SUCCESSIVE QUADRATIC UPPER-BOUNDING FOR DISCRETE MEAN-RISK MINIMIZATION AND NETWORK INTERDICTION

ALPER ATAMTÜRK, CARLOS DECK AND HYEMIN JEON

ABSTRACT. The advances in conic optimization have led to its increased utilization for modeling data uncertainty. In particular, conic mean-risk optimization gained prominence in probabilistic and robust optimization. Whereas the corresponding conic models are solved efficiently over convex sets, their discrete counterparts are intractable. In this paper, we give a highly effective successive quadratic upper-bounding procedure for discrete mean-risk minimization problems. The procedure is based on a reformulation of the mean-risk problem through the perspective of its convex quadratic term. Computational experiments conducted on the network interdiction problem with stochastic capacities show that the proposed approach yields solutions within 1-2% of optimality in a small fraction of the time required by exact search algorithms. We demonstrate the value of the proposed approach for constructing efficient frontiers of flow-at-risk vs. interdiction cost for varying confidence levels.

Keywords: Risk, polymatroids, conic integer optimization, quadratic optimization, stochastic network interdiction.

July 2017; June 2018
1. Introduction

Conic optimization problems arise frequently when modeling parametric value-at-risk (VaR) minimization, portfolio optimization, and robust optimization with ellipsoidal objective uncertainty. Although convex versions of these models are solved efficiently by polynomial interior-point algorithms, their discrete counterparts are intractable. Branch-and-bound and branch-and-cut algorithms require excessive computation time even for relatively small instances. The computational difficulty is exacerbated by the lack of effective warm-start procedures for conic optimization.

In this paper, we consider a reformulation of a conic quadratic discrete mean-risk minimization problem that lends itself to a successive quadratic optimization procedure benefiting from fast warm-starts and eliminating the need to solve conic optimization problems directly.

Let \( u \) be an \( n \)-dimensional random vector and \( x \) be an \( n \)-dimensional decision vector in a closed set \( X \subseteq \mathbb{R}^n \). If \( u \) is normally distributed with mean \( c \) and covariance \( Q \), the minimum value-at-risk for \( u^\prime x \) at confidence level \( 1 - \epsilon \), i.e.,

\[
\zeta(\epsilon) = \min \left\{ z : \text{Prob}(u^\prime x > z) \leq \epsilon, \ x \in X \right\},
\]

for \( 0 < \epsilon \leq 0.5 \), is computed by solving the mean-risk optimization problem

\[
(\text{MR}) \quad \min \left\{ c^\prime x + \Omega \sqrt{x^\prime Q x} : \ x \in X \right\},
\]

where \( \Omega = \Phi^{-1}(1 - \epsilon) \) and \( \Phi \) is the c.d.f. of the standard normal distribution \([18]\). If \( u \) is not normally distributed, but its mean and variance are known, (MR) yields a robust version by letting \( \Omega = \sqrt{(1 - \epsilon)/\epsilon} \), which provides an upper bound on the worst-case VaR \([16, 21]\). Alternatively, if \( u_i \)'s are independent and symmetric with support \([c_i - \sigma_i, c_i + \sigma_i]\), then letting \( \Omega = \sqrt{\ln(1/\epsilon)} \) with \( Q_{ii} = \sigma_i^2 \) gives an upper bound on the worst-case VaR as well \([13]\). The reader is referred to Ben-Tal et al. \([15]\) for an in-depth treatment of robust models through conic optimization.

Hence, under various assumptions on the uncertainty of \( u \), one arrives at different instances of the mean-risk model (MR) with a conic quadratic objective. Ahmed \([1]\) studies the complexity and tractability of various stochastic objectives for mean-risk optimization. Maximization of the mean-risk objective is \( \mathcal{NP} \)-hard even for a diagonal covariance matrix \([2, 5]\). If \( X \) is a polyhedron, (MR) is a special case of conic quadratic optimization \([4, 28]\), which can be solved by polynomial-time interior points algorithm \([3, 29, 14]\). Atamtürk and Gómez \([7]\) give simplex QP-based algorithms for this case.

The interest of the current paper is in the discrete case of (MR) with integrality restrictions: \( X \subseteq \mathbb{Z}^n \), which is \( \mathcal{NP} \)-hard. Atamtürk and Narayanan \([10]\) describe mixed-integer rounding cuts, and Çezik and Iyengar \([19]\) give disjunctive cuts for conic mixed-integer programming. The integral case is more predominantly addressed in the special case of independent random variables over binaries. In the absence of correlations, the covariance matrix reduces to a diagonal matrix \( Q = \text{diag}(q) \), where \( q \) is the vector of variances. In addition, when the decision
variables are binary, (MR) reduces to

\[
(DMR) \quad \min \left\{ c'x + \Omega \sqrt{q'x} : x \in X \subseteq \mathbb{B}^n \right\}.
\]

Several approaches are available for (DMR) for specific constraint sets \( X \). Ishii et al. [25] give an \( O(n^6) \) algorithm when the feasible set \( X \) is the set of spanning trees; Hassin and Tamir [22] utilize parametric linear programming to solve (DMR) when \( X \) defines a matroid in polynomial time. Atamtürk and Narayanan [11] give a cutting plane algorithm utilizing the submodularity of the objective; Atamtürk and Jeon [9] extend it to the mixed 0-1 case with indicator variables. Atamtürk and Narayanan [12] give an \( O(n^3) \) algorithm over a cardinality constraint. Shen et al. [34] provide a greedy \( O(n \log n) \) algorithm to solve the diagonal case over the unit hypercube. Nikolova [30] gives a fully polynomial-time approximation scheme (FPTAS) for an arbitrary set \( X \subseteq \mathbb{B}^n \) provided the deterministic problem with \( \Omega = 0 \) can be solved in polynomial time.

The reformulation we give in Section 2 reduces the general discrete mean-risk problem (MR) to a sequence of discrete quadratic optimization problems, which is often more tractable than the conic quadratic case [8]. The uncorrelated case (DMR) reduces to a sequence of binary linear optimization problems. Therefore, one can utilize the simplex-based algorithms with fast warm-starts for general constraint sets. Moreover, the implementations can benefit significantly for structured constraint sets, such as spanning trees, matroids, graph cuts, shortest paths, for which efficient algorithms are known for non-negative linear objectives.

A motivating application: Network interdiction with stochastic capacities. Our motivating problem for the paper is network interdiction with stochastic capacities; although, since the proposed approach is independent of the feasible set \( X \), it can be applied to any problem with a mean-risk objective (MR).

The deterministic interdiction problem is a generalization of the classical min-cut problem, where an interdictor with a limited budget minimizes the maximum flow on a network by stopping the flow on a subset of the arcs at a cost per interdicted arc. Consider a graph \( G = (N, A) \) with nodes \( N \) and arcs \( A \). Let \( s \) be the source node and \( t \) be the sink node. Let \( \alpha_a \) be the cost of interdicting arc \( a \in A \) and \( \beta \) be the total budget for interdiction. Then, given a set \( I \) of interdicted arcs, the maximum \( s-t \) flow on the remaining arcs is the capacity of the minimum cut on the arcs \( A \setminus I \). Wood [36] shows that the deterministic interdiction problem is \( \text{NP} \)-hard and gives integer formulations for it. Royset and Wood [32] give algorithms for a bi-criteria interdiction problem and generate an efficient frontier of maximum flow vs. interdiction cost. Cormican et al. [20], Janjarassuk and Linderoth [26] consider a stochastic version of the problem, where interdiction success is probabilistic. Held et al. [23] develop a decomposition approach for interdiction when network topology is stochastic. Network interdiction is a dual counterpart of survivable network design [17, 31], where one installs capacity to maximize the minimum flow against an adversary blocking the arcs. See Smith et al. [35] for a review of network interdiction models and algorithms.
When the arc capacities are stochastic, we are interested in an optimal interdiction plan that minimizes the maximum flow-at-risk. Unlike the expectation criterion used in previous stochastic interdiction models, this approach provides a confidence level for the maximum flow on the arcs that are not interdicted. Letting $c$ be the mean capacity vector and $Q$ the covariance matrix, the mean-risk network interdiction problem is modeled as

$$
\text{min } c^t x + \Omega \sqrt{x^t Q x} \\
\text{s.t. } By \leq x + z, \\
(\text{MRNI}) \quad \alpha^t z \leq \beta, \\
y_s = 1, \ y_t = 0, \\
x \in \{0, 1\}^A, y \in \{0, 1\}^N, z \in \{0, 1\}^A,$$

where $B$ is the node-arc incidence matrix of $G$. Here, $z_a$ is one if arc $a$ is interdicted at a cost of $\alpha_a$ and zero otherwise; and $x_a$ is one if arc $a$ is in the minimum mean-risk cut and zero otherwise. The optimal value of (MRNI) is the “flow-at-risk” for a given interdiction budget $\beta$. Note that when $\Omega = 0$, (MRNI) reduces to the deterministic network interdiction model of Wood [36]; and, in addition, if $z$ is a vector of zeros, it reduces to the standard $s-t$ min-cut problem. In a recent paper Lei et al. [27] give a scenario-based approach stochastic network interdiction under conditional value-at-risk measure. The following example underlines the difference of interdiction solutions between deterministic and mean-risk models with stochastic capacities.

**Example 1.** Consider the simple network in Figure 1 with two arcs from $s$ to $t$. Arc 1 has mean capacity 1 and 0 variance, whereas arc 2 has mean capacity 0.9 and variance $\sigma^2$. Suppose the budget allows a single-arc interdiction. Then, the deterministic model with $\Omega = 0$ would interdict arc 1 with higher mean and leave arc 2 with high variance intact. Consequently, the maximum $s-t$ flow would exceed $0.9 + 0.5\sigma$ with probability 0.3085 according to the normal distribution. On the other hand, the mean-risk model with $\Omega > 0.2$, interdicts arc 2 with lower mean, but high variance ensuring that the maximum $s-t$ flow to be no more than 1.

![Figure 1. Mean-risk network interdiction.](image)

The combinatorial aspect of network interdiction, coupled with correlations, make it extremely challenging to determine the least cost subset of arcs to interdict for a desired confidence level in the maximum flow even for moderate sized networks. Yet,
understanding the cost and benefit of an interdiction strategy is of critical interest for planning purposes. Toward this end, the proposed approach in the current paper allows one to quickly build efficient frontiers of flow-at-risk vs. interdiction cost, which would, otherwise, be impractical for realistic sizes. Figure 2 shows the flow-at-risk as a function of the interdiction cost for different confidence levels for a $15 \times 15$ grid graph shown in Figure 4. At 100% budget the network is interdicted completely allowing no flow. At lower budget levels the flow-at-risk increases significantly with the higher confidence levels. The vertical axis is scaled so that the deterministic min-cut value with $\Omega = 0$ at 0% budget (no interdiction) is 100. The green solid curve corresponding to 95% confidence level shows that, if no interdiction is performed, the flow on the network is more than 200% of the deterministic maximum flow with probability 0.05. The same curve shows with 40% interdiction budget the flow is higher than the deterministic maximum flow (100) with probability only 0.05.

Contributions and outline. In Section 2 we give a non-convex upper-bounding function for (MR) that matches the mean-risk objective value at its local minima. Then, we describe an upper-bounding procedure that successively solves quadratic optimization problems instead of conic quadratic optimization. The rationale behind the approach is that algorithms for quadratic optimization with linear constraints scale better than interior point algorithms for conic quadratic optimization. Moreover, simplex algorithms for quadratic optimization can be effectively integrated into branch-and-bound algorithms and other iterative procedures as they allow fast warm-starts. In Section 3 we test the effectiveness of the proposed approach on the network interdiction problem with stochastic capacities and compare it with exact algorithms. We conclude in Section 4 with a few final remarks.
2. A successive quadratic optimization approach

In this section, we present a successive quadratic optimization procedure to obtain feasible solutions to (MR). The procedure is based on a reformulation of (MR) using the perspective function of the convex quadratic term \( q(x) = x'Qx \). Atamtürk and Gómez [7] introduce

\[
(PO) \quad \min \left\{ c'x + \frac{\Omega}{2} h(x, t) + \frac{\Omega}{2} t : x \in X, \ t \geq 0 \right\},
\]

where \( \Omega \) is a positive scalar as before, \( h : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R} \cup \{\infty\} \) is the closure of the perspective function of \( q \) and is defined as

\[
h(x, t) := \begin{cases} 
\frac{x'Qx}{t} & \text{if } t > 0, \\
0 & \text{if } t = 0, \ x'Qx = 0, \\
+\infty & \text{otherwise}.
\end{cases}
\]

As the perspective of a convex function is convex [24], \( h \) is convex. Atamtürk and Gómez [7] show the equivalence of (MR) and (PO) for a polyhedral set \( X \). Since we are mainly interested in a discrete feasible region, we study (PO) for \( X \subseteq \mathbb{Z}^n \).

For \( t \geq 0 \), it is convenient to define the optimal value function

\[
f(t) := \min_{x \in X} \left\{ g(x, t) := c'x + \frac{\Omega}{2} h(x, t) + \frac{\Omega}{2} t \right\}.
\]

Given \( t \), optimization problem (1) has a convex quadratic objective function. Let \( x(t) \) be a minimizer of (1) with value \( f(t) \). Note that \( g \) is convex and \( f \) is a pointwise minimum of convex functions in \( t \) for each choice of \( x \in X \), and is, therefore, typically non-convex (see Figure 3). We show below that, for any \( t \geq 0 \), \( f(t) \) provides an upper bound on the mean-risk objective value for \( x(t) \).

**Lemma 1.** \( \sqrt{a} \leq \frac{1}{2}(a/t + t) \) for all \( a, t \geq 0 \).

**Proof.** Since \( \sqrt{a} \) is concave over \( a \geq 0 \), it is bounded above by its gradient line:

\[
\sqrt{a} \leq \sqrt{y} + \frac{1}{2\sqrt{y}}(a - y)
\]

at any point \( y \geq 0 \). Letting \( t = \sqrt{y} \) gives the result. \( \square \)

**Proposition 1.** For any \( t \geq 0 \), we have

\[
c'x(t) + \sqrt{x(t)'Qx(t)} \leq f(t).
\]

**Proof.** Applying Lemma 1 with \( a = x'Qx \) (\( \geq 0 \) as \( Q \) is positive semidefinite) gives

\[
\sqrt{x'Qx} \leq \frac{1}{2} h(x, t) + \frac{t}{2}, \ \forall x \in \mathbb{R}^n, \ \forall t \geq 0.
\]

First multiplying both sides by \( \Omega \geq 0 \) and then adding \( c'x \) shows

\[
c'x + \Omega \sqrt{x'Qx} \leq c'x + \frac{\Omega}{2} h(x, t) + \frac{\Omega}{2} t, \ \forall x \in \mathbb{R}^n, \ \forall t \geq 0.
\]

The inequality holds, in particular, for \( x(t) \) as well. \( \square \)
Example 2. Consider the mean-risk optimization problem

$$\min \left\{ x_2 + \sqrt{10}x_1^2 + 5x_2^2 : x \in X = \{(0, 1), (1, 0)\} \subseteq \mathbb{R}^2 \right\}$$

with two feasible points. Figure 3 illustrates the optimal value function $f$. The curves in red and green show $g((1, 0), t)$ and $g((0, 1), t)$, respectively, and $f(t) = \min\{g((1, 0), t), g((0, 1), t)\}$ is shown with a dotted line. As the red and green curves intersect at $t = 2.5$, $x(t)$ is $(0, 1)$ for $t \leq 2.5$, and $(1, 0)$ for $t \geq 2.5$.

In this example, $f$ has two local minima: $1 + \sqrt{5}$ attained at $t = \sqrt{5}$ and $\sqrt{10}$ at $t = \sqrt{10}$. Observe that the upper bound $f(t)$ matches the mean-risk objective at these local minima:

$$c'x(t) + \Omega \sqrt{x(t)'Qx(t)} = f(t), \quad t \in \{\sqrt{5}, \sqrt{10}\}.$$

The black step function shows the mean-risk values for the two feasible solutions of $X$. It turns out the upper bound $f(t)$ is tight, in general, for any local minima (Proposition 2).

In order to contrast the convex and discrete cases, we show with solid blue curve the lower bound $\hat{f}$ of $f$, where $f(t) = \min\{g(x, t) : x \in \hat{X}\}$ and $\hat{X} := \{(x_1, x_2) \in \mathbb{R}_+^2 : x_1 + x_2 = 1\}$ is the convex relaxation of $X$. Let $\hat{x}(t)$ be the solution of this convex problem. Then $\hat{f}(t)$ provides an upper bound on $c'\hat{x}(t) + \Omega \sqrt{\hat{x}(t)'Q\hat{x}(t)}$ (graph shown in dotted blue curve) at any $t \geq 0$, and the bound is tight at $t = \sqrt{25/7}$, where the minimum of $\hat{f}(t)$ is attained.

![Figure 3](image-url)  

**Figure 3.** The value function $f$ with two discrete feasible points.

Although, in general, $f(t)$ provides an upper bound, the next proposition shows that the mean-risk objective and $f$ match at local minima of $f$. 


Proposition 2. If \( f \) has a local minimum at \( \bar{t} > 0 \), then we have

\[
c'x(\bar{t}) + \Omega \sqrt{x'Qx(\bar{t})} = f(\bar{t}). \tag{2}
\]

Proof. Since \( f \) is the point-wise minimum of differentiable convex functions, it is differentiable at its local minima in the interior of its domain \((t > 0)\). Then, its vanishing derivative at \( \bar{t} \)

\[
f'(\bar{t}) = -\frac{x'(\bar{t})'Qx(\bar{t})}{\bar{t}^2} + 1 = 0
\]

implies \( \bar{t} = \sqrt{x'(\bar{t})'Qx(\bar{t})} \). Plugging this expression into \( f(\bar{t}) \) gives the result. \( \square \)

Finally, we show that problems (MR) and (PO) are equivalent. In other words, the best upper bound matches the optimal value of the mean-risk problem, which provides an alternative way for solving (MR).

Proposition 3. Problems (MR) and (PO) are equivalent; that is,

\[
\min \left\{ c'x + \Omega \sqrt{x'Qx} : x \in X \right\} = \min \{ f(t) : t \geq 0 \}.
\]

Proof. Let \( t^* \) be optimal for \( \min \{ f(t) : t \geq 0 \} \). By Proposition 1

\[
f(t^*) \geq c'x(t^*) + \Omega \sqrt{x(t^*)'Qx(t^*)} \geq \min \left\{ c'x + \Omega \sqrt{x'Qx} : x \in X \right\}.
\]

The other direction follows from the observation

\[
\min_{x \in X} \left\{ c'x + \Omega \sqrt{x'Qx} \right\} = \min_{x \in X, t \geq 0} \left\{ c'x + \frac{\Omega}{2} h(x, t) + \frac{\Omega}{2} t : t = \sqrt{x'Qx} \right\} \\
\geq \min_{x \in X, t \geq 0} \left\{ c'x + \frac{\Omega}{2} h(x, t) + \frac{\Omega}{2} t \right\} = \min_{t \geq 0} \{ f(t) \}.
\]

\( \square \)

The one-dimensional upper-bounding function \( f \) above suggests a local search algorithm that utilizes quadratic optimization to evaluate the function at any \( t \geq 0 \):

\[
f(t) = \min_{x \in X} \left\{ g(x, t) := c'x + \frac{\Omega}{2t} x'Qx + \frac{\Omega}{2} t \right\}
\]

and avoids the solution of a conic quadratic optimization problem directly.

Algorithm describes a simple binary search method that halves the uncertainty interval \([t_{\text{min}}, t_{\text{max}}]\), initiated as \( t_{\text{min}} = 0 \) and \( t_{\text{max}} = \sqrt{x'Qx} \), where \( \bar{x} \) is an optimal solution to (MR) with \( \Omega = 0 \). The algorithm is terminated either when a local minimum of \( f \) is reached or the gap between the upper bound \( f(t) \) and \( c'x(t) + \Omega \sqrt{x(t)'Qx(t)} \) is small enough. For the computations in Section 3 we use 1% gap as the stopping condition.
Algorithm 1 Binary local search.

**Input:** \( X \subseteq \mathbb{Z}^n; Q \text{ p.s.d. matrix}; c \text{ cost vector}; \Omega > 0 \)

**Output:** Local optimal solution \( x \)

1. Initialize \( t_{\text{min}} \) and \( t_{\text{max}} \)
2. \( \hat{z} \leftarrow \infty \) \( \triangleright \) best objective value found
3. **repeat**
   4. \( t \leftarrow \frac{t_{\text{min}} + t_{\text{max}}}{2} \)
   5. \( x(t) \leftarrow \arg \min \left\{ c'x + \frac{\Omega}{2} x'Qx + \frac{\Omega}{2} t : x \in X \right\} \)
   6. if \( \frac{\partial g}{\partial t}(x(t), t) \leq -\epsilon \) then
      7. \( t_{\text{min}} \leftarrow t \)
   8. else if \( \frac{\partial g}{\partial t}(x(t), t) \geq \epsilon \) then
      9. \( t_{\text{max}} \leftarrow t \)
   10. else
      11. return \( x(t) \)
   12. end if
4. **until** stopping condition is met
14. return \( \hat{x} \)

The uncorrelated case over binaries. The reformulation (PO) simplifies significantly for the special case of independent random variables over binaries. In the absence of correlations, the covariance matrix reduces to a diagonal matrix \( Q = \text{diag}(q) \), where \( q \) is the vector of variances. For

\[
(DMR) \quad \min \left\{ c'x + \Omega \sqrt{q'}x : x \in X \subseteq \mathbb{B}^n \right\}
\]

the upper bounding problem simplifies to

\[
f(t) = \min \left\{ c'x + \frac{\Omega}{t} q'x + \frac{\Omega}{2} t : x \in X \subseteq \mathbb{B}^n \right\}, \tag{3}
\]

which is a binary linear optimization problem for fixed \( t \). Thus, \( f \) can be evaluated fast for linear combinatorial optimization problems, such as the minimum spanning tree problem, shortest path problem, assignment problem, minimum cut problem [33], for which there exist polynomial-time algorithms. Even when the evaluation problem (3) is \( \mathcal{NP} \)-hard, simplex-based branch-and-bound algorithms equipped with warm-starts perform much faster than conic quadratic mean-risk minimization as demonstrated in the next section.

3. Computational Experiments

In this section we report on computational experiments conducted to test the effectiveness of the proposed successive quadratic optimization approach on the network interdiction problem with stochastic capacities. We compare the solution quality and the computation time with exact algorithms.

All experiments are carried out using CPLEX 12.6.2 solver on a workstation with a 3.60 GHz Intel R Xeon R CPU E5-1650 and 32 GB main memory and with a single
thread. Default CPLEX settings are used with few exceptions: dynamic search and presolver are disabled to utilize the user cut callback; the branch-and-bound nodes are solved using linear outer approximation for faster enumeration; and the time limit is set to one hour.

**Problem instances.** We generate instances of the mean-risk network interdiction problem (MRNI) on grid graphs similar to the ones used in Cormican et al. [20], Janjarassuk and Linderoth [26]. Let $p \times q$ grid be the graph with $p$ columns and $q$ rows of grid nodes in addition to a source and a sink node (see Figure 4). The source and sink nodes are connected to all the nodes in the first and last column, respectively. The arcs incident to source or sink have infinite capacity and are not interdictable. The arcs between adjacent columns are always directed toward the sink, and the arcs connecting two nodes within the same column are directed either upward or downward with equal probability.

![Figure 4. p \times q grid graph.](image)

We generate two types of data: uncorrelated and correlated. For each arc $a \in A$, the mean capacity $c_a$ and its standard deviation $\sigma_a$ are independently drawn from the integral uniform [1, 10], and the interdiction cost $\alpha_a$ is drawn from the integral uniform [1, 3]. For the correlated case, the covariance matrix is constructed via a factor model: 

$$Q = \text{diag}(\sigma_1^2, \ldots, \sigma_{|A|}^2) + EFE',$$

where $F$ is an $m \times m$ factor covariance matrix and $E$ is the exposure matrix of the arcs to the factors. $F$ is computed as $F = HH'$, where each $H_{ij}$ is drawn from uniform $[-100/pq, 100/pq]$, and each $E_{ij}$ from uniform $[0, 0.1]$ with probability 0.2 and set to 0 with probability 0.8. The interdiction budget $\beta$ is set to $\lfloor \frac{Y}{2} \rfloor$, and the risk averseness parameter $\Omega$ is set to $\Phi^{-1}(1 - \epsilon)$, where $\Phi$ is the c.d.f. of the standard normal distribution. Five instances are generated for each combination of graph sizes $p \times q : 10 \times 10, 20 \times 20, 30 \times 30$ and confidence levels $1 - \epsilon$: 0.9, 0.95, 0.975. The data set is available for download at [http://atamturk.ieor.berkeley.edu/data/prob.interdiction](http://atamturk.ieor.berkeley.edu/data/prob.interdiction).
For completeness, we state the corresponding perspective optimization for (MRNI):

\[
\begin{align*}
\min & \quad c'x + \Omega x'Qx/2t + \Omega t/2 \\
\text{s.t.} & \quad By \leq x + z, \\
(PO - MRNI) & \quad \alpha'z \leq \beta, \\
& \quad y_s = 1, \ y_t = 0, \\
& \quad x \in \{0,1\}^A, \ y \in \{0,1\}^N, \ z \in \{0,1\}^A, \ t \in \mathbb{R}_+.
\end{align*}
\]

**Computations.** Table 1 summarizes the performance of the successive quadratic optimization approach on the network interdiction instances. We present the number of iterations, the computation time in seconds, and the percentage optimality gap for the solutions, separately for the uncorrelated and correlated instances. Each row represents the average over five instances for varying grid sizes and confidence levels. One sees in the table that only a few number of iterations are required to obtain solutions within about 1% of optimality for both the correlated and uncorrelated instances. While the solution times for the correlated case are higher, even the largest instances are solved under 20 seconds on average. The computation time increases with the size of the grids, but is not affected by the confidence level $1 - \epsilon$.

**Table 1.** Performance of the binary local search.

<table>
<thead>
<tr>
<th>$p \times q$</th>
<th>$1 - \epsilon$</th>
<th>Uncorrelated</th>
<th>Correlated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iter</td>
<td>time</td>
<td>gap</td>
</tr>
<tr>
<td>10 x 10</td>
<td>0.9</td>
<td>2.8</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>3.4</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>2.8</td>
<td>0.05</td>
</tr>
<tr>
<td>20 x 20</td>
<td>0.9</td>
<td>3.0</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>2.8</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>2.8</td>
<td>0.45</td>
</tr>
<tr>
<td>30 x 30</td>
<td>0.9</td>
<td>3.0</td>
<td>2.24</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>3.0</td>
<td>2.54</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>3.0</td>
<td>2.58</td>
</tr>
<tr>
<td>avg</td>
<td>2.96</td>
<td>0.98</td>
<td>1.06</td>
</tr>
</tbody>
</table>

The optimal/best known objective values used for computing the optimality gaps in Table 1 are obtained with the CPLEX branch-and-bound algorithm. To provide a comparison with the successive quadratic optimization procedure, we summarize the performance for the exact algorithm in Table 2 for the uncorrelated and correlated instances, respectively. In each column, we report the percentage integrality gap at the root node (rgap), the time spent until the best feasible solution is obtained (stime), the total solution time in CPU seconds (time), the percentage gap between the best upper bound and the lower bound at termination (egap), and the number of nodes explored (nodes). If the time limit is reached before proving optimality, the number of instances unsolved (#) is shown next to egap. Each row of the tables represents the average for five instances.
Table 2. Performance of b&b and b&c algorithms.

<table>
<thead>
<tr>
<th>$p \times q$</th>
<th>$1 - \epsilon$</th>
<th>Uncorrelated instances</th>
<th>Correlated instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cplex</td>
<td>CPLEX + cuts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - gap</td>
<td>stime</td>
</tr>
<tr>
<td>$10 \times 10$</td>
<td>0.9</td>
<td>15.1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>17.1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>17.9</td>
<td>1</td>
</tr>
<tr>
<td>$20 \times 20$</td>
<td>0.9</td>
<td>17.9</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>21.0</td>
<td>469</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>21.6</td>
<td>404</td>
</tr>
<tr>
<td>$30 \times 30$</td>
<td>0.9</td>
<td>19.1</td>
<td>2,338</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>23.1</td>
<td>3,535</td>
</tr>
<tr>
<td></td>
<td>0.975</td>
<td>23.1</td>
<td>3,535</td>
</tr>
<tr>
<td>avg</td>
<td></td>
<td>19.2</td>
<td>1,125</td>
</tr>
</tbody>
</table>

Observe that the solution times with the CPLEX branch-and-bound algorithm are much larger compared to the successive quadratic optimization approach: 1,408 secs. vs. 1 sec. for the uncorrelated instances and 666 secs. vs. 7 secs. for the correlated instances. The difference in the performance is especially striking for the $30 \times 30$ instances, of which half are not solved to optimality within the time limit. Many of these unsolved instances are terminated with large optimality gaps (egap).

In order to strengthen the convex relaxation of 0-1 problems with a mean-risk objective, one can utilize the polymatroid inequalities [6]. Polymatroid inequalities exploit the submodularity of the mean-risk objective for the diagonal case. They are extended for the (non-digonal) correlated case as well as for mixed 0-1 problems in [11]. To improve the performance of the exact algorithm, we also test it by adding the polymatroid cuts. It is clear in Table 2 that the polymatroid cuts have a very positive impact on the exact algorithm. The root gaps are reduced significantly with the addition of the polymatroid cuts. Whereas 16 of the instances are unsolved within the time limit with default CPLEX, all but two instances are solved to optimality when adding the cuts. Nevertheless, the solution times even with the cutting planes are much larger compared to the successive quadratic optimization.
approach: 543 secs. vs. 1 sec. for the uncorrelated case and 179 secs. vs. 7 secs. for the correlated case.

Branch-and-bound and branch-and-cut algorithms spend a significant amount of solution time to prove optimality rather than finding feasible solutions. Therefore, for a fairer comparison, it is also of interest to check the time to the best feasible solution, which are reported under the column stime in Table 2. The average time to the best solution is 1,125 and 386 seconds for the branch-and-bound algorithm and 404 and 133 seconds for the branch-and-cut algorithm for the uncorrelated and correlated cases, respectively. Figure 5 presents the progress of the incumbent solution over time for one of the $30 \times 30$ instances. The vertical axis shows the distance to the optimal value (100%). The binary search algorithm finds a solution within 3% of the optimal under 3 seconds. It takes 1,654 seconds for the default branch-and-bound algorithm and 338 seconds for the branch-and-cut algorithm to find a solution at least as good.

Figure 5. Performance profile of the algorithms.

The next set of experiments are done to test the impact of the budget constraint on the performance of the algorithms. For these experiments, the instances with $1-\epsilon = 0.95$ and grid size $20 \times 20$ are solved with varying levels of budgets. Specifically, the budget parameter $\beta$ is set to $\bar{\alpha} \gamma$ for $\gamma \in \{2, 4, 6, 8, 10, 20\}$, where $\bar{\alpha}$ denotes the mean value of $\alpha$. As before, each row of the Tables 3–4 presents the averages for five instances. Observe that the binary search algorithm is not as sensitive to the budget as the exact algorithms. For the exact algorithms, while the root gap decreases with larger budget values, the solution time tends to increase, especially for the uncorrelated instances.

Next, we present the experiments performed to test the effect of the interdiction cost parameter $\alpha$. New instances with $1-\epsilon = 0.95$ and grid size $20 \times 20$ are generated with varying $\alpha_a$ drawn from integral uniform $[r; 3r]$ for $r \in \{5, 10, 15, 20, 25\}$. To keep the relative scales of the parameters consistent with the previous experiments,
Table 3. Performance of the binary search for varying budgets.

<table>
<thead>
<tr>
<th>η</th>
<th>iter</th>
<th>time</th>
<th>gap</th>
<th>iter</th>
<th>time</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3.0</td>
<td>0.30</td>
<td>0.00</td>
<td>4.0</td>
<td>1.33</td>
<td>0.27</td>
</tr>
<tr>
<td>4</td>
<td>2.8</td>
<td>0.36</td>
<td>1.06</td>
<td>4.0</td>
<td>2.68</td>
<td>0.44</td>
</tr>
<tr>
<td>6</td>
<td>2.8</td>
<td>0.38</td>
<td>1.02</td>
<td>4.0</td>
<td>1.70</td>
<td>0.54</td>
</tr>
<tr>
<td>8</td>
<td>3.0</td>
<td>0.35</td>
<td>0.17</td>
<td>4.0</td>
<td>1.69</td>
<td>0.26</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>0.36</td>
<td>0.03</td>
<td>4.0</td>
<td>1.48</td>
<td>1.50</td>
</tr>
<tr>
<td>avg</td>
<td>2.92</td>
<td>0.35</td>
<td>0.46</td>
<td>4.0</td>
<td>1.78</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 4. Performance of b&b and b&c for varying budgets.

<table>
<thead>
<tr>
<th></th>
<th>Uncorrelated instances</th>
<th></th>
<th>Correlated instances</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>η</td>
<td>rgap</td>
<td>stime</td>
<td>time</td>
<td>egap (#)</td>
</tr>
<tr>
<td>2</td>
<td>24.4</td>
<td>3</td>
<td>6</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>20.0</td>
<td>469</td>
<td>676</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>18.7</td>
<td>888</td>
<td>1,412</td>
<td>0.1(1)</td>
</tr>
<tr>
<td>8</td>
<td>18.6</td>
<td>343</td>
<td>1,618</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>18.1</td>
<td>898</td>
<td>1,705</td>
<td>1.1(1)</td>
</tr>
<tr>
<td>avg</td>
<td>20.0</td>
<td>520</td>
<td>1,083</td>
<td>0.2(2)</td>
</tr>
</tbody>
</table>

Table 5. Performance of the binary search for varying interdiction costs.

<table>
<thead>
<tr>
<th></th>
<th>Uncorrelated</th>
<th></th>
<th>Correlated</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>iter</td>
<td>time</td>
<td>gap</td>
<td>iter</td>
</tr>
<tr>
<td>5</td>
<td>3.0</td>
<td>0.39</td>
<td>3.93</td>
<td>4.0</td>
</tr>
<tr>
<td>10</td>
<td>2.8</td>
<td>0.42</td>
<td>1.94</td>
<td>4.0</td>
</tr>
<tr>
<td>15</td>
<td>2.8</td>
<td>0.41</td>
<td>3.43</td>
<td>4.0</td>
</tr>
<tr>
<td>20</td>
<td>2.8</td>
<td>0.41</td>
<td>1.20</td>
<td>4.0</td>
</tr>
<tr>
<td>25</td>
<td>2.8</td>
<td>0.44</td>
<td>1.76</td>
<td>4.0</td>
</tr>
<tr>
<td>avg</td>
<td>2.84</td>
<td>0.41</td>
<td>2.45</td>
<td>4.0</td>
</tr>
</tbody>
</table>

The budget parameter β is set to \( \frac{\alpha Y}{4} \). Tables 3–6 summarize the results. The optimality gaps for the binary search algorithm are higher for these experiments with similar run times. Both the binary search and the exact algorithms appear to be insensitive to the changes in the interdiction cost in our experiments.
Table 6. Performance of b&b and b&c for varying interdiction costs.

<table>
<thead>
<tr>
<th>r</th>
<th>Uncorrelated instances</th>
<th>Cplex</th>
<th>Cplex + cuts</th>
<th>Correlated instances</th>
<th>Cplex</th>
<th>Cplex + cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r</td>
<td>rgap</td>
<td>stime</td>
<td>time</td>
<td>egap (#)</td>
<td>nodes</td>
</tr>
<tr>
<td>5</td>
<td>22.4</td>
<td>.969</td>
<td>2,405</td>
<td>0.6(2)</td>
<td>299,432</td>
<td>705</td>
</tr>
<tr>
<td>10</td>
<td>22.5</td>
<td>1,299</td>
<td>2,552</td>
<td>1.8(2)</td>
<td>282,257</td>
<td>725</td>
</tr>
<tr>
<td>15</td>
<td>22.6</td>
<td>1,611</td>
<td>2,383</td>
<td>3.3(3)</td>
<td>267,595</td>
<td>686</td>
</tr>
<tr>
<td>20</td>
<td>22.2</td>
<td>1,436</td>
<td>2,905</td>
<td>4.4(4)</td>
<td>279,349</td>
<td>704</td>
</tr>
<tr>
<td>25</td>
<td>22.3</td>
<td>1,502</td>
<td>2,905</td>
<td>4.2(4)</td>
<td>339,789</td>
<td>691</td>
</tr>
<tr>
<td>avg</td>
<td>22.4</td>
<td>1,363</td>
<td>2,630</td>
<td>2.8(15)</td>
<td>293,684</td>
<td>702</td>
</tr>
</tbody>
</table>

Finally, we test the performance of the binary search algorithm for larger grid sizes up to 100 × 100 to see how it scales up. Five instances of each size are generated as in our original set of instances. The exact algorithms are not run for these large instances; therefore, the gap is computed against the convex relaxation of the problem and, hence, it provides an upper bound on the optimality gap. Table 7 reports the number of iterations, the time spent for the algorithm, and the percentage integrality gap, that is the gap between the upper bound found by the algorithm and the lower bound from the convex relaxation. Observe that the 100 × 100 instances have about 20,000 arcs. The correlated instances for this size could not be run due to memory limit. For the 20 × 20 instances, the reported upper bounds 20.73% and 17.31% on the optimality gap should be compared with the actual optimality gaps 1.06% and 0.44% in Table 1. The large difference between the exact gap in Table 1 and igap in Table 7 is indicative of poor lower bounds from the convex relaxations, rather than poor upper bounds. The binary search algorithm converges in a small number of iterations for these large instances as well; however, solving quadratic 0-1 optimization problems at each iteration takes significantly longer time.

4. Conclusion

In this paper we introduce a successive quadratic optimization procedure embedded in a bisection search for finding high quality solutions to discrete mean-risk minimization problems with a conic quadratic objective. The search algorithm is applied on a non-convex upper-bounding function that provides tight values at local
Table 7. Performance of the binary local search for larger networks.

<table>
<thead>
<tr>
<th>p × q</th>
<th>iter</th>
<th>time (s)</th>
<th>igap (%)</th>
<th>iter</th>
<th>time (s)</th>
<th>igap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 × 20</td>
<td>2.8</td>
<td>0.36</td>
<td>20.73</td>
<td>4.0</td>
<td>2.68</td>
<td>17.31</td>
</tr>
<tr>
<td>40 × 40</td>
<td>3.0</td>
<td>5.33</td>
<td>26.15</td>
<td>5.0</td>
<td>80.47</td>
<td>11.81</td>
</tr>
<tr>
<td>60 × 60</td>
<td>3.2</td>
<td>35.66</td>
<td>27.09</td>
<td>6.0</td>
<td>502.14</td>
<td>10.12</td>
</tr>
<tr>
<td>80 × 80</td>
<td>3.6</td>
<td>141.96</td>
<td>27.31</td>
<td>6.0</td>
<td>3,199.11</td>
<td>8.53</td>
</tr>
<tr>
<td>100 × 100</td>
<td>10.2</td>
<td>4,991.3</td>
<td>31.08</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>avg</strong></td>
<td><strong>4.6</strong></td>
<td><strong>1,034.92</strong></td>
<td><strong>26.47</strong></td>
<td><strong>5.3</strong></td>
<td><strong>945.97</strong></td>
<td><strong>11.94</strong></td>
</tr>
</tbody>
</table>

minima. Computations with the network interdiction problem with stochastic capacities indicate that the proposed method finds solutions within 1–4% optimal in a small fraction of the time required by exact branch-and-bound and branch-and-cut algorithms. Although we demonstrate the approach for the network interdiction problem with stochastic capacities, since method is agnostic to the constraints of the problem, it can be applied to any 0-1 optimization problem with a mean-risk objective.

ACKNOWLEDGEMENT

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

REFERENCES


27. Xiao Lei, Siqian Shen, and Yongjia Song. Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers & Operations Research*


