Abstract
Maximizing a convex function over convex constraints is an NP-hard problem in general. We prove that such a problem can be reformulated as an adjustable robust optimization (ARO) problem where each adjustable variable corresponds to a unique constraint of the original problem. We use ARO techniques to obtain approximate solutions to the convex maximization problem. In order to demonstrate the complete approximation scheme, we distinguish the case where we have just one nonlinear constraint and the case where we have multiple linear constraints. Concerning the first case, we give three examples where one can analytically eliminate the adjustable variable and approximately solve the resulting static robust optimization problem efficiently. More specifically, we show that norm constrained log-sum-exp (geometric) maximization problem can be approximated by (convex) exponential cone optimization techniques. Concerning the second case of multiple linear constraints, the equivalent ARO problem can be represented as an adjustable robust linear optimization (ARLO) problem. Then, using linear decision rules returns a safe approximation of the constraints. The resulting problem is a convex optimization problem, and solving this problem gives an upper bound on the global optimum value of the original problem. By using the optimal linear decision rule, we obtain a lower bound solution as well. We derive the approximation problems explicitly for quadratic maximization, geometric maximization, and sum-of-max-linear-terms maximization problems with multiple linear constraints. Numerical experiments show that, contrary to the state-of-the-art solvers, we can approximate large-scale problems swiftly with tight bounds for these problems. In several cases, we have equal upper and lower bounds, which concludes we have global optimality guarantees in these cases.
1 Introduction

In this paper, we propose a new approximation method for the convex maximization problem:

$$\max_{x \in \mathbb{R}^n} f(Ax + b) \quad \text{s.t.} \quad x \in U,$$

where $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is a closed convex function, and $U$ consists of convex constraints. It is well-known that this problem is in general NP-hard (see, e.g., (Zwart 1974)). The seminal paper of Tuy (1964) is regarded as the first approach to solve convex maximization problems, where $U$ is a polyhedron.

There are many real-life problems that can be reformulated as convex maximization problems. Rebennack et al. (2009) show that the fixed charge network flow problem can be formulated as a convex maximization problem. Moreover, Zwart (1974) shows that two important problem classes are equivalent to convex maximization problems, namely cost minimization problems with the cost function being subject to economies of scale, and linear optimization problems that involve ‘yes’ or ‘no’ decisions (binary variables). Many machine learning (ML) problems can be formulated as convex maximization problems, for instance Mangasarian (1996) shows that fundamental problems in ML, misclassification minimization and feature selection, are equivalent to convex maximization problems. The same author more recently proposes the use of absolute value inequalities for classifying unlabeled data, which results in a problem of minimizing a concave function on a polyhedral set (Mangasarian 2015). Other important examples from data science are variants of principal component analysis (PCA). Zass and Shashua (2007) show that the nonnegative PCA problem and the sparse PCA problem are convex quadratic maximization problems. Additionally, a popular approach to solve difference of convex functions (DC) programming is the convex-concave method, which iteratively solves convex minimization and convex maximization problems (assuming the constraints are convex) (Lipp and Boyd 2016). DC programming is being used to solve many problems in machine learning, data science, biology, security and transportation (Le Thi and Pham Dinh 2018). Also many problems arising in graph theory can be formulated as convex maximization problems, a well known example is the MAX-CUT problem (Goemans and Williamson 1994). A lot of variations of integer linear and integer quadratic optimization problems over polyhedra can be written as convex maximization problems (Benson 1995). Convex maximization naturally appears in Robust Optimization when finding the worst-case scenario of a constraint which is a convex function of the uncertain parameter. A similar problem appears when one applies the adversarial approach (Bienstock and Özbay 2008) to solve a robust convex optimization problem. In this approach, at the step of adding worst-case uncertainty realization to the discrete uncertainty set, one needs to maximize convex functions.

A local or global solution of the convex maximization problem is necessarily at an extreme point of the feasible region (Rockafellar 1970), hence there exist many methods to solve convex maximization problem by searching for extreme point solutions, but this approach is itself
very hard. It is shown that the convex maximization problem is NP-hard in very simple cases (e.g., quadratic maximization over a hypercube), and even verifying local optimality is NP-hard (Pardalos and Schnitger 1988). Hence, there are many papers to approximate the convex maximization problem (Benson 1995). The survey paper (Pardalos and Rosen 1986) collects such works until the 1980s. Most of the proposed methods use linear underestimator functions, which are derived by the so-called convex envelopes. These algorithms have a disadvantage, namely, the size of the sub problems grows in every new iteration, which makes them impractical algorithms in general. Moreover, the proposed methods in the literature are designed only for some specific cases (e.g., (Zwart 1974)). All of the well-accepted methods to solve the most studied convex maximization problem, quadratic maximization, are based on cutting plane methods, iterative numerical methods such as the element methods, and techniques of branches and borders based on the decomposition of the admissible set as summarized in (Audet et al. 2005, Andrianova et al. 2016). These papers also indicate that these methods do not provide solutions in reasonable time for practical problems.

Convex maximization is frequently being studied in the scope of DC programming in recent optimization research. As summarized by Lipp and Boyd (2016), the early approaches reformulated the DC programming problems as convex maximization problems (Tuy 1986, Tuy and Horst 1988, Horst et al. 1991). One can see how the methods in convex maximization are adopted for DC programming literature in the work of Horst and Thoai (1999). Lipp and Boyd (2016) provide a thorough literature review in convex maximization, and state that the literature to solve such problems mostly relies on branch and bound or cutting plane methods which are very slow in practice. To this respect, in order to be able to cope with large DC programming problems, Lipp and Boyd (2016) propose a heuristic algorithm to find a decent local solution of the convex maximization problem.

In this paper a new method to approximately solve the convex maximization problem is presented. The method starts with reformulating the convex maximization problem as an adjustable robust optimization (ARO) problem. The ARO problem has a number of adjustable variables equal to the number of the original constraints. The adjustable variables appear nonlinearly, hence the ARO problem is still a hard problem. Therefore, we apply approximation methods used in the ARO literature, in order to approximate the original convex maximization problem. This way we derive convex optimization problems which provide upper and lower bounds of the original convex maximization problem.

The rest of the paper is organized as follows. In Section 2, we present our main theorems and show how to reformulate the convex maximization problem as an ARO problem. We exploit the relationship between equivalent formulations to show how to obtain a solution in the original problem by using the solution of the ARO problem. We give three cases of convex maximization over a single norm constraint, show how to analytically eliminate the adjustable variable, and (approximately) solve the resulting static robust optimization problem. The approximate solution of the ARO problem gives an upper bound to the convex maximization problem, and by using this solution we show how to obtain a lower bound for the original problem. We derive the
ARO reformulation of a general convex maximization problem with arbitrary convex constraints, and show that one way to approximately solve this ARO problem is by relaxing adjustable variables to static variables. Convex maximization over multiple linear constraints is a special case, and we investigate this problem in Section 3. It is shown that the ARO reformulation of this case is an adjustable robust linear optimization (ARLO) problem. Adoption of linear decision rules for the adjustable variables enables one to derive efficient upper and lower bound approximation problems. Explicit approximations are obtained for convex quadratic, geometric, and sum-of-max-linear-terms maximization problems. The numerical experiments follow in Section 4, which illustrate that our approximation problems can be solved significantly faster than the state-of-the-art optimization solvers, and provide tight optimality gaps in most of the cases. In cases where the upper bound is equal to the lower bound, a guarantee of global maximizer is obtained. We conclude the paper in Section 5 by discussing our findings and giving future research directions.

2 ARO Reformulation of a Convex Maximization Problem

In this section it is shown how to reformulate the convex maximization problem as an equivalent ARO problem. We first consider problems with just one convex constraint, and then in the second subsection we consider problems with multiple constraints.

2.1 Convex Maximization over a Single Convex Constraint

Let \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) and \( g : \mathbb{R}^n \rightarrow \mathbb{R} \) be closed convex functions. We assume that \( \exists \bar{x} \in \mathbb{R}^n : g(\bar{x}) < \rho \) for scalar \( \rho \in \mathbb{R} \). Moreover, let \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^m \). We consider the following convex maximization problem with a single convex constraint:

\[
\sup_{x \in \mathbb{R}^n} f(Ax + b) \quad \text{s.t.} \quad g(x) \leq \rho.
\]  

(1)

The feasible set is denoted by \( U \), i.e., \( U = \{ x \in \mathbb{R}^n : g(x) \leq \rho \} \). The following theorem shows that problem (1) is equivalent to an ARO problem where \( w \in \text{dom} \ f^* \) is the uncertain parameter and \( \lambda > 0 \) is the adjustable variable.

**Theorem 1.** The optimal objective value of problem (1) is equal to the optimal objective value of the following adjustable robust optimization problem:

\[
\inf_{\tau \in \mathbb{R}} \quad \tau \quad \text{s.t.} \quad \forall w \in \text{dom} \ f^*; \quad \exists \lambda > 0 : \lambda \rho + \lambda g^* \left( \frac{A^T w}{\lambda} \right) + b^T w - f^*(w) \leq \tau.
\]  

(2)

**Proof.** Problem (1) can be written as:

\[
\inf_{\tau \in \mathbb{R}} \quad \tau \quad \text{s.t.} \quad f(Ax + b) \leq \tau, \quad \forall x \in U.
\]  

(3)
Since $f$ is a closed convex function, we have:

$$f(z) = f^{**}(z) = \sup_{w \in \text{dom } f^*} \{ z^T w - f^*(w) \},$$

where $f^*$ denotes the convex conjugate of $f$, and $f^{**}$ is the double conjugate. Hence, the constraint of problem (3) becomes:

$$\forall x \in U : f(Ax + b) \leq \tau \iff \forall x \in U : \sup_{w \in \text{dom } f^*} \{ (Ax + b)^T w - f^*(w) \} \leq \tau \tag{4a}$$

$$\iff \sup_{w \in \text{dom } f^*} \left\{ \sup_{x \in U} \{ (A^T w)^T x + b^T w - f^*(w) \} \right\} \leq \tau, \tag{4b}$$

where in step (4a) we exploit the fact that if a constraint holds for the worst-case then it holds for any case, and in step (4b) we change the order of the supremum operators. We replace the inner linear maximization problem with its Lagrangian dual problem and obtain:

$$\sup_x \{ (A^T w)^T x : g(x) \leq \rho \} = \inf_{\lambda \geq 0} \left\{ \sup_x \{ (A^T w)^T x - \lambda g(x) \} + \lambda \rho \right\}$$

$$= \inf_{\lambda > 0} \left\{ \lambda \sup_x \left\{ \frac{(A^T w)^T x}{\lambda} - g(x) \right\} + \lambda \rho \right\}, \tag{5a}$$

$$= \inf_{\lambda > 0} \left\{ \lambda g^* \left( \frac{A^T w}{\lambda} \right) + \lambda \rho \right\}. \tag{5b}$$

We substitute (5b) into (4b) to conclude:

$$\forall x \in U : f(Ax + b) \leq \tau$$

$$\iff \sup_{w \in \text{dom } f^*} \left\{ \inf_{\lambda > 0} \left\{ \lambda \rho + \lambda g^* \left( \frac{A^T w}{\lambda} \right) + b^T w - f^*(w) \right\} \right\} \leq \tau \tag{6a}$$

$$\iff \forall w \in \text{dom } f^*, \exists \lambda > 0 : \lambda \rho + \lambda g^* \left( \frac{A^T w}{\lambda} \right) + b^T w - f^*(w) \leq \tau. \tag{6b}$$

Minimizing $\tau$ over (6b) concludes the proof. \hfill \Box

The optimal solution of problem (1) can be retrieved from the optimal solution of ARO problem (2). Let $(\bar{\lambda}, \bar{w})$ denote the solution of the left-hand side problem in constraint (6a), where $\bar{\lambda}$ is a function of $\bar{w}$ due to the inner problem. By exploiting the equivalence of (5a) and (5b), we can retrieve the optimal $x$ value by solving the concave supremization problem in (5a):

$$\bar{x} = \arg \sup_x \left\{ \frac{(A^T \bar{w})^T}{\lambda} x - g(x) \right\}. \tag{7}$$

If $g$ is differentiable and the inverse of its gradient ($\nabla^{-1} g(\cdot)$) exists, then by the first order conditions with respect to $x$ we obtain:

$$\bar{x} = \nabla^{-1} g \left( \frac{A^T \bar{w}}{\lambda} \right). \tag{8}$$
In the ARO reformulation (2) the adjustable variable $\lambda$ appears nonlinearly, hence this is also a difficult problem. However, there is only one adjustable variable, and this appears in a single (semi-infinite) constraint. In the following, we consider three cases where one can derive explicit expressions for $\lambda$ (e.g., the analytic worst case for $\lambda$, as a function of $w$), which circumvents nonlinearity. In these examples we do not show how to derive the convex conjugate of various $g(x)$ functions, but these can be found in, e.g., (Boyd and Vandenberghe 2004).

**Case 1** (Squared 2-norm constraint). Let the feasible set of problem (1) be defined as (where $a \in \mathbb{R}^n$ is a parameter):

$$U_1 = \left\{ x \in \mathbb{R}^n : g(x) = \frac{1}{2} \|x-a\|_2^2 \leq \frac{1}{2} \rho^2 \right\}.$$

The conjugate of the squared norm is $g^*(z) = \frac{1}{2} \|z\|_2^2 + z^\top a$, hence, minimizing $\tau$ over the constraint (6) is minimizing $\tau$ subject to:

$$\sup_{w \in \text{dom} f^*} \left\{ \min_{\lambda > 0} \left\{ \frac{1}{2} \rho^2 \lambda + \lambda g^* \left( \frac{A^\top w}{\lambda} \right) \right\} + b^\top w - f^*(w) \right\} \leq \tau$$

$$\iff \sup_{w \in \text{dom} f^*} \left\{ \min_{\lambda > 0} \left\{ \frac{1}{2} \rho^2 \lambda + \frac{1}{2} \lambda \left\| \frac{A^\top w}{\lambda} \right\|_2^2 \right\} + a^\top A^\top w + b^\top w - f^*(w) \right\} \leq \tau$$

$$\iff \sup_{w \in \text{dom} f^*} \left\{ \rho \left\| A^\top w \right\|_2 + a^\top A^\top w + b^\top w - f^*(w) \right\} \leq \tau,$$

where the last step holds since the inner minimization problem is a convex problem with the optimal decision rule $\bar{\lambda} = \rho^{-1} \left\| A^\top w \right\|_2$. Therefore, for the feasible set $U_1$, problem (1) is equivalent to the static robust optimization problem:

$$\tau^* = \sup_{w \in \text{dom} f^*} \rho \left\| A^\top w \right\|_2 + a^\top A^\top w + b^\top w - f^*(w).$$

Notice that at the original problem (1) with $U = U_1$, the constraint is convex and the convexity of the objective function $f(Ax + b)$ makes the problem non-convex, whereas in problem (9) maximizing $-f^*(w)$ is fine and the 2-norm makes the problem non-convex.

Although we conclude that problem (9) is non-convex, there exist strong approximation methods. To give an example, suppose the domain of $f^*$ consists of linear inequalities, i.e.,

$$\text{dom} f^* = \{ w : \alpha_i^\top w \leq \beta_i, \ i = 1, \ldots, d \},$$

for $\alpha_i \in \mathbb{R}^m$, $\beta_i \in \mathbb{R}$. In this setting, the only difficult part is maximizing the sum of a convex function and a concave function over linear constraints. To be able to approximate problem (9) with a convex problem we use the reformulation-linearization technique (RLT), whose details can be found in (Sherali and Adams 2013). We also tighten the RLT approximation by using a ‘positive semi-definite cut’ (Sherali and Fraticelli 2002) and obtain (complete derivation is in
Appendix A):

\[
\sup_{\mathbf{V} \in \mathbb{S}^{m \times m}, \mathbf{w} \in \mathbb{R}^m} \rho \sqrt{\text{tr}(\mathbf{V}^\top \mathbf{V} A)} + a^\top \mathbf{A}^\top \mathbf{w} + b^\top \mathbf{w} - f^*(\mathbf{w})
\]

s.t.

\[
\alpha_i^\top \mathbf{w} - \beta_i \leq 0, \quad i = 1, \ldots, d
\]

\[
\alpha_i^\top \mathbf{V} \alpha_j - (\beta_i \alpha_j + \beta_j \alpha_i)^\top \mathbf{w} + \beta_i \beta_j \geq 0, \quad i \leq j = 1, \ldots, d
\]

\[
\left(\begin{array}{c}
\mathbf{V} \\
\mathbf{w}^\top \\
1
\end{array}\right) \succeq 0,
\]

\[(10)\]

where \(\text{tr}(\cdot)\) denotes the trace operator. For instance, suppose \(f\) is the log-sum-exp function. Since \(f^*(\mathbf{w})\) is the negative entropy of \(\mathbf{w}\) in its domain (standard \(m\)-dimensional simplex), problem (10) becomes an exponential-cone representable problem. It can further be proved that for this case the semi-definiteness constraint is redundant, and the optimal value of \(V\) is the diagonal matrix \(\text{Diag}(\mathbf{w})\). So the only variable is \(\mathbf{w} \in \mathbb{R}^m\), and we can solve problem (10) with the exponential cone solver of MOSEK (MOSEK ApS 2019b). The numerical experiments for this specific instance are presented in Section 4.

Going back to the general setting of \(f\), it is straightforward to see that problem (10) upper bounds the optimal value of problem (9) since an optimal solution \(\tilde{\mathbf{w}}\) in problem (9) is feasible in problem (10) by taking \(V = \tilde{\mathbf{w}} \tilde{\mathbf{w}}^\top\). We can also obtain a lower bound on problem (9) by using the upper bound solution. From equation (8), the optimal \(\bar{x}\) value can be recovered by \(\bar{x} = \nabla^{-1} g(\frac{\mathbf{A}^\top \tilde{\mathbf{w}}}{\tilde{\lambda}})\) and in this example of 2-norm constrained convex minimization we have \(\tilde{\lambda} = \rho^{-1} \|A^\top \tilde{w}\|_2\). Moreover we have \(g(x) = \frac{1}{2}\|x - a\|_2^2\) so \(\nabla^{-1} g(y) = y + a\). Therefore, the global maximizer of the original problem is \(\bar{x} = (A^\top \tilde{w}) \frac{\rho}{\|A^\top \tilde{w}\|_2} + a\) for \(\tilde{w}\) being the optimal solution of problem (9). Since we approximate \(\tilde{w}\) of problem (9) with \(w^*\) in problem (10), a lower bounding approximation of the optimal \(\bar{x}\) value is:

\[
x^* = (A^\top w^*) \frac{\rho}{\|A^\top w^*\|_2} + a.
\]

Moreover, the constraint of the original convex maximization problem, i.e., \(\frac{1}{2}\|x - a\|_2^2 \leq \frac{1}{2} \rho^2\), is indeed tight for \(x^*\). This shows us that this method gives us an extreme point lower bound solution (recall that the global maximizer is also an extreme point).

**Case 2** (Box constraints). Next, consider the feasible region defined by box constraints:

\[
U_2 = \{x \in \mathbb{R}^n : g(x) = \|x - a\|_\infty \leq \rho\}.
\]

Although box constraints are actually a collection of multiple constraints, by using \(\infty\)-norm we can apply our theorem for a single constraint. Using the conjugate

\[
g^*(z) = \begin{cases} 
  z^\top a & \text{if } \|z\|_1 \leq 1 \\
  \infty & \text{otherwise},
\end{cases}
\]

where \(\text{tr}(\cdot)\) denotes the trace operator. For instance, suppose \(f\) is the log-sum-exp function. Since \(f^*(\mathbf{w})\) is the negative entropy of \(\mathbf{w}\) in its domain (standard \(m\)-dimensional simplex), problem (10) becomes an exponential-cone representable problem. It can further be proved that for this case the semi-definiteness constraint is redundant, and the optimal value of \(V\) is the diagonal matrix \(\text{Diag}(\mathbf{w})\). So the only variable is \(\mathbf{w} \in \mathbb{R}^m\), and we can solve problem (10) with the exponential cone solver of MOSEK (MOSEK ApS 2019b). The numerical experiments for this specific instance are presented in Section 4.

Going back to the general setting of \(f\), it is straightforward to see that problem (10) upper bounds the optimal value of problem (9) since an optimal solution \(\tilde{\mathbf{w}}\) in problem (9) is feasible in problem (10) by taking \(V = \tilde{\mathbf{w}} \tilde{\mathbf{w}}^\top\). We can also obtain a lower bound on problem (9) by using the upper bound solution. From equation (8), the optimal \(\bar{x}\) value can be recovered by \(\bar{x} = \nabla^{-1} g(\frac{\mathbf{A}^\top \tilde{\mathbf{w}}}{\tilde{\lambda}})\) and in this example of 2-norm constrained convex minimization we have \(\tilde{\lambda} = \rho^{-1} \|A^\top \tilde{w}\|_2\). Moreover we have \(g(x) = \frac{1}{2}\|x - a\|_2^2\) so \(\nabla^{-1} g(y) = y + a\). Therefore, the global maximizer of the original problem is \(\bar{x} = (A^\top \tilde{w}) \frac{\rho}{\|A^\top \tilde{w}\|_2} + a\) for \(\tilde{w}\) being the optimal solution of problem (9). Since we approximate \(\tilde{w}\) of problem (9) with \(w^*\) in problem (10), a lower bounding approximation of the optimal \(\bar{x}\) value is:

\[
x^* = (A^\top w^*) \frac{\rho}{\|A^\top w^*\|_2} + a.
\]

Moreover, the constraint of the original convex maximization problem, i.e., \(\frac{1}{2}\|x - a\|_2^2 \leq \frac{1}{2} \rho^2\), is indeed tight for \(x^*\). This shows us that this method gives us an extreme point lower bound solution (recall that the global maximizer is also an extreme point).
in (6), we obtain that problem (1) is equivalent to minimizing $\tau$ over:

$$
\sup_{w \in \text{dom} f^*} \left\{ \inf_{\lambda > 0} \left\{ \lambda \rho + a^T A^T w : \| A^T w \|_1 \leq \lambda \right\} + b^T w - f^*(w) \right\} \leq \tau
$$

$$
\iff \sup_{w \in \text{dom} f^*} \left\{ \rho \| A^T w \|_1 + a^T A^T w + b^T w - f^*(w) \right\} \leq \tau,
$$

since the minimizer of the inner problem is $\bar{\lambda} = \| A^T w \|_1$. This also shows that the piece-wise linear decision rule (LDR) is optimal for this problem, as $\bar{\lambda}$ has the structure of a piece-wise LDR. We can conclude that if the uncertainty set is $U_2$, the optimal value of problem (1) is given by:

$$
\tau^* = \sup_{w \in \text{dom} f^*} \rho \| A^T w \|_1 + a^T A^T w + b^T w - f^*(w).
$$  \hfill (12)

Notice that (12) is still a hard problem due to the convexity of $\rho \| A^T w \|_1$. However, this 1-norm can be represented by linear terms using extra binary variables for absolute values (see, e.g., (Löfberg 2016)). There are many efficient solvers which can solve the resulting mixed-integer convex optimization problem. For example, if $f$ is a log-sum-exp function, then $f^*$ is the negative entropy (with linear domain constraints), hence problem (12) will be a mixed-integer exponential cone representable problem and MOSEK can efficiently solve these type of problems. Numerical experiments can be found in Section 4. \hfill $\blacksquare$

**Case 3** ($p$-norm constraint). Consider the $p$-norm constrained set for $p \geq 1$:

$$
U_3 = \{ x \in \mathbb{R}^n : g(x) = \| x - a \|_p \leq \rho \}.
$$

The dual norm is $q$-norm, where $q$ is given by $\frac{1}{q} + \frac{1}{p} = 1$. Therefore, similarly to Case 2, we can conclude the optimal objective value of problem (1) is:

$$
\tau^* = \sup_{w \in \text{dom} f^*} \rho \| A^T w \|_q + a^T A^T w + b^T w - f^*(w),
$$

where again the non-convexity of the problem is not due to $f^*(w)$, but the dual norm of the original $g(x)$. This generalizes the previous two cases, i.e., $p = 2, \infty$. \hfill $\blacksquare$

### 2.2 Convex Maximization over Multiple Convex Constraints

Next, we extend Theorem 1 to a convex maximization problem with multiple constraints. Let $f : \mathbb{R}^m \to \mathbb{R}$ be a closed convex objective function, and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, q$ be closed convex functions. We assume that $\exists \bar{x}_j \in \mathbb{R}^n : g_j(x_j) < \rho_j$ for $j = 1, \ldots, q$. Similarly to the previous setting, let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We consider the following convex maximization problem with multiple convex constraints:

$$
\sup_{x \in \mathbb{R}^n} f(Ax + b)
$$

$$
s.t. \quad g_j(x) \leq \rho_j, \quad j = 1, \ldots, q.
$$  \hfill (13)
Theorem 2. The optimal objective value of problem (13) is equal to the optimal objective value of the following adjustable robust optimization problem:

\[
\inf_{\tau} \quad \tau
\]

s. t. \(\forall w \in \text{dom } f^*, \exists \lambda \in \mathbb{R}_+^q, z \in \mathbb{R}^{n \times q} : \left\{ \begin{array}{l} \sum_{j=1}^q \lambda_j \rho_j + \sum_{i=1}^q \lambda_i g_i^* \left( \frac{z_i}{\lambda_i} \right) + b^\top w - f^*(w) \leq \tau \\ \sum_{j=1}^q \lambda_j = A^\top w, \end{array} \right. \]

(14)

where \(z_j\) denotes the \(j\)-th column of \(z\), and \(\mathbb{R}_+\) is the set of strictly positive real values.

**Proof.** From the proof of Theorem 1 we see that problem (13) can be written as minimizing \(\tau\) over:

\[
\sup_{w \in \text{dom } f^*} \left\{ \sup_{x \in U} \left\{ (A^\top w)^\top x \right\} + b^\top w - f^*(w) \right\} \leq \tau,
\]

(15)

where in problem (13) we have \(U = \{ x \in \mathbb{R}^n : g_i(x) \leq \rho_i, \ i = 1, \ldots, q \}\). Consider the inner maximization problem in constraint (15). Taking the Lagrangian dual problem gives us:

\[
\sup_{x \in U} \left\{ (A^\top w)^\top x \right\} = \inf_{\lambda \in \mathbb{R}_+^q} \left\{ \sup_{x \in \mathbb{R}^n} \left\{ (A^\top w)^\top x - \sum_{i=1}^q \lambda_i g_i(x) \right\} + \sum_{j=1}^q \lambda_j \rho_j \right\} = \inf_{\lambda \in \mathbb{R}_+^q} \left\{ \left( \sum_{i=1}^q \lambda_i g_i \right)^*(A^\top w) + \sum_{j=1}^q \lambda_j \rho_j \right\}. \tag{16}
\]

To simplify \(\left( \sum_{i=1}^q \lambda_i g_i \right)^*(A^\top w)\), we use the fact that the conjugate of sum of convex functions can be written as the infimal convolution of these functions (Rockafellar 1970). This equivalence gives:

\[
\left( \sum_{i=1}^q \lambda_i g_i \right)^*(A^\top w) = \inf_{z_1, \ldots, z_q} \left\{ \sum_{i=1}^q \frac{z_i}{\lambda_i} : \sum_{j=1}^q z_j = A^\top w \right\}
\]

\[
= \inf_{z_1, \ldots, z_q} \left\{ \sum_{i=1}^q \lambda_i \left( g_i^* \left( \frac{z_i}{\lambda_i} \right) \right) : \sum_{j=1}^q z_j = A^\top w \right\},
\]

where the last step holds since \(\lambda_i g_i^* \left( \frac{z_i}{\lambda_i} \right) = \lambda_i \sup_x \left\{ \frac{x^\top z_i}{\lambda_i} - g_i(x) \right\} = (\lambda_i g_i)^*(z_i)\). Therefore, (16) becomes:

\[
\inf_{\lambda \in \mathbb{R}_+^q} \left\{ \sum_{j=1}^q \lambda_j \rho_j + \inf_{z_1, \ldots, z_q} \left\{ \sum_{i=1}^q \lambda_i g_i^* \left( \frac{z_i}{\lambda_i} \right) : \sum_{j=1}^q z_j = A^\top w \right\} \right\},
\]

and so constraint (15) reduces to:

\[
\sup_{w \in \text{dom } f^*} \left\{ \inf_{z_1, \ldots, z_q} \left\{ \sum_{j=1}^q \lambda_j \rho_j + \sum_{i=1}^q \lambda_i g_i^* \left( \frac{z_i}{\lambda_i} \right) + b^\top w - f^*(w) : \sum_{j=1}^q z_j = A^\top w \right\} \right\} \leq \tau. \tag{17}
\]

Minimizing \(\tau\) over (17) gives us the global optimum value of problem (13), which concludes the proof. \(\square\)
One way to approximate the ARO problem (14) is to use static variables for the adjustable variables. Assume without loss of generality that at problem (13) we have that \( x \in \mathbb{R}^n_+ \). It is possible to verify that this problem can be approximated by problem (14), with the last constraint being \( \sum_{j=1}^q z_j \geq A^\top w \) (instead of equality). By relaxing the adjustable variables to static variables, we obtain a tractable approximation of the problem. Hence, the following convex optimization problem returns an upper bound to the original problem:

\[
\inf_{\tau \in \mathbb{R}, \lambda \in \mathbb{R}^q_+, z \in \mathbb{R}^n} \tau \\
\text{s.t.} \quad \sum_{j=1}^q \lambda_j \rho_j + \sum_{i=1}^q \lambda_i g_i^* \left( \frac{z_i}{\lambda_i} \right) + \sup_{w \in \text{dom } f^*} \left\{ b^\top w - f^*(w) \right\} \leq \tau \\
\sup_{w \in \text{dom } f^*} \left\{ A_i^\top w \right\} \leq \sum_{j=1}^q z_{ij}, \quad i = 1, \ldots, n.
\]

Moreover, we know that solving \( \arg \sup_{x \in \mathbb{R}^n_+} \left\{(A^\top \bar{w})^\top x - \sum_{i=1}^q \bar{\lambda}_i g_i(x)\right\}\) gives us the global optimum solution, where \( \bar{\lambda}, \bar{w} \) are the worst-case solution of (17). Since we used a safe-approximation for this problem, we can use the \( \lambda \) value which solves problem (18) as an approximation of \( \bar{\lambda} \), and try \( n + 1 \) many scenarios for \( w \) solving each supremum problem in (18), in order to obtain a lower bound \( \bar{x} \). Such a lower-bound approach is thoroughly described in Section 3 where we approximate the problem of maximizing a convex function over linear constraints.

When the constraints are linear, problem (14) is linear in the adjustable variables, and hence this deserves a separate treatment, which is done in the next section.

### 3 Convex Maximization Over a Polyhedron

In this section, we consider problem (13) in which the feasible set is a polyhedron with nonempty interior. This special case allows us to obtain an attractive adjustable robust linear optimization formulation, for which efficient approaches are known in the literature. We first illustrate how the problem can be approximated with lower and upper bounds. Then, we consider special cases of the objective function \( f \): quadratic, log-sum-exp (geometric), and sum of max terms.

Formally, we work on the following special form of problem (13):

\[
\max_x f(Ax + b) \\
\text{s.t.} \quad x \in U,
\]

where for \( D \in \mathbb{R}^{q \times n} \) and \( d \in \mathbb{R}^n \), the polyhedral constraint set is:

\[
U = \{ x \in \mathbb{R}^n_+ : Dx \leq d \}.
\]

Let \( D_{(j)} \) denote the \( j \)-th row of \( D \). Then, the \( j \)-th constraint of \( U \) is given by

\[
g_j(x) \leq \rho_j \iff D_{(j)} x \leq d_j.
\]
By Theorem 2, problem (19) becomes:

\[
\inf_{\tau \in \mathbb{R}} \tau \quad \text{s.t.} \forall w \in \text{dom } f^*, \exists \lambda \in \mathbb{R}^q_+, z \in \mathbb{R}^{n \times q}:
\begin{align*}
  & d^T \lambda + b^T w - f^*(w) \leq \tau \\
  & z_i \leq D(i) \lambda_i, \quad i = 1, \ldots, q \\
  & \sum_{j=1}^q z_j = A^T w.
\end{align*}
\]

Constraints \( \sum_{j=1}^q z_j = A^T w \) and \( z_i \leq D(i) \lambda_i, \quad i = 1, \ldots, q \) together can be written as \( A^T w \leq D^T \lambda \). Therefore, problem (19) has the same optimal objective value as the following ARO problem:

\[
\inf_{\tau \in \mathbb{R}} \tau \quad \text{s.t.} \forall w \in \text{dom } f^*, \exists \lambda \in \mathbb{R}^q: \begin{cases}
  d^T \lambda + b^T w - f^*(w) \leq \tau \\
  D^T \lambda \geq A^T w \\
  \lambda \geq 0.
\end{cases}
\]

Notice that the final problem is a linear ARO problem with fixed recourse (linearity is obtained by lifting the \( -f^*(w) \) term to the uncertainty set). There are many possible methods one can use to solve such a problem, for example, one can solve this problem to optimality by eliminating the adjustable variables via Fourier-Motzkin Elimination for ARO (Zhen et al. 2018), which is efficiently applicable for small-sized problems. We refer to Yanıkoglu et al. (2019) for a survey of alternative methods to solve this linear ARO problem. In the remaining of this section we show how to derive tractable problems to find upper and lower bounds on the optimal solution of problem (19).

By using linear decision rules, one obtains a (tractable) safe approximation of the constraints of the ARO problem (21). The following theorem exploits this to obtain an upper bound on the optimal objective value of problem (19), and the proof is in Appendix B.

**Theorem 3.** The optimal objective value of the following problem is an upper bound to the optimal objective value of problem (19):

\[
\min_{u \in \mathbb{R}^q, V \in \mathbb{R}^{n \times m}, \tau \in \mathbb{R}} \frac{d^T u + (1 + d^T r) f \left( \frac{V^T d + b}{1 + d^T r} \right)}{1 + d^T r} \leq \tau \quad \text{s.t.} \begin{cases}
  -D_i^T u + (-D_i^T r) f \left( \frac{A_i - V^T D_i}{-D_i^T r} \right) \leq 0 & i = 1, \ldots, n \\
  -D_i^T r \geq 0 & i = 1, \ldots, n \\
  -u_i + (-r_i) f \left( \frac{V_{(i)}}{r_i} \right) \leq 0 & i = 1, \ldots, q \\
  -r_i \geq 0.
\end{cases}
\]

Here, \( V_{(i)} \) stands for the \( i \)-th row of \( V \) where \( A_i, D_i \) are the \( i \)-th columns of the corresponding matrices.

\(^{1}z_j \) denotes the \( j \)-th column of \( z \), while \( D_{(i)} \) denotes the \( i \)-th row of \( D \).
Remark 1. If \( f \) is positively homogeneous, the perspective function is \( z_0 f \left( \frac{z}{z_0} \right) = f(z) \) which in turn makes problem (22) easier (without variable \( r \)). Moreover, if \( z_0 = 0 \) our understanding for \( z_0 f \left( \frac{z}{z_0} \right) \) is the recession function \( \lim_{z_0 \to 0} z_0 f \left( \frac{z}{z_0} \right) \) (Rockafellar 1970).

Remark 2. When problem (22) is hard to solve, the adversarial approach could be a valuable alternative. In Appendix B we show that using LDRs to approximate the ARO problem gives us the following problem:

\[
\begin{align*}
\min_{\tau,u,V,r} & \quad \tau \\
\text{s.t.} & \quad d^T(u + V w + r w_0) + b^T w + w_0 - \tau \leq 0, \quad \forall \begin{pmatrix} w_0 \\ w \end{pmatrix}^T \in W \quad (23a) \\
& \quad -D^T(u + V w + r w_0) + A^T w \leq 0, \quad \forall \begin{pmatrix} w_0 \\ w \end{pmatrix}^T \in W \quad (23b) \\
& \quad -(u + V w + r w_0) \leq 0 \quad \forall \begin{pmatrix} w_0 \\ w \end{pmatrix}^T \in W, \quad (23c)
\end{align*}
\]

where \( W = \left\{ \begin{pmatrix} w_0 \\ w \end{pmatrix}^T \in \mathbb{R}^{m+1} : w_0 + f^*(w) \leq 0 \right\} \). The adversarial approach avoids directly solving the safe approximation of the ARO, i.e., the upper bound problem. Instead, this approach introduces a finite set \( \overline{W} \) and replaces it with \( W \) in problem (23), and solves the resulting LP problem. Then, the \( \begin{pmatrix} w_0 \\ w \end{pmatrix}^T \in W \) values that violate each of the constraints the most are added to set \( \overline{W} \). Iterating this procedure until there is no violation guarantees that at termination the optimal solution will be obtained. Obviously, this approach does not use the perspective functions, but if the number of LPs solved is high, then this also becomes a hard problem. For a background of the adversarial approach, see (Bienstock and Özbay 2008).

The solution of the upper bound problem can be used to obtain a (potentially good) lower bound for problem (19) by using what was proposed for two-stage fixed-recourse robust constraints by Hadjiyiannis et al. (2011) and extended by Zhen et al. (2017). In Theorem 1 we showed the global optimum value of the convex maximization problem is given by:

\[
\sup_{w \in \text{dom} f^*} \left\{ \sup_{x \in U} \left\{ (A^T w)^T x \right\} + b^T w - f^*(w) \right\},
\]

where the solution of the inner problem gives us the optimal \( x \) value. In the lower bound approach we generate a finite set \( \overline{W} \) whose elements are selected such as to represent the optimal \( w \) to the above problem. Then, we solve

\[
\bar{x}^{(i)} = \arg \max_{x \in U} \{(Ax + b)^T \bar{w}_i\}, \quad \forall \bar{w}_i \in \overline{W}, \quad (24)
\]

and return the \( \bar{x}^{(i)} \) value giving the highest objective value \( f(A\bar{x}^{(i)} + b) \). The set \( \overline{W} \) can be generated in many ways, but we use our upper bound solution for this purpose, namely \((\hat{u}, \hat{r}, \hat{V}, \hat{\tau})\) standing for optimal \((u, r, V, \tau)\) values. So we use the LDR \( \lambda = \hat{u} + \hat{V} w + \hat{r} w_0 \) for finding the worst-case scenarios in the constraints of the ARO problem (23) with the optimal LDR substitution. With this approach we hope to find the worst-case scenarios similar to the worst-case scenarios of the original ARO problem (not the approximation). Finding the worst case uncertainty
realization for each of the ARO constraints is given by solving:

\[
\mathcal{W} \leftarrow \arg \max_{w \in \text{dom } f^*} \left\{ -(1 + d^T r) f^*(w) + (d^T \hat{V} + b^T) w + d^T \hat{u} - \hat{r} \right\},
\]

\[
\mathcal{W} \leftarrow \arg \max_{w \in \text{dom } f^*} \left\{ (D_i^T r) f^*(w) + (A_i^T - D_i^T \hat{V}) w - D_i^T \hat{u} \right\}, \quad i = 1, \ldots, n
\]

\[
\mathcal{W} \leftarrow \arg \max_{w \in \text{dom } f^*} \left\{ -\hat{u}_i - \hat{V}_i w + \hat{r}_i f^*(w) \right\}, \quad i = 1, \ldots, q.
\]

In the above notation \(\mathcal{W}\) is a finite set of \(n + q + 1\) many \(w\) scenarios, and \(\leftarrow\) is the sign of appending the set of critical values.

We next derive the approximations for specific problems, namely convex quadratic maximization, convex log-sum-exp (geometric) maximization, and convex sum-of-max-linear-terms maximization. The results are shared in Tables 1, 2, and 3. Complete derivations can be found in Appendix C. In summary, we see that the upper bound approximation of the convex quadratic maximization problem is found by solving a second-order cone optimization problem, and from the solution of this problem the lower-bound scenarios can be collected analytically. For geometric maximization, the upper bound problem is an exponential cone optimization problem, and the lower bound scenarios can also be collected by solving multiple exponential cone optimization problems. Finally, for sum-of-max-linear-terms maximization, the upper bound problem is a linear optimization problem, and the lower bound scenarios can be found analytically. All of the original problems are known to be very hard problems, while the approximation problems are mainstream convex optimization problems, and there exist many powerful solvers to solve such problems. In the numerical experiments, we use Mosek as a conic optimization solver, and CPLEX (IBM ILOG CPLEX 2014) as a linear optimization solver.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters and Assumptions</th>
<th>Convex Maximization Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic Maximization</td>
<td>(g(x) = x^T Q x + \ell^T x), convex quadratic function</td>
<td>(\max_{x \in \mathbb{R}^n} g(x) = x^T Q x + \ell^T x) s.t. (Dx \leq d)</td>
</tr>
<tr>
<td>Geometric Maximization</td>
<td>(f(z) = \log(\sum_{i=1}^m \exp(z_i))), log-sum-exp function, (A \in \mathbb{R}^{m \times n}), (b \in \mathbb{R}^m), linear coefficients matrix</td>
<td>(\max_{x \in \mathbb{R}^n} f(Ax + b) = \log(\sum_{i=1}^m \exp(A_i x + b))) s.t. (Dx \leq d)</td>
</tr>
<tr>
<td>Sum-of-Max-Linear-Terms Optimization</td>
<td>(f(z) = \sum_{i \in \ell} \max_{j \in \ell_i} {z_j}), sum-of-max-terms function, (K &gt; 0), (\ell_i \subset {1, \ldots, m}), (\ell_{k \cap \ell} = \emptyset), for (k \neq \ell), w.l.o.g., (\ell = {1, \ldots, m}), (A \in \mathbb{R}^{m \times n}), linear coefficients matrix, (b \in \mathbb{R}^m), linear constants vector</td>
<td>(\max_{x \in \mathbb{R}^n} f(Ax + b) = \sum_{i=1}^m \max_{j \in \ell_i} {A_{ij} x + b_j}) s.t. (Dx \leq d) Note: By using logical programming one can reformulate as mixed integer optimization</td>
</tr>
</tbody>
</table>

Table 1: Quadratic, Geometric, and Sum-of-Max-Linear-Terms Maximization problems. In each case we introduce the problem setting, and share the main convex maximization problem that we are interested in solving. All problems have the same linear constraints \(Dx \leq d\) for \(D \in \mathbb{R}^{q \times n}\), \(d \in \mathbb{R}^q\).
### Problem Type and Variables

#### Upper Bound Problem

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Variables</th>
</tr>
</thead>
</table>
| Quadratic Maximization | $\min \tau$
| Geometric Maximization | $\min \tau$
| Sum-of-Max-Linear-Terms Optimization | $\min \tau$

#### Problem Type and Variables

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Variables</th>
</tr>
</thead>
</table>
| Quadratic Maximization | $\min \tau$
| Geometric Maximization | $\min \tau$
| Sum-of-Max-Linear-Terms Optimization | $\min \tau$

**Table 2:** For each of the three problems, we show the upper bound approximation problems. These problems are special forms of upper bound problem (22) of Theorem 3.

### Analytic Scenarios

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Lower Bound Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic Maximization</td>
<td>$|L_{i} - V^{\top} D_{i}|_{2}$, $i = 1, \ldots, n$</td>
</tr>
</tbody>
</table>

Each scenario collection is an exponential cone problem.

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Lower Bound Scenarios</th>
</tr>
</thead>
</table>
| Sum-of-Max-Linear-Terms Optimization | $\arg\max_{u \in \mathbb{R}^n} \{d^{\top}(u + V^{\top} w) + b^{\top} w - \tau\}$
| Sum-of-Max-Linear-Terms Optimization | $\arg\max_{u \in \mathbb{R}^n} \{d^{\top}(u + V^{\top} w) + b^{\top} w - \tau\}$

Each optimization problem can be split to K optimization problems, that are analytically solvable.

**Table 3:** For each of the three problems, we share the lower bound scenario collection steps. These scenarios will help us to find a lower bound solution by solving linear optimization problems corresponding to each scenario.

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Lower Bound Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic Maximization</td>
<td>$|L_{i} - V^{\top} D_{i}|_{2}$, $i = 1, \ldots, n$</td>
</tr>
</tbody>
</table>

Each scenario collection is an exponential cone problem.

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Lower Bound Scenarios</th>
</tr>
</thead>
</table>
| Sum-of-Max-Linear-Terms Optimization | $\arg\max_{u \in \mathbb{R}^n} \{d^{\top}(u + V^{\top} w) + b^{\top} w - \tau\}$
| Sum-of-Max-Linear-Terms Optimization | $\arg\max_{u \in \mathbb{R}^n} \{d^{\top}(u + V^{\top} w) + b^{\top} w - \tau\}$

Each optimization problem can be split to K optimization problems, that are analytically solvable.
4 Numerical Experiments

In this section we present the numerical experiments to support the theory developed in this paper. We use YALMIP through MATLAB 2018b to call various solvers, and report the solver times below (we do not reflect the problem formulation times in YALMIP). Numerical experiments are held in a standard personal computer with an 8-th Generation Intel(R) Core(TM) i7-8750H processor. The details of data generation of these experiments are in Appendix F.

4.1 Log-Sum-Exp Maximization over a 2-norm Constraint

Consider problem (1) with the constraint $||x - a||_2^2 \leq \rho^2$. In Section 2.1 under Case 1 we discussed that for $f(z) = \log(\sum_{i=1}^{m} \exp(z_i))$, the upper bound approximation problem is exponential-cone representable, and we can analytically obtain the lower-bound solution immediately. The approximation problem (10) is solved by MOSEK.

To benchmark our approximation, we used general-purpose nonlinear optimization solvers Artelys Knitro (Byrd et al. 2006), IPOPT (Wächter and Biegler 2006), and BMIBNB of YALMIP Löfberg (2004). Knitro appeared to yield the best result, so we report Knitro as a benchmark. Although Knitro does not guarantee global optimality it can find the global optimum value faster and more often compared to the other solvers we tried. We also compare our approximation with the global optimization solver BARON. The results can be found in Table 4.

<table>
<thead>
<tr>
<th>Problem and Dimensions</th>
<th>Knitro</th>
<th>Knitro Multi-Start</th>
<th>BARON</th>
<th>Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td># 1 (# = 5, m = 5)</td>
<td>9.7890 0.05</td>
<td>-</td>
<td>-</td>
<td>9.7890 0.40</td>
</tr>
<tr>
<td># 2 (# = 20, m = 15)</td>
<td>54.0329 0.18</td>
<td>55.8913 2 0.43</td>
<td>25.0731 1800</td>
<td></td>
</tr>
<tr>
<td># 3 (# = 100, m = 120)</td>
<td>241.1606 1.61</td>
<td>-</td>
<td>-</td>
<td>241.1606 241.1606</td>
</tr>
<tr>
<td># 4 (# = 20, m = 40)</td>
<td>156.6875 0.40</td>
<td>179.1224 5 5.52</td>
<td>NA 1800</td>
<td></td>
</tr>
<tr>
<td># 5 (# = 50, m = 100)</td>
<td>324.8785 1.17</td>
<td>370.9066 3 6.55</td>
<td>NA 1800 370.9066</td>
<td></td>
</tr>
<tr>
<td># 6 (# = 100, m = 100)</td>
<td>428.7200 2.37</td>
<td>472.0475 2 8.86</td>
<td>NA 1800 472.0475</td>
<td></td>
</tr>
<tr>
<td># 7 (# = 200, m = 30)</td>
<td>551.3160 0.68</td>
<td>570.8467 20 18.45</td>
<td>NA 1800 570.8467</td>
<td></td>
</tr>
<tr>
<td># 8 (# = 400, m = 80)</td>
<td>570.7248 3.51</td>
<td>601.6886 4 14.35</td>
<td>NA 1800 601.6886</td>
<td></td>
</tr>
<tr>
<td># 9 (# = 50, m = 20)</td>
<td>error 0.62</td>
<td>-</td>
<td>-</td>
<td>error 1800 890.7722</td>
</tr>
<tr>
<td># 10 (# = 10,000, m = 100)</td>
<td>NA 1800</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># 11 (# = 1,000, m = 1,000)</td>
<td>156.4414 1311</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># 12 (# = 2,000, m = 700)</td>
<td>238.4521 1800</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4: Comparison of our approximation method with Knitro and BARON for squared 2-norm constrained log-sum-exp maximization problem. The first column addresses the indexed problems, and gives the dimension of these problems (recall the decision variable $x$ is an $n$-dimensional vector and the function $f(Ax + b)$ takes $m$-dimensional vector input, hence $A$ is a matrix of size $m \times n$). The ‘value’ column stands for the best value corresponding solver computed, and ‘time’ stands for how many seconds it took for the solver to compute this. ‘start’ means the minimum number of starting points for Knitro to find the corresponding value. The column ‘upper’ is our upper bound approximation, and ‘lower’ is the lower bound approximation. We allow 30-minutes for solvers, hence ‘NA’ means that the solver cannot find any feasible point in the time limitation, otherwise the best solution found is written. The entry ‘error’ means that there are numerical errors and the solver stops.
These results show that BARON solver can only find the global optimum value (and guarantee) in the first problem, which is the smallest one. For the other problems BARON is not usable, which is mainly due to the fact that the log-sum-exp function is highly nonlinear as well as evaluating it is a hard task due to the exponential-terms. Moreover, we see that our approximation method (except Problems 1 and 2) is faster than Knitro without multi-start which is designed to find a local optimum. For Problems 4-7 Knitro cannot find the global optimum without using the multi-start option (which needs manual tuning), and there is no guarantee of global optimality at the end, while our method solves the problem with a global optimality guarantee (upper bounds are equal to lower bounds). For Problem 1 and Problem 2 we cannot find the global optimum value, which shows our method does not necessarily find the global optimum in all of the cases. Problem 9 is a problem where the exponential summands in the logarithm operator get very large and the computers accept these terms as infinity, so the solvers cannot compute any solution, and they do not automatically scale the problem. However, since our upper-bound approximation does not compute the log-sum-exp function directly, it does not suffer from such numerical issues. To compute a lower bound we need function evaluations and to address the numerical issues we can automatically scale the problem. Problem 10 has 10,000 variables, so none of the solvers can find an initial solution within 1800 seconds, while our method finds the upper and lower bounds within 0.29 seconds. Problem 11 has $m = 1,000$ and $n = 1,000$. This problem has a larger approximation gap, which may be a result of increasing dimension (recall that the proposed approximation problem has $m$ variables). Finally, Problem 12 has a large dimension in terms of $m$ and $n$. Knitro returns a feasible solution within 1800 seconds but not locally optimum (large $n$ is harder in the main problem). We also have a gap between the upper and lower bound approximations.

4.2 Log-Sum-Exp Maximization over an $\infty$-norm Constraint

We consider problem (1) with constraint $\|x - a\|_\infty \leq \rho$. In Case 3 we discussed that we can find the global optimum value of problem (1) by solving a mixed-integer convex optimization problem. When $f$ is a log-sum-exp function, then in our approach we can represent this problem as a mixed-integer exponential-cone problem and MOSEK can solve these type of problems efficiently. We use the same solvers as in the previous subsection as benchmark. In this problem, we find the exact global optimum value, i.e., we do not approximate this value. However, we do not have the feasible solution which attains this value. The results are presented in Table 5.

In general, our method’s computation time is higher compared to the previous problem, mainly because we solve a mixed-integer exponential cone optimization problem. However, we get a guarantee that we obtain the global optimum value, where Knitro, even with multi-start, cannot guarantee this. Similarly as the previous example, BARON is not scalable to larger problems. Problem 3 results once again show that for a problem where function evaluation is subject to numerical issues, we can easily find the global optimum value without computing the function itself.
4.3 Convex Quadratic Maximization over Linear Constraints

We consider the first problem type at Table 1, e.g., maximizing a convex quadratic function with respect to linear constraints. The state-of-the-art solver for convex quadratic maximization is CPLEX (version 12.6 onwards), which uses a branch-and-bound method based on McCormick relaxations and SDP cuts. The algorithm terminates at the global optimum, but as for any branch and bound algorithm, this may take exponential number of steps (Boyd and Mattingley 2007). On the one hand, our upper bound approximation method is a second-order cone optimization problem, so we use MOSEK solver for this purpose. On the other hand, our lower-bound approximation method solves multiple linear optimization problems, hence we use CPLEX for this purpose. The results are given in Table 6.

Table 5: Comparison of our exact method with Knitro and BARON for ∞-norm constrained log-sum-exp maximization. The meaning of the columns are mostly same with Table 4, and time is in seconds. Here, ‘Exact’ section stands for our method which computes exactly the global optimum value.

Table 6: Comparison of our approximation method and CPLEX with multiple linear constrained convex quadratic maximization. Here the column names that are used in the previous tables have the same meaning, and time is similarly in seconds. Recall that $q$ is the number of (linear) constraints. The ‘Upper Bound’ section is our proposed upper bound, and ‘Lower Bound’ is the proposed lower bound. In ‘restricted’ column of CPLEX, we are giving a time limit of the total time needed by our approximation method (upper + lower) and see the best CPLEX can find. Note that if the ‘restricted’ case finds the same value as its ‘value’ (e.g., the unlimited time case), it means CPLEX finds the global optimum solution but cannot guarantee global optimality yet.
Problems 1-4 are small to medium-sized problems. Here, we see that CPLEX finds the global optimum faster than our approximation method’s total time. At Problem 5 CPLEX finds the global optimum slower than our approximation, and if we give a time limitation of our approximation time it still finds the global optimum without optimality guarantee. A similar argument holds for Problem 6, but the run-time of CPLEX starts to get significantly high. Finally at Problem 7, CPLEX cannot find the global optimum solution in an hour (3600 seconds). In general, we see that our lower-bound values are all equal to global optimum (except problem 7) in a reasonable run-time, however we cannot guarantee global optimality or give tight upper bounds, unlike the other experiments we do.

4.4 Log-Sum-Exp Maximization over Linear Constraints

We consider the second problem type at Table 1, where we are interested in maximizing a log-sum-exp function over linear constraints. We use the state-of-the-art general purpose global optimization solver BARON as the main solver. We also use Knitro as a local optimization solver. Since our upper and lower bound problems are exponential cone representable problems, we use MOSEK solver for approximating this problem. The results can be found in Table 7.

<table>
<thead>
<tr>
<th>Size</th>
<th>Upper Bound</th>
<th>Lower Bound</th>
<th>Knitro</th>
<th>Knitro Multi-Start</th>
<th>BARON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>value</td>
<td>time</td>
<td>value</td>
<td>time</td>
<td></td>
</tr>
<tr>
<td>1 (size= 10)</td>
<td>35.2008</td>
<td>0.05</td>
<td>35.2008</td>
<td>0.01</td>
<td>16.8763</td>
</tr>
<tr>
<td>1 (size= 40)</td>
<td>248.7589</td>
<td>1.13</td>
<td>248.7589</td>
<td>0.01</td>
<td>248.7589</td>
</tr>
<tr>
<td>1 (size= 60)</td>
<td>386.0733</td>
<td>2.95</td>
<td>386.0733</td>
<td>0.01</td>
<td>282.6632</td>
</tr>
<tr>
<td>1 (size= 100)</td>
<td>676.808</td>
<td>9.98</td>
<td>676.808</td>
<td>0.01</td>
<td>676.8081</td>
</tr>
<tr>
<td>2 (size= 10)</td>
<td>65.2268</td>
<td>0.37</td>
<td>65.4826</td>
<td>0.30</td>
<td>65.4826</td>
</tr>
<tr>
<td>3 (size= 50)</td>
<td>145.338</td>
<td>9.73</td>
<td>145.338</td>
<td>0.01</td>
<td>145.338</td>
</tr>
<tr>
<td>4 (size= 100)</td>
<td>176.1041</td>
<td>241.57</td>
<td>176.1074</td>
<td>0.01</td>
<td>176.1074</td>
</tr>
<tr>
<td>5 (size= 10)</td>
<td>45.0356</td>
<td>0.18</td>
<td>45.0356</td>
<td>0.01</td>
<td>45.0356</td>
</tr>
<tr>
<td>6 (size= 30)</td>
<td>76.1283</td>
<td>14.71</td>
<td>76.0362</td>
<td>4.37</td>
<td>76.0362</td>
</tr>
</tbody>
</table>

Table 7: Comparison of our approximation method with Knitro and BARON. Size of the problem means \( n = q = m \) and these are equal to the given size, i.e., the number of variables, number of constraints, and input of the log-sum-exp function are of equal length. A star (*) next to the solution-time of BARON means the solver encountered numerical issues and it returns the best solution in the reported time.

Except for Problem 2 and Problem 6, we find global optimum upper and lower bounds, so we guarantee global optimality. For Problem 2 and Problem 6 we obtain tight upper bounds, and our lower bounds are globally optimum. Our upper bound approximation problem works fast, however it is an exponential cone problem with \( 2q + qm \) variables in total. So depending on the size of the problem, it may get slower to solve our approximation, and Problem 4 is a nice example to show this. However, in such big problems Knitro also needs a lot of multiple starting points, and does not find an upper-bound to the solution obtained, therefore does not have an optimality guarantee. Knitro needs multi-start to find the global optimum in all the problems except Problem 1 with \( n = 40 \). Also, we would like to point out that BARON solver can not
guarantee global optimality in several cases. This is because BARON has numerical difficulties in these problems.

4.5 Sum-of-Max-Linear-Terms Maximization over Linear Constraints

Finally, we consider the third problem type at Table 1. For this problem, we solve our approximation problems (which are all linear problems) with CPLEX. Also, we compare our approximations with directly solving this convex maximization problem via GUROBI version 9.0 (Gurobi Optimization 2018). Although GUROBI does not solve sum-of-max maximization problems implicitly, YALMIP gives the mixed integer linear optimization problem reformulation of this problem successfully (by applying logical programming). The reason we choose GUROBI as a benchmark solver is because it gives a better performance than CPLEX in our numerical experiments. Therefore, we compare our linear approximation problems with the performance of GUROBI on mixed integer optimization to solve the problem to global optimality. We know that mixed integer optimization is NP-hard (Nemhauser and Wolsey 1988). The results are given in Table 8.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Upper Bound</th>
<th>Lower Bound</th>
<th>GUROBI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>value</td>
<td>time</td>
<td>value</td>
</tr>
<tr>
<td># 1</td>
<td>23.2885</td>
<td>0.01</td>
<td>23.2885</td>
</tr>
<tr>
<td># 2</td>
<td>233.9417</td>
<td>0.01</td>
<td>233.9417</td>
</tr>
<tr>
<td># 3</td>
<td>1,169.4</td>
<td>0.20</td>
<td>1,032.5</td>
</tr>
<tr>
<td># 4</td>
<td>4,499.1</td>
<td>0.72</td>
<td>3,909.4</td>
</tr>
<tr>
<td># 5</td>
<td>27,939.0</td>
<td>16.69</td>
<td>21,617.0</td>
</tr>
<tr>
<td># 6</td>
<td>83,466</td>
<td>146.22</td>
<td>61,579</td>
</tr>
<tr>
<td># 7</td>
<td>113,706.8</td>
<td>0.01</td>
<td>113,706.8</td>
</tr>
<tr>
<td># 8</td>
<td>3,052.9000</td>
<td>1.37</td>
<td>3,052.9000</td>
</tr>
<tr>
<td># 9</td>
<td>2,894.6</td>
<td>42.38</td>
<td>2,699.7</td>
</tr>
<tr>
<td># 10</td>
<td>3,323.5000</td>
<td>356.23</td>
<td>2,997.4000</td>
</tr>
<tr>
<td># 11</td>
<td>3,032.0</td>
<td>0.4760</td>
<td>3,002.4</td>
</tr>
<tr>
<td># 12</td>
<td>3,452.2000</td>
<td>0.43</td>
<td>3,336.6000</td>
</tr>
<tr>
<td># 13</td>
<td>17,568.000</td>
<td>10.06</td>
<td>16,973.000</td>
</tr>
</tbody>
</table>

Table 8: Comparison of our approximation method with GUROBI for multiple linear constrained sum-of-max-linear-terms maximization. The column descriptions are same as the previous tables. The size of the problem is defined by \(n\) (number of variables), \(|I_k|\) (number of elements in the set of each max-term), \(K\) (number of max-terms). GUROBI solves the problem’s mixed integer linear optimization reformulation, hence for large sized problems this can take a vast amount of time. Therefore we give 1,000 seconds time limitations for each run of GUROBI. The ‘value’ column of GUROBI shows the value computed by GUROBI within 1,000 seconds, thus if the time is 1,000*, it means GUROBI cannot compute the global optimum solution within the time limitation and returns the best solution found. The ‘restricted’ column gives the best value GUROBI can compute within a limitation of the time that it takes for our method to find the upper and lower bound values.

Our method does not necessarily yield global optimum contrary to solving the mixed-integer optimization reformulation of the problem, however, in large problems our method converges...
to a lower bound solution very swiftly. On the other hand, GUROBI suffers severely from the curse of dimensional in mixed integer optimization. Therefore, we see that in bigger problems our lower bound quality is better than the best solution found by GUROBI, while in small sized problems the solver converges to the global optima very quickly. The speed of our method makes it very practical to use it in this problem. We find the global optimum (and certify optimality) in problems 1,2,7,8. Note that in problem 1 the global optimality of our approximation is not a coincidence because in this problem we have $K = 1$, and we know linear decision rules are optimal in this case (Ardestani-Jaafari and Delage 2016). Moreover, GUROBI cannot find the global optimum in the allowed 1,000 seconds for problems 4,5,6,8,9,10,12,13. Except problems 3,4,5,6, our lower bound solutions are better than the ones found by GUROBI, or equal with a faster computation time. At problems 3 and 4, GUROBI finds a better value than our lower bound in the same time (time restricted case). In the largest problems, namely 6 and 13, the best solutions found by GUROBI in 1,000 seconds are considerably lower than our lower bound solutions, where our method computes the upper and lower bounds within 150 and 11 seconds, respectively.

5 Conclusions

Maximizing a convex function over convex constraints is known to be a hard problem. Even the simplest case, maximizing a nonlinear convex function over linear constraints, is NP-hard. Although there exist a great number of methods in the field of convex maximization, these methods are in general restricted to limited settings, and they are impractical for large scale problems. Currently the only convex maximization problem that one can use dedicated solvers is convex quadratic maximization over convex quadratic constraints (latest versions of (IBM ILOG CPLEX 2014, Gurobi Optimization 2018)). Although the solvers give impressive results on small-to-medium sized problems, in larger problems they cannot return the global optimum value and the best solution found in a practical time limitation is poor. This is due to the nature of branch-and-bound algorithms that are embedded in these solvers. For general convex maximization problems, e.g., norm constrained geometric maximization, one can either use general purpose local optimization solvers, or global optimization solvers. We show that, as the size of convex maximization problem gets larger, the local optimization solvers return solutions with large global optimality gaps, and in many cases the global optimization solvers cannot terminate.

In this work, we show how to use adjustable robust optimization techniques to approximately solve the convex maximization problem. More specifically, we show how to carry the difficulty of the main non-convex problem to the nonlinearity of an equivalent (convex) ARO problem. Once we obtain the ARO reformulation, we show how to exploit the rich literature in the ARO field for approximately solving hard uncertain constraints. We show that in presence of a single norm constraint one can analytically eliminate the adjustable variable to obtain a static robust optimization problem. Then, one can solve the resulting convex optimization problem whose complexity depends on the original optimization function and the original norm
constraint. Moreover, if the original formulation has multiple constraints, then there are many alternative ways to approximately solve the resulting ARO problem (e.g., static solution of the ARO problem, decision rules). We thoroughly investigate the case of multiple linear constraints and show that the equivalent problem is an ARLO problem. Therefore, this problem can be approximated by using linear decision rules, and we derive the upper and lower approximation problems of three special convex maximization problems.

When the original problem has box constraints, then we can recover the global optimum value by solving a mixed-integer convex optimization problem (which is a well-studied, efficiently solvable problem). For other cases, we obtain upper and lower bounds. The performance of the approximation can be measured by the gap between the upper and lower bounds. For linear constrained convex maximization problems, the performance of the approximation is closely related to the optimality of linear decision rules in special ARO settings. For example, we know that LDRs are optimal for linear constrained single-max-term maximization problem, hence we see that in such problems we obtain the global optimum value by solving the approximation problems.

In the numerical experiments we see that, in general, our approximation problems are solved very efficiently with tight approximations. To be more specific, in log-sum-exp maximization over a ball constraint, we find the global optimum in most of the problems (and the remaining have very tight bounds). Solving each experiment roughly takes less than half a second, where some of these problems cannot be solved by optimization solvers in half-hour. In log-sum-exp maximization over box constraints, contrary to the solvers, we obtain the global optimum value in all problems, as this is guaranteed. However, since we are solving mixed-integer exponential cone optimization, the computation times are higher compared to the ball constrained problem. In quadratic maximization over linear constraints, our lower bound values are very tight, and they are optimal in most of the cases. In these problem, however, the optimization solver works well in small-to-medium sized problems. Our novelty is for large problems, where the solver takes too much time to guarantee optimality (e.g., more than an hour), and even cannot find a solution better than the starting point of the algorithms. In log-sum-exp maximization over linear constraints, we find globally optimal upper and lower bounds except two cases, and the exceptions are also solved with very tight bounds. In this problem the solvers either face numerical difficulties, or need a lot of multiple starting points to find a good solution. Finally, in sum-of-max-linear-terms maximization over linear constraints, our approximation problems are being solved with tight bounds very swiftly since these are linear optimization problems. The solvers solve an equivalent mixed-integer reformulation of this problem, hence their termination guarantees global optimality, although in larger sized problems they cannot terminate and the best solution returned is worse than our lower bound solutions.

There are two directions as future work: application and theory. For the former one, we show that there are numerous real-life applications of convex maximization. A primary future work could be applying our approximation methods to these real-life problems. For example, adopting our approximations in DC programming applications would be very interesting, since
it is essential to obtain good lower-bound solutions at the convex maximization step of the convex-concave method. Another implementation can be integrating our methodology with the optimization solvers. In the numerical experiments we are comparing our methods with the solvers, but these are not necessarily alternative methods. For example, at some problems our approach finds upper and lower bound values very swiftly, but the optimization solver guarantees global optimality upon termination. So the solver can first solve our method, obtain a lower bound solution along with an upper bound value, and proceed from this point. As for theory, one can apply different approximation methods to the ARO reformulation of the convex maximization problem. Piecewise-linear decision rules or nonlinear decision rules may be fruitful in some cases. We do not give theoretical guarantees of the approximation gaps, hence developing such guarantees is essential. Furthermore, one can develop an algorithm by dividing the feasible region of the convex maximization problem to parts and solve our approximations in each part, which may give tighter bounds.

Acknowledgments

The research of the first author was funded by Center of Economic and Business Research (CentER) of Tilburg University during his time there.

References


Appendices

A RLT Approximation of Problem (9)

Suppose we have a symmetric matrix variable $V \in S_m^{m \times m}$ such that $V = ww^\top$. The following can be applied to replace the convex term $||A^\top w||_2$:

$$||A^\top w||_2 = \sqrt{w^\top AA^\top w} = \sqrt{\text{tr}(w^\top AA^\top w)} = \sqrt{\text{tr}(A^\top ww^\top A)}.$$ 

Therefore, we use the following concave reformulation of $||A^\top w||_2$:

$$||A^\top w||_2 = \sqrt{\text{tr}(A^\top VA)}.$$ 

This concave reformulation is of course based on the assumption $V = ww^\top$. Moreover, we use the main idea of RLT and multiply each of the original constraints $\alpha_i^\top w - \beta_i \leq 0, \alpha_j^\top w - \beta_j \leq 0$ to obtain:

$$\begin{align*}
(\alpha_i^\top w - \beta_i)(\alpha_j^\top w - \beta_j) & \geq 0 \\
\iff \alpha_i^\top w w^\top \alpha_j - (\beta_i \alpha_j + \beta_j \alpha_i)^\top w + \beta_i \beta_j & \geq 0 \\
\iff \alpha_i^\top V \alpha_j - (\beta_i \alpha_j + \beta_j \alpha_i)^\top w + \beta_i \beta_j & \geq 0.
\end{align*}$$

(28)

Although $V = ww^\top$ is assumed, it is a non-convex constraint, so we relax it as:

$$V \succeq ww^\top \iff \begin{pmatrix} V & w \\ w^\top & 1 \end{pmatrix} \succeq 0.$$ 

(30)

Thus, problem (9) is approximated by the following convex optimization problem:

$$\begin{align*}
\sup_{V \in S_m^{m \times m}, \ w \in \mathbb{R}^m} & \sqrt{\text{tr}(A^\top VA)} + a^\top A^\top w + b^\top w - f^*(w) \\
\text{s.t.} & \alpha_i^\top w - \beta_i \leq 0, \quad i = 1, \ldots, d \\
& \alpha_i^\top V \alpha_j - (\beta_i \alpha_j + \beta_j \alpha_i)^\top w + \beta_i \beta_j \geq 0, \quad i \leq j = 1, \ldots, d \\
& \begin{pmatrix} V & w \\ w^\top & 1 \end{pmatrix} \succeq 0,
\end{align*}$$

(31)

which concludes the proof.

B Proof of Theorem 3

We showed that problem (19) can be represented as ARO problem (21). As shown by Roos et al. (2018), we can lift the nonlinear term $f^*(w)$ to the uncertainty set by introducing an auxiliary uncertain parameter $w_0$. Hence, the set of constraints of the ARO problem is equivalent to:

$$\forall \begin{pmatrix} w_0 \\ w \end{pmatrix} \in W, \ \exists \lambda \in \mathbb{R}^q : \begin{cases} 
d^\top \lambda + b^\top w + w_0 \leq \tau \\
D^\top \lambda \geq A^\top w \\
\lambda \geq 0,
\end{cases}$$

(32a)

(32b)

(32c)

where we define the new uncertainty set as

$$W = \left\{ \begin{pmatrix} w_0 \\ w \end{pmatrix} \in \mathbb{R}^{m+1} : w_0 + f^*(w) \leq 0 \right\}.$$ 

(33)
A safe approximation of the constraint set is obtained by using a linear decision rule for the adjustable variable:

$$\lambda = u + Vw + rw_0,$$

where $u \in \mathbb{R}^q, V \in \mathbb{R}^{q \times m}$ and $r \in \mathbb{R}^q$. Substituting this LDR in (32a) leads to

$$d^\top \lambda + b^\top w + w_0 \leq \tau \quad \forall \left(\begin{array}{c} w_0 \\ w^\top \end{array}\right) \in W$$

$$\iff d^\top (u + Vw + rw_0) + b^\top w + w_0 \leq \tau \quad \forall \left(\begin{array}{c} w_0 \\ w^\top \end{array}\right) \in W$$

$$\iff d^\top u + \left(\begin{array}{c} w_0 \\ w^\top \end{array}\right)^\top \left(1 + d^\top r \right) \leq \tau \quad \forall \left(\begin{array}{c} w_0 \\ w^\top \end{array}\right) \in W$$

$$\iff d^\top u + \delta^* \left(\left(1 + d^\top r \right) \left| W \right. \right) \leq \tau. \quad (34)$$

To be able to find the tractable robust counterpart of (34), we derive the support function of the new uncertainty set $W$, which is

$$\delta^* \left(\left(\begin{array}{c} z_0 \\ z^\top \end{array}\right) \left| W \right. \right) = \sup_{(w_0, w)^\top \in W} \{z_0w_0 + z^\top w\}$$

$$= \begin{cases} 
\sup_{w_0 \in \mathbb{R}^m} \{z^\top w - z_0f^*(w)\} & \text{if } z_0 \geq 0 \\
+\infty & \text{otherwise}
\end{cases}$$

$$= \begin{cases} 
z_0f \left(\left(\begin{array}{c} z \\ \frac{z}{z_0} \end{array}\right) \right) & \text{if } z_0 \geq 0 \\
+\infty & \text{otherwise.}
\end{cases} \quad (35)$$

Note that when $z_0 = 0$, we take $\lim_{z_0 \to 0} z_0f \left(\left(\begin{array}{c} z \\ \frac{z}{z_0} \end{array}\right) \right)$ for the perspective, which is known as the recession function of $f$ evaluated at $z$ (Rockafellar 1970). By substituting (35) into (34) we obtain:

$$d^\top u + \delta^* \left(\left(1 + d^\top r \right) \left| W \right. \right) \leq \tau$$

$$\iff \begin{cases} 
d^\top u + (1 + d^\top r)f \left(\frac{V^\top d + b}{1 + d^\top r} \right) \leq \tau \\
1 + d^\top r \geq 0.
\end{cases}$$

Hence, by using LDRs (32a) becomes exactly (22a).

Following the same steps for (32b) yields us to (22b):

$$D^\top \lambda \geq A^\top w \quad \forall \left(\begin{array}{c} w_0, \\ w^\top \end{array}\right) \in W$$

$$\iff D_i^\top \lambda \geq A_i^\top w \quad \forall \left(\begin{array}{c} w_0, \\ w^\top \end{array}\right) \in W, \ i = 1, \ldots, n$$

$$\iff \begin{cases} 
-D_i^\top u + (-D_i^\top r)f \left(\frac{A_i - V^\top D_i}{-D_i^\top r} \right) \leq 0 \\
-D_i^\top r \geq 0
\end{cases} \quad i = 1, \ldots, n.
Similarly (32c) becomes (22c):

\[ \lambda \geq 0 \iff -u_i - V_i w - r_i w_0 \leq 0 \quad \forall \left( w_0, \ w^\top \right)^\top \in W, \ i = 1, \ldots, q \]

\[ \iff \begin{cases} 
- u_i + (-r_i) f \left( \frac{-V_i}{r_i} \right) \\
-r_i \geq 0 
\end{cases} \ i = 1, \ldots, q. \]

Since we use an LDR for the adjustable variable, the optimal objective value of (22) is an upper bound to (19).

C Complete Derivation of Specific Problems in Section 3

C.1 Quadratic Optimization

Here we consider problem (19) when the objective function is a convex quadratic function. For the problem of maximizing a convex quadratic function over a polyhedron, we can find an upper bound by solving a second-order cone optimization problem, and we can find a lower bound by solving a linear optimization problem.

Consider the convex quadratic function \( g : \mathbb{R}^n \mapsto \mathbb{R} \) defined by:

\[ g(x) = x^\top Q x + \ell^\top x, \]

where \( \ell \in \mathbb{R}^n \) and \( Q \) is a symmetric positive semi-definite (psd) matrix. Maximizing this function over a polyhedral set can be written as the robust optimization problem:

\[
\begin{align*}
\min & \quad \tau \\
\text{s.t.} & \quad x^\top Q x + \ell^\top x \leq \tau, \quad \forall x \in U,
\end{align*}
\]

(36)

where \( U = \{ x \in \mathbb{R}^n_+ : D x \leq d \} \) for \( D \in \mathbb{R}^{q \times n}, \ d \in \mathbb{R}^q \). We use the conic representation of the constraints of problem (36):

\[
\left\| \left(1 + \frac{\ell^\top x - \tau}{2} \right)Lx \right\|_2 - \left(1 - \frac{\ell^\top x + \tau}{2} \right) \leq 0,
\]

where \( L \) is the psd decomposition \( Q = L^\top L \). Therefore, the constraint of problem (36) can be written as a robust conic constraint. Define \( f : \mathbb{R}^{m+1} \times \mathbb{R} \mapsto \mathbb{R} \) by:

\[
f \left( \begin{array}{c} z \\ \tilde{z} \end{array} \right) = \left\| z \right\|_2 + \tilde{z},
\]

(37)

with \( z \in \mathbb{R}^{m+1} \) and \( \tilde{z} \in \mathbb{R} \). It can be verified that \( f \) is positively homogeneous and that the conjugate of this function for \( w \in \mathbb{R}^{m+1}, \ \tilde{w} \in \mathbb{R} \) is:

\[
f^* \left( \begin{array}{c} w \\ \tilde{w} \end{array} \right) = \begin{cases} 
0 & \text{if } \tilde{w} = 1 \text{ and } \|w\|_2 \leq 1 \\
+\infty & \text{otherwise.}
\end{cases}
\]

Defining

\[
A = \begin{bmatrix} L \\
\ell^\top / 2 \\
\ell^\top / 2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\
(1 - \tau) / 2 \\
(-1 - \tau) / 2 \end{bmatrix},
\]

(38)
it follows that:
\[
f(Ax + b) = f\left( \begin{pmatrix} Lx \\ (1 + \ell^\top x - \tau)/2 \\ (\ell^\top x - 1 - \tau)/2 \end{pmatrix} \right) = \left\| \begin{pmatrix} Lx \\ (1 + \ell^\top x - \tau)/2 \\ (\ell^\top x - 1 - \tau)/2 \end{pmatrix} \right\|_2 - (1 - \ell^\top x + \tau)/2.
\]
Hence, the constraint of problem (36) is equivalent to \( f(Ax + b) \leq 0 \). Therefore, problem (36) can be rewritten as:
\[
\begin{align*}
\min & \quad \tau \\
\text{s.t.} & \quad f\left( \begin{pmatrix} Lx \\ (1 + \ell^\top x - \tau)/2 \\ (\ell^\top x - 1 - \tau)/2 \end{pmatrix} \right) \leq 0, \quad \forall x \in U.
\end{align*}
\]  
(39)

An upper bound of this problem can now be obtained by applying Theorem 3 and exploiting the positive homogeneity of \( f \) (see Appendix D). The upper bound is the optimal value of the problem:
\[
\begin{align*}
\min & \quad \tau \\
\text{s.t.} & \quad d^\top u + \bar{v}^\top d - (1 + \tau)/2 + \left\| \begin{pmatrix} \tilde{V}^\top d \\ \hat{v}^\top d + (1 - \tau)/2 \end{pmatrix} \right\|_2 \leq 0 \\
& \quad -D_i^\top u + \ell_i/2 - \hat{v}^\top D_i + \left\| \begin{pmatrix} L_i - \tilde{V}^\top D_i \\ \ell_i/2 - \hat{v}^\top D_i \end{pmatrix} \right\|_2 \leq 0, \quad i = 1, \ldots, n \\
& \quad -u_i - \bar{v}_i + \left\| \begin{pmatrix} -\tilde{V}^\top(i) \\ -\hat{v}_i \end{pmatrix} \right\|_2 \leq 0, \quad i = 1, \ldots, q,
\end{align*}
\]  
(40)
in which the variables are \( \tau \in \mathbb{R}, u \in \mathbb{R}^q, \bar{v} \in \mathbb{R}^q, \hat{v} \in \mathbb{R}^q, \tilde{V} \in \mathbb{R}^{q \times m}. \)

In order to compute a lower bound, we use the optimal solution \((\tau, u, \bar{v}, \hat{v}, \tilde{V})\) to problem (40) by obtaining a collection of worst case scenarios \( \bar{W} \) from (25), (26), and (27). These problems can be solved analytically as explained in Appendix E. This yields the scenarios:
\[
\begin{align*}
\bar{W} & \leftarrow h \left( \begin{pmatrix} \tilde{V}^\top d \\ d^\top \bar{v} + (1 - \tau)/2 \end{pmatrix} \right) \\
\bar{W} & \leftarrow h \left( \begin{pmatrix} L_i - \tilde{V}^\top D_i \\ \ell_i/2 - \hat{v}^\top D_i \end{pmatrix} \right) \\
\bar{W} & \leftarrow h \left( \begin{pmatrix} -\tilde{V}^\top(i) \\ -\hat{v}_i \end{pmatrix} \right)
\end{align*}
\]  
(41)
where \( h(a) = a/\|a\|_2 \) normalizes its input. Using these worst-case scenarios, the candidate solutions \( \bar{x}^{(i)} \) are obtained by solving (24), and we can substitute them in the main objective function as \( f(A\bar{x}^{(i)} + b) \) to find the best lower bound.

### C.2 Geometric Optimization

Geometric Optimization (GO) is a class of optimization problems originally introduced by Duffin (1967). A practical tutorial can be found in the work of Boyd et al. (2007). Even though it can have many
representations, we focus on the GO variant where the objective is maximizing the convex log-sum-exp objective. The log-sum-exp function \( f : \mathbb{R}^m \to \mathbb{R} \) is defined as
\[
f(z) = \log \left( \sum_{i=1}^m \exp(z_i) \right),
\]
and we are interested in solving problems of the following type
\[
\max_x f(Ax + b) = \log \left( \sum_{i=1}^m \exp(A_{i(j)}x + b_i) \right)
\text{ s.t. } x \in U,
\]
where \( U = \{ x \in \mathbb{R}_+^n : Dx \leq d \} \). This problem may appear in robust geometric optimization problems. If one applies the adversarial approach for such problems, in the step of adding worst-case uncertainty realization to the discrete uncertainty set, one will need to maximize the convex geometric function.

The conjugate \( f^* : \mathbb{R}^n \to \mathbb{R} \) of the log-sum-exp function (42) is
\[
f^*(w) = \begin{cases} \sum_{i=1}^m w_i \log(w_i) & \text{if } w \in \mathbb{R}_+^m \text{ and } \sum_{i=1}^m w_i = 1 \\ \infty & \text{otherwise.} \end{cases}
\]

We observe that \( f^*(w) \) is the negative-entropy function of \( w \) on its domain, which is a standard \( m \)-dimensional simplex. It is well known that the negative entropy is a strictly convex function. Next we show that the upper bound and lower bound approximation problems (of problem (43)) are exponential cone representable. This allows one to use the power of today’s conic programming solvers, e.g., Mosek’s exponential cone optimization solver. We start by introducing the exponential cone, which is the following convex subset of \( \mathbb{R}^3 \):
\[
\mathcal{K}_{\exp} = \{(x_1, x_2, x_3) : x_1 \geq x_2 \exp(x_3/x_2), x_2 > 0 \} \cup \{(x_1, 0, x_3) : x_1 \geq 0, x_3 < 0 \}.
\]

So, the exponential cone is the set of points which satisfy \( x_1 \geq x_2 \exp(x_3/x_2), x_1, x_2 > 0 \). In the next corollary we show that the upper bound problem (22) is exponential cone representable.

**Corollary 1 (Upper Bound Approximation).** Upper bound problem (22) is exponential cone representable with the following problem with variables \( r \in \mathbb{R}^q, u \in \mathbb{R}^q, V \in \mathbb{R}^{q \times m}, \tau \in \mathbb{R}, z^{(1)} \in \mathbb{R}^n, z^{(2)} \in \mathbb{R}^{n \times n}, z^{(3)} \in \mathbb{R}^{q \times n} :$$
\[
\begin{align*}
1 + d^T r & \geq \sum_{j=1}^n z_{(1)}^{(j)} \\
\begin{pmatrix} z_{(1)}^{(j)}, 1 + d^T r, (V_j^T d + b_j - \tau + d^T u) \end{pmatrix} & \in \mathcal{K}_{\exp} \\
-D_i^T r & \geq \sum_{j=1}^n z_{(2)}^{(ij)} & i = 1, \ldots, n \\
\begin{pmatrix} z_{(2)}^{(ij)}, -D_i^T r, (A_{i(j)} - V_j^T D_i - D_i^T u) \end{pmatrix} & \in \mathcal{K}_{\exp} \\
-r_i & \geq \sum_{j=1}^n z_{(3)}^{(ij)} & i = 1, \ldots, q \\
\begin{pmatrix} z_{(3)}^{(ij)}, -r_i, (-V_{j(i)} - u_i) \end{pmatrix} & \in \mathcal{K}_{\exp}.
\end{align*}
\]

**Proof.** Roos et al. (2018) show that if a function is conically representable, so is its perspective in the same cone. Log-sum-exp is an exponential cone representable function (MOSEK ApS 2019a), and we
show how to represent a convex inequality system of its perspective with exponential cones. Consider the following set of constraints:

\[
\begin{aligned}
t &\geq x_0 \log(\exp(x_1/x_0) + \ldots + \exp(x_n/x_0)) \\
x_0 &> 0.
\end{aligned}
\]  

By using the proof in (Roos et al. 2018), we can write the following equivalent constraint set:

\[
\begin{aligned}
x_0 &\geq \sum_{i=1}^{n} z_i \\
(z_i, x_0, (x_i - t)) &\in K_{\exp}, \quad i = 1, \ldots, n.
\end{aligned}
\]  

Since constraints of type (46) appear in the upper bound approximation problem (22), we can use the equivalent representation (47) in each of the constraints to obtain problem (45).

The lower bound can also be obtained by solving exponential cone programs. The worst-case scenario collection is obtained by solving (25), (26), (27). Here \((r, u, V, \tau)\) are all parameters taken from the solution of the upper bound problem. This set of problems can be formulated as exponential conic problems, which is shown in the next corollary.

**Corollary 2 (Lower Bound Scenarios).** The problems (25), (26), and (27) can be written as the following exponential cone problems:

\[
\begin{align*}
(25) : \quad & \arg \max_{w, t} \left\{ (1 + d^T r)(\sum_{j=1}^{m} t_j) + (d^T V + b^T)w + d^T u - \tau \right\} \\
& \text{s.t.} \quad (1, w_j, t_j) \in K_{\exp}, \quad j = 1, \ldots, m \\
& \quad \sum_{j=1}^{m} w_j = 1, \\
(26) : \quad & \arg \max_{w, t} \left\{ (-D_i^T r)(\sum_{j=1}^{m} t_j) + (A_i^T - D_i^T V)w - D_i^T u \right\} \\
& \text{s.t.} \quad (1, w_j, t_j) \in K_{\exp}, \quad j = 1, \ldots, m \\
& \quad \sum_{j=1}^{m} w_j = 1, \\
(27) : \quad & \arg \max_{w, t} \left\{ (-r_i)(\sum_{j=1}^{m} t_j) + (-V(i))w - u_i \right\} \\
& \text{s.t.} \quad (1, w_j, t_j) \in K_{\exp}, \quad j = 1, \ldots, m \\
& \quad \sum_{j=1}^{m} w_j = 1.
\end{align*}
\]

**Proof.** Serrano (2015) shows that the negative entropy function is exponential conically representable, and the following problems are equivalent (here \(w \in \mathbb{R}^m\):

\[
\begin{aligned}
\max_w \{ c_0 ( - \sum_{i=1}^{m} w_i \log(w_i)) \} &\quad = \max_{w, t} \{ c_0 \sum_{i=1}^{m} t_i \} \\
& \text{s.t.} \quad w_i \geq 0, \quad i = 1, \ldots, m. \\
\text{s.t.} \quad (1, w_i, t_i) \in K_{\exp}, \quad i = 1, \ldots, m.
\end{aligned}
\]

The result now follows by substitution of the conjugate (44) in (25), (26), and (27), respectively and then applying equivalence (48).

**C.3 Sum-of-Max-Linear-Terms Optimization**

Formally, the sum-of-max-terms function \(f : \mathbb{R}^m \rightarrow \mathbb{R}\) is written as

\[
f(z) = \sum_{k=1}^{K} \max_{j \in \mathcal{I}_k} \{ z_j \},
\]

where the set \(\mathcal{I}_k \subseteq \{1, \ldots, m\}\) for each \(k \in \{1, \ldots, K\}\). Moreover, we can assume \(\mathcal{I}_k \cap \mathcal{I}_\ell = \emptyset\) for any \(k \neq \ell\) and \(\bigcup_{k=1}^{K} = \{1, \ldots, m\}\) without loss of generality, since otherwise we can add components to \(z\) to
make this statement hold. The sum-of-max-linear-terms function we cover at this section is represented as

\[ f(Ax + b) = \sum_{k=1}^{K} \max_{j \in I_k} \{ A_{(j)} x + b_j \}, \]

which is a convex and positively homogeneous function. The main convex maximization problem we are interested in is maximizing \( f(Ax + b) \) over \( U = \{ x \in \mathbb{R}^n_+ : D x \leq d \} \), formally:

\[
\begin{align*}
\max_x & \quad \sum_{k=1}^{K} \max_{j \in I_k} \{ A_{(j)} x + b_j \} \\
\text{s.t.} & \quad x \in U.
\end{align*}
\]

(50)

This problem naturally arises when one applies the adversarial approach to robust optimization problems with uncertain sum-of-max-linear-terms constraints.

The conjugate of sum-of-max-linear-terms (49) is given by Roos et al. (2018) as:

\[
\begin{align*}
\min_{w} & \quad 0 \quad \text{if } w_i \geq 0 \ \forall \ i = 1, \ldots, m, \ \sum_{j \in I_k} w_j = 1 \\
& \quad \infty \quad \text{otherwise}.
\end{align*}
\]

The formulation of the upper bound approximation for maximizing sum-of-max-linear-terms function over a polyhedron can be greatly simplified. This is due to the fact that sum-of-max-linear-terms function is a positively homogeneous function as well as the trick of introducing auxiliary variables which give us a linear optimization problem in return.

Since the function is a positively homogeneous function, we can write the upper bound approximation problem (22) as:

\[
\begin{align*}
\min_{u \in \mathbb{R}^q, V \in \mathbb{R}^{q \times m}, \tau} & \quad \tau \\
\text{s.t.} & \quad d^T u + \sum_{k=1}^{K} \max_{j \in I_k} \{ V_j^T d + b_j \} \leq \tau \\
& \quad -D_i^T u + \sum_{k=1}^{K} \max_{j \in I_k} \{ A_{(i)} - V_j^T D_i \} \leq 0 \quad i = 1, \ldots, n \\
& \quad -u_i + \sum_{k=1}^{K} \max_{j \in I_k} \{ V_{(i)j} \} \leq 0 \quad i = 1, \ldots, q.
\end{align*}
\]

(51)

Problem (51) can be reformulated as a linear optimization problem by using auxiliary variables. The worst-case scenarios for computing the lower bound are obtained from problems (25), (26), and (27), which simplify to:

\[
\begin{align*}
(25) : & \quad \arg \max_{w \in \text{dom} f^*} \left\{ d^T (u + V w) + b^T w - \tau \right\}, \\
(26) : & \quad \arg \max_{w \in \text{dom} f^*} \left\{ A_{(i)}^T w - D_i^T (u + V w) \right\}, \quad i = 1, \ldots, n \\
(27) : & \quad \arg \max_{w \in \text{dom} f^*} \left\{ -u_i - V_{(i)} w \right\}, \quad i = 1, \ldots, q,
\end{align*}
\]

where we do not have the conjugate terms since \( f \) is a homogeneous function so its conjugate takes value 0. Therefore, the worst-case scenarios of each constraint are given by the following problems:

- For (52):

\[
\begin{align*}
\max_w & \quad d^T (u + V w) + b^T w - \tau \\
\text{s.t.} & \quad w_j \geq 0 \quad \text{for } j = 1, \ldots, m \\
& \quad \sum_{j \in I_k} w_j = 1 \quad \text{for } k = 1, \ldots, K.
\end{align*}
\]

(55)
Recalling the only variable here is \( w \), this is a linear optimization problem. Moreover, since we have \( I_k \cap I_{k'} = \emptyset \) for \( k \neq k' \), we can separate this problem to \( K \) independent optimization problems, where each problem \( k \) is:

\[
c_k = \max_{w \geq 0} \quad d^T (u + V_{(j)} y) + b_{(j)}^T y
\]

s.t. \( \sum_{i=1}^{|I_k|} y_i = 1 \).

Here \( y \in \mathbb{R}^{|I_k|} \) is the \( w \) components corresponding to the \( k \)-th term in the sum-of-max-linear-terms function definition. Similarly, \( V_{(j)}, b_{(j)} \in \mathbb{R}^{|I_k|} \) are the components of \( V, b \) corresponding to the \( k \)-th term. Notice that problem (56) is a linear optimization problem over a simplex. The optimal value will have \( y_i = 1 \) for some \( i \) and \( y_i' = 0 \) for all \( i' \neq i \). Therefore, the solution is

\[
c_k = \max_{i = 1, \ldots, |I_k|} \{ d^T (u + V_{(j)},i) \} + b_{(j),i},
\]

where \( V_{(j),i}, b_{(j),i} \) represent the \( i \)-th columns of \( V_{(j)} \) and \( b_{(j)} \), respectively. Hence, the optimal value of (55) is given by \( -\tau + \sum_{k=1}^K c_k \). The arg max value can be retrieved easily by detecting which \( y_i \) variables took value 1; there will be exactly \( K \) ones in the result and the rest will be zeros.

- For (53), for all \( i = 1, \ldots, n \):

\[
\max_w \quad A_i^T w - D_i^T (u + V w)
\]

s.t.

\[
w_j \geq 0, \quad \text{for } j = 1, \ldots, m
\]

\[
\sum_{j \in I_k} w_j = 1, \quad \text{for } k = 1, \ldots, K.
\]

Similarly, this problem can be separated to \( K \) independent linear optimization problems over simplices. The optimal solution can be found analytically.

- For (54), for all \( i = 1, \ldots, q \):

\[
\max_w \quad -u_i - V_{(j)} w
\]

s.t.

\[
w_j \geq 0, \quad \text{for } j = 1, \ldots, m
\]

\[
\sum_{j \in I_k} w_j = 1, \quad \text{for } k = 1, \ldots, K.
\]

This problem can be solved analytically once again, concluding that all of the worst-case scenario finding procedure can be solved analytically.

## D Upper Bound Approximation of Quadratic Maximization via SOCO

We follow Theorem 3 to apply the upper bound approximation for problem (39). Because \( f \) is a positively homogeneous function, the upper bound problem (22) reduces to the following problem for the variables \( u \in \mathbb{R}^q, V \in \mathbb{R}^{q \times (m+2)}, \tau \in \mathbb{R} \):

\[
\min \tau
\]

s.t.

\[
d^T u + f (V^T d + b) \leq 0
\]

\[
-D_i^T u + f \left( A_i - V^T D_i \right) \leq 0, \quad i = 1, \ldots, n
\]

\[
u_i + f \left( -V_{(i)} \right) \leq 0, \quad i = 1, \ldots, q.
\]

Since \( V \) has \( m+2 \) columns, we represent it as:

\[
V = \begin{bmatrix} \tilde{V} & \hat{\nu} & \hat{\theta} \end{bmatrix} \quad \text{where} \quad \tilde{V} \in \mathbb{R}^{q \times m}, \hat{\nu} \in \mathbb{R}^q, \hat{\theta} \in \mathbb{R}^q.
\]
Constraints of problem (57) can be simplified to respectively:

\[
\begin{align*}
&d^\top u + f(V^\top d + b) \leq 0 \\
&= d^\top u + f\left(\frac{\tilde{V}^\top d}{\tilde{v}^\top d + 1/2 - \tau/2}\right) \\
&= d^\top u + \tilde{v}^\top d - \frac{1}{2} - \frac{\tau}{2} + \frac{\|\tilde{V}^\top d + 1/2 - \tau/2\|^2_2}{2} \leq 0
\end{align*}
\]

\[
\begin{align*}
-D_i^\top u + f(A_i - V^\top D_i) &\leq 0 \\
&= -D_i^\top u + f\left(\frac{L_i - \tilde{V}^\top D_i}{\ell_i/2 - \tilde{v}^\top D_i}\right) \leq 0 \\
&= -D_i^\top u + \frac{\ell_i}{2} - \tilde{v}^\top D_i + \frac{\|L_i - \tilde{V}^\top D_i\|^2_2}{\ell_i/2 - \tilde{v}^\top D_i} \leq 0
\end{align*}
\]

Thus the upper bound approximation problem can be represented as the following second-order conic program:

\[
\min_{\tau} \quad \tau \\
\text{s.t.} \quad d^\top u + \tilde{v}^\top d - (1 + \tau/2) + \|\tilde{V}^\top d - \tilde{v}^\top d + (1 - \tau)/2\|^2_2 \leq 0 \\
-D_i^\top u + \frac{\ell_i}{2} - \tilde{v}^\top D_i + \frac{\|L_i - \tilde{V}^\top D_i\|^2_2}{\ell_i/2 - \tilde{v}^\top D_i} \leq 0, \quad i = 1, \ldots, n \\
-u_i - \tilde{v} + \frac{\|\tilde{V}^\top(i)\|^2_2}{\tilde{v}^\top D_i} \leq 0, \quad i = 1, \ldots, q.
\]

### E Lower Bound Scenarios for Quadratic Maximization

In the light of problems (25), (26), (27), the worst-case scenarios are collected by:

\[
(25): \quad \arg \max_{(w, \hat{w}) \in \text{dom } f^*} \left\{-\left(1 + d^\top r\right) f^*\left(\frac{w}{\hat{w}}\right) + (d^\top V + b^\top) \left(\frac{w}{\hat{w}}\right) + d^\top u\right\}, \tag{59}
\]

\[
(26): \quad \arg \max_{(w, \hat{w}) \in \text{dom } f^*} \left\{(D_i^\top r) f^*\left(\frac{w}{\hat{w}}\right) + (A_i^\top - D_i^\top V) \left(\frac{w}{\hat{w}}\right) - D_i^\top u\right\}, \quad i = 1, \ldots, n \tag{60}
\]

\[
(27): \quad \arg \max_{(w, \hat{w}) \in \text{dom } f^*} \left\{-u_i - V_{(i)} \left(\frac{w}{\hat{w}}\right) + r_i f^*\left(\frac{w}{\hat{w}}\right)\right\}, \quad i = 1, \ldots, q. \tag{61}
\]
We already showed the convex conjugate of $f$ takes value 0 in its domain. Recalling

$$A = \begin{bmatrix} L \\ \ell^T/2 \\ \ell^T/2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ (1-\tau)/2 \\ (-1-\tau)/2 \end{bmatrix}, \quad V = \begin{bmatrix} \tilde{V} & \hat{v} & \bar{v} \end{bmatrix},$$

we can rewrite problem (59) as:

$$\max_{w \in \mathbb{R}^{m+1}, \tilde{w} \in \mathbb{R}^d} \left( d^T \begin{bmatrix} \tilde{V} & \hat{v} & \bar{v} \end{bmatrix} w + \begin{bmatrix} 0 \\ (1-\tau)/2 \\ (-1-\tau)/2 \end{bmatrix} \right) + d^T u$$

s.t. $\tilde{w} = 1$, $||w||_2 \leq 1$.

Using $\tilde{w} = 1$, we can eliminate $\tilde{w}$ from the problem. Moreover, $w$ only appears as a linear term, so we can change $||w||_2 \leq 1$ constraint to be $||w||_2 = 1$ instead, i.e., $w$ is a unit vector. So the problem becomes finding the value of:

$$\max_{w : ||w||_2 = 1} \left\{ \begin{bmatrix} \tilde{V}^T d \\ \hat{v}^T d + (1-\tau)/2 \end{bmatrix}^T w \right\} + d^T u + d^T \bar{v} - (1+\tau)/2.$$ 

Hence we need to maximize a linear function over the unit ball, which can be solved analytically. This yields the objective value:

$$\left\| \begin{bmatrix} \tilde{V}^T d \\ d^T \hat{v} + (1-\tau)/2 \end{bmatrix} \right\|_2 + d^T u + d^T \bar{v} - (1+\tau)/2, \quad (62)$$

and the maximizer is:

$$\tilde{w} \leftarrow h \left( \begin{bmatrix} \tilde{V}^T d \\ d^T \hat{v} + (1-\tau)/2 \end{bmatrix} \right),$$

where $h(a) = a/||a||$ normalizes its input. Notice that the last element 1 comes since $\tilde{w} = 1$ is in the domain of convex conjugate. The worst-case of constraints (60) and (61) are obtained via similar calculations.

F Problems in Numerical Experiments

F.1 Experiments of Section 4.1

Problem 1 The problem data is:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad a = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \rho = 3.$$

For the next problems we generate bigger problems by uniform random sampling (denoted simply as $\sim$). Remember that $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $a \in \mathbb{R}^n$. We let $A_{ij}$ denote the elements of $A$.

Problem 2: $m = 15, n = 20, A_{ij} \sim \{0, 1\}, b_i \sim [-5, 5], a_j \sim [0, 3], \rho = 8$
Problem 3: \( m = 120, n = 100, A_{ij} \sim \{0, 1\}, b_i \sim [-5, 5], a_j \sim [0, 4], \rho = 14 \)

Problem 4: \( m = 40, n = 20, A_{ij} \sim \{-4, -3, \ldots, 3, 4\}, b_i \sim [-5, 5], a_j \sim [0, 4], \rho = 10 \)

Problem 5: \( m = 100, n = 50, A_{ij} \sim [-5, 5], b_i \sim [-2, 2], a_j \sim [-4, 4], \rho = 12 \)

Problem 6: \( m = 100, n = 100, A_{ij} \sim [-4, 4], b_i \sim [-3, 3], a_j \sim [-4, 4], \rho = 15 \)

Problem 7: \( m = 30, n = 200, A_{ij} \sim [-4, 2], b_i \sim [-1, 1], a_j \sim [-3, 3], \rho = 16 \)

Problem 8: \( m = 80, n = 400, A_{ij} \sim [-2, 1], b_i \sim [-\frac{1}{2}, \frac{1}{2}], a_j \sim [-1, 1], \rho = 12 \)

Problem 9: \( m = 20, n = 50, A_{ij} \sim [0, 8], b_i \sim [-1, 1], a_j \sim [0, 4], \rho = 14 \)

Problem 10: \( m = 100, n = 10,000, A_{ij} \sim [-\frac{1}{2}, \frac{1}{2}], b_i \sim [-\frac{1}{2}, \frac{1}{2}], a_j \sim [-\frac{7}{4}, \frac{7}{4}], \rho = 15 \)

Problem 11: \( m = 1,000, n = 1,000, A_{ij} \sim [-\frac{1}{2}, \frac{1}{2}], b_i \sim [-\frac{1}{4}, \frac{1}{4}], a_j \sim [-\frac{1}{4}, \frac{1}{4}], \rho = 18 \)

Problem 12: \( m = 700, n = 2,000, A_{ij} \sim [-\frac{1}{2}, \frac{1}{2}], b_i \sim [-\frac{1}{2}, \frac{1}{2}], a_j \sim [-\frac{1}{2}, \frac{1}{2}], \rho = 24 \)

F.2 Experiments of Section 4.2

Problem 1: \( A = \begin{bmatrix} -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, b = 0_{5 \times 1}, a = 0_{5 \times 1}, \rho = 3 \)

Problem 2: \( m = 5, n = 20, A_{ij} \sim [-1, 1], b_i \sim [-2, 2], a_j \sim [0, 1], \rho = 5 \)

Problem 3: \( m = 20, n = 50, A_{ij} \sim [-10, 10], b_i \sim [-3, 3], a_j \sim [-2, 2], \rho = 6 \)

Problem 4: \( m = 20, n = 180, A_{ij} \sim [-1, 0.5], b_i = 0, a_j \sim [0, 1], \rho = 1 \)

Problem 5: \( m = 300, n = 30, A_{ij} \sim [-1, 1], b_i = 0, a_j \sim [0, 1], \rho = 2 \)

F.3 Experiments of Section 4.3

Problem 1 (Enkhbat et al. 2006) In this example, we solve:

\[
\max_{x \in \mathbb{R}_{+}^{20}} \frac{1}{2} \sum_{i=1}^{20} (x_i - 2)^2
\]

s.t. \( Dx \leq d \)

where \( D^T = \)

\[
\begin{bmatrix}
-3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & 1 \\
7 & 0 & -5 & 1 & 1 & 0 & 2 & -1 & 1 & 1 \\
0 & -5 & 1 & 1 & 0 & 2 & -1 & -1 & 9 & 1 \\
-5 & 1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 1 \\
1 & 1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 1 \\
1 & 0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 1 \\
0 & 2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 \\
2 & -1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 1 \\
-1 & -1 & -9 & 3 & 5 & 0 & 0 & 1 & 7 & 1 \\
-1 & 9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & 1 \\
-9 & 3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & 1 \\
3 & 5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & 1 \\
5 & 0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 1 \\
0 & 0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 1 \\
0 & 1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & 1 \\
1 & 7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 \\
7 & -7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 \\
-7 & -4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 1 \\
-4 & -6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 1 \\
-6 & -3 & 7 & 0 & -5 & 1 & 1 & 0 & 2 & 1 \\
\end{bmatrix}, \quad d = \begin{bmatrix} -5 \\
2 \\
5 \\
4 \\
1 \\
0 \\
9 \\
0 \\
40 \end{bmatrix}
\]
Problem 2 Same as problem 1, but the objective function is \( \frac{1}{4} \sum_{i=1}^{20} (x_i + 5)^2 \).

Problems 3-7 use the following problem:
\[
\max_{x \in \mathbb{R}_+^n} \quad x^\top L^\top Lx \\
\text{s.t.} \quad Dx \leq d \\
\quad x \leq x_u,
\]
where \( L \in \mathbb{R}^{m \times n} \) is a matrix generated randomly where all the entries are sampled uniformly from \([0, 1)\). Moreover, let \( D \) be a similar matrix with all \([0, 1)\) random coefficients (except the last problem uses \([0, 2)\) random coefficients) and \( d \) to have integer entries uniformly distributed in range \([d_l, d_u]\). Denote \( q \) to be the number of total constraints.

**Problem 3** \( x_u = 5, d_l = 20, d_u = 60, n = 10, q = 15 \)

**Problem 4** \( x_u = 3, d_l = 30, d_u = 60, n = 50, q = 62 \)

**Problem 5** \( x_u = 2, d_l = 80, d_u = 120, n = 100, q = 130 \)

**Problem 6** \( x_u = 2, d_l = 160, d_u = 240, n = 200, q = 240 \)

**Problem 7** \( x_u = 1, d_l = 150, d_u = 300, n = 240, q = 280 \).

### F.4 Experiments of Section 4.4

**Problem 1** considers the following problem:
\[
\max_{x \in \mathbb{R}_+^n} \quad \log \left( \sum_{i=1}^{m} \exp(A_{ij} x) \right) \\
\text{s.t.} \quad \frac{i}{n} \leq x_i \leq \frac{n}{i},
\]
where \( A_{ij} \sim [-3, 3] \). In the numerical experiments \( n \) will vary.

Problems 2-4 consider the following problem:
\[
\max_{x \in \mathbb{R}_+^n} \quad \log \left( \sum_{i=1}^{m} \exp(A_{ij} x + b_i) \right) \\
\text{s.t.} \quad Dx \leq d.
\]

**Problem 2** \( n = q = m = 10, A_{ij} \sim [-3, 3], b_i \sim [-2, 2], D_{i,j} \sim [0, 1], d_i \sim [10, 25] \)

**Problem 3** \( n = q = m = 50, A_{ij} \sim [-3, 3], b_i \sim [-1, 1], D_{i,j} \sim [0, 1], d_i \sim [20, 50] \)

**Problem 4** \( n = q = m = 100, A_{ij} \sim [-3, 3], b_i \sim [-1, 1], D_{i,j} \sim [0, 1], d_i \sim [25, 62.5] \)

Problems 5-6 consider:
\[
\max_{x \in \mathbb{R}_+^n} \quad \log \left( \sum_{i=1}^{m} \exp(A_{ij} x + b_i) \right) \\
\text{s.t.} \quad x_i \leq c, \quad i = 1, \ldots, n \\
\quad x_i + x_j \leq u_{ij}, \quad i, j = 1, \ldots, n, \ i \neq j.
\]

**Problem 5** \( n = m = 10, A_{ij} \sim [-3, 3], b_i \sim [-1, 1], u_{ij} \sim [5, 12.5], c = 8 \)

**Problem 6** \( n = m = 30, A_{ij} \sim [-3, 3], b_i \sim [-1, 1], u_{ij} \sim [4, 10], c = 6 \).

### F.5 Experiments of Section 4.5

For the easiness of bookkeeping, we generate problem with every max-term having the same number of elements, i.e., \( |\mathcal{I}_k| = |\mathcal{I}_{k'}| \forall k, k' \in \{1, \ldots, K\} \).
Problems 1-6 are defined by:

\[
\max_{x \in \mathbb{R}_+^n} \sum_{k=1}^K \max_{j \in I_k} \{A(j)x\}
\]
\[
s.t. \quad \frac{i}{n} \leq x_i \leq \frac{n}{i},
\]

where:

Problem 1 \(n = 5, \ A_{ij} \sim [-5, 5], \ |I_k| = 5, \ K = 1\)
Problem 2 \(n = 5, \ A_{ij} \sim [-5, 5], \ |I_k| = 5, \ K = 10\)
Problem 3 \(n = 20, \ A_{ij} \sim [-5, 5], \ |I_k| = 10, \ K = 10\)
Problem 4 \(n = 30, \ A_{ij} \sim [-5, 5], \ |I_k| = 20, \ K = 20\)
Problem 5 \(n = 100, \ A_{ij} \sim [-5, 5], \ |I_k| = 40, \ K = 30\)
Problem 6 \(n = 200, \ A_{ij} \sim [-4, 4], \ |I_k| = 50, \ K = 50\).

Problems 7-10 are defined by:

\[
\max_{x \in \mathbb{R}_+^n} \sum_{k=1}^K \max_{j \in I_k} \{A(j)x + b_j\}
\]
\[
s.t. \quad Dx \leq d,
\]

Problem 7 \(n = 10, \ A_{ij} \sim [-5, 5], \ b_j \sim [-10, 10], \ |I_k| = 5, \ K = 2\)
Problem 8 \(n = 10, \ A_{ij} \sim [-5, 5], \ b_j \sim [-10, 10], \ D_{ij} \sim [0, 1], \ d_i \sim [5, 15], \ |I_k| = 50, \ K = 50\)
Problem 9 \(n = 30, \ A_{ij} \sim [-5, 5], \ b_j \sim [-10, 10], \ D_{ij} \sim [0, 1], \ d_i \sim [5, 15], \ |I_k| = 50, \ K = 50\)
Problem 10 \(n = 50, \ A_{ij} \sim [-5, 5], \ b_j \sim [-10, 10], \ D_{ij} \sim [0, 1], \ d_i \sim [5, 15], \ |I_k| = 60, \ K = 60\).

Problems 10-13 consider the same problem as above, but \(D\) and \(d\) are as given in (63) with \(n = 20\).

The objective function varies as:

Problem 11 \(A_{ij} \sim [-5, 10], \ b_j \sim [-10, 10], \ |I_k| = 10, \ K = 10\)
Problem 12 \(A_{ij} \sim [-5, 10], \ b_j \sim [-10, 10], \ |I_k| = 50, \ K = 10\)
Problem 13 \(A_{ij} \sim [-5, 10], \ b_j \sim [-10, 10], \ |I_k| = 100, \ K = 10\).