BCOL RESEARCH REPORT 15.05

Industrial Engineering & Operations Research
University of California, Berkeley, CA 94720-1777

Forthcoming in *Operations Research*

MAXIMIZING A CLASS OF UTILITY FUNCTIONS OVER THE VERTICES OF A POLYTOPE

ALPER ATAMTÜRK AND ANDRÉS GÓMEZ

Abstract. Given a polytope X, a monotone concave univariate function g, and two vectors c and d, we study the discrete optimization problem of finding a vertex of X that maximizes the utility function $c'x + g(d'x)$. This problem has numerous applications in combinatorial optimization with a probabilistic objective, including estimation of project duration with stochastic times, in reliability models, in multinomial logit models and in robust optimization. We show that the problem is $\mathcal{NP}$-hard for any strictly concave function g even for simple polytopes, such as the uniform matroid, assignment and path polytopes; and propose a 1/2-approximation algorithm for it. We discuss improvements for special cases where g is the square root, log utility, negative exponential utility and multinomial logit probability function. In particular, for the square root function, the approximation ratio is 4/5. We also propose a 1.25-approximation algorithm for a class of minimization problems in which the maximization of the utility function appears as a subproblem. Although the worst case bounds are tight, computational experiments indicate that the suggested approach finds solutions within 1-2% optimality gap for most of the instances, and can be considerably faster than the existing alternatives.

Keywords: PERT, value-at-risk, multinomial logit, reliability, assortment, submodularity, combinatorial optimization, conic quadratic optimization, robust optimization.

October 2015; May 2016

The research has been supported, in part, by grant FA9550-10-1-0168 from the Office of Assistant Secretary of Defense for Research and Engineering.

A. Atamtürk: Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720-1777 USA atamturk@berkeley.edu

A. Gómez: Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720-1777 USA a.gomez@berkeley.edu .

1. Introduction

For a rational polytope $X \subseteq \mathbb{R}^n$, let $V_X \subseteq X$ denote the set of vertices of X. We consider the discrete optimization problem

$$\max_{x \in V_X} f(x) := c'x + g(d'x), \tag{1}$$

where c and d are rational vectors in $\mathbb{R}^n$ and $g : \mathbb{R} \to \mathbb{R}$ is a monotone concave function. We refer to f as the utility function. Problem (1) includes combinatorial optimization problems, where X is an integral polytope, e.g., trees, flows, matchings, with an objective function that exhibits diminishing returns. The concave utility function f is often used to model probabilistic objectives.

In Section 2 we present four applications of problem (1) involving different utility functions. The first one is in modeling reliability of a parallel system, where g takes a negative exponential form. The second application is in the assortment planning application, where g is the multinomial logit probability function. The third application is on estimating project duration with stochastic task durations. We propose an improvement over PERT (project evaluation and review technique) by solving a maximum value-at-risk problem of the form (1) to determine a critical path, where g is the square root function. Finally, as the fourth application, we present a class of robust conic quadratic optimization problems, where g is the square root function.

The case where g is the square root function is particularly interesting, since it is commonly used to model the standard deviation. For example, in reinforcement learning, Dani et al. (2008) and Rusmevichientong and Tsitsiklis (2010) propose policies for multi-armed bandit problems that require computing

$$\max_{x \in V_X} c'x + \beta\sqrt{x'Qx}, \tag{2}$$

where $\beta > 0$ and Q is positive semi-definite. When $V_X \subseteq \{0,1\}^n$ and Q is diagonal, problem (2) is a special case of (1).

Connections to submodularity. Independently, the square root function also arises in approximating submodular functions. Given a non-negative, monotone, submodular function h, which is accessible through either a value oracle or random sampling, Goemans et al. (2009) and Balcan and Harvey (2010) consider constructing an approximation $\hat{h}$ such that $\hat{h}(S) \leq h(S) \leq \alpha\hat{h}(S)$ for $S \subseteq \{1, \cdots, n\}$[1]. They give a construction of a function of the form $\hat{h}(S) = \sqrt{\sum_{i \in S} d_i}$ with an approximation ratio of $O(\sqrt{n} \log n)$ that uses a polynomial number of queries to h, and show that the approximation ratio is close to the best possible.

Note that when $V_X \subseteq \{0,1\}^n$ and $d \geq 0$, the objective function f is submodular. Maximization of submodular functions over special structured binary polytopes has received considerable attention. Fisher et al. (1978) show that when X is a matroid polytope and the objective is any non-negative, monotone submodular function accessible through a value oracle, the *greedy* algorithm yields a 1/2-approximation, and Nemhauser et al. (1978) prove that the approximation ratio of the greedy algorithm is $(1 - e^{-1})$ if X is the uniform matroid. Other $(1 - e^{-1})$-approximation algorithms have been given when the objective is non-negative, monotone and X is either a down-monotone polytope defined with a single knapsack constraint (Sviridenko 2004) or an arbitrary matroid (Calinescu et al. 2011), and a $(1-e^{-1}-\epsilon)$-approximation algorithm is known for the case when X is a down-monotone polytope defined with multiple knapsack constraints (Kulik et al. 2009). For a non-monotone objective, Buchbinder et al. (2012) give a 1/2-approximation algorithm for the unconstrained case, and Vondrák et al. (2011) give a 0.325-approximation algorithm for down-monotone polytopes. Ahmed and Atamtürk (2011) study the polytope induced by problem (1) when $V_X = \{0,1\}^n$, $d \geq 0$ and g is strictly concave and increasing, and use submodularity to derive a strong formulation for exact algorithms. Atamtürk and Narayanan (2009) give valid inequalities for the lower level set of non-decreasing f, whereas Atamtürk and Bhardwaj (2015) give valid inequalities for the lower level set of non-increasing f.

[1]When h is accessible through sampling, the condition $\hat{h}(S) \leq h(S) \leq \alpha\hat{h}(S)$ needs to hold only in most of the sets with high probability.

In this paper we do not make use of submodularity, but exploit the structure of the function f to derive an approximation algorithm for *any* polytope X. The algorithm rounds an optimal solution to the continuous relaxation $\max_{x\in X} f(x)$ to a vertex of X. When $c'x$ and $g(d'x)$ are non-negative on V_X and g is monotone, we show that the gap of the continuous relaxation is at most 100%, and the approximation ratio of the algorithm is 1/2. Moreover, when g is the square root function, the gap is at most 25%, and the approximation ratio improves to 4/5. Both of these bounds are tight.

Exploiting the structure of the utility function f leads to a number of advantages compared to relying on submodularity alone. For the square root case and monotone f, the approximation ratio of 4/5 is better than $(1 - e^{-1}) \approx 0.63$. In the general case, the proposed approximation algorithm can be used with arbitrary polytopes. We give examples with path and assignment polytopes, which are neither down-monotone nor matroids. Moreover, we do not require that f be monotone – even if g is monotone, the function f may be non-monotone – and the approximation ratios of 4/5 for the square root or 1/2 for the general case are better than 0.325. Moreover, unlike approximation algorithms based on submodularity, we also get a tight upper bound on the optimal objective value (which is used in the robust conic quadratic optimization application discussed in Section 2.4).

The paper is organized as follows. In Section 2 we give applications in reliability modeling, assortment planning, in estimating project duration with stochastic times and in robust optimization that motivate problem (1). In Section 3 we prove $\mathcal{NP}$-hardness of (1) for simple polytopes. In Section 4 we give a high-level description of the approximation algorithm and show that the gap of the continuous relaxation of problem (1) is tight. In Section 5 we propose efficient implementations of the proposed approximation algorithms, and compare with alternatives found in the literature. In Section 6 we illustrate the empirical performance of the approximation algorithms through computational experiments for varying utility functions and polytopes. In Section 7 we conclude the paper with a few final remarks.

2. Applications

2.1. **Reliability modeling.** Given a parallel system with components N, where each component has an independent failure/malfunction probability q_i, $i \in N$, the reliability of the system is the probability that not all components malfunction simultaneously, i.e., $1 - \prod_{i=1}^{n} q_i$.

Now given a set of candidate components N with revenue r_i and malfunction probability q_i, $i \in N$, consider the problem of finding a subset $S \subseteq N$ with a revenue and reliability tradeoff. Letting $x_i = 1$ if component i is selected and 0 otherwise, the problem can be formulated as

$$\begin{aligned} &\max_{x\in V_X} \sum_{i\in N} r_i x_i + \beta \left(1 - \prod_{i\in N} q_i^{x_i}\right) \\ &= \max_{x\in V_X} \sum_{i\in N} r_i x_i + \beta \left(1 - \exp\Big(\sum_{i\in N} x_i \ln(q_i)\Big)\right), \end{aligned} \tag{3}$$

where $\beta > 0$ is the weight given to the reliability and X is an integral 0-1 polytope with additional restrictions (e.g. a cardinality constraint). Note that (3) is a special case of problem (1), where $g(d'x) = \beta\,(1 - \exp(-d'x))$ is the negative exponential function and $d_i = -\ln(q_i)$.

Problems of the form of (3) arise when considering how to allocate resources either to defend a parallel system from threats such as terrorist attacks or cyber attacks, or to attack a series system to ensure that the system is disrupted. Bier et al. (2005) and Hausken (2008) study continuous versions of such problems, in which they derive the Karush-Kuhn-Tucker conditions explicitly, and Levitin and Hausken (2008) consider a discrete version when all components are identical. Formulations similar to (3) also arise as substructures of more complicated systems (Ahmed and Papageorgiou 2013, Gen and Yun 2006).

2.2. **Assortment with multinomial logit choice model and fixed costs.** Given a set of products N, consider the problem of choosing an assortment $S \subseteq N$ satisfying a set of constraints so as to maximize the profit. In this context consumer preferences are commonly modeled with a

multinomial logit (MNL) choice model (Van Ryzin and Mahajan 1999, Chong et al. 2001). In the MNL choice model the utility of products $i \in N$ are modeled as $u_i = \mu_i + \zeta_i$, where $\mu_i \in \mathbb{R}$ is a known parameter, and ζ_i are i.i.d. standard Gumbel random variables. For an assortment S, the probability that a customer purchases item $i \in S$ is given by

$$p_i(S) = \frac{e^{\mu_i}}{1 + \sum_{j \in S} e^{\mu_j}}.$$

Consider an online advertiser that earns a variable profit c_i for displaying ad $i \in N$ as well as constant profit β if the ad is clicked. Constant profit margins have been considered by Van Ryzin and Mahajan (1999) among others. Then, an optimal set of up to k ads to display in order to maximize the expected profit is formulated as

$$\max_{S:\ |S| \leq k} \sum_{i \in S} (c_i + \beta p_i(S)).$$

More generally, the problem is stated as

$$\max_{x \in V_X} c'x + \beta \frac{d'x}{1 + d'x}, \tag{4}$$

which is a special case of problem (1) with $g(d'x) = \beta \frac{d'x}{1+d'x}$.

Various versions of the assortment problem with multinomial logit choice model have been studied in the literature (Rusmevichientong et al. 2010). For many of these the constraint set X_V is the uniform matroid (corresponding to maximum cardinality constraint). Note that formulation (4) allows, among others, differential pricing and probabilities according to the position of the ad when X_V is the assignment polytope.

2.3. **PERT and VaR critical paths.** Given a set of activities V with duration $\ell_i \geq 0$ for $i \in V$ and a set A of dependencies between pairs of activities, the Critical Path Method (CPM) computes the completion time of the overall project by finding a longest path on the directed acyclic graph $G = (V, A)$. The duration of the project is the length of the longest path, referred to as the *critical path*, between a source node s representing the start time of the first activity and a destination node t representing the completion of the last activity. The activities on the critical path are monitored carefully as delays in those activities result in delays in the completion time of the project.

The Project Evaluation and Review Technique (PERT) is a simple and prevalent generalization of CPM to incorporate the uncertainty in activity durations. The traditional PERT computes the deterministic critical path using the expected duration of the activities, and then makes an assessment of the probability of completion time of the project based on the deterministic critical path (e.g. Nahmias 2001, Chapter 9). Since the uncertainty is not taken into account when computing the deterministic critical path, the estimation of PERT can be poor. In order to incorporate the uncertainty, we consider an alternative where we compute a path with maximum value-at-risk.

The Value-at-Risk at confidence level $\alpha \in (0.5, 1)$ of duration D is

$$\mathrm{VaR}_\alpha(D) = \sup\{\ell \in \mathbb{R} : \mathbf{Pr}(D \leq \ell) \leq \alpha\}. \tag{5}$$

Observe that the value-at-risk of any path is at most the value at risk of the project. Let $\mathcal{P}$ be the set of feasible paths, let D_p be the duration of a path $p \in \mathcal{P}$ and let $D^* = \max_{p \in \mathcal{P}} D_p$ be the duration of the project. Since $\mathbf{Pr}(D^* \leq \ell) \leq \mathbf{Pr}(D_p \leq \ell)$, we have

$$\mathrm{VaR}_\alpha(D^*) = \sup\{\ell \in \mathbb{R} : \mathbf{Pr}(D^* \leq \ell) \leq \alpha\} \geq \sup\{\ell \in \mathbb{R} : \mathbf{Pr}(D_p \leq \ell) \leq \alpha\} = \mathrm{VaR}_\alpha(D_p).$$

A path with the largest value-at-risk at confidence level α thus provides the best lower bound on the value-at-risk of the project. We call such a path a VaR_α-critical path.

Let $x_{ij} = 1$ if arc $(i, j) \in A$ belongs to the path and 0 otherwise, $\delta^+(i)$ denote the set incoming arcs to i and $\delta^-(i)$ denote the set of outgoing arcs from i. If the durations are independent and

normally distributed with mean μ_{ij} and variance σ_{ij}^2, then the VaR_α-critical path corresponds to the optimal solution to

$$\begin{aligned} \max \quad & \sum_{(i,j)\in A} \mu_{ij} x_{ij} + \Phi^{-1}(\alpha) \sqrt{\sum_{(i,j)\in A} \sigma_{ij}^2 x_{ij}} \\ \text{s.t.} \quad & \sum_{(j,i)\in\delta^+(i)} x_{ji} - \sum_{(i,j)\in\delta^-(i)} x_{ij} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = t \\ 0 & \text{if } i \in V \setminus \{s,t\} \end{cases} \\ & x_{ij} \in \{0,1\}, \quad \forall (i,j) \in A, \end{aligned} \tag{6}$$

where Φ is the c.d.f of the standard normal distribution. Problem (6) is a case of problem (1), where the objective is submodular but the feasible region is neither down-monotone nor a matroid.

Example. Figure 1 illustrates a network with four activities. Using the traditional PERT method, we find that the (deterministic) critical path is given by activities $(s,2)$ and $(2,t)$, and the duration of the project is estimated to be exactly 2 at any confidence interval. On the other hand, for $\delta \geq 0.08$, the VaR-critical path is given by activities $(s,3)$ and $(3,t)$, with corresponding value-at-risk $1.8 + 2.77\delta$ at 97.5% confidence level. The $\text{VaR}_{0.975}$-critical path provides a better assessment of the risk of the project duration.

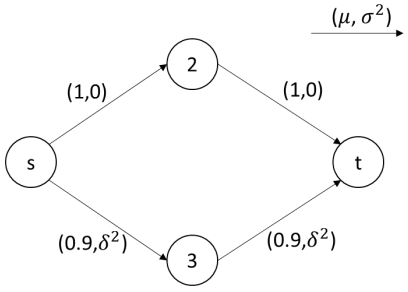

FIGURE 1. PERT network with four activities.

2.4. **Robust optimization with conic quadratic objective.** Given a feasible region $Z \subseteq \mathbb{R}^p$ and matrix $Q \in \mathcal{S}_+^p$, consider the problem with the conic quadratic objective

$$\min_{z\in Z} c'z + \sqrt{z'Qz}. \tag{7}$$

Problem (7) arises when minimizing value-at-risk as a mean-risk objective (e.g. Ahmed 2006, Atamtürk and Narayanan 2008). The reader is referred to Lobo et al. (1998) and Alizadeh and Goldfarb (2003) for other applications conic quadratic optimization.

We describe a robust formulation of problem (7), generalizing the approach given by Bertsimas and Sim (2003, 2004) for linear objectives. Let c_0 and $Q_0 \succeq 0$ be the nominal mean and covariance and $N = \{1,\dots,n\}$ be a set of potential events, each of which may increase the mean and covariance by $c_i \geq 0$ and $Q_i \succeq 0$, $\in N$. For example, Q_i may represent the increase in volatility and correlations in financial markets for a particular stress scenario. Then $c(S) = c_0 + \sum_{i\in S} c_i$ and $Q(S) = Q_0 + \sum_{i\in S} Q_i$ are the mean and covariance when events $S \subseteq N$ are realized. The goal of the robust optimization is to find a solution that minimizes the worst objective given that only a small number, $k \leq n$, events are realized, i.e.,

$$\min_{z\in Z} \max_{S\subseteq N:|S|\leq k} c(S)'z + \sqrt{z'Q(S)z}. \tag{8}$$

Letting

$$X = \left\{ x \in [0,1]^n : \sum_{i=1}^n x_i \leq k \right\}, \tag{9}$$

problem (8) can be equivalently stated as

$$\min_{z\in Z}\max_{x\in V_X} c_0'z + \sum_{i\in N}(c_i'z)\,x_i + \sqrt{z'Q_0z + \sum_{i\in N}(z'Q_iz)\,x_i}. \tag{10}$$

Observe that for a given value of z the inner maximization problem is of the form (1) with $g(x) = \sqrt{\xi + x}$, where $\xi \geq 0$ does not depend on x.

3. Complexity

In this section we show that problem (1) is $\mathcal{NP}$-hard for the uniform matroid polytope, the path polytope, and the assignment polytope. Note that the results are valid even when g is given explicitly.

Proposition 1. *For $c \geq 0$, $d \geq 0$ and any strictly concave function g, problem*

$$\max_{x\in\{0,1\}^n}\left\{c'x + g(d'x) : \sum_{i=1}^n x_i = k\right\} \tag{11}$$

is $\mathcal{NP}$-hard.

Proof. Ahmed and Atamtürk (2011) show that the *unconstrained* problem

$$\max_{x\in\{0,1\}^n} -c'x + g(d'x) \tag{12}$$

is $\mathcal{NP}$-hard when g is the negative exponential function. We first extend their proof for *any* strictly concave functions. The proof is by reduction from Partition: Given positive numbers c_i, $i \in N$, Partition calls for a set $S \subseteq N$ such that $\sum_{i\in S} c_i = \sum_{i\in N\setminus S} c_i$. If $\sum_{i\in N} c_i = 0$, then $S = \emptyset$ is a trivial answer. Otherwise, let $y^* = \arg\max_{y\in\mathbb{R}}\{g(y) - g(1)y\}$, which can be computed in polynomial time since it is the solution of a convex problem (note that if g is differentiable, y^* can be computed in closed form). If necessary, by negating and/or scaling the data, we may assume without loss of generality that $\sum_{i\in N} c_i = 2y^*$. Then, there exists $S \subseteq N$ with the desired property if and only if

$$\max_{x\in\{0,1\}^n} -g(1)c'x + g(c'x) = -g(1)y^* + g(y^*),$$

which is true if and only if $c'x = y^*$.

Next we show that (11) is $\mathcal{NP}$-hard. Given an instance of (12), let $\bar{c} = \max_{i\in N} c_i$. Then problem (12) can be written as

$$\begin{aligned}&\max_{S\subseteq N} -\bar{c}\,|S| + \sum_{i\in S}(\bar{c} - c_i) + g\left(\sum_{i\in S} d_i\right)\\ &= \max_{k=0,\cdots,n}\left(-\bar{c}k + \max_{x\in\{0,1\}^n}\left\{(\bar{c}-c)'x + g(d'x) : \sum_{i=1}^n x_i = k\right\}\right),\end{aligned}$$

which can be solved by solving n problems of the form (11). □

We will prove $\mathcal{NP}$-hardness of problem (1) for special polytopes by reduction from problem (11).

3.1. Uniform matroid polytope.

The uniform matroid polytope is given by

$$X = \left\{x \in [0,1]^n : \sum_{i=1}^n x_i \leq k\right\}, \tag{13}$$

with $k \in \mathbb{N}$.

Proposition 2. *Problem (1) is $\mathcal{NP}$-hard when X is the uniform matroid polytope and g is strictly concave.*

Proof. Observe that if $c, d \geq 0$, g is non-decreasing and X is the uniform matroid polytope, then any optimal solution x^* satisfies

$$\sum_{i=1}^{n} x_i = k \tag{14}$$

and is also an optimal solution for (11). In general, given an instance of (11), let

$$\bar{d} = \min_{j=1,\ldots,n} \left\{ c_j + g\left(\sum_{i=1}^{n} d_i\right) - g\left(-d_j + \sum_{i=1}^{n} d_i\right) \right\}$$

and consider the optimization problem

$$\max_{x \in \{0,1\}^n} \left\{ \sum_{i=1}^{n} (c_i - \bar{d}) x_i + g\left(\sum_{i=1}^{n} d_i x_i\right) : \sum_{i=1}^{n} x_i \leq k \right\}. \tag{15}$$

Observe that any optimal solution x^* of (15) satisfies (14) by construction. Moreover, the objective value of any solution satisfying (14) is $c'x + g(d'x) - \bar{d}k$ (where $\bar{d}k$ is a constant), and therefore we see that x^* is an optimal solution of (11).

□

3.2. **Path polytope.** The path polytope on an acyclic directed graph is described in (6).

Proposition 3. *Problem* (1) *is $\mathcal{NP}$-hard when X is the path polytope and g is strictly concave.*

Proof. Consider an instance of problem (11). We construct an equivalent path instance on a directed graph $G = (V, A)$ with $O(n^2)$ vertices and arcs. Let $V = \{(i, j) \in \mathbb{Z}^2 : 1 \leq i \leq n+1, 0 \leq j \leq k\}$ be the set of vertices. G has an arc for the vertex pair (i, j), $(i+1, j)$ with value $(0, 0)$, which corresponds to setting $x_i = 0$ in problem (11); G has an arc for pair (i, j), $(i+1, j+1)$ with value (c_i, d_i), which corresponds to setting $x_i = 1$ in problem (11). Note that a path between $(1, 0)$ and (i, j) corresponds to a choice of $x_1, \ldots, x_{i-1}$ such that $\sum_{l=1}^{i-1} x_l = j$, thus paths between $(1, 0)$ and $(n+1, k)$ correspond to feasible solutions of problem (11). Therefore, we have that the set of optimal solutions of problem (11) correspond to set of longest paths between $(1, 0)$ and $(n+1, k)$.

□

3.3. **Assignment polytope.** The $m \times m$ assignment polytope is given by

$$X = \left\{ x \in \mathbb{R}^{m \times m} : \sum_{i=1}^{m} x_{ij} = 1, \sum_{j=1}^{m} x_{ij} = 1, x_{ij} \geq 0 \right\}.$$

Proposition 4. *Problem* (1) *is $\mathcal{NP}$-hard when X is the assignment polytope and g is strictly concave.*

Proof. Consider an instance of problem (11). We construct an equivalent $n \times n$ assignment instance. Let x_{ij} have objective values (c_j, d_j) for $i = 1, \cdots, k$, and values $(0, 0)$ for $i = k+1, \cdots, n$. Note that given any assignment we can construct a feasible solution $\bar{x}$ to problem (11) with same objective value: set $\bar{x}_j = 1$ if $x_{ij} = 1$ for some $i \leq k$, and set $\bar{x}_j = 0$ otherwise. Moreover, any feasible solution to (11) corresponds to at least one assignment by construction. Therefore, we get that an optimal solution to (11) can be obtained by finding an optimal assignment. □

4. Approximation analysis

In this section we discuss approximation algorithms for problems (1) and (10) that use the continuous relaxation

$$\max_{x \in X} c'x + g(d'x). \tag{16}$$

In Section 4.1 we describe an algorithm for the maximization problem (1), in Section 4.2 we describe the approach used for the robust problem (10), in Section 4.3 we prove constant approximation ratios for both methods, and in Section 4.4 we give a simple extension for unbounded polyhedra.

4.1. **Approximation algorithm.** The approximation algorithm for (1) consists of rounding an optimal solution to the continuous relaxation (16) to particular extreme points X if the continuous optimal solution is not an extreme point itself.

Proposition 5. *If there exists an optimal solution to problem* (16)*, then there exists an optimal solution on an edge of* X*.*

Proof. Let x_0 be an optimal solution to (16). Consider the linear optimization problem

$$\begin{aligned} &\max c'x \\ &\text{s.t. } d'x = d'x_0 \\ &\quad x \in X. \end{aligned} \tag{17}$$

Observe that $\bar{X} = X \cap \{x \in \mathbb{R}^n : d'x = d'x_0\}$ is a nonempty polytope. Let x^* be an optimal extreme point of $\bar{X}$. Note that x^* is also an optimal solution to problem (16). Let F_0 be the zero-dimensional face of $\bar{X}$ defined by active constraints at x^*. Removing constraint (17) from F_0 results in a face F_1 of X of dimension at most 1. Therefore $x^* \in F_1$ and F_1 is either an extreme point or an edge of X, as desired. □

If the face F_1 described in Proposition 5 has zero-dimension, then F_1 is an optimal solution to problem (1) and the approximation algorithm returns it as the solution. Otherwise, F_1 is an edge of X, and the approximation algorithm computes the two extreme points of F_1 and returns the one with the higher objective value.

Proposition 6. *Given a rational polytope* X *and a rational optimal solution* x_0 *to the continuous problem* (16)*, there exists a polynomial-time algorithm that finds an optimal solution* x^* *of problem* (16) *on an edge* E *of* X *and, if* x^* *is not an extreme point of* X*, two extreme points* x_1 *and* x_2 *of* E*.*

Proof. Computing F_1 as described in Proposition 5 requires finding an optimal extreme point to a linear program, which can be done using a polynomial time algorithm for linear programs that finds an (interior) optimal primal-dual pair, and then using the polynomial algorithm of Megiddo (1991) to find an optimal extreme point. Therefore, computing x^* on an edge E of X can be done in polynomial time. If x^* is not an extreme point of X, the two extreme points of E can be found similarly by solving two auxiliary linear programs by converting tight inequalities defining F_1 into equalities. □

The approximation algorithm is displayed in Algorithm 1. We first solve the continuous relaxation (line 1) and compute an optimal x^* on an edge of X. If optimal x^* is a vertex, then we have found an optimal solution to the discrete problem (lines 3-4). Otherwise, we compute the two extreme points of the edge (line 6) and then we select the best vertex (line 7).

Algorithm 1 Approximation algorithm.

Input: X, polytope; f, objective function with $f(x) = c'x + g(d'x)$.
Output: x, a vertex of X.
1: $x_0 \leftarrow \max_{x \in X} f(x)$
2: Compute an optimal x^* on edge E.
3: **if** x^* is vertex **then**
4: $\quad x \leftarrow x^*$
5: **else**
6: $\quad (x_1, x_2) \leftarrow \texttt{vertices}(E)$
7: $\quad x \leftarrow \arg\max_{\{x_i : i=1,2\}} f(x_i)$
8: **end if**
9: **return** x

Corollary 1. *If the convex problem* (16) *can be solved in polynomial time, then Algorithm 1 is a polynomial-time algorithm.*

Interior point algorithms can be used to solve convex problems in polynomial time if used with a self-concordant barrier functions (Nemirovski and Todd 2008). In particular, if g is a negative exponential, logarithmic or power function, then (16) can be solved in polynomial time (see chapters 5.3.1 and 5.3.2 of Nesterov and Nemirovskii (1994)).

4.2. **Approximation for robust conic quadratic programs.** Note that in order to have an approximation algorithm to problem (10), an *upper bound* to problem (1) needs to be computed (instead of a lower bound). To this end, we propose to solve the *approximate robust optimization problem*

$$\min_{z\in Z}\max_{x\in X} c_0'z + \sum_{i\in N}(c_i'z)\,x_i + \sqrt{z'Q_0z + \sum_{i\in N}(z'Q_iz)\,x_i}, \tag{18}$$

where the feasible region of the inner maximization problem is relaxed from V_X to X. We evaluate the strength of formulation (18) in terms of the *translated approximation ratio*

$$\Delta = \frac{\omega_c - \omega_n}{\omega_d - \omega_n}, \tag{19}$$

where ω_n, ω_c and ω_d are the optimal objective values of the nominal problem (7), the continuous relaxation (18) and the discrete problem (10), respectively. Note that when $\omega_n \geq 0$, a translated approximation ratio Δ implies a Δ-approximation algorithm in the usual sense for the minimization problem (10)[2].

4.3. **Approximation ratio.** We now characterize the approximation ratio of Algorithm 1. In this section we assume that $c'x \geq 0$ and $g(d'x) \geq 0$ over V_X. If x^* is a vertex of X, then the algorithm is exact. Now let x_1 and x_2 be two vertices of X such that $x^* = (1-\lambda)x_1 + \lambda x_2$ with $0 < \lambda < 1$, $a_i = c'x_i$ and $b_i = d'x_i$, $i = 1, 2$. Without loss of generality, assume that $g(b_1) \leq g(b_2)$. We can bound the gap between the optimal objective of (16) and the best extreme point solution by

$$\begin{aligned}\rho(\lambda) :=& \frac{f\left((1-\lambda)x_1 + \lambda x_2\right) - \max\left\{f(x_1), f(x_2)\right\}}{\max\left\{f(x_1), f(x_2)\right\}} \\ =& \frac{a_1 + \lambda(a_2 - a_1) + g\left(b_1 + \lambda(b_2 - b_1)\right) - \max\left\{a_1 + g(b_1), a_2 + g(b_2)\right\}}{\max\left\{a_1 + g(b_1), a_2 + g(b_2)\right\}}.\end{aligned} \tag{20}$$

For any $\lambda \in [0, 1]$ we have

$$\max\left\{a_1 + g(b_1), a_2 + g(b_2)\right\} \geq a_1 + g(b_1) + \lambda\left(a_2 + g(b_2) - a_1 - g(b_1)\right), \tag{21}$$

and

$$\max\left\{a_1 + g(b_1), a_2 + g(b_2)\right\} \geq a_2 + g(b_2) \geq g(b_2). \tag{22}$$

Using (21) in the numerator of (20), and (22) in the denominator, we get

$$\rho(\lambda) \leq \frac{g\left(b_1 + \lambda(b_2 - b_1)\right) - g(b_1) - \lambda(g(b_2) - g(b_1))}{g(b_2)} =: \rho_+(\lambda). \tag{23}$$

Remark 1. A sufficient condition for $a_2 \geq 0$ is that $c \geq 0$ and x is nonnegative. More generally, if $c'x \geq -k$ for $x \in V_X$ and $k \geq 0$, we can use (23) to bound the gap of the problem

$$\max_{x\in V_X} k + c'x + g(d'x). \tag{24}$$

Lemma 1. *If g is concave, non-decreasing, differentiable, for any $\ell \leq b_1$,*

$$\rho(\lambda) \leq \frac{g\left(\ell + \lambda(b_2 - \ell)\right) - g(\ell) - \lambda(g(b_2) - g(\ell))}{g(b_2)}.$$

[2]It is not possible to have an approximation algorithm for (10) when $\omega_n < 0$: it is possible to construct non-trivial instances where $\omega_d = 0$, and every other feasible solution is arbitrarily bad in comparison.

Proof. Since g is non-decreasing, as by assumption $g(b_1) \leq g(b_2)$, we have $b_1 \leq b_2$. Taking the derivative of (23) with respect to b_1, we get that

$$\rho'_+(\lambda) = \frac{1}{g(b_2)}(1-\lambda)\big(g'\left(b_1 + \lambda(b_2 - b_1)\right) - g'(b_1)\big).$$

Since g is concave, g' is non-increasing and $g'\left(b_1 + \lambda(b_2 - b_1)\right) - g'(b_1) \leq 0$. The function ρ_+ is then non-increasing in b_1, and setting $b_1 = \ell$, we get the upper bound. □

Lemma 2. *If g is concave, non-increasing, differentiable, for any $\ell \geq b_2$,*

$$\rho(\lambda) \leq \frac{g\left(\ell + \lambda(b_2 - \ell)\right) - g(\ell) - \lambda(g(b_2) - g(\ell))}{g(b_2)}.$$

Proof. The proof is analogous to the proof of Lemma 1. □

We now give approximation ratios for different forms of function g.

Proposition 7 (Root function)**.** *If $g(z) = z^p$ with $0 < p < 1$ and $d'x \geq 0$ for all $x \in V_X$, the approximation ratio of Algorithm 1 is*

$$\frac{1}{1 + \left(\frac{1}{p}\right)^{\frac{p}{p-1}}(1-p)}.$$

Proof. We use Lemma 1 with $\ell = 0$ to get

$$\rho(\lambda) \leq \frac{(\lambda b_2)^p - \lambda b_2^p}{b_2^p} = \lambda^p - \lambda. \tag{25}$$

Expression (25) is maximized at $\lambda^* = \left(\frac{1}{p}\right)^{\frac{1}{p-1}}$, and $\rho(\lambda^*) \leq \left(\frac{1}{p}\right)^{\frac{p}{p-1}}(1-p)$. The continuous relaxation has then a gap of at most $\left(\frac{1}{p}\right)^{\frac{p}{p-1}}(1-p)$ and the result follows. □

Corollary 2. *If $g(z) = \sqrt{z}$ and $d'x \geq 0$ for all $x \in V_X$, Algorithm 1 has 4/5 approximation ratio.*

Corollary 3. *The translated approximation ratio for problem* (10) *satisfies $\Delta \leq 1.25$.*

Remark 2. The gap of the continuous relaxation of the square root function is tight: consider the case in which X has only two extreme points x_1 and x_2, with $(a_1, b_1) = (1, 0)$ and $(a_2, b_2) = (0, 1)$; the value of an optimal extreme point solution is 1, while the optimal solution to the continuous relaxation is $0.75x_1 + 0.25x_2$, with value 1.25. We can similarly prove that the approximation ratios of 0.72 and 0.68 (approximately) of the cubic and quartic roots are tight.

Proposition 8 (Monotone function)**.** *When g is nonnegative and monotone, Algorithm 1 has an approximation ratio 1/2.*

Proof. From (23), we have that

$$\begin{aligned}
\rho(\lambda) \leq & \frac{g\left(b_1 + \lambda(b_2 - b_1)\right) - g(b_1) - \lambda(g(b_2) - g(b_1))}{g(b_2)} \\
\leq & \frac{g(b_2) - g(b_1) - \lambda(g(b_2) - g(b_1))}{g(b_2)} && ((g\left(b_1 + \lambda(b_2 - b_1)\right) \leq g(b_2) \text{ by monotonicity})) \\
\leq & \frac{g(b_2) - g(b_1)}{g(b_2)} && ((\text{since } g(b_1) \leq g(b_2) \text{ w.l.o.g})) \\
= & 1 - \frac{g(b_1)}{g(b_2)} \leq 1.
\end{aligned}$$

The continuous relaxation has therefore a gap of at most 100%. In other words, selecting the better of the two extreme points solutions results in a 1/2-approximation guarantee. □

Example: Exponential utility. Consider the exponential utility function, $g(z) = 1 - e^{-z}$. Consider the case where X has only two extreme points x_1 and x_2, with $(a_1, b_1) = (1, 0)$ and $(a_2, b_2) = (0, b)$, and let $\lambda^* = -\frac{1}{b} \ln\left(\frac{1-e^{-b}}{b}\right)$. It can be shown that the gap $\rho(\lambda^*)$ defined in (20) approaches 1 as b goes to infinity. For instance, when $b = 100$, $\rho(\lambda^*) \approx 0.94$.

Example: Logarithmic utility. Consider the logarithmic utility function, $g(z) = \ln(1 + z)$. Consider the case where X has only two extreme points x_1 and x_2, with $(a_1, b_1) = (\ln(1 + b), 0)$ and $(a_2, b_2) = (0, b)$, and let $\lambda^* = \frac{1}{\ln(1+b)} - \frac{1}{b}$. It can be shown that the gap approaches 1 as b goes to infinity. For instance, when $b = 10^{40}$, $\rho(\lambda^*) \approx 0.94$.

Example: MNL probability function. Consider the MNL probability function, $g(z) = 1 - \frac{1}{1+z}$. Consider the case where X has only two extreme points x_1 and x_2, with $(a_1, b_1) = (\frac{b}{1+b}, 0)$ and $(a_2, b_2) = (0, b)$, and let $\lambda^* = \frac{\sqrt{1+b}-1}{b}$. It can be shown that the gap approaches 1 as b goes to infinity. For instance, when $b = 1000$, $\rho(\lambda^*) \approx 0.94$.

4.4. **Unbounded polyhedra with integrality constraints.** When X is an unbounded polyhedron, the solution x^* given by Proposition 5 may lie on an extreme ray, in which case it is not possible to find two extreme points. We consider, in this section, the particular case when X is a polyhedron with integral extreme points and rational rays and the feasible region consists of all integer points of X.

In this case we can still find an optimal solution to the continuous relaxation x^* on a face F_1 with dimension at most one as before (Proposition 5). If F_1 is an extreme point or an edge, then the algorithm is identical to the polytope case. If F_1 is a ray, let x_1 be its extreme point. In this case, we let $x_2 = x_1 + \gamma(x^* - x_1)$, where γ is the least common multiple of the denominators of the entries of $(x^* - x_1)$. Since x^* is a convex combination of x_1 and x_2, the analysis of Section 4.3 holds.

5. Implementation

The general algorithm proposed in Section 4.1, despite being polynomial, may not be efficient in practice. In this section we discuss improvements of Algorithm 1 that exploit additional information on X or function g, and we also give an explicit formulation of problem (18).

5.1. **0-1 polytopes.** In many of the applications of problem (1), including those discussed in Section 2, X is a binary polytope, i.e., a polytope with 0-1 valued vertices. Given an optimal solution x^* on an edge of the polytope, we describe a simple $O(n)$ procedure for finding the extreme points x_1 and x_2, provided that $x^* \neq \frac{x_1+x_2}{2}$. Observe that because x^* is a convex combination of only two binary extreme points, i.e., $x^* = \lambda x_1 + (1 - \lambda)x_2$, each component of x^* is either 0, 1, λ or $1 - \lambda$ for some $\lambda \in (0, 1)$.

Denote by $x(i)$ the i-th coordinate of vector x. Then for each $i = 1, \ldots, n$, one of the following cases is true:

$x^*(i) = 0$**:** Let $x_1(i) = 0$ and $x_2(i) = 0$.
$x^*(i) = \lambda$**:** Let $x_1(i) = 1$ and $x_2(i) = 0$.
$x^*(i) = 1 - \lambda$**:** Let $x_1(i) = 0$ and $x_2(i) = 1$.
$x^*(i) = 1$**:** Let $x_1(i) = 1$ and $x_2(i) = 1$.

Note that if $\lambda = 1/2$, it is not obvious how to round the components to 0 and 1 consistently for each component. From the previous observation, the condition $x^* \neq \frac{x_1+x_2}{2}$ is equivalent to having no $x^*_i = \frac{1}{2}$, which is easily verifiable.

5.2. **Lagrangian relaxation.** In many cases there may be efficient algorithms for maximizing a linear function over X, while directly solving the nonlinear problem (16) may be computationally more challenging. Here we give a Lagrangian relaxation approach that exploits an oracle for the linear problem.

Suppose g is concave and increasing, the inverse h of g is differentiable and the derivative h' has an inverse h'^{-1}. Then

$$\begin{aligned}\max_{x\in X} f(x) &= \max_{x\in X} c'x + g(d'x)\\ &= \max_{x\in X,\ t\in\mathbb{R}} \{c'x + t : t \le g(d'x)\}\\ &= \max_{x\in X,\ t\in\mathbb{R}} \{c'x + t : h(t) \le d'x\}. \end{aligned} \tag{26}$$

Let $y \ge 0$ be the dual variable associated with the constraint $h(t) \le d'x$. Since h is convex, the Lagrangian dual with respect to this constraint has no duality gap. We obtain

$$\max_{x\in X} f(x) = \min_{y\ge 0} \max_{x\in X, t\in\mathbb{R}} (c' + yd')x + t - yh(t) = \min_{y\ge 0} \theta(y). \tag{27}$$

Note that $\theta(y)$ is a one-dimensional convex function, which can be optimized using line search methods.

We now describe how to evaluate $\theta(y)$. Taking derivatives in (27) with respect to t, we find that $1 - yh'(t) = 0$, or $t = h'^{-1}\left(\frac{1}{y}\right)$. Replacing in (27), we obtain

$$\theta(y) = h'^{-1}\left(\frac{1}{y}\right) - yh\left(h'^{-1}\left(\frac{1}{y}\right)\right) + \max_{x\in X}(c' + yd')x,$$

which can be computed efficiently using the linear oracle.

Example: Square root function. For $g(z) = \beta\sqrt{\xi+z}$, we have that $h(z) = \left(\frac{z}{\beta}\right)^2 - \xi$, $h'(z) = \frac{2z}{\beta^2}$, $h'^{-1}(z) = \frac{\beta^2 z}{2}$ and

$$\theta(y) = \frac{\beta^2}{4y} + y\xi + \max_{x\in X}(c' + yd')x.$$

Example: Exponential utility. For $g(z) = 1 - e^{-z}$, we have that $h(z) = -\ln(1-z)$, $h'(z) = \frac{1}{1-z}$, $h'^{-1}(z) = 1 - \frac{1}{z}$ and

$$\theta(y) = 1 - y + y\ln(y) + \max_{x\in X}(c' + yd')x.$$

Example: Logarithmic utility. For $g(z) = \ln(1+z)$, we have that $h(z) = e^z - 1$, $h'(z) = e^z$, $h'^{-1}(z) = \ln(z)$ and

$$\theta(y) = y - \ln(y) - 1 + \max_{x\in X}(c' + yd')x.$$

Example: MNL probability function. For $g(z) = \frac{z}{1+z}$, we have that $h(z) = \frac{z}{1-z}$, $h'(z) = \frac{1}{(1-z)^2}$, $h'^{-1}(z) = 1 - \frac{1}{\sqrt{z}}$ and

$$\theta(y) = (1 - \sqrt{y})^2 + \max_{x\in X}(c' + yd')x.$$

Proposition 9. *Problem* (27) *can be solved with* $O\left(\ln\left(\frac{g'(d'x_L)}{\epsilon}\right)\right)$ *calls to the linear optimization oracle, where* $\epsilon > 0$ *is the precision and* $x_L = \arg\max_{x\in X} c'x$.

Proof. Let $y^* = \arg\min_{y\ge 0}\theta(y)$, and let x^* and t^* be optimal solutions to (26). Since $t^* = g(d'x^*)$ and $t^* = h'^{-1}\left(\frac{1}{y^*}\right)$, we have that $y^* = \frac{1}{h'(g(d'x^*))}$. Using the inverse function theorem, we get that $y^* = g'(d'x^*)$. Moreover, $c'x_L \ge c'x^*$ and $d'x_L \le d'x^*$ and, since g is concave, g' is non-increasing

and in particular $g'(d'x^*) \leq g'(d'x_L)$. Therefore $0 \leq y^* \leq g'(d'x_L)$, and solving problem (27) using golden section search requires

$$\left\lceil \frac{\ln\left(\frac{g'(d'x_L)}{\epsilon}\right)}{-\ln(0.618)} \right\rceil + 2$$

calls to the linear oracle. □

Remark 3. Let $y^* = \arg\min_{y\geq 0}\theta(y)$. If $\max_{x\in X}(c+y^*d)'x$ has a unique optimal solution x^*, then x^* is an optimal solution to problems (16) and (1). Otherwise, if $\max_{x\in X}(c+y^*d)'x$ has optimal extreme point solutions x_1 and x_2, then there exists an optimal solution x^* to problem (16) which can be written as a convex combination of x_1 and x_2. Moreover, note that each evaluation of $\theta(y)$ yields a vertex of X. Therefore, instead of selecting the best vertex among x_1 and x_2, we can select the best among all evaluated vertices, resulting in a practical improvement of Algorithm 1.

Remark 4. The Lagrangian relaxation provides an upper bound on the optimality gap, given by

$$\left|\frac{V_{\text{cont}} - V_{\text{int}}}{V_{\text{int}}}\right|, \tag{28}$$

where $V_{\text{cont}} = \min_{y\geq 0}\theta(y)$ is the value of the continuous relaxation and V_{int} is the objective value of the best extreme point found.

5.3. **Approximate robust formulation.** In this section we discuss how to solve the approximate robust conic quadratic optimization problem

$$\begin{aligned}(\text{ARCQO})\quad \zeta &:= \min_{z\in Z}\max_{x\in\mathbb{R}^n}\left\{c_0'z + \sum_{i\in N}(c_i'z)\,x_i + \sqrt{z'Q_0z + \sum_{i\in N}(z'Q_iz)\,x_i} : Ax\leq b, x\geq 0\right\}\\ &= \min_{z\in Z, y\geq 0}\left\{\frac{1}{4y} + c_0'z + yz'Q_0z + \max_{Ax\leq b, x\geq 0}\sum_{i=1}^{n}(c_i'z + yz'Q_iz)\,x_i\right\},\end{aligned}$$

where the equality follows from the analysis in Section 5.2. Let w be the dual variables associated with the inner maximization problem. Using standard LP duality, we get that

$$\begin{aligned}\zeta = \min\ & \frac{1}{4y} + c_0'z + yz'Q_0z + b'w & \\ (\text{ARCQO}')\qquad \text{s.t}\ & A_i'w \geq c_i'z + yz'Q_iz & i=1,\dots,n\\ & z\in Z, y\geq 0, w\geq 0,\end{aligned}$$

where A_i' denotes the transpose of the i-th column of A. We now substitute $y = 1/\hat{y}$, introduce additional nonnegative variables v_i, $i = 0,\dots,n$, and enforce the rotated second order cone constraints $z'Qz \leq \hat{y}v_i$ to get the equivalent formulation

$$\begin{aligned}\zeta = \min\ & \frac{1}{4}\hat{y} + c_0'z + v_0 + b'w & \\ (\text{ARCQO}'')\qquad \text{s.t}\ & A_i'w \geq c_i'z_i + v_i & i=1,\dots,n\\ & z'Q_iz \leq \hat{y}v_i & i=0,\dots,n\\ & z\in Z, \hat{y}\geq 0, w\geq 0, v\geq 0.\end{aligned}$$

A case of particular interest is when Z is a SOCP-representable convex set.

Proposition 10. *If Z is SOCP-representable, then the approximate robust conic quadratic optimization problem (ARCQO) is SOCP-representable.*

Note that there are efficient interior point algorithms for the special case of SOCPs (e.g. Nesterov and Todd 1998).

Remark 5. Unlike the linear case studied in Bertsimas and Sim (2004), the *exact* robust counterpart (10) is $\mathcal{NP}$-hard. Note that most robust counterparts of conic quadratic programs are $\mathcal{NP}$-hard, and *safe tractable approximations* are used instead (Ben-Tal et al. 2009, chapters 5-7). In our case, the approximation ratio is constant (1.25), and the approximation belongs to the same complexity class as the nominal problem (7).

5.4. **Computational complexity.** We now discuss the theoretical computational complexity of the approximation algorithm for specific polytopes, and compare it with the existing approaches in the literature.

5.4.1. *Computational complexity for the uniform matroid.* The greedy algorithm, with complexity $O(nk)$, is a well-known approach to tackle a monotone submodular maximization problem over the uniform matroid. We now argue that the Lagrangean version of the approximation algorithm described in 5.2 may be preferable in large instances.

First note that maximizing a linear function over the uniform matroid can be done in $O(n)$ time using quickselect. The complexity of the approximation algorithm is thus $O\left(n \ln\left(\frac{g'(d'x_L)}{\epsilon}\right)\right)$. Note that ϵ corresponds to the required precision of a lagrangean multiplier related with the nonlinear term, and *does not depend on* n *or* k. Moreover, for the exponential utility, logarithmic utility and MNL probability function we have that $g'(d'x) \leq 1$ whenever $d'x \geq 0$ (a similar dimension-independent upper bound can be obtained for the square root function if $\xi > 0$). Therefore, we see in small instances with high precision (i.e., $k < \ln \frac{1}{\epsilon}$) the greedy algorithm is preferable, but in instances with large values for k and n the proposed approximation algorithm is faster.

Badanidiyuru and Vondrák (2014) proposed another algorithm for maximizing a submodular function over the uniform matroid that does not depend on k. However, the complexity of their algorithm is $O\left(\frac{n}{\epsilon} \ln \frac{n}{\epsilon}\right)$, which is worse than the complexity of our approximation algorithm for all values of n.

5.4.2. *Computational complexity for matroids.* We consider the case when X is a matroid which is known through an independence oracle. There are randomized algorithms for maximizing a monotone submodular function, which rely on *the multilinear extension*, but such algorithms are computationally expensive. For example, the algorithm of Calinescu et al. (2011) runs in $\tilde{O}(n^8)$ time, and the authors comment that the high complexity is due to the number of samples necessary to achieve high probability bounds. A more efficient algorithm was proposed by Badanidiyuru and Vondrák (2014), with complexity $O\left(\frac{1}{\epsilon^4} nk \ln^2 \frac{n}{\epsilon} + T\left(\frac{1}{\epsilon^2} n \ln \frac{n}{\epsilon} + \frac{1}{\epsilon} k^2\right)\right)$, where k is the maximum cardinality of a feasible solution and T is the time complexity of the independence oracle. Observe that the time complexity of the Lagrangean version of our approximation algorithm is better: finding a maximum weight independent set (linear oracle) can be done in $O\left(n \ln n + Tn\right)$ using the greedy algorithm, and the overall complexity of the algorithm is thus $O\left((n \ln n + Tn) \ln \frac{1}{\epsilon}\right)$.

5.4.3. *Computational complexity for 0-1 down-monotone polytopes.* Vondrák et al. (2011) propose a general framework for maximizing a submodular function over down-monotone polytopes using *contention resolution schemes*, which are general randomized rounding techniques that convert a fractional solution obtained from solving a relaxation into a feasible integer solution while preserving the quality of the solution. However, finding such a resolution scheme (which depends on fractional solution found) is computationally challenging, even when the polytope is a matroid: it requires solving an LP which is only known through a separation oracle (using the ellipsoid method, which is known to perform poorly in practice), and in which each evaluation of the separation oracle is noisy (if ε is the maximum violation allowed for a given constraint, then we require $poly(1/\varepsilon)$ calls to the separation oracle to guarantee feasibility with high probability). In contrast, using the techniques described in Section 5.1, we *deterministically* recover a feasible solution in only $O(n)$ time.

5.4.4. *General case.* Most approximation algorithms in the literature rely on *rounding down* fractional solutions, and such algorithms do not guarantee feasibility if the polytope is not down-monotone (to the best of our knowledge there is no other approximation algorithm for general

polyhedra). In contrast our approach exploits the structure of the objective function, but it makes no assumption about the polyhedron besides the existence of an efficient optimization oracle.

6. Computational experiments

In this section we study the empirical performance of the approximation algorithm. First we test the approximation algorithm in instances for which there are other approximation methods in the literature (Section 6.1). Then in Section 6.2 we test the approximation algorithm in the assignment polytope, in Section 6.3 we test the approximation algorithm for the PERT application discussed in Section 2.3, and in Section 6.4 we test the approximation algorithm for the robust optimization application as discussed in Section 2.4. Note that for the assignment polytope, PERT, and robust optimization applications, methods based on submodularity are not applicable. All experiments are conducted on one thread of a Dell computer with a 2.2 GHz Intel®Core™ i7-2670QM CPU and 8 GB main memory.

6.1. Experiments for the uniform matroid polytope.

In this section we conduct computational experiments over the uniform matroid, comparing the proposed approximation algorithm and the greedy algorithm. We solve the problems with the Lagrangian approach described in Section 5.2, using golden section for the line search and *quickselect* as the linear oracle.

Each coefficient c_i is generated uniformly in $[0, 1]$ and scaled so that $\sum_{i=1}^{n} c_i = 1$. Each coefficient d_i is generated uniformly in $[0, 1/c_i]$, and scaled so that $\sum_{i=1}^{n} d_i = 1$. The scaling ensures that the weights of the linear and nonlinear parts of the objective are similar, and the coefficients are generated so that there is a tradeoff between the linear and nonlinear contributions. Both of these choices contribute to making the instances more challenging. We set $k = n/10$ and we use $g(x) = 1 - e^{-d'x}$, as in the reliability problem described in Section 2.1[3]. Table 1 shows, for different values of n, the time required by the approximation algorithm and the greedy algorithm to solve the instances in milliseconds and the gap between the solutions found, computed as

$$\frac{f^*_{\text{Greedy}} - f^*_{\text{Approx.}}}{f^*_{\text{Greedy}}},$$

where f^*_{Greedy} and $f^*_{\text{Approx.}}$ are the objective values of the solutions found by the greedy algorithm and the approximation algorithm, respectively. Each column represents the average over five instances generated with the same parameters. We also computed an upper bound of the gap with respect of the optimal solution using (28), and the gaps are under 0.1% for both approaches.

Table 1. Results in the uniform matroid polytope.

	$n = 100$	$n = 1,000$	$n = 10,000$	$n = 100,000$	$n = 500,000$
Time approximation (ms)	2	4	26	102	305
Time greedy (ms)	3	18	1,184	112,533	2,819,163
Gap	-1.5×10^{-16}	2.4×10^{-15}	-3.1×10^{-15}	8.1×10^{-9}	1.7×10^{-8}

We see that the solutions generated by both approaches are very similar in terms of the objective value. Yet, the proposed approximation algorithm is faster than the greedy algorithm in all cases, and is considerably so for the larger instances.

As we observed in Section 5.4, the time complexity of existing approximation algorithms for other down-monotone polytopes is much larger than the complexity of the greedy algorithm. The results for the uniform matroid polytope suggest the performance of the approximation algorithm may be orders of magnitude faster than existing approaches in other polytopes in practice.

[3]Experiments with other functions yield similar results.

6.2. **Experiments for the assignment polytope.** In this section we conduct computational experiments over the $n \times n$ assignment polytope. We solve the problems with the Lagrangian approach described in Section 5.2, using golden section for the line search and the the *Hungarian method* as the linear oracle. The coefficients are generated as described in Section 6.1, and we use $g(x) = \frac{d'x}{1+d'x}$, as in the assortment problem described in Section 2.2.

Table 2 shows, for different values of n, the time required by the approximation algorithm to solve the instances in milliseconds and the upper bound of the optimality gap given by (28). Each column represents the average over five instances generated with the same parameters.

TABLE 2. Results in the assignment polytope.

	$n = 3$	$n = 10$	$n = 100$	$n = 1000$
Time approximation (ms)	1	2	66	2,892
Gap	0.24%	0.01%	0.00%	0.00%

We observe that in problems with $n \geq 100$ the solutions to approximation algorithm are very close to optimal, and in the smaller instances the average gap is at most 0.24%. In both cases the approximation the reported gap is far from the worst case bound. Moreover the approximation algorithm can solve instances with $n = 1,000$ (i.e. 1,000,000 variables) in under three seconds.

6.3. **PERT.** The networks for the critical path experiments are constructed as follows. Let $V = \{0, 1, \cdots, r\}$ and p be the density parameter. For each pair (i, j), $i < j$, construct an arc with probability p. For each arc the expected duration c_{ij} is drawn from $U[0.5(j-i), 1.5(j-i)]$, and the standard deviation $\sqrt{d_{ij}}$ from $U[0.25c_{ij}, 0.75c_{ij}]$.

The goal is to find the critical 0-r path that best approximates the Value-at-Risk at confidence level α of the completion time of the project (as defined in (5)). We compare the two approaches described in Section 2.3:

Det-critical: We use the Value-at-Risk of the deterministic critical path.

VaR-critical: We use the Lagrangian version of the approximation algorithm to solve problem (6), and use the resulting path to approximate the Value-at-Risk.

For each instance we compute the simulated Value-at-Risk (VaR_{sim}) of the completion time of the project using Monte Carlo simulation with 2,000 replications. In each replication we generate samples for the duration of each arc according to the normal distributions. We then compute $\text{VaR}_\alpha(D) = \sup\{\ell \in \mathbb{R} : \mathbf{Pr}(D \leq \ell) \leq 1 - \alpha\}$ using binary search on ℓ, and we evaluate $\mathbf{Pr}(D \leq \ell)$ by the fraction of replications in which $D \leq \ell$ (where D is the length of the critical path).

Table 3 presents the results for instances with 50 nodes. It shows the density p; the solution approach used; and for varying confidence levels the estimated Value-at-Risk $\text{VaR}_{\text{est.}}$, and the gap between $\text{VaR}_{\text{est.}}$ and VaR_{sim}, computed as

$$\frac{\text{VaR}_{\text{sim.}} - \text{VaR}_{\text{est.}}}{\text{VaR}_{\text{sim.}}}.$$

Each row represents the average over five instances generated with the same parameters. In all cases, the solution of approximation algorithm is within 0.1% of optimal (computed using (28)) and computing the VaR_α-critical path takes less than 0.1 seconds.

The computational experiments suggest that explicitly using the risk information to select the critical path results in paths that better estimate the Value-at-Risk of the project compared to using the deterministic critical path by PERT: the gap between the estimated and true Value-at-Risk of the project is reduced by almost half. Note that the quality of the approximation for both approaches decreases as the density of the network increases, and the VaR_α-critical path is comparatively better for higher values of confidence.

These experiments on estimating project duration indicate that the approximation algorithm is effective in finding optimal solutions to VaR-critical path problem.

TABLE 3. Duration estimates in networks with 50 nodes.

Density	Method	90% CL Est.	90% CL Gap	97.5% CL Est.	97.5% CL Gap	99% CL Est.	99% CL Gap
0.3	Det	10,137	26.4%	12,031	16.5%	13,203	14.6%
	VaR	10,811	20.2%	13,008	9.8%	14,375	7.1%
	Sim	13,008	-	14,414	-	15,469	-
0.5	Det	10,002	33.4%	11,563	30.3%	12,344	30.1%
	VaR	11,699	22.2%	14,453	12.9%	15,625	11.5%
	Sim	15,029	-	16,593	-	17,656	-
0.8	Det	10,479	35.4%	12,383	30.2%	13,281	29.2%
	VaR	12,036	25.8%	15,078	15.0%	16,250	13.3%
	Sim	16,230	-	17,734	-	18,750	-
Average	**Det**		**31.7%**		**25.7%**		**24.6%**
	VaR		**22.7%**		**12.6%**		**10.6%**

6.4. **Robust portfolio optimization.** We illustrate the method for robust conic quadratic programs proposed in Section 4.2 using a canonical finance application. Given a set of assets $N = \{1, \ldots, n\}$, with expected return μ and covariance matrix Σ, the portfolio with minimum value-at-risk can be found by solving the optimization problem

$$\min_{z \in Z} \; -\mu' z + \beta \sqrt{z' \Sigma z},$$

where $Z = \left\{ z \in \mathbb{R}^n_+ : \sum_{i \in N} z_i = 1 \right\}$ is the set of long-only budget-constrained portfolios.

For simplicity we test the case where Σ is a diagonal matrix with entries $\Sigma_{ii} = \sigma_i^2$. In the robust version, we consider the events that may reduce the expected return of asset i to $\mu_i - c_i$ and increase the variance of it to $\sigma_i^2 + d_i$. The decision-maker wishes to select a robust portfolio, given that k of the events happen simultaneously.

For the particular case of diagonal Σ and each event corresponding to a single variable, the approximate robust formulation in Section 5.3 can be further simplified to

$$\begin{aligned}
\min \; & \frac{1}{4}\hat{y} - \sum_{i \in N} \mu_i z_i + \sum_{i \in N} \sigma_i^2 v_i + kw \\
\text{s.t } & w \geq c_i z_i + d_i v_i && i = 1, \ldots, n \\
& z_i^2 \leq \hat{y} v_i && i = 1, \ldots, n \\
& z \in Z, \hat{y} \geq 0, w \geq 0, v \geq 0.
\end{aligned}$$

In order to compute the optimality gap, we solve the robust problem (10) exactly by enumerating all extreme points of the uniform matroid polytope, i.e.,

$$\begin{aligned}
& \min \; t \\
& \text{s.t. } t \geq -\sum_{i \in N} (\mu_i - c_i x_i) z_i + \sqrt{\sum_{i \in N} (\sigma_i^2 + d_i x_i) z_i^2} \quad \forall x \in V_X \\
& \qquad z \in Z, t \in \mathbb{R}
\end{aligned} \tag{29}$$

which requires $\binom{n}{k}$ conic quadratic constraints.

We solve both formulations using CPLEX 12.6.2. In our experiments we use $k = 3$, $\beta = 2$, μ_i is drawn from $U[0, 1]$, σ_i is drawn from $U[0, 2\mu_i]$ (thus risky assets have on average high expected return) and c_i and $\sqrt{d_i}$ are drawn form $U[0, 2]$. Table 4 presents the results. It shows, for n between 10 and 70, the time required to solve the problems for both formulations in milliseconds, and the value of the approximation ratio (19). Each column represents the average over five instances generated with the same parameters. For instances with $n > 70$, formulation (29) is impractical due to its high memory requirement.

TABLE 4. Robust portfolio results.

	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$
Time approximation (ms)	8	10	17	35	36	39	59
Time exact (ms)	59	1,092	7,904	33,252	101,113	266,131	601,215
Δ	1.000	1.008	1.004	1.001	1.000	1.000	1.000

We see that the approximate robust formulation is very accurate in practice, finding optimal solutions (with $\Delta = 1$) in most of the cases. Its performance is far from worst case bound of 1.25. We also observe that the approximate robust problem scales very well in practice.

7. Conclusions

In this paper we consider the problem of maximizing a class of concave utility functions over the extreme points of a polytope, which is $\mathcal{NP}$-hard for many classes of polytopes. Such problems naturally arise in combinatorial problems in which the feasible region is composed of the extreme points of an integral polytope. We exploit the property that there exists a solution of the continuous relaxation of the problem on an edge of the polytope to develop an approximation algorithm. The algorithm requires either an oracle for the continuous relaxation of the nonlinear problem, or an oracle for linear programs over the polytope, and is polynomial time under mild assumptions on the function g and the polytope X. We prove that the proposed approach is a 1/2-approximation. When the concave function in the objective is the square root function, the proposed approach is a 4/5-approximation. We also propose a simple 1.25-approximation algorithm for a class of robust conic quadratic minimization problems. Computational experiments suggest that both approaches find solutions with very small optimality gap, are much more efficient than alternatives found in the literature in some cases, and are the first approximation algorithms proposed for other cases.

References

Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, 106:433–446.

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

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

Alizadeh, F. and Goldfarb, D. (2003). Second-order cone programming. *Mathematical Programming*, 95:3–51.

Atamtürk, A. and Bhardwaj, A. (2015). Supermodular covering knapsack polytope. *Discrete Optimization*, 18:74–86.

Atamtürk, A. and Narayanan, V. (2008). Polymatroids and risk minimization in discrete optimization. *Operations Research Letters*, 36:618–622.

Atamtürk, A. and Narayanan, V. (2009). The submodular 0-1 knapsack polytope. *Discrete Optimization*, 6:333–344.

Badanidiyuru, A. and Vondrák, J. (2014). Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1497–1514. SIAM.

Balcan, M. and Harvey, N. J. A. (2010). Learning submodular functions. *CoRR*, abs/1008.2159.

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

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52:35–53.

Bier, V. M., Nagaraj, A., and Abhichandani, V. (2005). Protection of simple series and parallel systems with components of different values. *Reliability Engineering & System Safety*, 87:315 – 323.

Buchbinder, N., Feldman, M., Naor, J., and Schwartz, R. (2012). A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 649–658.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40:1740–1766.

Chong, J.-K., Ho, T.-H., and Tang, C. S. (2001). A modeling framework for category assortment planning. *Manufacturing & Service Operations Management*, 3:191–210.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *COLT*.

Fisher, M., Nemhauser, G., and Wolsey, L. (1978). An analysis of approximations for maximizing submodular set functions II. In Balinski, M. and Hoffman, A., editors, *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 73–87. Springer Berlin Heidelberg.

Gen, M. and Yun, Y. (2006). Soft computing approach for reliability optimization: State-of-the-art survey. *Reliability Engineering & System Safety*, 91:1008 – 1026. Special Issue - Genetic Algorithms and ReliabilitySpecial Issue - Genetic Algorithms and Reliability.

Goemans, M. X., Harvey, N. J. A., Iwata, S., and Mirrokni, V. S. (2009). Approximating submodular functions everywhere. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 535–544.

Hausken, K. (2008). Strategic defense and attack for series and parallel reliability systems. *European Journal of Operational Research*, 186:856 – 881.

Kulik, A., Shachnai, H., and Tamir, T. (2009). Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 545–554, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Levitin, G. and Hausken, K. (2008). Protection vs. redundancy in homogeneous parallel systems. *Reliability Engineering & System Safety*, 93:1444–1451.

Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H. (1998). Applications of second-order cone programming. *Linear algebra and its applications*, 284:193–228.

Megiddo, N. (1991). On finding primal- and dual-optimal bases. *INFORMS Journal on Computing*, 3:63–65.

Nahmias, S. (2001). *Production and Operations Analysis*. McGraw Hill.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14:265–294.

Nemirovski, A. and Todd, M. (2008). Interior-point methods for optimization. *Acta Numerica*, 17:191–234.

Nesterov, Y. and Nemirovskii, A. (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics.

Nesterov, Y. E. and Todd, M. J. (1998). Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on optimization*, 8:324–364.

Rusmevichientong, P., Shen, Z.-J. M., and Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research*, 58:1666–1680.

Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35:395–411.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41 – 43.

Van Ryzin, G. and Mahajan, S. (1999). On the relationship between inventory costs and variety benefits in retail assortments. *Management Science*, 45:1496–1509.

Vondrák, J., Chekuri, C., and Zenklusen, R. (2011). Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792. ACM.