BCOL RESEARCH REPORT 16.01

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

NETWORK DESIGN WITH PROBABILISTIC CAPACITIES

ALPER ATAMTÜRK AND AVINASH BHARDWAJ

Abstract. We consider a network design problem with random arc capacities and give a formulation with a probabilistic capacity constraint on each cut of the network. To handle the exponentially-many probabilistic constraints a separation procedure that solves a nonlinear minimum cut problem is introduced. For the case with independent arc capacities, we exploit the supermodularity of the set function defining the constraints and generate cutting planes based on the supermodular covering knapsack polytope. For the general correlated case, we give a reformulation of the constraints that allows to uncover and utilize the submodularity of a related function. The computational results indicate that exploiting the underlying submodularity and supermodularity arising with the probabilistic constraints provides significant advantages over the classical approaches.

Keywords. Probabilistic constraints, submodularity, supermodularity, polymatroids, correlation.

August 2016

1 Introduction

Designing networks with random components often necessitates building costly extra capacity to satisfy a desired service level. Examples include communication networks, air and ground traffic networks, supply-chain networks. It is of significant interest to policy makers and other stakeholders to understand the trade-off between the network design cost and the service level to assess the value of additional capacity investments. To this end, we give an optimal network design model with probabilistic capacity constraints and propose effective solution methods that exploit the underlying combinatorial properties.

Let $N(V, A)$ be a network with vertices V and arcs A. Given random arc capacities $\xi_a,\ a \in A$, we are interested in the problem of selecting a minimum-cost subset of the arcs to satisfy the demand on the network with a high probability. Without loss of generality, we may assume that there is a single source s and a single sink t with

A. Atamtürk: Department of Industrial Engineering & Operations Research, University of California, Berkeley, CA 94720. atamturk@berkeley.edu
A. Bhardwaj: Department of Industrial Engineering & Operations Research, University of California, Berkeley, CA 94720. avinash@ieor.berkeley.edu .

demand d, as one can collate sources and sinks into a single node each respectively via arcs with deterministic bounds. Let h_a be the fixed "design" cost of arc $a \in A$ for inclusion in the selected sub-network.

We model the problem with a probabilistic capacity constraint for each cut of the network. That is, we look for a minimum cost subset of the arcs so that each cut of the selected sub-network satisfies the demand with probability at least $1-\epsilon$, where $0 < \epsilon \leq 0.5$. Then, letting x_a be 1 if arc a is selected, 0 otherwise, the probabilistic network design problem is formulated as:

$$\textbf{(PND)} \quad \min_{\mathbf{x} \in \{0,1\}^m} \mathbf{h}'\mathbf{x} : \mathbf{Pr}\left(\sum_{a \in C} \xi_a x_a \geq d\right) \geq 1-\epsilon, \quad \forall\, C \in \mathcal{C}, \tag{1}$$

where $\mathcal{C}$ denotes the set of all $s-t$ cuts in the network $N(V,A)$.

If $\boldsymbol{\xi}$ is normally distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, problem (1) is equivalently stated with conic quadratic constraints:

$$\textbf{(CQND)} \quad \min_{\mathbf{x} \in \{0,1\}^m} \mathbf{h}'\mathbf{x} : \boldsymbol{\mu}_C' \mathbf{x}_C - \Omega \parallel \boldsymbol{\Sigma}_C^{1/2} \mathbf{x}_C \parallel \geq d, \forall\, C \in \mathcal{C}, \tag{2}$$

where $\Omega = \phi^{-1}(1-\epsilon)$ and ϕ is the standard normal c.d.f., $\boldsymbol{\mu}_C$ is the vector of mean capacities for the arcs in cut C and $\boldsymbol{\Sigma}_C$ is the covariance matrix of the arcs in C. The term $\Omega \parallel \boldsymbol{\Sigma}^{1/2} \mathbf{x} \parallel$ is used to build sufficient slack into the constraint to accommodate the variability of $\boldsymbol{\xi}$ around its mean $\boldsymbol{\mu}$. Problem (2) is $\mathcal{NP}$-hard as the the deterministic network design problem with $\Omega = 0$ is $\mathcal{NP}$-hard. We use σ_{ij} to denote the (ij)th coefficient of the covariance matrix $\boldsymbol{\Sigma}$ and σ_i^2 for the ith variance (σ_{ii}).

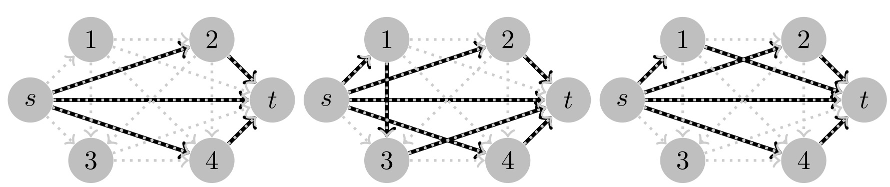
If the distribution of $\boldsymbol{\xi}$ is unknown, robust versions can be modeled as (2) by using appropriate choices of Ω. If ξ_i's are known only through their first two moments, then inequalities in (2) with $\Omega = \sqrt{(1-\epsilon)/\epsilon}$ imply the probabilistic constraints in (1) [8, 14]. Alternatively, if ξ_i's are independent and symmetric with support $[u_i - \sigma_i, u_i + \sigma_i]$, then inequalities in (2) with $\Omega = \sqrt{\ln(1/\epsilon)}$ and diagonal $\boldsymbol{\Sigma} = \boldsymbol{D}$, where $D_{ii} = \sigma_i^2$, imply the probabilistic constraints in (1) [6, 7]. Hence, under different assumptions on $\boldsymbol{\xi}$, one arrives at different instances of the model (2) with conic quadratic constraints $(\Omega \geq 0)$.

Contributions. There are two main challenges in solving **(CQND)**. The first one is that the formulation has an exponentially many conic quadratic constraints – one for each cut of the network. We use a separation approach to generate only a small subset of the conic quadratic inequalities. Finding a violated conic quadratic constraint (2) is modeled as a minimum cut problem with a concave objective, which includes the $\mathcal{NP}$-hard max-cut problem as a special case. We give alternative formulations and compare them for the independent and correlated cases. The second challenge is that there is often a large gap between the optimal objective and its convex relaxation. In order to strengthen the convex relaxations, we derive strong cutting planes that make use of submodularity. In particular, we show that when the capacities are independent, under a mild assumption, the feasible set of **(CQND)** is the intersection of monotone supermodular covering knapsacks. For the general correlated case, we give a reformulation to uncover the combinatorial structure of another set function for the underlying constraints. This reformulation allows us to make use of submodularity even if the original constraint function is not submodular. The proposed reformulation is independent from the network structure and, therefore, it can be useful for other combinatorial problems with a risk objective or risk constraints.

Literature review. Uncertainties in network capacities may be caused a variety of natural or artificial factors. In the context of air or ground transportation network infrastructure, the link (arc) capacity uncertainties are due to weather conditions and traffic along with other factors. Similar link uncertainties arise in the design and operation of communication network infrastructure, power grid, and related applications [15]. Transportation networks have been studied well with respect to variations in the link capacities. Chen et al. [9] study the capacity reliability of a road network, that is, the probability that a network can support a given demand at a given service level when the link capacities are uncertain. Lo and Tung [16] formulate a probabilistic user equilibrium model for maximizing the network flow capacity while accounting for the effects of travel time reliability due to degradable links in terms of predetermined link performance function (proxy for capacity distributions). We refer the reader to Chen et al. [10] for a survey on the stochastic transportation network design problem. In the context of communication networks, Claßen et al. [11] study a broadband wireless network which uses microwave links to deliver data, where the channel conditions account for uncertainty in link capacities. Yang and Wen [21] consider the transmission expansion problem in a power network for handling uncertainties corresponding to future plant locations in terms of generation expansion and load growth. Fortz and Poss [13] study the commodity packing problem on a network with uncertain capacities and linearize the probabilistic linear constraint (2) when ξ_a's are independent and $\xi_a \sim \mathcal{N}(\mu_a, \lambda\mu_a)\ \forall\, a \in A$. Song et al. [20] give a scenario-based approach to multi-row chance-constraints on binary variables. We do not address joint probabilistic constraints here.

Example. We end this section with a motivating example that highlights the value of model **(CQND)** in constructing a trade-off curve of service level versus design cost. To do so we construct networks with varying service levels and design costs; and with simulations compute the observed service levels. The data for the sample six-node network is presented in Table 7 of Appendix A.

First using reformulation **(CQND)** we construct minimum cost networks for six service levels: 50%, 70%, 80%, 97.5%, 99%, 99.9% . The 50% service level is the nominal case corresponding to the deterministic network design with $\Omega = 0$. The network configurations obtained for $50\%, 80\%$ and 97.5% service levels are shown in Figure 1. Note that the networks are not nested, that is, the arcs used for lower service levels are not necessarily included in the networks with higher service levels.



(A) service level 50% (B) service level 80% (C) service level 97.5%

FIGURE 1. Minimum cost networks corresponding to different service levels.

The complete set of network configurations for all service levels are displayed in Appendix A.

Next for each network configuration, we generate 10,000 samples from the corresponding arc capacity distributions. For each network sample we check whether that network satisfies the demand, i.e., whether the minimum cut capacity is greater than or equal to the demand to compute the simulated service level. The results of this simulation are summarized in Table 1. The frequency distributions shown in Appendix A indicate that they are skewed and non-normal for all network configurations. Nevertheless, the simulated service levels are reasonably close to the model service levels and increase monotonically with $1-\epsilon$.

Table 1. Simulation results.

$1-\epsilon$	cost	minimum cut			simulated
		min	mean	max	service level
50%	100%	121.9	222.1	274.6	39.81%
70%	104%	124.0	238.4	293.2	70.44%
80%	127%	154.0	249.2	306.5	82.68%
97.5%	135%	195.2	301.4	356.4	99.68%
99%	177%	209.6	303.6	360.2	99.85%
99.9%	186%	209.6	313.4	368.5	99.96%

Figure 2 shows the trade-off between the cost of the network design and the service level. The red curve is for the predicted model service level, whereas the blue curve is for the simulated service level. As expected, the cost of the design increases with the service level, at steeper rates for higher service levels. The trade-off curve is not convex as the network configuration changes in discrete steps.

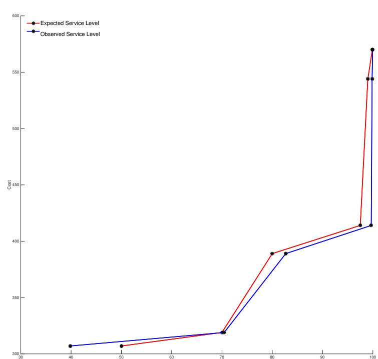


Figure 2. Network cost versus service level, model versus simulated.

The remainder of this paper is organized as follows: In Section 2 we give reformulations and strong valid inequalities for the independent and correlated capacities, separately. In Section 3 we discuss the separation problem for the capacity constraints for each case and compare alternative formulations. In Section 4 we present computational experiments on testing the effectiveness of the valid inequalities in solving the probabilistic network design problem.

2 Strengthening the formulation

In this section we describe how to strengthen the convex relaxation of model **(CQND)**. We consider each constraint (2) separately and, for simplicity of notation, drop the subscript C. The case with independent arc capacities warrants special consideration as it leads to a specific combinatorial structure. Therefore, we discuss the cases of independent capacities and correlated capacities separately in Sections 2.1 and 2.2.

2.1 Independent arc capacities

When the arc capacities of the network are independent, the covariance matrix $\mathbf{\Sigma}$ reduces to a diagonal matrix $\boldsymbol{D}$ of variances; that is, $D_{ii} = \sigma_i^2$. Then, as $x_i^2 = x_i$ for binary $\mathbf{x}$, each constraint in (2) reduces to

$$f(\boldsymbol{x}) = \boldsymbol{\mu}'\mathbf{x} - \Omega\sqrt{\boldsymbol{D}\,\mathbf{x}} \;\geq\; d.$$

Given a finite ground set A, a set function $g : 2^A \to \mathbb{R}$ is supermodular if its difference function

$$\rho_i(S) := g(S \cup i) - g(S) \;\text{ for } S \subseteq N \setminus i$$

is non-decreasing on $N \setminus i$; that is, $\rho_i(S) \leq \rho_i(T)$, $\forall\, S \subseteq T \subseteq N \setminus i$ and $i \in N$ [19]. If f is supermodular, then $-f$ is submodular and vice-versa.

In a recent paper, Atamtürk and Bhardwaj [3] study the polyhedral structure of the supermodular covering knapsack set

$$K_g \;:=\; \{\mathbf{x} \in \{0,1\}^n \;:\; g(\mathbf{x}) \geq d\}$$

for a non-decreasing supermodular set function g. They give strong valid inequalities, refereed to as pack inequalities and extended pack inequalities, for K_g.

It is easy to check that $f(\boldsymbol{x}) = \boldsymbol{\mu}'\mathbf{x} - \Omega\sqrt{\boldsymbol{D}\,\mathbf{x}}$ is supermodular on $\{0,1\}^A$ [e.g. 1]. Moreover, it is reasonable to assume that the network capacity increases with the addition of new arcs. In particular, f is non-decreasing if

$$\text{(A.1)} \qquad \mu_a \geq \Omega\sigma_a \text{ for all } a \in A.$$

For the majority of the paper we will assume that (A.1) holds. In order to strengthen the continuous relaxation of **(CQND)**for the independent case, we utilize the extended pack inequalities of Atamtürk and Bhardwaj [3] for K_f. The computational effectiveness of the pack inequalities and their extensions for the probabilistic network design problem is detailed in Section 4. In Section 4.3, we show that the results can be applied and useful even if assumption (A.1) does not hold for all arcs.

Unfortunately supermodularity is lost once correlations are introduced to the covariance matrix $\mathbf{\Sigma}$. In the following section we show how to exploit additional combinatorial structure when taking into account the correlations among the arc capacities.

2.2 Correlated arc capacities

For a non-diagonal covariance matrix $\mathbf{\Sigma}$, the cut capacity function

$$f(\boldsymbol{x}) = \boldsymbol{\mu}'\mathbf{x} - \Omega\sqrt{\mathbf{x}'\mathbf{\Sigma}\,\mathbf{x}}$$

is not supermodular and the pack inequalities and their extensions that are valid for the diagonal case are no longer applicable. Nevertheless, we show below that under assumption (A.1), we can utilize submodularity of a related function as opposed to supermodularity of the original cut capacity function.

First, observe that for $\mathbf{x} : \boldsymbol{\mu}'\mathbf{x} \geq d$, we have

$$f(\mathbf{x}) \geq d \iff \Omega^2\mathbf{x}'\mathbf{\Sigma}\mathbf{x} \leq (\boldsymbol{\mu}'\mathbf{x} - d)^2.$$

However, since $\Omega \geq 0$, for any feasible solution, $\boldsymbol{\mu}'\mathbf{x} \geq d$ holds. Therefore, we may use the more convenient quadratic constraint

$$q(\mathbf{x}) := \Omega^2\mathbf{x}'\mathbf{\Sigma}\mathbf{x} - (\boldsymbol{\mu}'\mathbf{x} - d)^2 \leq 0 \tag{3}$$

instead of the original $f(\mathbf{x}) \geq d$. The function q is the difference of two convex functions; it is neither convex, not concave. However, it turns out that q is submodular under assumption (A.1).

Proposition 1. *[18] Let $p : \{0,1\}^n \to \mathbb{R}$ be the quadratic function*

$$p(\mathbf{x}) \;:=\; \mathbf{x}'\,\mathbf{Q}\,\mathbf{x},$$

where $\mathbf{Q}$ is a symmetric matrix. p is submodular if $Q_{ij} \leq 0,\ 1 \leq i < j \leq n$.

Proposition 2. *If (A.1) holds, then the set function $q(\mathbf{x}) = \Omega^2\mathbf{x}'\mathbf{\Sigma}\mathbf{x} - (\boldsymbol{\mu}'\mathbf{x} - d)^2$ is submodular.*

Proof. Note that by observing $x_i = x_i^2$ for binary $\mathbf{x}$, $q(\mathbf{x})$ can be written as

$$q(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_i + 2\sum_{i=1}^{n}\sum_{j>i}^{n} \beta_{ij} x_i x_j - d^2, \tag{4}$$

where $\alpha_i = \Omega^2\sigma_i^2 + 2\mu_i d - \mu_i^2$ and $\beta_{ij} = \Omega^2\sigma_{ij} - \mu_i\mu_j$. As $\sigma_{ij} \leq \sigma_i\sigma_j$ and by assumption (A.1), we have

$$\beta_{ij} = \Omega^2\sigma_{ij} - \mu_i\mu_j \leq \Omega^2\sigma_i\sigma_j - \mu_i\mu_j \leq 0.$$

Then from Proposition 1 it follows that q is submodular. □

Submodular pack [3] and cover [5] inequalities and their extensions have been used to strengthen the continuous relaxations of submodular knapsack sets when the underlying set function is monotone. Unfortunately, q is neither non-increasing nor non-decreasing and, therefore, these inequalities are not applicable. Instead, here we will consider linear underestimates from the extended polymatroid associated with submodular q.

Definition 1. For a submodular set function $g : \{0,1\}^A \to \mathbb{R}$, the polyhedron

$$EP_g \;:=\; \left\{\mathbf{v} \in \mathbb{R}^A \;:\; \mathbf{v}(S) \leq g(S),\ S \subseteq A\right\}$$

is called the extended polymatroid associated with g.

Theorem 3. *[12] For a submodular function g with $g(\emptyset) = 0$, each vertex of the extended polymatroid EP_g is given by*

$$v_j^{\boldsymbol{\pi}} = g(S_j^{\boldsymbol{\pi}}) - g(S_{j-1}^{\boldsymbol{\pi}})$$

for a permutation $\boldsymbol{\pi}$ of A, where $S_j^{\boldsymbol{\pi}}$ denotes the set consisting of the first j elements of the permutation $\boldsymbol{\pi}$.

Let $\mathbf{V}_g$ be the set of all extreme points of EP_g and consider the discrete epigraph of the set function g

$$B_g := \left\{ (\mathbf{x}, z) \in \{0,1\}^A \times \mathbb{R} \ : \ g(x) \leq z \right\}.$$

It follows from polarity that any inequality

$$\mathbf{v}'\mathbf{x} \leq z, \ \mathbf{v} \in EP_g$$

is valid for B_g [e.g. 4]. In particular, $\mathbf{v}'\mathbf{x} \leq z$ defines a facet of $\operatorname{conv}(B_g)$ for all $\mathbf{v} \in \mathbf{V}_g$.

Proposition 4. *For $\mathbf{v} \in \mathbf{V}_g$ and fixed $\gamma \in \mathbb{R}$, let $K_{\mathbf{v}} := \{\mathbf{x} \in \{0,1\}^n \ : \ \mathbf{v}'\mathbf{x} \leq \gamma\}$. Then*

$$K_g := \{\mathbf{x} \in \{0,1\}^n \ : \ g(\mathbf{x}) \leq \gamma\} = \bigcap_{\mathbf{v} \in \mathbf{V}_g} K_{\mathbf{v}}.$$

Proposition 4 allows us to linearize the submodular capacity constraint (3) using inequalities from the vertices of the extended polymatroid EP_q. Although the number of such inequalities is factorial in $|A|$, we can use a separation algorithm to generate violated ones on the fly. Given a point $\bar{\mathbf{x}}$, we solve

$$\max\{\bar{\mathbf{x}}'\mathbf{v} : \mathbf{v} \in \mathbf{V}_q\}$$

exactly to find the strongest polymatroid knapsack inequality $\mathbf{v}'\mathbf{x} \leq \gamma$ using the greedy algorithm of Edmonds [12]. Note that each $K_{\mathbf{v}}$ is a $0-1$ knapsack set, for which a large class of valid inequalities are known. To strengthen the formulation further, we generate cover inequalities for $K_{\mathbf{v}}$ and lift them using the superadditive lifting functions [2]. As an additional step, we aggregate multiple polymatroid knapsack inequalities and generate cover inequalities for the aggregated knapsack set and lift them in the same way. We present our computational analysis of the aforementioned approach in Section 4.

3 Separation

As **(CQND)** contains exponentially many probabilistic capacity constraints, one needs to utilize a separation approach to find violated constraints for a given solution $\bar{\mathbf{x}}$ to a relaxation of the problem; i.e., to find a cut C of the network for which the corresponding probabilistic capacity constraint is violated:

$$\boldsymbol{\mu}_C'\bar{\mathbf{x}}_C - \Omega\sqrt{\bar{\mathbf{x}}_C'\boldsymbol{\Sigma}_C\,\bar{\mathbf{x}}_C} < d. \tag{5}$$

We now formalize the separation problem for the probabilistic capacity constraints. Let $\operatorname{diag}(\bar{\mathbf{x}})$ be the diagonalization of the vector $\bar{\mathbf{x}}$ and $\boldsymbol{\mu}_{\bar{\mathbf{x}}} = \operatorname{diag}(\bar{\mathbf{x}})\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{\bar{\mathbf{x}}} = \operatorname{diag}(\bar{\mathbf{x}})\boldsymbol{\Sigma}\operatorname{diag}(\bar{\mathbf{x}})$. Further, let the binary variable z_{ij} be 1 if arc $(ij) \in A$ is in the chosen cut and 0 otherwise. Then, a cut C achieving the smallest left-hand-side for inequality (5) can be found by solving the following nonlinear min-cut problem:

$$
(\mathbf{SEP}) \quad
\begin{aligned}
\min \quad & \Theta(\mathbf{z}) = \boldsymbol{\mu}'_{\bar{\mathbf{x}}}\mathbf{z} - \Omega\sqrt{\mathbf{z}'\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}\mathbf{z}} & & \\
\text{s.t.} \quad & \lambda_i - \lambda_j \le z_{ij}, & (ij) \in A & \quad (6)\\
& \lambda_i \ge z_{ij}, & (ij) \in A & \quad (7)\\
& \lambda_j \le 1 - z_{ij}, & (ij) \in A & \quad (8)\\
& \lambda_s = 1 & & \quad (9)\\
& \lambda_t = 0 & & \quad (10)\\
& \lambda_i, \lambda_j \ge 0, & (ij) \in A & \quad (11)\\
& z_{ij} \in \{0,1\}, & (ij) \in A. & \quad (12)
\end{aligned}
$$

Note that $\Theta(\mathbf{z})$ is a concave function as $\Omega \ge 0$ and $\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}$ is positive semidefinite. Constraints (6)–(8) ensure that only the arcs from the source (s) set to the sink (t) set are included in the chosen cut. Notice that for the usual (deterministic) min-cut problem with $\Omega = 0$, as the objective is nonnegative, inequality (6) is sufficient. However, one needs constraints (7) and (8) to ensure the correctness of the formulation with $\Omega > 0$. Observe that (**SEP**) is $\mathcal{NP}$-hard, since for $\boldsymbol{\mu} = \mathbf{0}$ and diagonal $\boldsymbol{\Sigma}$ it reduces to the max-cut problem.

In order to have an effective method for solving **(CQND)**, it is imperative to be able generate violated probabilistic capacity inequalities fast. In the following subsections we discuss solving the separation problem (**SEP**) for the independent capacities and correlated capacities separately as the correlations change the structure of the separation problem significantly.

3.1 Independent arc capacities

When the arc capacities are independent, the covariance matrix is diagonal; therefore, the separation problem reduces to

$$
(\mathbf{SEP\text{-}D}) \quad
\begin{aligned}
\min \quad & \Theta(\mathbf{z}) = \boldsymbol{\mu}'_{\bar{\mathbf{x}}}\mathbf{z} - \Omega\sqrt{\sum_{a\in A}\sigma_a^2\bar{x}_a^2\, z_a} \\
& \text{s.t. } (6) - (12).
\end{aligned}
$$

In this case the objective function Θ is convex, moreover, a non-decreasing supermodular function. Nevertheless, minimizing the set function Θ over binaries is $\mathcal{NP}$-hard [1]. Nemhauser et al. [18] show that any supermodular function minimization problem can be formulated using linear inequalities:

$$
\begin{aligned}
\min w \ \ \text{s.t.} & \\
w \ge \Theta(S) - \sum_{i\in S}\rho_i(A\setminus i)(1-z_i) + \sum_{i\in N\setminus S}\rho_i(S)z_i,\ \forall S \subseteq A & \qquad (13)\\
w \ge \Theta(S) - \sum_{i\in S}\rho_i(S\setminus i)(1-z_i) + \sum_{i\in N\setminus S}\rho_i(\emptyset)z_i,\ \forall S \subseteq A & \qquad (14)
\end{aligned}
$$

However, the linear mixed 0–1 formulation with inequalities (13)–(14) is computationally ineffective as its convex relaxation is usually weak [1].

Alternatively, (**SEP-D**) can be formulated as a quadratically-constrained mixed 0–1 optimization problem by bringing the nonlinear term in the objective into the constraints and squaring it:

$$(\textbf{SEP-QCQP}) \quad \begin{aligned} \min \quad & \Theta(\mathbf{z}, t) = \boldsymbol{\mu}'_{\bar{\mathbf{x}}}\mathbf{z} - \Omega t \\ \text{s.t.} \quad & t^2 \le \sum_{a \in A} \sigma_a^2 \bar{x}_a^2 \; z_a \\ & (6) \; - \; (12). \end{aligned} \tag{15}$$

Note that (15) is a convex constraint with only single quadratic term and can be readily handled by commercial solvers. Table 2 shows a comparison of the solution of the two formulations with `CPLEX` version 12.6 on a 2.93GHz Pentium Linux workstation with 8GB main memory. Each row shows the average number of branch-and-bound nodes till optimality and the time to the first feasible solution and the time to proving optimality (in seconds) for five instances with varying network sizes and Ω. Observe that (**SEP-QCQP**) is solved to optimality much faster than the linearization of (**SEP-D**) with (13)–(14). For all most all instances it requires no branching at all. Furthermore, in all cases feasible solutions are found much earlier. Therefore, we use formulation (**SEP-QCQP**) for the independent case in the computational experiments presented in Section 4.

Table 2. Separation formulations: independent case.

nodes	arcs	Ω	SEP-D with ineqs. (13)–(14)			SEP-QCQP		
			nodes	feasible	optimal	nodes	feasible	optimal
		1	15	0.4	0.9	0	0.1	0.4
50	1225	3	18	0.8	1.1	0	0.2	0.4
		5	21	0.8	1.5	0	0.2	0.4
		1	12	1.8	3.2	0	0.9	1.4
70	2415	3	17	2.9	4.4	0	0.6	1.4
		5	30	5.1	6.4	0	0.9	1.4
		1	14	9.1	16.9	0	2.7	5.3
100	4950	3	25	12.9	20.3	0	2.9	5.6
		5	41	18.4	26.6	0	2.4	5.6
avg			**21.4**	**5.8**	**9.0**	**0**	**1.2**	**2.4**
stdev			**9.3**	**6.4**	**9.7**	**0**	**1.1**	**2.3**

3.2 Correlated arc capacities

For the general case with correlated arc capacities, as the covariance matrix is not diagonal, the objective of the separation problem is no longer supermodular nor can it be formulated as (**SEP-QCQP**) due to the bilinear terms. To overcome the difficulty, we present two approaches. The first one is the usual McCormick linearization of the bilinear terms, whereas the second one is a two-step approach that allows one to convert the separation problem into one with a quadratic constraint on binary variables.

McCormick linearization

The classical approach of McCormick [17] to solving non-convex quadratic optimization problems is to linearize the bilinear terms using additional variables and constraints. In particular, a bilinear term $z = x_1 x_2$ is re-formulated with the *McCormick inequalities*:

$$z \ge \max(x_1 + x_2 - 1, 0), \qquad z \le \min(x_1, x_2). \tag{16}$$

Now consider the objective function of the separation problem (**SEP**):

$$\Theta(\mathbf{z}) = \boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - \Omega\sqrt{\mathbf{z}'\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}\mathbf{z}} = \boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - \Omega\sqrt{\sum_i (\sigma_i \bar{x}_i)^2 z_i^2 + 2\sum_i \sum_{j>i} \sigma_{ij}\bar{x}_i\bar{x}_j z_i z_j}. \quad (17)$$

Introducing a new variable y_{ij} for each distinct pair $\{i,j\}$, we formulate the separation problem as

$$(\textbf{SEP-MC}) \quad \begin{array}{lll} \min & \Theta(\mathbf{z},\mathbf{y},t) = \boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - \Omega t & \\ \text{s.t.} & t^2 \leq \sum_i (\sigma_i\bar{x}_i)^2 z_i + 2\sum_i\sum_{j>i}\sigma_{ij}\bar{x}_i\bar{x}_j y_{ij} & (18) \\ & z_i + z_j \leq y_{ij} + 1, \quad i,j \in A & (19) \\ & y_{ij} \leq z_i, \quad i,j \in A & (20) \\ & y_{ij} \leq z_j, \quad i,j \in A & (21) \\ & z_i \in \{0,1\}, \quad i \in A & (22) \\ & (6) - (12). & \end{array}$$

Observe that while (17) is a concave function in z, (18) is a convex constraint with a single quadratic term, and, thus, (**SEP-MC**) can be readily solved by conic quadratic MIP solvers. Also note that constraints (19)–(22) implicitly restrict y_{ij} to be binary. However, a challenge with this McCormick linearization is that it often yields a weak relaxation of the original formulation. Moreover, the large number (quadratic in $|A|$) of auxiliary variables and constraints introduced for the linearization makes it difficult to scale up the dimension of the problem.

A quadratic reformulation

In this section we give an alternative reformulation of the separation problem that does not require auxiliary variables. In particular, we reformulate the separation problem with correlations using a quadratic set function as in Section 2.2.

Note that for a given $\bar{\mathbf{x}}$, the cut corresponding to a feasible solution $\mathbf{z}$ to (**SEP**) is satisfied if and only if

$$\boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - \Omega\sqrt{\mathbf{z}'\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}\mathbf{z}} \geq d. \quad (23)$$

Assuming $\boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} \geq d$, this is equivalent to

$$\left(\boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - d\right)^2 \geq \Omega^2\mathbf{z}'\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}\mathbf{z}. \quad (24)$$

Observe that it is easy to satisfy the condition $\boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} \geq d$ as it corresponds to solving a standard min-cut problem. Therefore, in the first stage, we ensure that $\bar{\mathbf{x}}$ satisfies that standard (linear) cut capacity constraints $\boldsymbol{\mu}'\mathbf{x} \geq d$ and then, in the second stage, we look for a violated probabilistic capacity constraint by solving

$$(\textbf{SEP-BQC}) \quad \begin{array}{lll} \text{minimize} & \boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} & \\ \text{s.t.} & (\boldsymbol{\mu}_{\bar{\mathbf{x}}}'\mathbf{z} - d)^2 \leq \Omega^2 \cdot \mathbf{z}'\boldsymbol{\Sigma}_{\bar{\mathbf{x}}}\mathbf{z} - \epsilon & (25) \\ & (6) - (12) & \end{array}$$

for a sufficiently small $\epsilon > 0$. Note that any feasible solution to (**SEP-BQC**) corresponds to a violated probabilistic cut constraint. As (25) is a quadratic constraint on binary variables, it can be readily solved by commercial MIP solvers.

In the following, we compare the computational performance for three solution approaches:

(1) (**SEP-MC**) with full McCormick linearization
(2) (**SEP-BQC**) with partial McCormick linearization of only the non-convex right-hand-side of constraint (25).
(3) (**SEP-BQC**) with no linearization.

Each row of Table 3 shows the average for five randomly generated instances with varying network size and Ω. We compare the number of nodes explored in the search tree and the time (in seconds) to the first feasible solution and optimality.

Table 3. Separation formulations: correlated case.

nodes	arcs	Ω	Full McCormick			Partial McCormick			SEP-BQC		
			nodes	feasible	optimal	nodes	feasible	optimal	nodes	feasible	optimal
20	105	1	2	1.4	3.6	5	1.1	5	0	0	0.2
		3	0	1.8	4.3	14	3.1	10.9	0	0.1	0.2
		5	2	1.7	4.2	31	4.4	12.7	1	0.1	0.2
30	231	1	11	42.1	54.6	32	22.9	61.2	5	0.2	0.6
		3	17	51.7	95.2	366	54.5	88.6	4	0.6	0.6
		5	20	61.6	122.1	1093	36.6	190	3	1.0	5.4
40	432	1	36	65.2	676.7	39	188.9	283	1	4.3	5.3
		3	51	66.7	816.3	150	446.3	498	2	2.0	9.8
		5	25	72.8	736.5	291	447.1	628	5	3.9	11.2
avg			**18.2**	**40.6**	**279.3**	**224.6**	**133.9**	**197.6**	**2.3**	**1.0**	**3.7**
stdev			**17.1**	**30.5**	**352.1**	**351**	**186.5**	**229.1**	**2.0**	**1.5**	**4.4**

It is evident from the computations displayed in Table 3 that (**SEP-BQC**) reformulation leads to a much better computational performance compared to the partial or full McCormick linearizations. As the separation problem is called many times during the solution of the probabilistic network design problem, shorter separation times can make a significant impact on the overall solution times. Therefore, we use formulation (**SEP-BQC**) for the separation for the correlated case.

4 Computations

In this section we present our computational experiments for solving the network design problem with probabilistic capacities. We compare the impact of the strengthening methods discussed in Section 2 in the computational performance. We utilize the solution approaches for the separation problems discussed in Section 3. For the computational experiments we use the MIP solver of `CPLEX` version 12.6 that solves conic quadratic relaxations at the nodes of the branch-and-bound tree. `CPLEX` heuristics are turned off, and a single thread is used. The MIP search strategy is set to the traditional branch-and-bound method as it is not possible to add users cuts in `CPLEX` with the dynamic search strategy. Since the solution approaches are significantly different for the independent and correlated cases, we address them separately. For the case of independent capacities the time limit and the memory limit are set to 1800 secs. and 500 MB, respectively, whereas for the harder correlated case, the limits are 3600 secs. and 1 GB. All experiments are performed on a 2.93GHz Pentium Linux workstation with 8GB main memory.

We report the results of the computational experiments for varying number of nodes (n) and arcs (a), values of Ω, and demand. For each combination, five random networks are generated with mean arc capacities drawn from uniform $[0, 100]$ and σ_i from uniform $[0, \mu_i/\Omega]$. The demand $d = \beta \cdot \phi$, where ϕ is the maximum flow possible through the network for the given value of Ω. So that the networks are not completely dense, we set the probability of having an arc between two nodes as $1/\sqrt{n}$, while ensuring the connectedness of the network. In the case of correlated capacities, for each problem instance, the covariance matrix is generated using MATLAB moler matrix library which generates random symmetric positive definite matrices. The eigenvalues of each generated covariance matrix are then translated by half the spectral radius to reduce the condition number. The mean arc capacities in the correlated case are generated from uniform $[\Omega\, \sigma_i,\, 2\,\Omega\, \sigma_i]$. In Section 4.3, we drop the assumption (A.1) on the coefficient of variation and the mean capacities are generated from uniform $[0,\, 2\,\Omega\, \sigma_i]$. The data set is available for download at `http://ieor.berkeley.edu/∼atamturk/data/` .

4.1 Independent arc capacities

In Table 4 we compare the root relaxation gap (*rgap*), the numbers of cuts generated (*cuts*), the number of nodes explored *(nodes)*, the CPU time in seconds (*time*) and the number of instances solved to optimality (`#`) with four cut generation options. The *rgap* is computed as $(z_r - z_o)/z_o$, where z_r denotes the objective value at the root node and z_o denotes the optimal objective value. If none of the algorithms solve a given instance to optimality within the computation limits, then z_o is the objective value of the best found solution across all algorithms. Furthermore, if an algorithm is unable to solve all of the five instances, we provide the end gap (*egap*), which is computed as $(z_e - z_o)/z_o$, where z_e is the best objective value found with the algorithm.

The columns under the heading 'conic cuts' show the performance with the `CPLEX` barrier algorithm when the conic quadratic constraints (2) added to the formulation through the separation routine. We compare it with two linearizations: the first one is the CPLEX outer approximation, where CPLEX automatically generates gradients cuts and solves linear programs iteratively instead of conic quadratic programs with a barrier algorithm; whereas the second one is our implementation of the gradient cuts. Both linearizations perform better than using the barrier algorithm. CPLEX generates additional cutting planes to these linear formulations to improve the relaxations. Note that our implementation of the gradient cuts is more aggressive and generates significantly more cuts and reduces the root gaps and the number of nodes substantially; however, it is not as efficient as the native CPLEX implementation and takes longer time. Therefore, for the rest of the computations, we use the CPLEX outer approximation. The last set of columns under the heading 'extended packs' show the performance with the addition of extended pack inequalities [3]. The use of extended pack cuts strengthens the formulations substantially by reducing the average root gap to 12% and leads to much shorter solution times.

4.2 Correlated arc capacities

In Table 5 we compare the performance with conic cuts, CPLEX outer aproximation as well as with adding the extended extended polymatroid cuts, and the aggregated lifted cover cuts to the formulations. As in the independent case, solving iterative linear programs via CPLEX outer approximation is substantially faster compared to solving conic the quadratic programs with a barrier algorithm in the search tree. The user cuts are generated throughout the search tree using the respective separation algorithms on top of the CPLEX outer approximation. Table 5 clearly shows the positive impact of strengthening the formulations with the extended extended polymatroid cuts and the aggregated lifted cover cuts. All instances are solved to optimality with the addition of user cuts and the average solution time across all instances reduces from 1587 seconds to merely 4 seconds.

4.3 Relaxing the assumption on the coefficient of variation

Throughout the paper, we assumed an upper bound on the coefficient of variation (A.1). While assumption (A.1) is a sufficient condition leading to the submodularity of the set functions used in the analysis, we show in this section, that it is possible to drop this assumption and still utilize the results where applicable.

Recall from Section 2.2 that if (A.1) holds, the coefficients β_{ij} of the function

$$q(\mathbf{x}) = \sum_i \alpha_i x_i + 2\sum_i \sum_{j>i} \beta_{ij} x_i x_j + d^2,$$

are non-positive, and consequently, q is submodular. We can drop this assumption by using the McCormick linearization for pairs (i, j) where $\beta_{ij} > 0$. Introducing $z_{ij} = x_i x_j$ for such pairs only, consider the function

$$\tilde{q}(\mathbf{x}, \mathbf{z}) = \sum_i \alpha_i x_i + 2 \sum_{ij:j>i,\beta_{ij}\le 0} \beta_{ij} x_i x_j + 2 \sum_{ij:j>i,\beta_{ij}>0} \beta_{ij} z_{ij} + d^2.$$

As $\tilde{q}$ is submodular, we can employ the results using $\tilde{q}$ instead of q.

In Table 6 we present the computational results for the general probabilistic network design instances without the assumption (A.1). In this data set, 29% of the coefficients β_{ij} are positive. The results are largely consistent with Table 5 with slight degradation in the performance. The cutting planes added reduce the average solution time from 2334 seconds to merely 10 seconds and confirms their effectiveness for the general case as well.

Acknowledgement

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

References

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

[2] A. Atamtürk. Cover and pack inequalities for (mixed) integer programming. *Annals of Operations Research*, 139:21–38, 2005.

[3] A. Atamtürk and A. Bhardwaj. Supermodular covering knapsack polytope. *Discrete Optimization*, 18:74–86, 2015.

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

[5] A. Atamtürk and V. Narayanan. The submodular knapsack polytope. *Discrete Optimization*, 6:333–344, 2009.

[6] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.
[7] A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming*, 92:453–480, 2002.
[8] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM J. on Optimization*, 15:780–804, 2005.
[9] A. Chen, H. Yang, H. K. Lo, and W. H. Tang. Capacity reliability of a road network: an assessment methodology and numerical results. *Transportation Research Part B: Methodological*, 36:225–252, 2002.
[10] A. Chen, Z. Zhou, P. Chootinan, S. Ryu, C. Yang, and S. Wong. Transport network design problem under uncertainty: a review and new developments. *Transport Reviews*, 31:743–768, 2011.
[11] G. Claßen, D. Coudert, A. M. Koster, and N. Nepomuceno. A chance-constrained model and cutting planes for fixed broadband wireless networks. In *Network Optimization*, pages 37–42. Springer, 2011.
[12] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial OptimizationEureka, You Shrink!*, pages 11–26. Springer, 2003.
[13] B. Fortz and M. Poss. Easy distributions for combinatorial optimization problems with probabilistic constraints. *Operations Research Letters*, 38:545–549, 2010.
[14] L. E. Ghaoui, M. Oks, and F. Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51:543–556, 2003.
[15] J. L. Kennington and C. D. Nicholson. The uncapacitated time-space fixed-charge network flow problem: An empirical investigation of procedures for arc capacity assignment. *INFORMS Journal on Computing*, 22:326–337, 2010.
[16] H. K. Lo and Y.-K. Tung. Network with degradable links: capacity analysis and design. *Transportation Research Part B: Methodological*, 37:345–363, 2003.
[17] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I : Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.
[18] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14: 265–294, 1978.
[19] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, Berlin, 2003.
[20] Y. Song, J. R. Luedtke, and S. Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing, volume = 26, year = 2014, pages = 735-747,*.
[21] N. Yang and F. Wen. A chance constrained programming approach to transmission system expansion planning. *Electric Power Systems Research*, 75(2): 171–177, 2005.

TABLE 4. Network design: independent case.

				Conic cuts					CPLEX outer approximation					User outer approximation					Extended packs				
nodes	arcs	β	Ω	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]
			1	66.2	3	123	1	5	66.2	3	82	1	5	60.42	87	24	76	5	0	7	0	4	5
		0.3	3	68.5	7	394	1	5	68.5	7	226	3	5	64.9	89	21	71	5	12.0	11	1	6	5
			5	60.4	4	349	1	5	60.4	4	163	2	5	53.4	106	19	80	5	1.3	9	0	3	5
			1	54.4	5	264	1	5	49.0	5	158	0	5	46.4	77	20	62	5	20.5	9	1	5	5
10	18	0.5	3	59.7	6	408	3	5	59.7	6	161	4	5	51.5	66	19	60	5	0.2	11	0	6	5
			5	44	8	1817	8	5	44	8	1187	2	5	39.0	125	31	94	5	2.3	10	1	7	5
			1	42.2	6	386	1	5	40.9	6	278	2	5	38.9	82	18	68	5	3.0	11	1	4	5
		0.7	3	39.0	3	394	1	5	41.4	3	206	0	5	32.6	134	27	101	5	9.6	9	3	3	5
			5	56.9	11	1251	6	5	53.1	11	876	5	5	52.3	113	23	88	5	0.8	12	1	4	5
			1	70.8	21	28146	628	4 [26.1]	68.7	34	16424	16	5	60.9	836	81	398	5	8.0	102	6	8	5
		0.3	3	68.5	27	19620	591	5	68.5	29	3570	5	5	61.9	562	61	285	5	1.5	46	2	12	5
			5	67.3	23	19950	773	3 [25.3]	67.2	28	3236	6	5	51.6	469	42	237	5	7.2	60	3	12	5
			1	56.5	16	17876	639	4 [19.3]	52.0	32	22242	26	5	44.3	815	97	368	5	9.2	164	13	36	5
20	54	0.5	3	61.1	26	29287	887	3 [23.2]	56.9	53	51336	54	5	53.1	763	93	315	5	11.1	141	17	20	5
			5	59.5	22	21371	760	4 [18.6]	56.0	40	25241	25	5	50.5	1124	120	490	5	13.3	298	30	28	5
			1	51.8	21	38152	769	3 [22.1]	47.5	43	49404	49	5	41.7	801	122	416	5	8.7	150	13	27	5
		0.7	3	51.4	24	33386	761	4 [20.2]	46.7	36	32124	22	5	45.1	673	102	326	5	7.6	125	12	26	5
			5	51.3	22	53640	762	3 [20.8]	44.6	46	47598	40	5	38.2	579	93	300	5	9.3	80	6	10	5
			1	64.7	18	46877	1450	2 [25.5]	58.63	122	297893	624	4 [19.7]	42.0	2558	127	1729	1 [15.2]	14.5	819	46	66	5
		0.3	3	66.6	17	46226	1440	2 [27.2]	60.3	162	457997	936	3 [20.3]	48.3	1938	119	1323	2 [15.9]	14.8	1230	69	97	5
			5	68.3	16	49432	1473	1 [31.2]	59.7	138	586878	1118	2 [20.8	52.4	2391	124	1629	1 [16.4]	21.7	737	47	53	5
			1	58.7	18	87071	1800	1 [27.6]	48.1	142	898215	1222	4 [18.7]	37.8	2526	138	1602	2 [10.7]	20.5	1458	89	131	5
40	153	0.5	3	62.3	17	82869	1800	1 [29.2]	54.5	179	1008149	1603	3 [19.1]	44.7	2608	143	1740	1 [15.1]	27.9	1485	169	177	5
			5	61.2	16	59570	1800	0 [29.8]	51.3	178	906390	1581	2 [20.9]	41.9	2845	189	1946	1 [14.8]	24.4	2155	247	424	4 [4.3]
			1	52.6	16	64761	1800	2 [25.6]	42.2	148	806376	1184	4 [17.1]	36.2	2378	129	1639	1 [11.4]	27.1	2357	227	428	4 [5.2]
		0.7	3	57.8	16	69720	1800	1 [27.2]	42.7	121	998075	1125	3 [18.0]	35.6	2169	121	1398	2 [10.5]	26.4	2304	204	436	4 [4.5]
			5	55.7	15	83062	1800	0 [30.3]	45.1	152	772617	1023	3 [18.6]	35.31	2281	175	1515	3 [10.5]	24.9	2201	254	432	4 [3.8]
			Avg	**58.4**	**15**	**31719**	**806**		**53.8**	**64**	**258782**	**396**		**46.7**	**1081**	**84**	**670**		**12.1**	**593**	**54**	**91**	
			Stdev	**8.3**	**7**	**29239**	**710**		**9.1**	**64**	**378896**	**574**		**8.9**	**1011**	**53**	**691**		**8.9**	**835**	**85**	**150**	

TABLE 5. Network design: correlated case.

				Conic cuts					CPLEX outer approximation					Extended polymatroid cuts					Extended polymatroid & aggregated lifted cover cuts					
nodes	arcs	β	Ω	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	covers	nodes	time	# [egap]
			1	84.1	6	474	2.2	5	84.1	6	232	0.1	5	80.4	18	67	0.1	5	64.2	16	9	33	0.1	5
		0.3	3	79.1	6	675	3.4	5	79.1	5	280	0.1	5	77.4	16	71	0.1	5	61.2	14	10	27	0.1	5
			5	82.7	4	616	1.7	5	82.7	4	202	0.1	5	81.4	33	89	0.1	5	62.6	20	9	37	0.1	5
			1	72.8	7	468	2.6	5	72.8	7	314	0.1	5	66.1	43	104	0.1	5	50.2	39	17	58	0.1	5
10	21	0.5	3	79.1	7	1314	5.5	5	79.1	7	430	0.1	5	74.2	30	119	0.3	5	59	30	16	62	0.1	5
			5	73.9	8	901	10.5	5	73.9	8	427	0.2	5	70.2	54	174	0.2	5	48	39	21	70	0.2	5
			1	68.3	7	1523	7.6	5	68.3	7	721	0.2	5	66.8	44	190	0.22	5	45.3	56	25	84	0.2	5
		0.7	3	80	11	2670	20.4	5	80	11	855	0.3	5	73.5	53	204	0.4	5	52.6	52	26	98	0.2	5
			5	76.1	12	5614	97.8	5	76.1	12	2125	0.8	5	70.4	98	359	0.3	5	49.3	101	42	141	0.2	5
			1	73.1	29	25429	1327.9	4 [24]	72.2	11	5671	3.5	5	72	279	1834	1.3	5	55	208	141	248	0.8	5
		0.3	3	73.8	16	7690	238.7	5	73.8	16	3837	2.2	5	72	164	792	0.7	5	55.4	154	121	256	0.6	5
			5	72.4	10	4283	81.5	5	72.4	10	2784	1.2	5	70.7	118	395	0.5	5	48.3	90	62	155	0.4	5
			1	67.3	51	59031	2176.4	2 [9.9]	69.5	29	41528	33.7	5	65.3	423	4758	3.2	5	46	466	200	1132	2.5	5
20	54	0.5	3	70.4	41	36523	1712.4	4 [4.1]	70.5	31	14067	8.4	5	66.9	418	1793	1.7	5	49.4	344	167	531	1.5	5
			5	68.6	84	34812	1583.5	3 [6.6]	70.4	50	129362	727.3	4 [11.2]	68.5	258	1074	1	5	50.6	190	120	310	0.8	5
			1	60.7	25	38103	1254.5	4 [13.3]	62.2	33	46481	32.5	5	62.1	719	8237	10.8	5	43.4	397	197	1270	2.6	5
		0.7	3	67.3	66	102159	3600	0 [9.7]	68.4	44	28176	19.6	5	65.5	352	2192	2.1	5	46.3	238	134	647	1.2	5
			5	67	48	47892	1698.7	4 [14.9]	67.7	38	24963	24.3	5	65.7	346	2658	1.9	5	47	393	224	828	2.4	5
			1	68.1	46	32337	2854.5	2 [18.4]	71.3	68	56284	171.4	5	70.6	231	2304	1.7	5	51	387	222	672	2.9	5
		0.3	3	66.6	54	28538	2586.2	2 [23.1]	72.2	108	137584	844.8	4	67.9	507	5620	5.2	5	50	397	246	814	3.2	5
			5	67.6	100	33951	2735.2	2 [21.2]	71.6	87	77529	385.8	5	69.3	305	2582	2.2	5	49.4	191	139	460	1.5	5
			1	57.6	90	109833	3600	0 [18.8]	65.7	106	541575	2219.5	3 [4.0]	64.1	1244	24314	24.2	5	42	937	566	2596	9.5	5
30	101	0.5	3	59.6	21	65734	2850.7	2 [16.7]	64.3	83	330311	1612.2	4 [8.1]	60.9	1297	14424	15.9	5	42.8	1155	721	2999	14.7	5
			5	60.8	40	133861	3600	0 [12.3]	65.2	98	262583	1123.6	5	61.5	963	5186	5.5	5	39.8	866	479	1242	6.3	5
			1	44.4	67	83056	3600	0 [30.7]	59.3	108	644090	2892.9	2 [9.2]	61.5	1790	51076	63.1	5	36.6	1686	962	4759	28.8	5
		0.7	3	56.9	63	90960	3600	0 [10.8]	62.7	63	540993	1811.8	3 [22.4]	62.2	1977	49073	81.9	5	42	2168	1118	3411	27	5
			5	51	83	76914	3600	0 [13.5]	56.6	91	314039	984.8	4 [6.1]	57	1175	13124	15.7	5	41.2	1379	630	2022	11.3	5
			avg	**68.5**	**37**	**37976**	**1587**		**70.8**	**42**	**118794**	**478**		**68.3**	**480**	**7141**	**9**		**49.2**	**445**	**245**	**925**	**4**	
			stdev	**9.5**	**30**	**39441**	**1476**		**6.8**	**38**	**190837**	**801**		**5.9**	**560**	**13578**	**19**		**5.8**	**563**	**302**	**1216**	**8**	

TABLE 6. Network design: the general case.

				Conic Cuts					CPLEX Outer Approximation					Extended Polymatroids					Extended Polymatroids with Aggregated Covers					
nodes	arcs	β	Ω	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	nodes	time	# [egap]	rgap	cuts	covers	nodes	time	# [egap]
			1	59.2	13	413	4.7	5	59.5	13	276	0.1	5	59	11	56	0	5	43.4	8	4	22	0	5
		0.3	3	55.9	11	866	52.7	5	57.5	11	204	0.1	5	56.2	13	98	0.1	5	40.2	17	10	41	0	5
			5	52.8	9	363	16.1	5	54.2	9	164	0.1	5	53.4	52	47	0.1	5	37.4	63	29	19	0	5
			1	57.3	7	631	28.1	5	57.5	7	60	0.1	5	57.5	19	68	0.1	5	39.7	22	9	33	0	5
10	26	0.5	3	51.4	10	933	78.3	5	52.6	10	168	0.1	5	52.1	29	103	0.1	5	33.7	26	13	66	0.1	5
			5	48.2	8	854	43.5	5	49.1	8	95	0	5	48.8	32	91	0.1	5	31.1	38	20	48	0.1	5
			1	53.8	6	816	51.6	5	54.1	6	171	0.1	5	53.6	36	99	0.1	5	37.2	30	15	39	0	5
		0.7	3	48.9	9	485	14.6	5	48.8	9	337	0.7	5	49.2	60	65	0.1	5	29.7	78	42	36	0	5
			5	44.6	7	1432	128.4	5	44.1	7	300	0.5	5	44	37	161	0.2	5	25.2	51	25	71	0.1	5
			1	47.4	67	55692	2582	3 [2.5]	48.6	80	17546	13.3	5	47.6	577	6453	5.9	5	32	692	350	1021	3.8	5
		0.3	3	43.5	60	111141	3287	2 [14.2]	44.3	65	12956	10.2	5	44.1	682	11769	16.1	5	24.5	697	325	1497	4.7	5
			5	47.2	81	68984	2803	3 [8.0]	47	67	8234	7.5	5	47.5	749	7314	6.2	5	30.3	616	275	937	3.9	5
			1	45.6	80	101183	3527	1 [12.7]	45.6	54	159690	1251.3	3 [7.5]	45.3	310	18110	19.2	5	25.7	250	136	1541	5.0	5
20	56	0.5	3	44.2	62	141726	3600	0 [19.9]	44.1	58	196375	2101.1	3 [16.8]	44.2	259	23994	24.1	5	28.3	373	174	2313	8.6	5
			5	49.8	72	107933	3600	0 [28]	50	46	186240	1874.2	3 [16.3]	49.8	753	20107	20.8	5	32.9	850	458	2579	9.47	5
			1	58.3	56	124710	3600	0 [43.3]	58.6	71	156478	1441.6	3 [39.2]	58.8	725	20302	22.4	5	40.7	482	274	1875	5.84	5
		0.7	3	47.1	38	104174	3600	0 [16.3]	48.2	83	24724	15.2	5	48	569	11657	15.07	5	28.4	606	305	1654	5.8	5
			5	49.6	80	126946	3600	0 [19.9]	50.9	40	11811	10.1	5	50.7	640	13289	18.2	5	33.3	818	342	997	4.1	5
			1	68.9	109	106396	3600	0 [43.7]	68.6	188	578013	3496.2	1 [9.7]	68.3	3346	39550	42.7	5	49.6	3180	1652	4101	21.7	5
		0.3	3	70.1	124	190983	3600	0 [50.5]	70.8	157	483580	3284.1	2 [4.5]	70	3220	43285	49.7	5	50.1	3068	1353	4638	26.3	5
			5	65.3	141	174411	3600	0 [26.7]	67.1	142	532954	3600	0 [25.1]	66.3	2265	44096	39.3	5	48.4	2292	1373	3855	19.7	5
			1	89.7	146	73266	3600	0 [81]	89	154	591382	3600	0 [50.6]	89.3	3145	36896	84.9	5	74	2470	1013	3192	15.7	5
30	134	0.5	3	68.2	118	180719	3600	0 [38.1]	68.9	144	637194	3600	0 [24.8]	69	2767	49934	80.3	5	50.1	3152	1526	5215	33.9	5
			5	65.5	136	184461	3600	0 [19.5]	66.2	121	581235	3600	0 [31.1]	66.2	2531	47512	77.2	5	49.3	1535	858	4827	29.9	5
			1	72.3	124	110874	3600	0 [45.7]	72.4	145	628966	3600	0 [44.3]	72.1	2243	42543	66.7	5	52.1	2800	1204	3977	20.9	5
		0.7	3	63.3	97	169244	3600	0 [24.5]	63	110	514648	3600	0 [15.5]	63.1	2240	42657	69.1	5	48	2933	1302	4022	21.8	5
			5	87.8	110	186748	3600	0 [70.5]	87.7	187	513009	3600	0 [33.4]	88.1	2820	44332	91.8	5	48.9	1627	661	4691	29.6	5
			avg	**57.6**	**66**	**86162**	**2334**		**58.1**	**74**	**216178**	**1434**		**57.9**	**1116**	**19429**	**28**		**39.4**	**1066**	**509**	**1974**	**10**	
			stdev	**12.6**	**49**	**70804**	**1666**		**12.4**	**61**	**258096**	**1639**		**12.4**	**1215**	**18773**	**31**		**11.3**	**1163**	**554**	**1844**	**11**	

Appendix A. Simulation Details

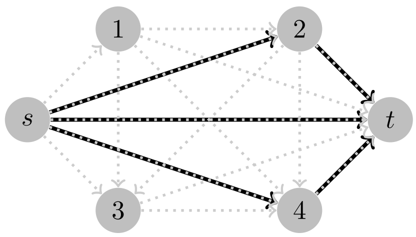
A.1 Network data

For the example in Section 1, we use a six-node acyclic complete network with demand 230 s to t is 230. Mean capacities, variances and costs for the arcs are presented in Table 7.

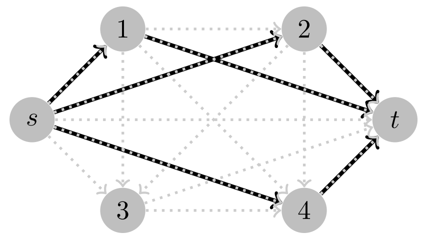
TABLE 7. Candidate arcs and data.

edge#	from	to	mean capacity	variance	cost
1	s	1	81	16	14
2	s	2	90	676	42
3	s	3	12	4	91
4	s	4	91	289	79
5	s	t	63	9	95
6	1	2	9	4	65
7	1	3	27	25	3
8	1	4	54	36	84
9	1	t	95	256	93
10	2	3	96	169	67
11	2	4	15	1	75
12	2	t	97	64	74
13	3	4	95	16	39
14	3	t	48	9	65
15	4	t	80	49	17

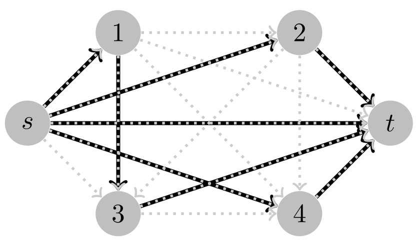
A.2 Minimum cost network configurations for different service levels



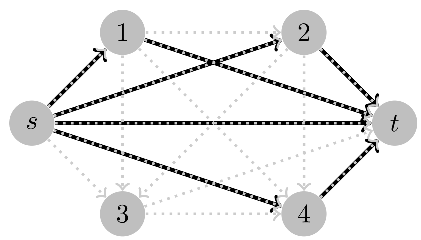
(A) service level 50%



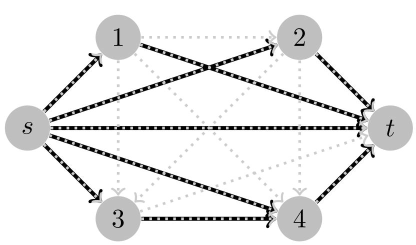
(B) service level 70%



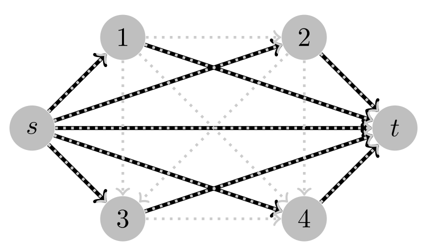
(C) service level 80%



(D) service level 97.5%



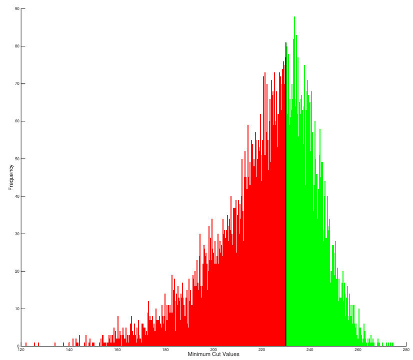
(E) service level 99%



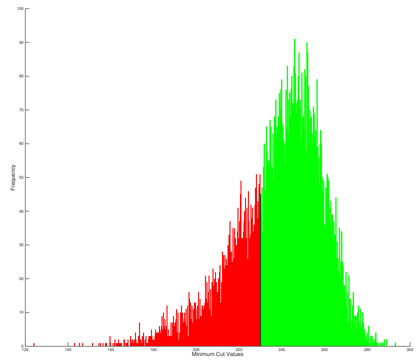
(F) service level 99.9%

FIGURE 3. Network configurations corresponding to different service levels.

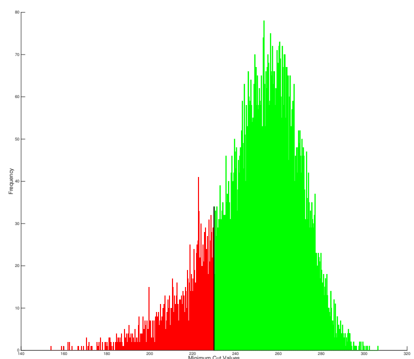
A.3 Distribution of minimum cut capacities for the simulated instances



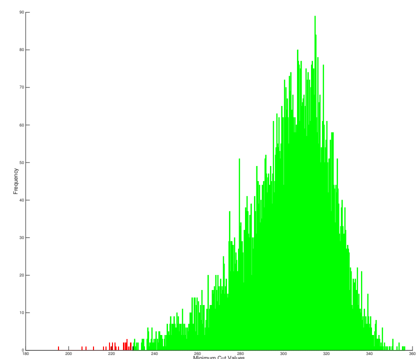
(A) $1 - \epsilon = 50\%$



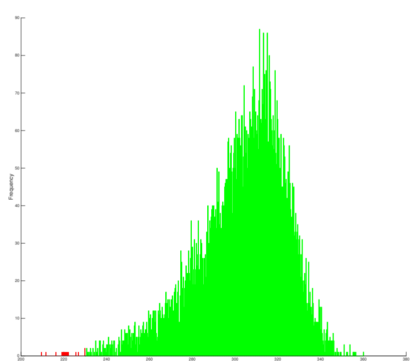
(B) $1 - \epsilon = 70\%$



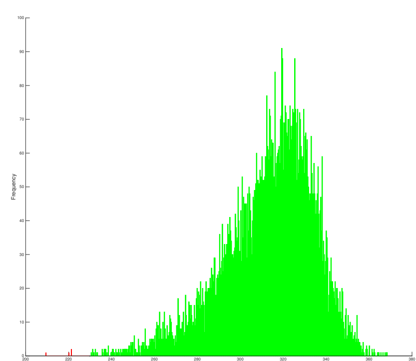
(C) $1 - \epsilon = 80\%$



(D) $1 - \epsilon = 97.5\%$



(E) $1 - \epsilon = 99\%$



(F) $1 - \epsilon = 99.9\%$

FIGURE 4. Minimum cut capacities for simulated instances.