−32.51%	896.29	−3930.35	−32.11%	1137.78
			1-4-multi-3000-2	−5779.51	−3774.00	−34.70%	1081.53	−3914.96	−32.26%	1172.65	−5847.85	1.18%	46.24
			1-4-multi-3000-3	−5769.41	−5850.97	1.41%	36.34	−5851.30	1.42%	50.44	−5852.15	1.43%	45.80
			1-4-multi-3000-4	−5783.57	−5855.41	1.24%	853.19	−5363.52	−7.26%	1120.67	−5855.88	1.25%	42.42
			1-4-multi-3000-5	−5770.51	−5847.71	1.34%	759.02	−5847.90	1.34%	48.89	−5848.12	1.34%	79.78
	39	5	1-6-multi-3000-1	−9847.60	−4715.17	−52.12%	1050.76	−5552.73	−43.61%	1958.64	−9963.15	1.17%	482.06
			1-6-multi-3000-2	−9866.40	−9986.35	1.22%	110.18	−9985.54	1.21%	756.33	—	—	3600.00
			1-6-multi-3000-3	−9871.75	−8897.43	−9.87%	275.73	−3778.40	−61.73%	294.53	−3453.30	−65.02%	339.25
			1-6-multi-3000-4	−9883.64	−10008.02	1.26%	126.00	−6258.29	−36.68%	1639.07	—	—	3600.00
			1-6-multi-3000-5	−9852.74	−9984.75	1.34%	136.82	−5755.42	−41.59%	970.66	—	—	3600.00
	50	5	1-7-multi-3000-1	−15778.48	−15902.91	0.79%	279.86	−15902.83	0.79%	82.59	—	—	3600.00
			1-7-multi-3000-2	−15750.66	−15886.40	0.86%	155.41	−15878.77	0.81%	225.76	—	—	3600.00
			1-7-multi-3000-3	−15782.39	−10966.13	−30.52%	344.26	−10540.66	−33.21%	660.41	−10191.29	−35.43%	2740.78
			1-7-multi-3000-4	−15774.55	−15901.23	0.80%	160.82	−15913.76	0.88%	316.42	−15901.05	0.80%	408.35
			1-7-multi-3000-5	−15744.71	−15881.81	0.87%	72.37	−15873.20	0.82%	101.42	−15874.82	0.83%	58.05
					Average	**−7.66%**			**−18.77%**			(**−12.45%**)	
0.100333	20	10	1-4-multi-3000-1	−5831.68	−5910.98	1.36%	960.78	−5907.93	1.31%	105.92	−5908.13	1.31%	65.95
			1-4-multi-3000-2	−5828.17	−5912.97	1.46%	53.91	−5913.35	1.46%	38.30	−5913.26	1.46%	82.27
			1-4-multi-3000-3	−5821.63	−4277.83	−26.52%	1553.48	−4103.27	−29.52%	1206.77	−5905.92	1.45%	57.44
			1-4-multi-3000-4	−5829.83	−5909.92	1.37%	42.35	−5911.07	1.39%	50.75	−5911.35	1.40%	54.37
			1-4-multi-3000-5	−5819.61	−5894.85	1.29%	51.62	−5895.32	1.30%	60.99	−4112.11	−29.34%	1120.31
	39	5	1-6-multi-3000-1	−9925.97	−10061.82	1.37%	220.22	−10062.52	1.38%	34.90	−10062.83	1.38%	46.56
			1-6-multi-3000-2	−9945.20	−10086.21	1.42%	457.22	−10084.84	1.40%	132.32	—	—	3600.00
			1-6-multi-3000-3	−9956.57	−10107.55	1.52%	88.06	−10106.39	1.50%	1129.46	—	—	3600.00
			1-6-multi-3000-4	−9970.29	−10121.97	1.52%	65.15	−10119.67	1.50%	1103.21	−10120.66	1.51%	185.91
			1-6-multi-3000-5	−9937.59	−6681.68	−32.76%	2564.60	−10083.03	1.46%	459.03	—	—	3600.00
	50	5	1-7-multi-3000-1	−15863.65	−16019.40	0.98%	90.13	−16017.69	0.97%	1704.50	−16016.49	0.96%	176.73
			1-7-multi-3000-2	−15838.68	−10946.83	−30.89%	2114.74	—	—	3600.00	−16005.89	1.06%	1223.53
			1-7-multi-3000-3	−15864.52	−12869.22	−18.88%	2280.27	−16002.81	0.87%	72.19	−16003.88	0.88%	757.04
			1-7-multi-3000-4	−15860.93	−16019.43	1.00%	86.14	−16019.11	1.00%	135.34	−16012.86	0.96%	150.48
			1-7-multi-3000-5	−15833.63	−15992.87	1.01%	126.80	—	—	3600.00	—	—	3600.00
					Average	**−6.32%**			(**−1.07%**)			(**−1.54%**)	
0.200333	20	10	1-4-multi-3000-1	−5882.23	−5980.90	1.68%	60.33	−5981.29	1.68%	92.64	−5982.58	1.71%	79.71
			1-4-multi-3000-2	−5882.54	−5975.39	1.58%	118.26	−5944.34	1.05%	1335.93	−3775.14	−35.82%	1254.68
			1-4-multi-3000-3	−5878.65	—	—	3600.00	−5981.66	1.75%	79.12	−3953.03	−32.76%	1008.98
			1-4-multi-3000-4	−5885.45	−5985.57	1.70%	80.63	−5985.85	1.71%	72.83	−3801.25	−35.41%	1060.49
			1-4-multi-3000-5	−5873.65	−5969.43	1.63%	96.34	−5969.97	1.64%	70.32	−5970.48	1.65%	90.51
	39	5	1-6-multi-3000-1	−10021.39	−10193.10	1.71%	72.18	−10192.75	1.71%	738.18	−10194.72	1.73%	224.50
			1-6-multi-3000-2	−10041.39	−10214.84	1.73%	86.70	−10212.85	1.71%	215.10	−10212.41	1.70%	444.14
			1-6-multi-3000-3	−10060.69	−10245.91	1.84%	77.05	—	—	3600.00	−10246.20	1.84%	2295.06
			1-6-multi-3000-4	−10077.44	—	—	3600.00	−10274.30	1.95%	199.66	−10270.80	1.92%	380.29
			1-6-multi-3000-5	−10037.88	−10229.68	1.91%	944.48	−10230.54	1.92%	144.37	−10228.89	1.90%	209.22
	50	5	1-7-multi-3000-1	−15971.22	−16161.54	1.19%	80.40	−16160.78	1.19%	110.50	−16160.89	1.19%	264.81
			1-7-multi-3000-2	−15953.23	—	—	3600.00	−16154.10	1.26%	115.48	−16155.65	1.27%	246.37
			1-7-multi-3000-3	−15966.21	−16161.73	1.22%	110.24	−16160.10	1.21%	128.82	−16160.41	1.22%	264.63
			1-7-multi-3000-4	−15973.34	−16166.21	1.21%	77.53	—	—	3600.00	—	—	3600.00
			1-7-multi-3000-5	−15948.75	−16150.54	1.27%	126.65	−16149.75	1.26%	102.94	—	—	3600.00
					Average	(**1.56%**)			(**1.54%**)			(**−6.76%**)	
0.300333	20	10	1-4-multi-3000-1	−5918.46	−5529.07	−6.58%	1109.47	−5967.94	0.84%	1219.53	−6035.95	1.99%	115.28
			1-4-multi-3000-2	−5918.62	−6031.59	1.91%	83.70	−6031.64	1.91%	66.21	−6032.01	1.92%	160.50
			1-4-multi-3000-3	−5917.11	−6032.30	1.95%	68.59	−6032.78	1.95%	540.16	−4373.25	−26.09%	1084.91
			1-4-multi-3000-4	−5923.33	−6037.02	1.92%	299.47	−6037.10	1.92%	134.30	−6036.37	1.91%	103.80
			1-4-multi-3000-5	−5911.10	−4329.24	−26.76%	1755.72	−6030.15	2.01%	139.35	−3832.07	−35.17%	1209.61
	39	5	1-6-multi-3000-1	−10091.29	−10301.80	2.09%	185.77	−10302.85	2.10%	1137.36	−10303.76	2.11%	290.70
			1-6-multi-3000-2	−10110.44	−10319.13	2.06%	78.86	−10320.36	2.08%	108.68	−10320.34	2.08%	299.29
			1-6-multi-3000-3	−10136.15	−10350.59	2.12%	66.52	−10351.18	2.12%	107.02	−10352.26	2.13%	248.15
			1-6-multi-3000-4	−10154.42	−10393.37	2.35%	79.72	−10395.64	2.38%	105.96	−10394.16	2.36%	193.27
			1-6-multi-3000-5	−10112.49	−10328.75	2.14%	103.98	−10328.43	2.14%	112.36	—	—	3600.00
	50	5	1-7-multi-3000-1	−16046.86	−16272.00	1.40%	134.34	—	—	3600.00	—	—	3600.00
			1-7-multi-3000-2	−16035.80	−16280.56	1.53%	96.06	−16284.22	1.55%	126.57	−16285.86	1.56%	279.63
			1-7-multi-3000-3	−16043.21	−16271.90	1.43%	105.23	−16272.40	1.43%	160.19	−16275.51	1.45%	880.99
			1-7-multi-3000-4	−16050.82	−16270.86	1.37%	100.12	−16271.18	1.37%	96.51	−16272.31	1.38%	212.77
			1-7-multi-3000-5	−16030.74	−16266.31	1.47%	132.44	−15314.50	−4.47%	120.63	—	—	3600.00
					Average	**−0.64%**			(**1.38%**)			(**−3.53%**)	