An Exact Method for Constrained Maximization of the Conditional Value-at-Risk of a Class of Stochastic Submodular Functions

Hao-Hsiang Wu

Department of Management Science, National Chiao Tung University, Hsinchu, Taiwan
hhwu2@nctu.edu.tw

Simge Küçükyavuz[*]

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL
simge@northwestern.edu

April 16, 2020

ABSTRACT: We consider a class of risk-averse submodular maximization problems (RASM) where the objective is the conditional value-at-risk (CVaR) of a random nondecreasing submodular function at a given risk level. We propose valid inequalities and an exact general method for solving RASM under the assumption that we have an efficient oracle that computes the CVaR of the random function. We demonstrate the proposed method on a stochastic set covering problem that admits an efficient CVaR oracle for the random coverage function.

Keywords: conditional value-at-risk; stochastic programming; oracle; stochastic set covering; lifting; submodular maximization

1. Introduction   We consider a class of risk-averse submodular maximization problems (RASM) recently formulated by Maehara [13]. Formally, let $V = \{1, \ldots, n\}$ be a finite set, and $\bar{\Omega}$ be a probability space. We define an outcome mapping $\sigma : 2^{|V|} \times \bar{\Omega} \to \mathbb{R}$. The random outcome $\sigma(X) : \bar{\Omega} \to \mathbb{R}$ is defined by $\sigma(X)(\omega) = \sigma(X, \omega)$ for all $X \subseteq V$ and $\omega \in \bar{\Omega}$. We assume that $\sigma(X)(\omega)$ is a nondecreasing submodular set function. With a slight abuse of notation, we refer to the submodular function $\sigma(X)$ for $X \subseteq V$ as $\sigma(x)$ for $x \in \{0,1\}^n$ interchangeably, where x is the characteristic vector of X, and the usage will be clear from the context. We measure the risk associated with this function using *conditional value-at-risk* (CVaR), where larger function values correspond to less risky random outcomes. In this context, risk measures are referred to as *acceptability functionals*. CVaR was first introduced by Artzner et al. [3] and has been widely used, especially in finance, due to its desirable properties (e.g., coherence and tractability).

Formally, let $[z]_+ = \max(z, 0)$ be the positive part of a number $z \in \mathbb{R}$. Let η be a variable that measures the *value-at-risk* (VaR) at a given risk level. For a given $\bar{x}$, the value-at-risk at a risk level $\alpha \in (0, 1]$ is defined as

$$\mathrm{VaR}_\alpha(\sigma(\bar{x})) = \max\left\{\eta : \mathbb{P}(\sigma(\bar{x}) \geq \eta) \geq 1 - \alpha, \eta \in \mathbb{R}\right\}. \tag{1}$$

For a given $\bar{x}$, the conditional value-at-risk at a risk level $\alpha \in (0, 1]$ is defined as

$$\mathrm{CVaR}_\alpha(\sigma(\bar{x})) = \max\left\{\eta - \frac{1}{\alpha}\mathbb{E}([\eta - \sigma(\bar{x})]_+) : \eta \in \mathbb{R}\right\}, \tag{2}$$

and it provides the conditional expected value of $\sigma(\bar{x})$ that is no larger than the value-at-risk at the risk level α [17]. In general, the risk level α is small, such as $\alpha = 0.05$ or 0.01. Let $\mathcal{X}$ be the deterministic constraints on the variables x. Given a risk level $\alpha \in (0, 1]$, a class of risk-averse submodular maximization problems (RASM) is defined as

$$\max_{x \in \mathcal{X}} \mathrm{CVaR}_\alpha(\sigma(x)). \tag{3}$$

Let $\mathbb{E}[\sigma(x)]$ represent the expectation of $\sigma(x)$ with respect to ω. In the next observation, we review properties of CVaR that will be useful in developing valid inequalities for RASM.

OBSERVATION 1.1 *From the definition of CVaR in* (2), *for a given $\bar{x} \in \mathcal{X}$, we have*

(i) *$CVaR_1(\sigma(\bar{x})) = \mathbb{E}[\sigma(\bar{x})]$, and because $\mathbb{E}[\sigma(\bar{x})]$ is submodular (see, e.g., [24]), so is $CVaR_1(\sigma(\bar{x}))$.*

(ii) *$CVaR_{\alpha_1}(\sigma(\bar{x})) \leq CVaR_{\alpha_2}(\sigma(\bar{x}))$ for $\alpha_1, \alpha_2 \in (0, 1]$ and $\alpha_1 \leq \alpha_2$.*

[*]Corresponding author

Due to Observation 1.1(i), when $\alpha = 1$, problem (3) is equivalent to the $\mathcal{NP}$-hard *risk-neutral* submodular maximization problem $\max_{x \in \mathcal{X}} \mathbb{E}[\sigma(x)]$ (see, e.g., [10, 11, 20] for the associated applications).

Note that for a nondecreasing submodular function $\sigma(\cdot)$, Proposition 2 of [14] shows that the *submodular inequality*

$$\sigma(x) \leq \sigma(S) + \sum_{j \in V \setminus S} (\sigma(\{j\} \cup S) - \sigma(S))x_j, \quad \forall S \subseteq V, \tag{4}$$

is valid. Ahmed and Atamtürk [1] and Yu and Ahmed [26] strengthen the submodular inequalities by lifting, under the assumption that the submodular utility function is strictly concave, increasing, and differentiable, assumptions we do not make in this paper. However, Maehara [13] shows that CVaR of a stochastic submodular function is no longer submodular for any risk level $\alpha \in (0, 1)$. The author proves that unless $\mathcal{P} = \mathcal{NP}$, there is no polynomial-time approximation algorithm with a multiplicative error for RASM under some reasonable assumptions on the given risk level. Hence it is theoretically intractable to find a solution that is close to optimal in polynomial time without any assumption on the feasible region or the probability distribution. Along this line of work, Zhou and Tokekar [28] propose a sequential greedy algorithm for the CVaR maximization problem in [13]. The authors show that the proposed algorithm gives a solution that is within a constant factor of optimal with an additional additive term that depends on the optimal value and a parameter related to the curvature of the submodular set function. However, the running time of this algorithm is exponential as it depends quadratically on the cardinality of the feasible set.

Instead of solving the generic CVaR maximization problem in [13] directly, Ohsaka and Yoshida [16] consider a specific risk-averse submodular optimization problem arising in social networks. The authors relax the problem by replacing the combinatorial decisions that determine a single set of a predetermined size with a choice of a portfolio (convex combination) of sets of the given size. They also give a polynomial-time algorithm that obtains a portfolio with a CVaR value that has a provable guarantee with a specified probability. Wilder [22] generalizes these results and proposes an approximation algorithm for maximizing the CVaR of a generic monotone *continuous* submodular function (or a portfolio of dicrete sets) with a worst-case guarantee within $(1 - \frac{1}{e})$ factor of the optimal solution. That is, a function $F : \bar{V} \to \mathbb{R}$ is continuous submodular if and only if $F(a) + F(b) \geq F(a \vee b) + F(a \wedge b)$ for any $a, b \in \bar{V} \subseteq \mathbb{R}^n_+$, where $\bar{V}_i$ is a compact subset of $\mathbb{R}$, $\bar{V} = \prod_{i=1}^n \bar{V}_i \subseteq \mathbb{R}^n_+$, and notations $\vee$ and $\wedge$ denote the coordinate-wise minimum and maximum operations, respectively (see, e.g., [6] and [7]). In contrast to this line of work that considers a portfolio of discrete sets as decision variables, we consider the discrete case, where we must choose one set of decisions for implementability purposes. In another line of work, Wu and Küçükyavuz [25] propose an exact method for the minimal cost selection of $x \in \mathcal{X}$ that satisfies the chance constraint $\mathbb{P}(\sigma(x) \geq \tau)$ for a given τ, where the authors assume that there exists a probability oracle for evaluating the chance constraint for a given solution. However, the exact method of [25] cannot be applied to the CVaR maximization problem (or its VaR maximization counterpart) directly.

Note that for a given $\bar{x} \in \mathcal{X}$ and a finite probability space $(\Omega, 2^\Omega, \mathbb{P})$ with a set of $N \in \mathbb{N}$ realizations (scenarios) $\Omega = \{\omega_1, \ldots, \omega_N\}$, $\mathbb{P}(\omega_i) = p_i$ for $i = 1, \ldots N$, CVaR is given by [17]

$$\text{CVaR}_\alpha(\sigma(\bar{x})) = \max \left\{ \eta - \frac{1}{\alpha} \sum_{i \in [N]} p_i w_i : w_i \geq \eta - \sigma_i(\bar{x}), \quad \forall i \in [N], w \in \mathbb{R}^N_+, \eta \in \mathbb{R} \right\}, \tag{5}$$

where $[N] = \{1, \ldots, N\}$ represents the set of the first N positive integers, $\sigma_i(\bar{x}) = \sigma(x)(\omega_i)$, and $\sigma_q(\bar{x}) = \text{VaR}_\alpha(\sigma(\bar{x}))$ for at least one $\omega_q \in \Omega$. Therefore, when an oracle for evaluating CVaR is not available, we can take a sampling-based approach. To this end, in [23], we apply the stochastic submodular optimization methods of [24] to problem (3) based on the CVaR definition (5). The resulting solutions are then tested out-of-sample and statistical performance guarantees are provided.

In this paper, our goal is to give (near-)optimal solutions for RASM without sampling. We start by considering high-quality (ideally optimal) feasible solutions to RASM under a true (non-trivial) distribution of the uncertain parameters assuming that there is an efficient oracle to evaluate the true value of CVaR of the random function at a given risk level. To solve RASM, we propose a decomposition method with various

classes of valid inequalities. We demonstrate our proposed methods on a risk-averse set covering problem (RASC) that admits an efficient CVaR oracle in our computational study.

2. RASM with an Exact CVaR Oracle In this section, we assume that for a given incumbent solution $\bar{x} \in \mathcal{X}$, we have an efficient oracle that computes $\text{CVaR}_\alpha(\sigma(\bar{x}))$ exactly. Under this assumption, we solve problem (3) without sampling by using a two-stage optimization model and an exact algorithm with various valid inequalities. In the proposed method, we solve a relaxed master problem (RMP) at any iteration in the form

$$\max\{\psi : (x, \psi) \in \mathcal{C}, x \in \mathcal{X}, \psi \in \mathbb{R}\}, \tag{6}$$

where ψ is a variable that is an upper bounding approximation of $\text{CVaR}_\alpha(\sigma(x))$ for a solution $x \in \mathcal{X}$, and $\mathcal{C}$ is a set of optimality cuts to be defined later. It is an approximation in that not all necessary optimality cuts may have been generated at an intermediate iteration. A *full* master problem includes all optimality cuts in $\mathcal{C}$ such that for any $x \in \mathcal{X}$, we have $\psi \leq \text{CVaR}_\alpha(\sigma(x))$. Therefore, solving the full master problem is equivalent to solving the original problem (3). In Algorithm 1, we propose a delayed constraint generation algorithm for solving problem (3) using RMP (6). The algorithm starts with a set of optimality cuts $\mathcal{C}$ (could be empty). In the while loop, we solve RMP (6) and obtain an incumbent solution (Line 3). Based on the incumbent solution, we add an optimality cut to RMP (6) (Line 5). In this algorithm, ϵ is a user-defined optimality tolerance. Let UB be the upper bound obtained from the optimal objective value of RMP (6) at each iteration. Let LB be the lower bound equal to $\text{CVaR}_\alpha(\sigma(\bar{x}))$, obtained from calling the CVaR oracle with the given incumbent solution $\bar{x} \in \mathcal{X}$ as input. If the optimality gap is below ϵ, then we terminate the algorithm and return the near-optimal solution.

Algorithm 1: A Delayed Constraint Generation Algorithm with a CVaR oracle

1 Start with an initial set of optimality cuts in $\mathcal{C}$ (could be empty), UB$= \infty$ and LB$= -\infty$;
2 **while** *UB-LB $> \epsilon$* **do**
3 $\quad$ Solve RMP (6) and obtain $(\bar{\psi}, \bar{x})$. ;
4 $\quad$ UB $\leftarrow$ the optimal objective value of RMP (6), LB$\leftarrow \text{CVaR}_\alpha(\sigma(\bar{x}))$;
5 $\quad$ Add an optimality cut to $\mathcal{C}$;
6 **end**
7 Output $\bar{x}$ as the optimal solution.

As we mentioned earlier, $\text{CVaR}_\alpha(\sigma(x))$ is not submodular in x even if $\sigma(x)$ is submodular [13]. Hence, there is no direct way use the submodular inequality (4) as a class of optimality cuts in $\mathcal{C}$. Therefore, in this section, we propose new optimality cuts that are valid for (6). Throughout, we let $\mathbf{e}_j$ be a unit vector of dimension $|V|$ whose jth component is 1, and let $\mathbf{1}$ be a $|V|$-dimensional vector with all entries equal to 1. For a given $\bar{x} \in \mathcal{X}$, we first propose an optimality cut given by

$$\psi \leq \text{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{j \in V \setminus \bar{X}} \left(\text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) - \text{CVaR}_\alpha(\sigma(\bar{x})) \right) x_j. \tag{7}$$

Before formally proving the validity of inequality (7), a few remarks are in order. It may be tempting to think that inequality (7) is in the form of a submodular inequality (4). However, we highlight that the coefficients $\left(\text{CVaR}_1(\sigma(\bar{x}+\mathbf{e}_j)) - \text{CVaR}_\alpha(\sigma(\bar{x})) \right)$ for $j \in V \setminus \bar{X}$ are different from their submodular counterparts $\left(\text{CVaR}_\alpha(\sigma(\bar{x} + \mathbf{e}_j)) - \text{CVaR}_\alpha(\sigma(\bar{x})) \right)$ for $j \in V \setminus \bar{X}$ and because $\text{CVaR}_\alpha(\sigma(\cdot))$ is not submodular, the latter coefficients are not valid. To provide some intuition for the validity of the coefficients of the variables $x_j, j \in V \setminus \bar{X}$ in inequality (7), we consider the upper bound on ψ provided by the right-hand side of the inequality (recall that $\psi \leq \text{CVaR}_\alpha(\sigma(x))$ for all $x \in \mathcal{X}$). First, observe that for $x = \bar{x}$, inequality (7) holds trivially, because $x_j = 0$ for $j \in V \setminus \bar{X}$. Now consider the point $x = \bar{x} + \mathbf{e}_j$ for some $j \in V \setminus \bar{X}$. Then the right-hand side of inequality (7) is $\text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j))$, which is a valid upper bound on ψ (because $\psi \leq \text{CVaR}_\alpha(\sigma(\bar{x} + \mathbf{e}_j)) \leq \text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j))$ from Observation 1.1 (ii)). The validity for other x is proven by exploiting the properties of $\text{CVaR}_\alpha(\sigma(x))$ given in Observation 1.1 as we show next.

PROPOSITION 2.1 *Inequality* (7) *for a given* $\bar{x} \in \mathbb{B}^n$ *is valid for RMP* (6).

PROOF. Consider a feasible point $(\hat{\psi}, \hat{x})$ to the full master problem, in other words, let $\hat{x} \in \mathcal{X}$ such that $\hat{\psi} \leq \text{CVaR}_\alpha(\sigma(\hat{x}))$. We show that $(\hat{\psi}, \hat{x})$ satisfies inequality (7) written for any $\bar{x} \in \mathcal{X}$. Let $\hat{X} = \{i \in V : \hat{x}_i = 1\}$. From the definition of CVaR in (2), it follows that because $\sigma(x)(\omega)$ is nondecreasing in x for all $\omega \in \bar{\Omega}$, $\text{CVaR}_\alpha(\sigma(x))$ is also nondecreasing in x. For the case that $\hat{X} \subseteq \bar{X}$, since $\text{CVaR}_\alpha(\sigma(x))$ is a monotonically nondecreasing function in x and $\hat{x}_j = 0$ for all $j \in V \setminus \bar{X}$, we have $\hat{\psi} \leq \text{CVaR}_\alpha(\sigma(\hat{x})) \leq \text{CVaR}_\alpha(\sigma(\bar{x}))$, which shows that inequality (7) is valid for $\hat{X} \subseteq \bar{X}$. For the case that $\hat{X} \setminus \bar{X} \neq \emptyset$, we select an arbitrary $j' \in \hat{X} \setminus \bar{X}$, where $\hat{x}_{j'} = 1$. Then

$$\hat{\psi} \leq \text{CVaR}_\alpha(\sigma(\hat{x}))$$

$$\leq \text{CVaR}_1(\sigma(\hat{x})) \tag{8a}$$

$$\leq \text{CVaR}_1(\sigma(\bar{x})) + \sum_{j \in V \setminus \bar{X}} [\text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) - \text{CVaR}_1(\sigma(\bar{x}))] \hat{x}_j \tag{8b}$$

$$= \sum_{j \in V \setminus \bar{X}} \text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) \hat{x}_j - \sum_{j \in V \setminus (\bar{X} \cup \{j'\})} \text{CVaR}_1(\sigma(\bar{x})) \hat{x}_j \tag{8c}$$

$$\leq \sum_{j \in V \setminus \bar{X}} \text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) \hat{x}_j - \sum_{j \in V \setminus (\bar{X} \cup \{j'\})} \text{CVaR}_\alpha(\sigma(\bar{x})) \hat{x}_j \tag{8d}$$

$$= \left(\sum_{j \in V \setminus \bar{X}} \text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) \hat{x}_j - \sum_{j \in V \setminus \bar{X}} \text{CVaR}_\alpha(\sigma(\bar{x})) \hat{x}_j \right) + \text{CVaR}_\alpha(\sigma(\bar{x})) \hat{x}_{j'} \tag{8e}$$

$$= \text{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{j \in V \setminus \bar{X}} (\text{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) - \text{CVaR}_\alpha(\sigma(\bar{x}))) \hat{x}_j. \tag{8f}$$

Inequality (8a) follows from Observation 1.1 (ii) that $\text{CVaR}_\alpha(\sigma(x))$ is nondecreasing in α. Inequality (8b) follows from the submodular inequality (4) written for the monotone submodular function $\text{CVaR}_1(\sigma(x))$ (see Observation 1.1 (i)) and for $S = \bar{X}$ evaluated at the point $x = \hat{x}$. Arranging terms in inequality (8b) and recalling that $\hat{x}_{j'} = 1$, we obtain equality (8c). Inequality (8d) follows from Observation 1.1 (ii), this time observing that $-\text{CVaR}_\alpha(\sigma(x))$ is nonincreasing in α. To obtain equality (8e), we add $\text{CVaR}_\alpha(\sigma(\bar{x})) \hat{x}_{j'} - \text{CVaR}_\alpha(\sigma(\bar{x})) \hat{x}_{j'}$ to inequality (8d) and reorganize the terms. Finally, equality (8f) follows from $\hat{x}_{j'} = 1$. This completes the proof. $\square$

Next, we introduce a class of valid inequalities obtained by a sequential lifting procedure [15, Proposition 1.1 in Section II.2.1]. Given $\bar{x} \in \mathcal{X}$, which is a characteristic vector of the set $\bar{X}$, consider a restriction with $x_j = 0$ for $j \in V \setminus \bar{X}$. For this restriction, we know that the *base inequality* $\psi \leq \text{CVaR}_\alpha(\sigma(\bar{x}))$ is valid, because for any x satisfying this restriction, we have $X \subseteq \bar{X}$, and $\psi = \text{CVaR}_\alpha(\sigma(x)) \leq \text{CVaR}_\alpha(\sigma(\bar{x}))$ due the property that $\text{CVaR}_\alpha(\sigma(x))$ is monotonically nondecreasing in x. However, this base inequality is not valid when the restriction is lifted. To obtain a valid inequality, we sequentially lift the base inequality with the variables $x_j, j \in V \setminus \hat{X}$. Let $j_1, \ldots, j_r$ be an ordering of the elements in $V \setminus \bar{X}$, where $r = |V \setminus \bar{X}|$. Sequential lifting following this order produces a valid inequality

$$\psi \leq \text{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{i=1}^{r} \delta_{j_i}(\bar{x}) x_{j_i}, \tag{9}$$

where $\delta_{j_t}(\bar{x})$ for $t = 1, \ldots, r$ is an *exact* lifting function given by the t-th lifting problem

$$\delta_{j_t}(\bar{x}) = -\text{CVaR}_\alpha(\sigma(\bar{x})) + \max \quad \text{CVaR}_\alpha(\sigma(x)) - \sum_{i=1}^{t-1} \delta_{j_i}(\bar{x}) x_{j_i} \tag{10a}$$

$$\text{s.t.} \quad x_{j_t} = 1 \tag{10b}$$

$$x_{j_i} = 0, \quad i = t+1, \ldots, r \tag{10c}$$

$$x \in \mathcal{X}. \tag{10d}$$

In (10a), we use the convention that for $t = 1$, the term $\sum_{i=1}^{t-1}(\cdot) = 0$. Note that in the t-th lifting problem, to obtain the coefficient of the variable x_{j_t} in inequality (9), we let $x_{j_t} = 1$ in constraint (10b), we remove the restriction on the variables preceding this variable in the lifting sequence, while keeping the restriction that the variables following x_{j_t} in the lifting sequence are fixed to zero in constraint (10c). The exact lifting problem is hard to solve since it is related to the submodular maximization problem (3). Instead of solving problem (10) exactly, we propose to solve a relaxation that provides an upper bound for the lifting coefficients. Then, we can obtain another valid inequality by using the upper bounds on the coefficients instead of the exact coefficients. We describe this approach next.

Given $\bar{x} \in \mathcal{X}$, we define $\bar{\delta}_{j_t}(\bar{x})$ as an upper bound of $\delta_{j_t}(\bar{x})$ for $t = 1, \ldots, r$, where $\bar{\delta}_{j_t}(\bar{x})$ is given by solving the following relaxation of the exact lifting problem (10):

$$\bar{\delta}_{j_t}(\bar{x}) = \max\{\text{CVaR}_\alpha(\sigma(x)), (10b), (10c), x \in \mathbb{B}^n\} - \text{CVaR}_\alpha(\sigma(\bar{x})); \tag{11}$$

here we relax constraints (10d) and remove the term $-\sum_{i=1}^{t-1} \delta_{j_i}(\bar{x})x_{j_i}$ from the objective function in problem (10). This is a valid relaxation, because $\delta_{j_i}(x) \geq 0$ when $\text{CVaR}_\alpha(\sigma(x))$ is monotonically nondecreasing in x. Furthermore, because $x_{j_t} = 1$ and $x_{j_i} = 0$ for $i = t + 1, \ldots, r$, the feasible solution with $x_{j_i}^* = 1$ for $i = 1, \ldots, t$ has the largest $\text{CVaR}_\alpha(\sigma(x^*))$ in the objective function of (11). Thus, the optimal solution of the relaxed lifting problem (11) is given by $\bar{x}^{j_t} = \mathbf{1} - \sum_{i=t+1}^{r} \mathbf{e}_{j_i}$, where we can use an efficient oracle for $\text{CVaR}_\alpha(\sigma(x))$ to obtain the optimal value of $\bar{\delta}_{j_t}(\bar{x})$ efficiently.

PROPOSITION 2.2 *Given $\bar{x} \in \mathcal{X}$, its support $\bar{X}$ and an ordering of $V \setminus \bar{X}$ given by $\{j_1, \ldots, j_r\}$, inequality*

$$\psi \leq CVaR_\alpha(\sigma(\bar{x})) + \sum_{i=1}^{r} \bar{\delta}_{j_i}(\bar{x})x_{j_i}, \tag{12}$$

where $\bar{\delta}_{j_i}(\bar{x}) = CVaR_\alpha(\sigma(\bar{x}^{j_i})) - CVaR_\alpha(\sigma(\bar{x}))$, is valid for RMP (6).

PROOF. Consider a feasible point $(\hat{\psi}, \hat{x})$. We show that $(\hat{\psi}, \hat{x})$ satisfies inequality (12) written for $\bar{x} \in \mathbb{B}^n$.

$$\psi \leq \text{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{i=1}^{r} \delta_{j_i}(\bar{x})\hat{x}_{j_i} \tag{13a}$$

$$\leq \text{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{i=1}^{r} \bar{\delta}_{j_i}(\bar{x})\hat{x}_{j_i}. \tag{13b}$$

Inequality (13a) follows from the valid inequality (9). Inequality (13b) follows from $\bar{\delta}_{j_i}(\bar{x}) \geq \delta_{j_i}(\bar{x})$ for $i = 1, \ldots, r$ since problem (11) is a relaxation of problem (10). This completes the proof. $\square$

While inequality (12) is derived by approximate lifting, next we provide some intuition on the validity of the coefficients of the variables $x_j, j \in V \setminus \bar{X}$ in inequality (12). Consider the upper bound on $\psi \leq \text{CVaR}_\alpha(\sigma(x))$ for some $x \in \mathcal{X}$ provided by the right-hand side of the inequality. First, observe that for $x = \bar{x}$, inequality (12) holds trivially, because $x_j = 0$ for $j \in V \setminus \bar{X}$. Now consider the point $x = \bar{x} + \mathbf{e}_{j_t}$, for some $t \in \{1, \ldots, r\}$. Then the right-hand side of inequality (12) is $\text{CVaR}_\alpha(\sigma(\bar{x}^{j_t}))$, which is a valid upper bound on $\psi = \text{CVaR}_\alpha(\sigma(\bar{x} + \mathbf{e}_{j_t}))$ from Observation 1.1 (ii). The approximate lifting argument establishes that the inequality is valid for other choices of x as well, because the coefficients of the variables are no smaller than those obtained from an exact lifting problem.

In Algorithm 2, we propose a greedy method that generates an inequality (12) given an incumbent $\bar{x}$. In the for loop, Line 3 determines the entry j_i by choosing the candidate index s for which $\bar{\delta}_{j_i=s}(\bar{x})$ attains its smallest value, where the candidate s has not been chosen previously.

Finally, we show the correctness of Algorithm 1 based on the inequalities (7) or (12).

PROPOSITION 2.3 *Algorithm 1 with optimality cuts (7) and/or (12) finitely converges to an optimal solution.*

Algorithm 2: GreedyUp-lifting($\alpha, \bar{x}$)

---

1 $S \leftarrow V \setminus \bar{X}$;
2 for $i = 1$ *to* r **do**
3 $\quad$ $j_i \leftarrow \arg\min_{s \in S} \bar{\delta}_{j_i = s}(\bar{x})$;
4 $\quad$ $x^* \leftarrow \mathbf{1} - \sum_{i=t+1}^{r} \mathbf{e}_{j_i}$;
5 $\quad$ $\bar{\delta}_{j_i}(\bar{x}) \leftarrow \bar{\delta}_{j_i}(x^*)$;
6 $\quad$ $S \leftarrow S \setminus \{j_i\}$;
7 end
8 Output an inequality (12) as the solution.

PROOF. From Proposition 2.1 and 2.2, for all $\bar{X} \subseteq V$ and $x \in \mathcal{X}$, we have

$$\psi \leq \mathrm{CVaR}_\alpha(\sigma(\bar{x})) \leq \mathrm{CVaR}_\alpha(\sigma(x)) + \sum_{j \in V \setminus \bar{X}} \big(\mathrm{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) - \mathrm{CVaR}_\alpha(\sigma(\bar{x}))\big)x_j,$$

and

$$\psi \leq \mathrm{CVaR}_\alpha(\sigma(\bar{x})) \leq \mathrm{CVaR}_\alpha(\sigma(x)) + \sum_{i=1}^{r} \bar{\delta}_{j_i}(\bar{x})x_{j_i}.$$

Hence, the following two linear integer programs,

$$\max\{\psi : \psi \leq \mathrm{CVaR}_\alpha(\sigma(x)) + \sum_{j \in V \setminus \bar{X}} \big(\mathrm{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) - \mathrm{CVaR}_\alpha(\sigma(\bar{x}))\big)x_j \quad \forall \bar{X} \subseteq V, x \in \mathcal{X}, \psi \in \mathbb{R}\}$$

and

$$\max\{\psi : \psi \leq \mathrm{CVaR}_\alpha(\sigma(x)) + \sum_{i=1}^{r} \bar{\delta}_{j_i}(\bar{x})x_{j_i} \quad \forall \bar{X} \subseteq V, x \in \mathcal{X}, \psi \in \mathbb{R}\},$$

are equivalent to Problem (3). Since the number of feasible solutions and the number of inequalities (7) and (12) are finite, the result follows. $\qquad\square$

Next we report our computational experience with the proposed method.

3. An Application: the Risk-Averse Set Covering Problem (RASC) We demonstrate the proposed methods on a variant of a risk-averse set covering problem (RASC). We represent RASC on a bipartite graph $G = (V_1 \cup V_2, E)$. There are two groups of nodes V_1 and V_2 in G, where all arcs in E are from V_1 to V_2. Let $V_2 := \{1, \ldots, m\}$ be a set of items. Let $S_j \subseteq V_2, j \in V_1 := \{1, \ldots, n\}$ be a collection of n subsets, where $\bigcup_{j=1}^{n} S_j = V_2$. There exists an arc $(i, j) \in E$ if $j \in S_i$ for $i \in V_1$ representing the covering relationship. In RASC, there is uncertainty on whether an arc appears in the graph. To formulate RASC, first consider a deterministic set covering problem with a feasibility set $\{x \in \mathbb{B}^n | \sum_{j \in V_1} t_{ij}x_j \geq h_i, \quad \forall i \in V_2\}$, where $h_i = 1$ for all $i \in V_2$, and $t_{ij} = 1$ if $i \in S_j$; otherwise, $t_{ij} = 0$ for all $i \in V_2 \setminus S_j, j \in V_1$. We say that an item $i \in V_2$ is covered, if there exists $x_j = 1$ for $j \in V_1$ and $t_{ij} = 1$. Now suppose that there is uncertainty on whether a chosen subset can cover an item, where constraint $t_{ij}x_j \geq h_i$ has random constraint coefficients t_{ij} or random right-hand side h_i for $i \in V_2, j \in V_1$. A related class of risk-averse problems, referred to as the probabilistic set covering problems, consider a feasible set $\{x \in \mathbb{B}^n | \mathbb{P}(\sum_{j \in V_1} t_{ij}x_j \geq h_i, \quad \forall i \in V_2) \geq 1 - \alpha\}$, where α is a given risk level and t_{ij} and/or h_i are random variables for $i \in V_2, j \in V_1$. Here the objective is to select a minimum cost selection of subsets such that the probability of covering all items is at least the risk threshold $1 - \alpha$ (see, e.g., [2, 5, 8, 19]). In contrast, in this paper, we assume that $h_i = 1$ for all $i \in V_2$, and that there is uncertainty on t_{ij} for all $i \in V_2$ and $j \in V_1$ and consider a different type of risk aversion, where we aim to choose at most k subsets from the collection so that the CVaR of the number of covered items at a risk level α is maximized. In this section, we consider an *independent probability coverage model*, where each node j has an independent probability a_{ij} of being covered by node $i \in V_1$ for $j \in S_i$, i.e., $\mathbb{P}(t_{ij} = 1) = a_{ij}$.

Let $\sigma(x)$ be a random variable representing the number of covered items in V_2 for a given x, i.e., $\sigma(x) :=$ $|\{i \in V_2 : \exists j \in V_1 \text{ with } x_j = 1 \text{ and } t_{ij} = 1\}|$. It is known that $\sigma(x)(\omega)$ is submodular for $\omega \in \Omega$ [25]. Given an integer k and $\alpha \in (0,1]$, problem (3) for RASC is

$$\max\{\text{CVaR}_\alpha(\sigma(x)) : \sum_{i \in V_1} x_i \leq k, x \in \mathbb{B}^n\}, \tag{14}$$

which is in the form of (3), where $\mathcal{X}$ is given by $\mathcal{X} = \{x \in \mathbb{B}^n : \sum_{i \in V_1} x_i \leq k\}$. Next we propose an efficient CVaR oracle to evaluate the objective function in problem (14) for a given $x \in \mathcal{X}$ under the probability distribution of interest.

PROPOSITION 3.1 *There exists a polyniamial-time oracle that computes the function $CVaR_\alpha(\sigma(x))$ for $x \in \mathcal{X}$ for RASC under the independent probability coverage model.*

PROOF. We follow the notation described in [25] to describe the CVaR oracle. From [9, 18, 21], we know that function $\mathbb{P}(\sigma(x) = b)$ is equal to the probability mass function of a Poisson binomial distribution and use a dynamic program (DP) to evaluate $A(x, i, j)$, which is defined as the probability that the selection x covers j nodes among the first i nodes of V_2 for $0 \leq j \leq i, i \in V_2$. Barlow and Heidtmann [4], Zhang et al. [27], and Wu and Küçükyavuz [25] use the DP to calculate $\mathbb{P}(\sigma(x) \geq b) := \sum_{j=b}^{m} A(x, m, j)$. Next, we show that we can use the same recursion to evaluate $\text{CVaR}_\alpha(\sigma(x))$. From the definition of $\text{VaR}_\alpha(\sigma(x))$ in (1), we have $\text{VaR}_\alpha(\sigma(x)) = \min\{j \in \mathbb{Z}_+ : \sum_{i=0}^{j} A(x, m, i) \geq \alpha\}$. The function $\text{CVaR}_\alpha(\sigma(x))$ is given by

$$\text{CVaR}_\alpha(\sigma(x)) = \text{VaR}_\alpha(\sigma(x)) - \frac{1}{\alpha}\left(\sum_{j=0}^{\text{VaR}_\alpha(\sigma(x))-1} \mathbb{P}(\sigma(x) = j)(\text{VaR}_\alpha(\sigma(x)) - j)\right)$$

$$= \text{VaR}_\alpha(\sigma(x))\left(1 - \frac{1}{\alpha}\sum_{j=0}^{\text{VaR}_\alpha(\sigma(x))-1} A(x, m, j))\right) + \frac{1}{\alpha}\sum_{j=0}^{\text{VaR}_\alpha(\sigma(x))-1} jA(x, m, j).$$

For a given x, the running time of the DP is $\mathcal{O}(nm + m^2)$, because obtaining $P(x, j)$ for all $j \in V_2$ is $\mathcal{O}(nm)$, and computing the recursion is $\mathcal{O}(m^2)$. $\square$

Table 1: Algorithm 1 with different inequalities

			Oracle-LShape			Oracle-Ineq (7)			Oracle-Ineq (12)			Oracle-Ineq (7) and (12)		
$|\mathcal{V}|$	α	k	Time	Cuts	Nodes	Time	Cuts	Nodes	Time	Cuts	Nodes	Time	Cuts	Nodes
50	0.025	3	9	2315	2363	3	801	1915	≤ 1	224	336	≤ 1	398	667
50	0.025	5	≥ 1800	33988	36405	681	16683	52677	3	726	1960	5	1462	2264
50	0.05	3	10	2319	2363	2	799	1734	≤ 1	225	366	≤ 1	397	787
50	0.05	5	≥ 1800	32548	36569	492	14978	43320	3	713	1859	5	1484	2095
100	0.025	3	1152	19661	20300	11	1845	4287	24	1511	2321	15	1816	1252
100	0.025	5	≥ 1800	21834	53054	1154	17741	97510	825	13156	46789	560	15648	45091
100	0.05	3	1205	19634	20685	9	1486	4476	24	1542	2400	15	1808	1224
100	0.05	5	≥ 1800	21980	52479	505	11408	74808	826	12911	47823	238	10684	21191
150	0.025	3	≥ 1800	18922	22045	25	1832	12753	457	6775	10868	91	3046	1389
150	0.025	5	≥ 1800	19364	43796	1640	17781	109795	≥ 1800	13468	72593	1603	20278	92923
150	0.05	3	≥ 1800	20394	21649	6	1021	1115	445	6705	10815	59	1802	8489
150	0.05	5	≥ 1800	20609	49523	≥ 1800	10832	150285	≥ 1800	14759	68158	1582	19978	75957

We study the computational performance of our proposed methods on RASC. All instances were executed on a Windows 8.1 operating system with an Intel Core i5-4200U 1.60 GHz CPU, 8 GB DRAM, and x64 based processor using C++ with IBM ILOG CPLEX 12.7. We set up the mixed-integer programming search method as traditional branch-and-cut with the lazycallback function of CPLEX, where the presolve process is turned off. The number of threads is equal to one. All other CPLEX options are set to their default values. The time limit is set to 1800 seconds. For RASC, we generate a complete bipartite graph with arcs from all nodes $i \in V_1$ to all $j \in V_2$. Let $\mathcal{V} = V_1 \cup V_2$. (Note that $V = V_1$ for RASC, because we only select nodes from V_1.) We follow the data generation scheme of [25] to set n, m, and a_{ij} for each arc (i, j). We let $k \in \{3, 5\}$ and $\alpha \in \{0.025, 0.05\}$. The size of the bipartite graphs is $|\mathcal{V}| \in \{50, 100, 150\}$.

In Table 1, we demonstrate the performance of Algorithm 1, referred to as "Oracle", using three types of valid inequalities, the L-shaped cut of [12], inequality (7) and lifted inequality (12). Note that for a given incumbent solution, $\bar{x}$, which is a characteristic vector of the set $\bar{X}$, we consider the L-shaped cut

$$\psi \leq \mathrm{CVaR}_\alpha(\sigma(\bar{x})) + \sum_{j \in V \setminus \bar{X}} \big(\mathrm{CVaR}_1(\sigma(V)) - \mathrm{CVaR}_\alpha(\sigma(\bar{x}))\big) x_j. \tag{15}$$

For $x_j = 1$ for any $j \in V \setminus \bar{X}$, inequality (15) gives a valid upper bound for $\psi = \mathrm{CVaR}_\alpha(\sigma(\bar{x}))$, given by $\mathrm{CVaR}_1(\sigma(\bar{x}))$ for any $\alpha \in (0, 1]$, from Observation 1.1 (ii). Clearly, inequalities (7) are stronger than inequalities (15), because $\mathrm{CVaR}_1(\sigma(\bar{x} + \mathbf{e}_j)) \leq \mathrm{CVaR}_1(\sigma(V))$, for $j \in V \setminus \bar{X}$. Column "Oracle-LShape" denotes Algorithm 1 with the L-shaped cut (15). Column "Oracle-Ineq (7)" denotes Algorithm 1 with inequality (7). Column "Oracle-Ineq (12)" denotes Algorithm 1 with inequality (12). Column "Oracle-Ineq (7) and (12)" denotes Algorithm 1 with both inequalities (7) and (12). Column "Time" denotes the total solution time for each instance for RMP, in seconds. Column "Cuts" denotes the total number of user cuts added to RMP. Column "Nodes" denotes the number of branch-and-bound nodes traced in RMP. From Table 1, we observe that the solution time increases as k and $|\mathcal{V}|$ increase for all methods. We observe that Oracle-LShape cannot solve most instances within the time limit. Oracle-Ineq (7) and Oracle-Ineq (12) are faster than Oracle-LShape for the instances that are solvable by Oracle-LShape within the time limit. In addition, Oracle-Ineq (7) or Oracle-Ineq (12) generates a fewer number of optimality cuts and traces a fewer number of nodes compared to Oracle-LShape.

For most of the instances with $|\mathcal{V}| \leq 100$, Oracle-Ineq (12) generates a fewer number of optimality cuts and traces a fewer number of nodes compared to Oracle-Ineq (7). However, for the instances with $|\mathcal{V}| = 150$, the solution time of Oracle-Ineq (12) is more than Oracle-Ineq (7). Recall that for a given incumbent solution $\bar{x}$, inequality (12) is generated by Algorithm 2. In Algorithm 2, we observe that for a given $\bar{x}$ and $1 \leq i \leq i' \leq r$, we have $\bar{\delta}_{j_i}(\bar{x}) \leq \bar{\delta}_{j_i'}(\bar{x}) \leq \mathrm{CVaR}_\alpha(\sigma(V_1)) - \mathrm{CVaR}_\alpha(\sigma(\bar{x}))$. From the above relation, if the size of $|\mathcal{V}|$ is large, then there may exist a large number of nodes with a high value of the lifting function $\delta_{j_i}(\bar{x})$, which is close to the coefficients in the L-shaped cut (15), i.e. $\mathrm{CVaR}_\alpha(\sigma(V_1)) - \mathrm{CVaR}_\alpha(\sigma(\bar{x}))$. This explains why Oracle-Ineq (12) does not perform well in instances with large $|\mathcal{V}|$. To benefit from the complementary strengths of inequalities (7) and (12), we add both classes of inequalities at each iteration in Algorithm 2. In Table 1, we observe that for the instances that are not solvable by either Oracle-Ineq (12) or Oracle-Ineq (7) within 1800 seconds, the solution time of Oracle-Ineq (7) and (12) is shorter. Furthermore, for a hard instance $(|V|, \alpha, k) = (150, 0.05, 5)$, only Oracle-Ineq (7) and (12) provides an optimal solution within the time limit. In conclusion, our proposed inequalities enable the effective solution of difficult instances of the RASC problem that cannot be solved with existing methods within the set time limit.

References

[1] S. Ahmed and A. Atamtürk. Maximizing a class of submodular utility functions. *Mathematical Programming*, 128(1):149–169, 2011.

[2] S. Ahmed and D. J. Papageorgiou. Probabilistic set covering with correlations. *Operations Research*, 61(2):438–452, 2013.

[3] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9 (3):203–228, 1999.

[4] R. E. Barlow and K. D. Heidtmann. Computing k-out-of-n system reliability. *IEEE Transactions on Reliability*, 33(4):322–323, 1984.

[5] P. Beraldi and A. Ruszczyński. The probabilistic set-covering problem. *Operations Research*, 50(6): 956–967, 2002.

[6] A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause. Guaranteed non-convex optimization:

Submodular maximization over continuous domains. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 111–120, 2017.

[7] L. Chen, H. Hassani, and A. Karbasi. Online continuous submodular maximization. In *Proceedings of the 21th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2018.

[8] M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273, 2012.

[9] W. Hoeffding. On the distribution of the number of successes in independent trials. *Ann. Math. Statist.*, 27(3):713–721, 1956.

[10] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM. doi: 10.1145/956750.956769.

[11] A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19), 2012.

[12] G. Laporte and F. Louveaux. The integer *L*-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[13] T. Maehara. Risk averse submodular utility maximization. *Operations Research Letters*, 43:526–529, 2015.

[14] G. Nemhauser and L. Wolsey. Maximizing submodular set functions: Formulations and analysis of algorithms. In P. Hansen, editor, *Annals of Discrete Mathematics (11) Studies on Graphs and Discrete Programming*, volume 59 of *North-Holland Mathematics Studies*, pages 279 – 301. North-Holland Mathematics Studies, 1981.

[15] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, USA, 1988.

[16] N. Ohsaka and Y. Yoshida. Portfolio optimization for influence spread. *In Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 977–985, 2017.

[17] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[18] S. M. Samuels. On the number of successes in independent trials. *The Annals of Mathematical Statistics*, 36(4):1272–1278, 1965.

[19] A. Saxena, V. Goyal, and M. A. Lejeune. MIP reformulations of the probabilistic set covering problem. *Mathematical Programming*, 121(1):1–31, 2010.

[20] R. V. Vohra and N. G. Hall. A probabilistic analysis of the maximal covering location problem. *Discrete Applied Mathematics*, 43(2):175–183, 1993.

[21] Y. H. Wang. On the number of successes in independent trials. *Statistica Sinica*, 3(2):295–312, 1993.

[22] B. Wilder. Risk-sensitive submodular optimization. *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 977–985, 2018.

[23] H.-H. Wu. *Stochastic Combinatorial Optimization with Applications in Graph Covering*. PhD thesis, University of Washington, 2018.

[24] H.-H. Wu and S. Küçükyavuz. A two-stage stochastic programming approach for influence maximization in social networks. *Computational Optimization and Applications*, 69(3):563–595, 2018.

[25] H.-H. Wu and S. Küçükyavuz. Probabilistic partial set covering with an oracle for chance constraints. *SIAM Journal on Optimization*, 29(1):690–718, 2019.

[26] J. Yu and S. Ahmed. Maximizing a class of submodular utility functions with constraints. *Mathematical Programming*, 162(1-2):145–164, 2017.

[27] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1306–1315, New York, NY, USA, 2014. ACM.

[28] L. Zhou and P. Tokekar. An approximation algorithm for risk-averse submodular optimization. *Arxiv:1807.09358v2*, 2018.