Branch-and-cut Approaches for Chance-constrained Formulations of Reliable Network Design Problems

Yongjia Song
James R. Luedtke *

August 9, 2012

Abstract

We study solution approaches for the design of reliably connected networks. Specifically, given a network with arcs that may fail at random, the goal is to select a minimum cost subset of arcs such the probability that a connectivity requirement is satisfied is at least $1-\epsilon$, where ϵ is a risk tolerance. We consider two types of connectivity requirements. We first study the problem of requiring an s-t path to exist with high probability in a directed graph. Then we consider undirected graphs, where we require the graph to be fully connected with high probability. We model each problem as a stochastic integer program with a joint chance constraint, and present two formulations that can be solved by a branch-and-cut algorithm. The first formulation uses binary variables to represent whether or not the connectivity requirement is satisfied in each scenario of arc failures and is based on inequalities derived from graph cuts in individual scenarios. We derive additional valid inequalities for this formulation and study their facet-inducing properties. The second formulation is based on probabilistic graph cuts, an extension of graph cuts to graphs with random arc failures. Inequalities corresponding to probabilistic graph cuts are sufficient to define the set of feasible solutions and can be separated efficiently at integer solutions, allowing this formulation to be solved by a branch-and-cut algorithm. Computational results demonstrate that the approaches can effectively solve instances on large graphs with many failure scenarios. In addition, we demonstrate that, by varying the risk tolerance, our model yields a rich set of solutions on the efficient frontier of cost and reliability.

1 Introduction

We study solution methods for the design of reliably connected networks. Specifically, given a graph $G = (V, A)$ with arcs that may fail at random, the goal is to select a minimum cost subset of arcs such that a connectivity requirement is satisfied with probability at least $1 - \epsilon$. We focus on two types of connectivity requirements. For directed graphs, we consider s-t connectivity, which requires that there exists an s-t path in the graph. For undirected graphs, we consider full connectivity, which requires all nodes in the graph to be connected. The chance-constrained model we use provides a probabilistic alternative to existing deterministic models for the design of reliably connected networks, such as the the survivable network design model of Grötschel, Monma, and Stoer [20].

We model each of these problems as a stochastic integer program with a joint chance constraint (sometimes referred to as a probabilistic constraint). We show that these problems are both $\mathcal{NP}$-hard, even when all failure scenarios are equally likely. We first study a formulation that introduces

*Department of Industrial and Systems Engineering, University of Wisconsin, Madison, Wisconsin 53706 (jrluedt1@wisc.edu).

binary variables to model whether or not the connectivity constraint is satisfied in each failure scenario. We describe methods for separating the valid inequalities required to make the formulation valid, enabling it to be solved by a branch-and-cut algorithm. We also describe new classes of valid inequalities and study their facet-inducing properties. We next consider an alternative formulation that uses fewer binary decision variables. This formulation is based on a class of inequalities derived from *probabilistic graph cuts*, a probabilistic extension of graph cuts in deterministic graphs to graphs with random arc failures. This exponential class of inequalities can be separated efficiently for integral solutions, again allowing this formulation to be solved by a branch-and-cut algorithm. An advantage of this approach is that the number of variables in the formulation is independent of the number of scenarios used to model the random arc failures.

A related model has been studied by Beraldi et al. [9], in which design of a network that is fully connected with high probability is considered. The approach in [9] is based on an extended formulation that introduces a set of variables and constraints for every failure scenario, making this formulation impractical for instances with large graphs or many failure scenarios. While [9] provides heuristic solution approaches that have reasonable computational time, the exact solution methods based on this extended formulation require thousands of hours for their test instances. We show that our approach is able to solve similar instances, but having many more failure scenarios, in less than two hours. A similar problem has also been studied by Fischetti and Monaci [19], where they assume that arc failures are independent and separate reliability constraints are imposed on each connected component of the graph. In contrast, we directly require that the graph be connected with high probability, and do not assume that arc failures are independent. Andreas and Smith [2] study an extension of the two arc-disjoint paths model to a probabilistic setting. They also assume that arcs fail independently, and the solution approach depends on this assumption. Our model does not use the two arc-disjoint paths constraint, and instead uses the chance constraint alone to ensure that solutions of sufficient reliability are constructed. Sorokin et al. [40] study a fixed charge network flow problem with random arc failures. Their model uses conditional value-at-risk as a risk measure to control the flow losses. We do not consider flow volumes, and instead focus on design of networks that are reliably connected.

In general, chance-constrained stochastic programs are hard to solve [33], although promising results have been obtained for some special cases. Classically, Charnes et al. [14, 15] propose a convex deterministic reformulation of a single-row chance constraint with joint normal random coefficients. Another special case that has received considerable study is a joint chance constraint where only the right-hand side is random, that is, the chance constraint takes the form $\mathbb{P}\{Tx \geq \xi\} \geq 1-\epsilon$ where ξ is a random vector and T is deterministic. If ξ has a continuous and log-concave distribution, this constraint defines a convex set and hence nonlinear programming techniques can be applied [35, 36]. When ξ is a discrete random variable, approaches based on certain non-dominated points (known as p-efficient points) of the distribution have been studied [10, 18, 38]. If ξ further is assumed to have finite support, an approach based on strengthening the corresponding mixed-integer programming formulation using inequalities from [5, 21] has been successful [23, 28]. Methods for more general chance-constrained problems, in which either the constraint matrix T may be random, or recourse decisions can be made to satisfy a set of constraints after observing the random outcomes, have been studied in [8, 25, 26, 37, 41]. Our approach is most similar to [25, 26] in that it allows decomposition by scenario, but exploits the graph structure to obtain an alternative formulation and new classes of valid inequalities.

To further motivate the model we use, we compare it in more detail to the survivable network design (SND) model [20]. (See [29] for a more detailed survey of deterministic models and formulations for reliable network design problems.) In particular, for the case of s-t connectedness, an SND model would require the selected network to include k arc-disjoint paths between s and t,

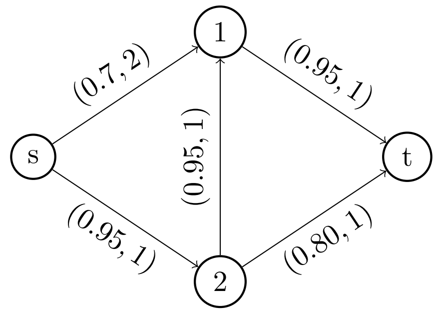

Figure 1: An example graph with known arc failure probabilities and costs.

for some integer $k \geq 1$. A potential drawback of this approach is that it cannot take advantage of information on the probability distribution of arc failures, if this information is available. As an example, consider the graph depicted in Figure 1. The labels (α, β) on each arc represent the probability α that the arc is available and the cost β. In this example the arcs are assumed to fail independently. Using the SND model, there are two possible optimal solutions, corresponding to whether we set $k = 1$, or $k = 2$.

- For $k = 1$, the optimal solution to the SND model is the minimum cost path: $(s, 2)$ and $(2, t)$. The corresponding cost is 2, and the reliability is 0.76.
- For $k = 2$, the optimal solution to the SND model is: $\{(s, 1), (s, 2), (1, t), (2, t)\}$. The corresponding cost is 5, and the reliability is 0.9196.

In the chance-constrained model that we study, we choose a minimum cost set of arcs subject to a constraint that the probability that a path exists from s to t using these arcs is at least $1-\epsilon$. Here, varying the minimum reliability $1-\epsilon$ allows the decision-maker to explore the trade-off between cost and reliability. Table 1 shows the optimal solutions of the chance-constrained model for different values of $1-\epsilon$. Each row presents the minimum specified reliability $1-\epsilon$, the optimal set of arcs, the cost of these arcs, and the actual reliability of these arcs (which may be higher than the minimum). Of these, only the shortest path solution, obtained with $1-\epsilon = 0.7$, can be obtained as an optimal solution of the SND model. Also, the solution obtained using $1-\epsilon = 0.9$, which has actual reliability 0.9315, dominates the SND model solution with $k = 2$ in terms of both reliability and cost. Thus, we see that, by varying the target reliability, the chance-constrained model may yield a richer set of possible solutions than a deterministic model.

Outline. In Section 2 we study a chance-constrained s-t connected network design problem in a directed graph, give the problem description and an extended mixed-integer programming (MIP)

Minimum Reliability $(1-\epsilon)$	Optimal set of arcs	Cost	Actual Reliability
0.7	$(s, 2), (2, t)$	2	0.76
0.8	$(s, 2), (2, 1), (1, t)$	3	0.857
0.9	$(s, 2), (2, 1), (2, t), (1, t)$	4	0.9315
0.95	All arcs	6	0.971

Table 1: Solutions of chance-constrained network design model for graph in Figure 1 for varying values of $1-\epsilon$.

formulation, and show that the problem is $\mathcal{NP}$-hard. We present and study the first formulation in Section 2.1 and the second formulation, based on probabilistic s-t cuts, in Section 2.2. In Section 3 we extend our approaches to the chance-constrained fully connected network design problem. In Section 4 we present computational results.

2 Chance-constrained s-t connected network design

Let $G = (V, A)$ be a directed graph with node set V and arc set A. Each arc $a \in A$ has a nonnegative cost c_a, and the cost of selecting a subset of arcs $Q \subseteq A$ is $c(Q) := \sum_{a \in Q} c_a$. The arcs of the graph fail randomly according to some known distribution. Thus, the set of available arcs is a random subset $\tilde{A}$ of A. Since the total number of possible failure outcomes is finite, without loss of generality we model the random failures with a finite set of scenarios N. Specifically, we assume $\mathbb{P}\{\tilde{A} = A_k\} = p_k > 0$ for $k \in N$, where $A_k \subseteq A$ is the set of available arcs in each scenario k, and $\sum_{k \in N} p_k = 1$. Given a set of scenarios $F \subseteq N$, we let $\mathbb{P}(F) = \sum_{k \in F} p_k$. Let $\tilde{G} = (V, \tilde{A})$ be the random graph having node set V and random set of arcs $\tilde{A}$. For a selected set of arcs Q, we define the graph $G_k(Q)$ to be the graph with node set V and arc set $Q \cap A_k$. For the special case of $Q = A$, we abbreviate $G_k(A)$ by G_k, which represents the graph obtained in scenario k if every arc is selected. For $Q \subseteq A$, let $S^Q = \{k \in N \mid \exists \text{ an } s\text{-}t \text{ path in } G_k(Q)\}$ be the set of "success" scenarios if arc set Q is selected. Then, for $s, t \in V, s \neq t$, the chance-constrained minimum cost s-t connected network design problem is:

$$\min_{Q \subseteq A} \big\{ c(Q) \mid \mathbb{P}(S^Q) \geq 1 - \epsilon \big\}. \tag{1}$$

We assume throughout this paper that $|N|$ is of moderate size (i.e., $O(|A|)$), so that, e.g., it is computationally feasible to evaluate $\mathbb{P}(S^Q)$ for a given Q by simply performing a graph search in each of the $|N|$ scenarios. In general random graphs, the number of arc failure scenarios might grow exponentially in $|A|$. In this case, we assume that we have used Monte Carlo sampling to obtain a sample average approximation (SAA) having a moderate number of scenarios, and that the problem (1) that we wish to solve is this SAA problem. Because the set of feasible solutions of this problem is finite, the results of [27] imply that the probability that such an SAA problem yields an optimal solution to the original problem approaches one exponentially fast with sample size. The SAA problem can also be used to derive statistical confidence intervals on the optimal value of the true problem [32]. Additional results on SAA for chance constraints can be found in [11, 12, 13].

To formulate (1) as an integer program, we introduce binary variables x_a for $a \in A$, where $x_a = 1$ if and only if arc a is selected. For a binary vector $x \in \{0,1\}^{|A|}$, let $A^x = \{a \in A \mid x_a = 1\}$ be the set of arcs induced by x. We also introduce binary variables z_k for $k \in N$, where $z_k = 1$ indicates that the graph $G_k(A^x)$ contains an s-t path. We can then formulate (1) as:

$$\min \sum_{a \in A} c_a x_a \tag{2a}$$

$$\text{subject to} \sum_{k \in N} p_k z_k \geq 1 - \epsilon \tag{2b}$$

$$z_k = 1 \Rightarrow \ G_k(A^x) \text{ contains an } s\text{-}t \text{ path}, \quad \forall\, k \in N \tag{2c}$$

$$x \in \{0,1\}^{|A|}, z \in \{0,1\}^{|N|}. \tag{2d}$$

We assume $\epsilon < 1$ to prevent (2b) from being redundant, and further assume that $p_k \leq \epsilon$ for all

$k \in N$, since otherwise we could fix $z_k = 1$. Finally, we assume that G_k contains an s-t path for all $k \in N$, since otherwise we could fix $z_k = 0$.

One way to formulate the logical constraints (2c) using linear constraints is to introduce second-stage continuous variables $\{y_a^k\}_{a \in A}$ for each scenario $k \in N$, which are used in the following flow balance constraints that enforce the desired graph connectivity. (See [29] for further information on flow formulations for network design under connectivity constraints.)

$$\sum_{a \in \delta_k^+(i)} y_a^k - \sum_{a \in \delta_k^-(i)} y_a^k = \begin{cases} 0, & i \neq s,t \\ z_k, & i = s \\ -z_k, & i = t \end{cases} \qquad \forall k \in N \tag{3a}$$

$$0 \leq y_a^k \leq x_a, \qquad \forall a \in A_k, k \in N. \tag{3b}$$

Here $\delta_k^+(i)$ and $\delta_k^-(i)$ denote, respectively, the sets of exiting and entering arcs to node $i \in V$ for each graph G_k, $k \in N$. For any $x \in \{0,1\}^{|A|}$, if $z_k = 1$, then (3) requires that there exists a flow of one unit from s to t, using only arcs $a \in A^x$, and hence a path from s to t exists in $G_k(A^x)$. If $z_k = 0$, the right-hand side of (3a) becomes 0, so it is feasible to assign $y_a^k = 0$ for all $a \in A_k$, and hence the constraints corresponding to scenario k do not constrain the choice of arcs. Thus, replacing (2c) with (3) in (2) yields a valid MIP formulation of (2). We refer to this MIP formulation as the *extended formulation* of (2).

Before proceeding to our solution approaches, we note that (1) is $\mathcal{NP}$-hard, even in the special case that all scenarios are equally likely.

Theorem 1. *The chance-constrained s-t connected network design problem* (1) *is $\mathcal{NP}$-hard in the special case in which $p_k = 1/|N|$ for all $k \in N$.*

Proof. First observe that (1) is equivalent to formulation (2): for every Q feasible to (1), $\exists\ (x,z)$ feasible to (2) with the same objective value, and conversely, for every (x,z) feasible to (2) there is a Q feasible to (1) with the same objective value. Therefore, it is sufficient to prove the problem formulated in (2) is $\mathcal{NP}$-hard. By letting $q = \lfloor \epsilon |N| \rfloor$, the decision version of (2) can be stated as follows.

> (D-(2)) Given a directed graph $G = (V, A)$, costs $c_a \in \mathbb{Z}_+$ on each arc $a \in A$, a positive integer J, a set of scenarios N with a set of available arcs $A_k \subseteq A$ for each $k \in N$, and a nonnegative integer $q \leq |N|$, is there a solution (x,z) that satisfies (2c), (2d), $\sum_{k \in N} z_k \geq |N| - q$, and $\sum_{a \in A} c_a x_a \leq J$?

Our proof is based on a reduction of the following $\mathcal{NP}$-complete decision problem [28] to D-(2):

> (DPCLP) Given binary numbers $\xi_{kj}, k = 1,2,...,n, j = 1,2,...,m$, and integers $K \leq n, L > 0$, is there a subset $I \subseteq \{1,\ldots,n\}$ such that $|I| \geq K$ and $\sum_{j=1}^m \max_{i \in I}\{\xi_{ij}\} \leq L$?

Now given an instance of (DPCLP): (ξ_{kj}, K, L), we reduce it to an instance of D-(2) as follows.

(a) Let V and A be given by:

$$V = \{s, t, r_1, ..., r_{m+1}, v_1, ..., v_m\}, \qquad A = B \cup D_1 \cup D_2 \cup \{(s, v_1)\} \cup \{(r_{m+1}, t)\},$$

where B is a set of "bypass" arcs, and D_1, D_2 are sets of "direct" arcs:

$$B = \cup_{j=1}^m (r_j, r_{j+1}), \quad D_1 = \cup_{j=1}^m (r_j, v_j), \quad D_2 = \cup_{j=1}^m (v_j, r_{j+1}).$$

(b) The cost c_a on each arc $a \in A$ is 1 if $a \in D_1$, and 0 otherwise.

(c) The set of scenarios is $N = \{1, \ldots, n\}$, and the arc set A_k in scenario k is

$$A_k = D_1 \cup D_2 \cup \{(s, r_1)\} \cup \{(r_{m+1}, t)\} \cup B_k,$$

where $B_k = \{(r_j, r_{j+1}) \mid \xi_{kj} = 0, j = 1, \ldots, m\}$.

(d) $q = n - K$, $J = L$.

See Figure 2 for an example of this construction with $m = 2$, and with the bypass arcs illustrated for some $k \in \{1, \ldots, n\}$ with $\xi_{k1} = 0$ and $\xi_{k2} = 1$ (the dashed line on arc (r_2, r_3) indicates the arc is not present in scenario $k = i$).

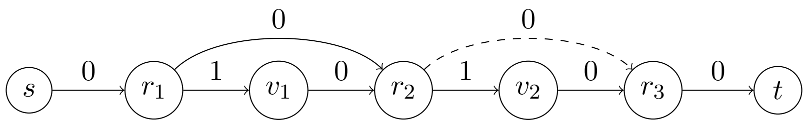

Figure 2: An example of construction of the graph $G_k = (V, A_k)$ in the proof of Theorem 1.

First, suppose the DPCLP instance is "Yes". Then there exists a subset $I \subseteq N$ such that $|I| \geq K$ and $\sum_{j=1}^{m} \max_{k \in I}\{\xi_{kj}\} \leq L$. We claim that D-(2) is also "Yes". Indeed, let $z_k = 1$ for all $k \in I$ and $z_k = 0$ otherwise. Then $\sum_{k \in N} z_k = |I| \geq K = n - q$. Also, we assign the values of x as follows:

- For $j = 1, 2, ..., m$: If $\max_{k \in I}\{\xi_{kj}\} = 0$, let $x_{(r_j, r_{j+1})} = 1$ and $x_{(r_j, v_j)} = x_{(v_j, r_{j+1})} = 0$; If $\max_{k \in I}\{\xi_{kj}\} = 1$, let $x_{(r_j, r_{j+1})} = 0$ and $x_{(r_j, v_j)} = x_{(v_j, r_{j+1})} = 1$. In either case, the contribution to the cost of selected arcs is $\max_{k \in I}\{\xi_{kj}\}$.
- $x_{(s, r_1)} = 1$ and $x_{(r_{m+1}, t)} = 1$.

Clearly (x, z) satisfies (2d). Also, $\sum_{a \in A} c_a x_a = \sum_{j=1}^{m} \max_{k \in I}\{\xi_{kj}\} \leq L = J$. Now consider (2c). In any graph $G_k(A^x)$, node r_1 is connected from s, and node t is connected from r_{m+1}. In addition, for all $k \in I$, node r_{j+1} is connected from node r_j, for all $j = 1, 2, ..., m$. Indeed, if $\max_{k \in I}\{\xi_{kj}\} = 0$, then $x_{(r_j, r_{j+1})} = 1$, that is, we have selected a "bypass" arc (r_j, r_{j+1}) that is in B_k (and hence A_k) for all $k \in I$. On the other hand, if $\max_{k \in I}\{\xi_{kj}\} = 1$, then $x_{(r_j, v_j)} = 1$, and $x_{(v_j, r_{j+1})} = 1$, we have selected the "direct" arcs (r_j, v_j) and (v_j, r_{j+1}) that are available in all scenarios and connect node r_{j+1} from node r_j. Thus, for any k with $z_k = 1$ ($k \in I$), there exists an s-t path in $G_k(A^x)$, and hence D-(2) is "Yes".

Now suppose the constructed D-(2) instance is "Yes". We show that the DPCLP instance is "Yes". Let (x, z) certify that D-(2) is "Yes" and let $I = \{k \in N \mid z_k = 1\}$. By construction, $|I| \geq n - q = K$. For any $k \in I$, there exists an s-t path in $G_k(A^x)$. Thus, for each $j = 1, \ldots, m$, if $x_{(r_j, v_j)} = 0$, then it must hold that $x_{(r_j, r_{j+1})} = 1$ and $\max_{k \in I}\{\xi_{kj}\} = 0$ (otherwise for some $k \in I$ an s-t path would not exist in $G_k(A^x)$). Thus, for each j, $\max_{k \in I}\{\xi_{kj}\} \leq c_{(r_j, r_{j+1})} x_{(r_j, r_{j+1})} + c_{(r_j, v_j)} x_{(r_j, v_j)} + c_{(v_j, r_{j+1})} x_{(v_j, r_{j+1})}$ and hence $\sum_{j=1}^{m} \max_{k \in I}\{\xi_{kj}\} \leq \sum_{a \in A} c_a x_a \leq J = L$, showing that indeed the DPCLP instance is "Yes". $\square$

2.1 A formulation based on scenario-based s-t cuts

The extended formulation with explicit flow constraints (3) may be very large when the sample size, $|N|$, is large, as the second stage recourse variables y have $|N|$ different copies, each of which must satisfy a set of flow balance constraints. To avoid introducing these recourse variables and constraints explicitly, we seek a decomposition approach that implicitly enforces (3) using cutting planes, while working with a formulation that uses only the x and z variables.

Recall that an s-t cut in graph $G = (V, A)$ is a set of arcs $C \subseteq A$ that intersects every s-t path in G. For each scenario $k \in N$, let $\mathcal{C}^k$ be the set of s-t cuts in G_k. To emphasize that these are cuts derived from the deterministic graphs associated with individual scenarios, we call the s-t cuts from scenario graphs G_k, $k \in N$, *scenario-based s-t cuts*. Applying the max-flow min-cut theorem to the extended formulation defined by (3) leads to the following integer programming (IP) formulation for (2):

$$\min \sum_{a \in A} c_a x_a \tag{4a}$$

$$\text{subject to} \sum_{k \in N} p_k z_k \geq 1 - \epsilon, \tag{4b}$$

$$\sum_{a \in C} x_a \geq z_k, \qquad \forall C \in \mathcal{C}^k,\ k \in N, \tag{4c}$$

$$x \in \{0,1\}^{|A|}, z \in \{0,1\}^{|N|}. \tag{4d}$$

Proposition 2. (4) *is a valid formulation for* (2).

Proof. It is sufficient to show that the constraints (4c) are equivalent to (3), since we have already argued that (3) provides a valid formulation of (2c). Observe that the constraints (3) enforce that there exist an s-t flow of value z_k in graph G_k, where the capacities on the arcs in $a \in A_k$ are given by x_a. By the max-flow min-cut theorem, such a flow exists if and only if the capacity of every s-t cut in G_k is at least z_k, which is the condition enforced in (4c). □

We can relax the integrality on z variables in (4) and still have a valid formulation. Our computational experience is that this relaxation improves the performance of the branch-and-cut algorithm based on (4).

Corollary 3. *Consider problem* (4) *with z relaxed to $z \in [0,1]^{|N|}$. There exists an optimal solution to this relaxed problem with $z \in \{0,1\}^{|N|}$, i.e., it is optimal for* (4).

Proof. Consider an optimal solution for the relaxed problem with $z \notin \{0,1\}^{|N|}$, i.e., $0 < z_k < 1$ for some $k \in N$. As $x \in \{0,1\}^{|A|}$, the left-hand side of (4c) is an integer. Thus, we can round up z_k to 1 without impacting feasibility of (4b) and (4c). The rounded-up solution has the same objective value, and is feasible to (4), hence is optimal. □

We call (4c) *Benders inequalities*, because (4) can be derived from the idea of Benders decomposition, where (4c) are feasibility cuts derived from (3).

Now we derive additional valid inequalities for the feasible set Y of (2) (or equivalently, the feasible set of (4)):

$$Y := \big\{(x,z) \in \{0,1\}^{|A|} \times \{0,1\}^{|N|} \;\big|\; (x,z) \text{ satisfies (2b),(2c)}\big\}. \tag{5}$$

For any set of arcs $C \subseteq A$, define F^C as the *failure set* of scenarios corresponding to C:

$$F^C := \{k \in N \mid G_k(A \setminus C) \text{ has no } s\text{-}t \text{ path}\}.$$

In other words, F^C is the set of scenarios $k \in N$ that fail to have an s-t path if the set of arcs C is removed from the scenario graph G_k.

Definition 4. *A set $C \subseteq A$ is an* ϵ-probabilistic s-t cut *if* $\mathbb{P}(F^C) > \epsilon$. *$C$ is* minimal *if* $\mathbb{P}(F^{C'}) \leq \epsilon$ *for any $C' \subsetneq C$.*

Thus, C is an ϵ-probabilistic s-t cut if the probability that C is an s-t cut in the random graph is greater than ϵ. Since ϵ is fixed, we often refer to ϵ-probabilistic s-t cuts simply as probabilistic s-t cuts. It is immediate that probabilistic s-t cuts yield valid inequalities for Y.

Proposition 5. *If C is an ϵ-probabilistic s-t cut, then the* probabilistic s-t cut inequality

$$\sum_{a \in C} x_a \geq 1 \tag{6}$$

is valid for Y.

Because of our assumption that $|N|$ is not too large, given a set of arcs C, one can immediately check whether $\mathbb{P}(F^C) > \epsilon$, and hence determine whether (6) is valid.

Valid inequalities for Y can also be derived from a set of arcs C with $\mathbb{P}(F^C) \leq \epsilon$. The idea is that, in order to satisfy the knapsack constraint (2b), we must either select an arc in C or set $z_k = 1$ for some scenario k in which a path does exist in $G_k(A \setminus C)$, i.e., from the set $N \setminus F^C$.

Proposition 6. *Let $C \subseteq A$ be such that $P(F^C) \leq \epsilon$, and let $T \subseteq N \setminus F^C$ be a subset of scenarios that satisfies $\mathbb{P}(T) > \epsilon - \mathbb{P}(F^C)$. Then the* partial cut inequality*:*

$$\sum_{a \in C} x_a + \sum_{k \in T} z_k \geq 1, \tag{7}$$

is valid for Y.

Proof. Let $(x, z) \in Y$. If $\sum_{a \in C} x_a \geq 1$, (7) is trivially satisfied. So assume $\sum_{a \in C} x_a = 0$, which implies by (2c) that $z_k = 0$ for all $k \in F^C$. If, in addition $z_k = 0$ for all $k \in T$, then

$$\sum_{k \in N} p_k z_k = \sum_{k \in N \setminus F^C} p_k z_k = \sum_{k \in N \setminus (F^C \cup T)} p_k z_k < 1 - \mathbb{P}(F^C) - (\epsilon - \mathbb{P}(F^C)) = 1 - \epsilon,$$

which violates (2b). Thus, $\sum_{k \in T} z_k \geq 1$. □

For a deterministic graph, the inequality $\sum_{a \in C} x_a \geq 1$ determines a facet of the dominant of the s-t path polytope if and only if C is a minimal s-t cut [39]. (Recall that an s-t cut is minimal if $\nexists C' \subsetneq C$ such that C' is an s-t cut.) Similarly, we next provide conditions for the Benders inequalities, partial cut inequalities, and probabilistic s-t cut inequalities to be facet-defining for $\text{conv}(Y^\uparrow)$, the convex hull of $Y^\uparrow$, where

$$Y^\uparrow = \{(x, z) \in \mathbb{Z}_+^{|A|} \times \{0, 1\}^{|N|} \mid \exists x' \leq x \text{ with } (x', z) \in Y\}.$$

Studying $\text{conv}(Y^\uparrow)$ as opposed to $\text{conv}(Y)$ is justified because for nonnegative cost vectors c, $\min\{cx \mid (x, z) \in Y\} = \min\{cx \mid (x, z) \in Y^\uparrow\}$.

We assume for the following facet-defining results that all scenarios are equally likely, i.e., $p_k = 1/|N|$ for all $k \in N$, and let $q = \lfloor \epsilon |N| \rfloor$. Then the inequality (2b) in the definition of Y becomes $\sum_{k \in N} z_k \geq |N| - q$, the condition $\mathbb{P}(F^C) \leq \epsilon$ in Proposition 6 is equivalent to $|F^C| \leq q$ and the set T in Proposition 6 can be any subset of $N \setminus F^C$ with $|T| \geq q - |F^C| + 1$. Also, to state the sufficient conditions, we define for any $C \subseteq A$ a bipartite graph (C, F^C, B), where C and F^C represent the two node sets in the bipartition, and the edge set B is defined by

$$B := \big\{\{a,k\} \in C \times F^C \mid k \notin F^{C\setminus\{a\}}\big\}.$$

That is, an edge $\{a,k\}$ exists in the bipartite graph (C, F^C, B) if and only if $C \setminus \{a\}$ is not an s-t cut in G_k. Also, for $a \in C$ let $\delta^B(a) \subseteq F^C$ be the set of scenarios adjacent to a in (C, F^C, B), i.e., the scenarios $k \in F^C$ for which $G_k((A \setminus C) \cup \{a\})$ contains an s-t path. Similarly, let $\delta^B(k) \subseteq C$ be the set of arcs adjacent to k in (C, F^C, B), i.e., the arcs $a \in C$ for which $G_k((A \setminus C) \cup \{a\})$ contains an s-t path.

Theorem 7. *Assume $\epsilon > 0$ (so that $q \geq 1$). Let $C \subseteq A$, and assume $|F^C| \leq q$.*

(a) For any $k \in F^C$, the Benders inequality, $\sum_{a \in C} x_a \geq z_k$, is facet-defining for $\text{conv}(Y^\uparrow)$ *if and only if C is a minimal scenario-based s-t cut in G_k and $\delta^B(k') \neq \emptyset$ for every $k' \in F^C$.*

(b) Let $T \subseteq N \setminus F^C$ be such that $|T| = q - |F^C| + 1$ and assume that for every connected component (C', F', B') of (C, F^C, B), there exists $a^* \in C'$ such that $|\delta^B(a^*)| \geq 2$. Then the partial cut inequality (7)*, is facet-defining for* $\text{conv}(Y^\uparrow)$.

Now assume $|F^C| \geq q + 1$.

(c) Assume:

(i) C is minimal, i.e. $|F^{C\setminus\{a\}}| \leq q$ for all $a \in C$, and

(ii) for every connected component (C', F', B') of (C, F^C, B), there exists $a^* \in C'$ such that $|\delta^B(a^*)| \geq |F^C| - q + 1$.

Then the probabilistic s-t cut inequality (6)*, is facet-defining for* $\text{conv}(Y^\uparrow)$.

Proof. Let e^j be a unit vector with a one in component j, and zero elsewhere. The dimension of e^j will be either $|A|$ or $|N|$, and will be understood from context. For a given set of arcs $Q \subseteq A$, let $\mathbf{1}_Q := \sum_{a \in Q} e^a$ and likewise for $S \subseteq N$, let $\mathbf{1}_S := \sum_{k \in S} e^k$. We also use the notation $\mathbf{x}$ to denote a set of vectors of the form (x^j, z^j) where $x^j \in \mathbb{R}^{|A|}$ and $z^j \in \mathbb{R}^{|N|}$.

First, observe that $\text{conv}(Y^\uparrow)$ is full-dimensional, since by definition the rays $(e^a, \mathbf{0})$ are feasible directions for $\text{conv}(Y^\uparrow)$, and by assumption the point $(\mathbf{1}_A, \mathbf{1}_N)$ and the points $(\mathbf{1}_A, \mathbf{1}_N - e^k)$ are feasible for all $k \in N$. Thus $\text{conv}(Y^\uparrow)$ contains a set of $|A| + |N| + 1$ affinely independent points.

(a) We first prove the conditions are necessary. If C is not a minimal scenario-based s-t cut in G_k, then for some $C' \subsetneq C$, the Benders inequality defined by C' dominates the given inequality. If $\delta^B(k') = \emptyset$ for some $k' \in F^C$, then $\sum_{a \in C} x_a \geq z_k + z_{k'}$ is a valid inequality that dominates the Benders inequality. This inequality is clearly valid when $z_{k'} = 0$, and is also valid when $z_{k'} = 1$, because $\delta^B(k') = \emptyset$ implies that at least two arcs in C must be selected in order for an s-t path to exist in scenario k'. Therefore, both conditions in (a) are necessary.

Now, to prove the conditions are sufficient, fix any $a' \in C$ and for each $k' \in F^C \setminus \{k\}$ let $a_{k'} \in \delta^B(k')$. We prove that the following $|A| + |N|$ points are feasible, satisfy the inequality with equality, and are affinely independent:

- $\mathbf{x}^1 := \{(\mathbf{1}_{A\setminus C}, \mathbf{1}_{N\setminus F^C})\}$, 1 point;
- $\mathbf{x}^2 := \{(e^a + \mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus F^C}) \mid a \in C\}$, $|C|$ points;
- $\mathbf{x}^3 := \{(e^{a'} + e^a + \mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus F^C}) \mid a \in A \setminus C\}$, $|A| - |C|$ points;

- $\mathbf{x}^4 := \{(e^{a'} + \mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus F^C} - e^{k'}) \mid k' \in N \setminus F^C\}, \quad |N| - |F^C|$ points;
- $\mathbf{x}^5 := \{(e^{a_{k'}} + \mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus F^C} + e^{k'}) \mid k' \in F^C \setminus \{k\}\}, \quad |F^C| - 1$ points.

It is immediate that all the points satisfy $\sum_{a\in C} x_a = z_k$. We next prove that the points in $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^5$ are in $Y^\uparrow$. First, observe that for each point, $\sum_{k'\in N} z_{k'} \geq |N \setminus F^C| \geq |N| - q$, and thus all the points satisfy (2b). Thus, it remains to check (2c): if $z_{k'} = 1$, then $G_{k'}(A^x)$ contains an s-t path. For $\mathbf{x}^1$, this follows by definition of the set F^C and because $|F^C| \leq q$. For $\mathbf{x}^2$, we use minimality of C: $G_k(A \setminus C \cup \{a\})$ has an s-t path for all $a \in C$, because otherwise $C \setminus \{a\}$ is a scenario-based s-t cut in G_k that has smaller cardinality than C. The points in $\mathbf{x}^3$ all dominate a point in $\mathbf{x}^2$ in the x values and are identical to the same point in the z values, and hence are in $Y^\uparrow$ by definition. The points in $\mathbf{x}^4$ are feasible by the same argument as for the points in $\mathbf{x}^2$, with the only difference being that the points in $\mathbf{x}^4$ have $z_{k'} = 0$ for some $k' \in N \setminus F^C$, which only makes it easier to satisfy (2c). For the points in $\mathbf{x}^5$, there exists an s-t path in $G_k(A \setminus C \cup \{a_{k'}\})$ for each $k' \in F^C$ because $a_{k'} \in \delta^B(k')$.

Next, to prove these points are affinely independent, we subtract the point in $\mathbf{x}^1$ from all the points in the other sets, to obtain $|A|+|N|-1$ points that we show are linearly independent. Placing these points in a matrix, with the corresponding variables labeled above the columns, yields the matrix:

$$\begin{array}{ccccc} z_k & x_a, a \in C & x_a, a \in A \setminus C & z_{k'}, k' \in N \setminus F^C & z_{k'}, k' \in F^C \setminus \{k\} \\ \hline \end{array}$$
$$\begin{pmatrix} \mathbf{1} & I_{|C|} & 0 & 0 & 0 \\ \mathbf{1} & * & I_{|A\setminus C|} & 0 & 0 \\ \mathbf{1} & * & 0 & -I_{|N\setminus F^C|} & 0 \\ \mathbf{1} & * & 0 & 0 & I_{|F^C|-1} \end{pmatrix}$$

where I_r represents a $r \times r$ identity matrix, 0 represents an appropriately sized matrix of zeros, and $*$ represents an appropriately sized unspecified matrix. As this matrix clearly has full-row rank, these points are linearly independent.

(b) Now consider a partial cut inequality, $\sum_{a\in C} x_a + \sum_{k\in T} z_k \geq 1$. Let $k^* \in T$ be fixed ($T \neq \emptyset$ because $|T| = q - |F^C| + 1 \geq 1$) and let $a^* \in C$ be such that $|\delta^B(a^*)| \geq 2$. Consider the following set of points:

- $\mathbf{x}^1 = (\mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus(F^C\cup T)})$, for all $k \in T$;
- $\mathbf{x}^2 = (e^a + \mathbf{1}_{A\setminus C}, \mathbf{1}_S + \mathbf{1}_{N\setminus(F^C\cup T)})$, for all $S \subseteq \delta^B(a), S \neq \emptyset, a \in C$;
- $\mathbf{x}^3 = (e^a + \mathbf{1}_{A\setminus C}, e^{k*} + \mathbf{1}_{N\setminus(F^C\cup T)})$, for all $a \in A \setminus C$;
- $\mathbf{x}^4 = (e^{a^*} + \mathbf{1}_{A\setminus C}, \mathbf{1}_{\delta^B(a^*)} + \mathbf{1}_{N\setminus(F^C\cup T)} - e^k)$, for all $k \in N \setminus (F^C \cup T)$.

First, observe that all these points satisfy $\sum_{a\in C} x_a + \sum_{k\in T} z_k = 1$. We next show that the points in $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^4$ are points in $Y^\uparrow$. In all points, $\sum_{k\in N} z_k \geq |N| - |F^C| - |T| + 1 = |N| - |F^C| - (q - |F^C| + 1) + 1 = |N| - q$, and hence they satisfy (2b). Thus, it remains to check (2c): if $z_k = 1$, then $G_k(A^x)$ contains an s-t path. This holds for the points in $\mathbf{x}^1$ by definition of the set F^C, and for the points in $\mathbf{x}^2$ by the definition of $\delta^B(a)$. For each point (x^j, z^j) in $\mathbf{x}^3$ there is a point $(x', z') \in \mathbf{x}^1$ for which $z' = z^j$ and $x^j \geq x'$, and hence (x^j, z^j) is in $Y^\uparrow$ by definition. The points in $\mathbf{x}^4$ are in $Y^\uparrow$ by the definition of $\delta^B(a^*)$.

Now, let $(\lambda, \mu, \beta) \in \mathbb{R}^{|A|+|N|+1}$ define an arbitrary equation, $\lambda x + \mu z = \beta$, satisfied by all these points. We show that this equation must be a scalar multiple of $\sum_{a\in C} x_a + \sum_{k\in T} z_k = 1$, which proves the partial cut inequality is facet-defining. Substituting the points into this equation yields the following equations:

$$\sum_{a' \in A\setminus C} \lambda_{a'} + \sum_{k' \in N \setminus (F^C \cup T)} \mu_{k'} + \mu_k = \beta, \quad \forall k \in T, \tag{8}$$

$$\sum_{a' \in A\setminus C} \lambda_{a'} + \lambda_a + \sum_{k' \in N \setminus (F^C \cup T)} \mu_{k'} + \sum_{k' \in S} \mu_{k'} = \beta, \quad \forall S \subseteq \delta^B(a), S \neq \emptyset, a \in C, \tag{9}$$

$$\sum_{a' \in A\setminus C} \lambda_{a'} + \lambda_a + \sum_{k' \in N \setminus (F^C \cup T)} \mu_{k'} + \mu_{k*} = \beta, \quad \forall a \in A \setminus C, \tag{10}$$

$$\sum_{a' \in A\setminus C} \lambda_{a'} + \lambda_{a^*} + \sum_{k' \in N \setminus (F^C \cup T)} \mu_{k'} + \sum_{k' \in \delta^B(a^*)} \mu_{k'} - \mu_k = \beta, \quad \forall k \in N \setminus (F^C \cup T). \tag{11}$$

Let $\alpha = \mu_{k^*}$. Subtracting (8) for $k^* \in T$ from (8) for all $k \in T \setminus \{k^*\}$ yields $\mu_k = \mu_{k^*} = \alpha$ for all $k \in T$. Subtracting (8) for k^* from (10) yields $\lambda_a = 0$ for all $a \in A \setminus C$. Subtracting (8) for k^* from all equations in (9) with $|S| = 1$ yields

$$\lambda_a + \mu_k - \alpha = 0, \quad \forall k \in \delta^B(a), a \in C. \tag{12}$$

This implies that for $a, a' \in C$, if $\exists k \in \delta^B(a) \cap \delta^B(a')$, then $\lambda_a = \lambda_{a'}$ and similarly for $k, k' \in F^C$, if $\exists a \in \delta^B(k) \cap \delta^B(k')$ then $\mu_k = \mu_{k'}$. It follows that if (C', F', B') is a connected component of (C, F^C, B) then $\lambda_a = \lambda_{a'}$ for all $a, a' \in C'$ and $\mu_k = \mu_{k'}$ for all $k, k' \in F'$. Now, let $a' \in C'$ be such that $|\delta^B(a')| \geq 2$, and subtracting (8) for k^* from (9) for a' and $S = \delta^B(a')$ yields $\lambda_{a'} + \sum_{k \in \delta^B(a')} \mu_k - \mu_{k^*} = 0$. Combined with (12), this implies that for any $k' \in \delta^B(a')$, $(|\delta^B(a')| - 1)\mu_{k'} = 0$, and hence $\mu_k = 0$ for all $k \in F'$. This same argument applies for all connected components, implying $\mu_k = 0$ for all $k \in F^C$, and so by (12), $\lambda_a = \alpha$ for all $a \in C$. Next, subtracting (8) for k^* from each equation in (11) yields $(\lambda_{a^*} - \mu_{k^*}) + \sum_{k' \in \delta^B(a^*)} \mu_{k'} - \mu_k = 0$, and hence $\mu_k = 0$ for all $k \in N \setminus (F^C \cup T)$ since $\lambda_{a^*} - \mu_{k^*} = 0$ and $\mu_{k'} = 0$ for all $k' \in \delta^B(a^*)$. Then, (11) reduces to $\lambda_{a^*} = \beta$, and hence $\beta = \alpha$. In summary, we have shown: $\lambda_a = \alpha$, for all $a \in C$, $\mu_k = \alpha$ for all $k \in T$, $\beta = \alpha$, $\lambda_a = 0$ for all $a \in A \setminus C$ and $\mu_k = 0$ for all $k \in N \setminus T$, which completes the proof of (b).

(c) Consider a probabilistic s-t cut inequality, $\sum_{a \in C} x_a \geq 1$. Note that the condition (i) is equivalent to $|F^C| - |\delta^B(a)| \leq q$ for all $a \in C$. Let $a^* \in C$ be such that $|\delta^B(a^*)| \geq |F^C| - q + 1$ and consider the following set of points:

- $(e^a + \mathbf{1}_{A\setminus C}, \mathbf{1}_S + \mathbf{1}_{N \setminus F^C})$, for all $S \subseteq \delta^B(a), S \neq \emptyset, a \in C$;
- $(e^a + e^{a^*} + \mathbf{1}_{A\setminus C}, \mathbf{1}_{\delta^B(a^*)} + \mathbf{1}_{N \setminus F^C})$, for all $a \in A \setminus C$;
- $(e^{a^*} + \mathbf{1}_{A\setminus C}, \mathbf{1}_{\delta^B(a^*)} + \mathbf{1}_{N \setminus F^C} - e^k)$, for all $k \in N \setminus F^C$.

All of these points are in $Y^{\uparrow}$, and satisfy $\sum_{a \in C} x_a = 1$. As in the proof of part (b), we show that if these points satisfy an arbitrary equation, $\sum_{a \in A} \lambda_a x_a + \sum_{k \in N} \mu_k z_k = \beta$, then this equation is a scalar multiple of $\sum_{a \in C} x_a = 1$. Indeed, using arguments identical to those in part (b), it can be shown that $\lambda_a = 0$ for all $a \in A \setminus C$, $\mu_k = 0$ for all $k \in N$, and $\lambda_a = \alpha = \beta$ for all $a \in C$. □

We have not been able to show that all conditions in parts (b) and (c) of Theorem 7 are necessary. The condition that $|T| = q - |F^C| + 1$ is clearly necessary for a partial cut inequality to be facet-defining, since otherwise taking a set $T' \subsetneq T$ with $|T'| = q - |F^C| + 1$ would yield a dominating inequality. It is also necessary that $\delta^B(a) \neq \emptyset$ for all $a \in C$ and $\delta^B(k) \neq \emptyset$ for all $k \in F^C$. Indeed, if $\delta^B(a) = \emptyset$ for some $a \in C$, then the partial cut inequality corresponding to

$T' = T$ and $C' = C \setminus \{a\}$, having $F^{C'} = F^C$, dominates the partial cut inequality defined by C and T. If $\delta^B(k') = \emptyset$ for some $k' \in F^C$, then the inequality $\sum_{a\in C} x_a + \sum_{k\in T} z_k \geq 1 + z_{k'}$ is valid: when $z_{k'} = 0$, the standard argument for partial cut inequalities applies, and when $z_{k'} = 1$, we need $\sum_{a\in C} x_a \geq 2$ since $C \setminus \{a\}$ is still an s-t cut in G_k for each arc $a \in C$. This inequality clearly dominates the partial cut inequality defined by C and T. For the probabilistic s-t cut inequality, minimality of C is necessary for it to be facet-defining, since otherwise the inequality is dominated by $\sum_{j\in C'} x_j \geq 1$ for some $C' \subsetneq C$. Similar to the partial cut inequality, it is also necessary that $\delta^B(k) \neq \emptyset$ for all $k \in F^C$, since otherwise the inequality is dominated by the valid inequality $\sum_{j\in C} x_j \geq 1 + z_k$ for some $k \in F^C$.

We next consider an alternative dominant of Y, $Y^{\uparrow\uparrow}$, defined by letting both sets of variables x and z be dominated:

$$Y^{\uparrow\uparrow} = \{(x,z) \in \mathbb{Z}_+^{|A|} \times \mathbb{Z}_+^{|N|} \mid \exists (x',z'), x' \leq x, z' \leq z, \text{with } (x',z') \in Y\}. \tag{13}$$

$\text{conv}(Y^{\uparrow\uparrow})$ is also an interesting set to study, since for $c \in \mathbb{R}_+^{|A|}$, it again holds that $\min\{cx \mid (x,z) \in Y\} = \min\{cx \mid (x,z) \in Y^{\uparrow\uparrow}\}$. The Benders inequalities are not valid for $Y^{\uparrow\uparrow}$. However, partial cut inequalities and probabilistic s-t cut inequalities are valid for $Y^{\uparrow\uparrow}$, and the conditions for them to be facet-defining for $\text{conv}(Y^{\uparrow\uparrow})$ are simpler than those for the set $\text{conv}(Y^{\uparrow})$.

Theorem 8. *(a) Let $C \subseteq A$. If $|F^C| \geq q+1$, then the probabilistic s-t cut inequality* (6) *is facet-defining for* $\text{conv}(Y^{\uparrow\uparrow})$ *if and only if C is minimal, i.e., $|F^{C\setminus\{a\}}| \leq q,\ \forall a \in C$.*

(b) Let $C \subseteq A$. If $|F^C| \leq q$, then the partial cut inequality (7) *with $T \subseteq N \setminus F^C$ that $|T| = q - |F^C| + 1$ is facet-defining for* $\text{conv}(Y^{\uparrow\uparrow})$ *if and only if $|F^{C\setminus\{a\}}| < |F^C|,\ \forall a \in C$.*

Proof. We adopt the same notations in the proof of Theorem 7. The "only if" part is trivial in both cases, since otherwise, inequality (6) and inequality (7) with some $C' \subsetneq C$ dominate the original inequality (6) and (7) with C. We then focus on the "if" part. First, observe that $\text{conv}(Y^{\uparrow\uparrow})$ is full-dimensional, since by definition the rays $(e^a, \mathbf{0}), \forall a \in A$ and $(\mathbf{0}, e^k), \forall k \in N$ are feasible directions for $\text{conv}(Y^{\uparrow\uparrow})$, and point $(\mathbf{1}_A, \mathbf{1}_N)$ is feasible.

(a) Fix any $a' \in C$. By minimality of C we have $\delta^B(a') \neq \emptyset$. We prove that the following $|N|+|A|$ points are feasible, satisfy the inequality (6) as an equation, and are affinely independent:

- $\mathbf{x}^1 := \{(e^a + \mathbf{1}_{A\setminus C}, \mathbf{1}_{\delta^B(a)} + \mathbf{1}_{N\setminus F^C}) \mid a \in C\}$, $|C|$ points;
- $\mathbf{x}^2 := \{(e^{a'} + \mathbf{1}_{A\setminus C} + e^a, \mathbf{1}_{\delta^B(a')} + \mathbf{1}_{N\setminus F^C}) \mid a \in A \setminus C\}$, $|A| - |C|$ points;
- $\mathbf{x}^3 := \{(e^{a'} + \mathbf{1}_{A\setminus C}, \mathbf{1}_{\delta^B(a')} + \mathbf{1}_{N\setminus F^C} + e^k) \mid k \in N\}$, $|N|$ points.

First, observe that all the above points satisfy $\sum_{a\in C} x_a = 1$. We next show that the points in $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ are points in $Y^{\uparrow\uparrow}$. The points in $\mathbf{x}^1$ satisfy the knapsack inequality (2b) by minimality of C. Also, by construction, for each point $(x,z) \in \mathbf{x}^1$, if $z_k = 1$, then $G_k(A^x)$ contains an s-t path, which shows that logical constraints (2c) are satisfied. For each point $(x,z) \in \mathbf{x}^2 \cup \mathbf{x}^3$, there exists a point $(x',z') \in \mathbf{x}^1$ such that $x \geq x'$, and $z \geq z'$, and so $(x,z) \in Y^{\uparrow\uparrow}$. Next, to prove these points are affinely independent, we subtract point $(e^{a'} + \mathbf{1}_{A\setminus C}, \sum_{k'\in\delta^B(a')} e^{k'} + \mathbf{1}_{N\setminus(F^C)})$ from all the other points, and the obtained $|A|+|N|-1$ points that are linearly independent.

(b) Fix any $a' \in C$. By minimality of C we have $\delta^B(a') \neq \emptyset$. We prove that the following $|N|+|A|$ points are feasible, satisfy the inequality (7) as an equation, and are affinely independent:

- $\mathbf{x}^1 := \{(e^a + \mathbf{1}_{A\setminus C}, \sum_{k'\in\delta^B(a)} e^{k'} + \mathbf{1}_{N\setminus(T\cup F^C)}) \mid a \in C\}$, $|C|$ points;

- $\mathbf{x}^2 := \{(e^{a'} + \mathbf{1}_{A\setminus C} + e^a, \sum_{k'\in\delta^B(a')} e^{k'} + \mathbf{1}_{N\setminus(T\cup F^C)}) \mid a \in A \setminus C\}$, $|A| - |C|$ points;
- $\mathbf{x}^3 := \{\mathbf{1}_{A\setminus C}, e^k + \mathbf{1}_{N\setminus(T\cup F^C)}) \mid k \in T\}$, $|T|$ points.
- $\mathbf{x}^4 := \{(e^{a'} + \mathbf{1}_{A\setminus C}, \sum_{k'\in\delta^B(a')} e^{k'} + e^k + \mathbf{1}_{N\setminus(T\cup F^C)}) \mid k \in N \setminus T\}$, $|N| - |T|$ points;

Observe that all the above points satisfy $\sum_{a\in C} x_a + \sum_{k\in T} z_k = 1$. We next show that the points in $\mathbf{x}^1, ..., \mathbf{x}^4$ are in $Y^{\uparrow\uparrow}$. First, consider a point $(x, z) \in \mathbf{x}^1$. (x, z) satisfies the knapsack inequality (2b) by minimality of C and satisfies the logical constraints (2c) by the definition of $\delta^B(a)$. Therefore, $(x, y) \in Y$. Similarly, every point $(x, z) \in \mathbf{x}^3$ is a point in Y. For any $(x, z) \in \mathbf{x}^2 \cup \mathbf{x}^4$, there exists $(x', z') \in \mathbf{x}^1 \cup \mathbf{x}^3 \subseteq Y$, such that $x \geq x'$ and $z \geq z'$. Next, to prove these points are affinely independent, we subtract point $(e^{a'} + \mathbf{1}_{A\setminus C}, \sum_{k'\in\delta^B(a')} e^{k'} + \mathbf{1}_{N\setminus(T\cup F^C)})$ from all the points in the other sets, and the obtained $|A| + |N| - 1$ points are linearly independent. □

2.2 A formulation based on probabilistic s-t cuts

The IP formulation (4) derived in the last section is based on using scenario-based s-t cuts to ensure that when $z_k = 1$ the graph in scenario k defined by the vector $x \in \{0, 1\}^{|A|}$ contains an s-t path. In this section, inspired by the concept of a probabilistic s-t cut given in Definition 4, we derive an IP formulation that is based directly on an inequality characterization of when a *random* graph contains an s-t path with sufficiently high probability.

Given a random graph $\tilde{G}$, let $\mathcal{C}^\epsilon$ be the set of all ϵ-probabilistic s-t cuts in $\tilde{G}$. Also, for any $Q \subseteq A$, let χ^Q be the incidence vector of Q, that is $\chi^Q = (\chi^Q_a)_{a\in A} \in \mathbb{R}^{|A|}$ with $\chi^Q_a = 1$ if $a \in Q$ and 0 otherwise.

Proposition 9. *A set of arcs $Q \subseteq A$ satisfies $\mathbb{P}(S^Q) \geq 1 - \epsilon$ if and only if*

$$\chi^Q \in X := \{x \in \{0, 1\}^{|A|} \mid \sum_{a\in C} x_a \geq 1 \quad \forall C \in \mathcal{C}^\epsilon\}. \tag{14}$$

Proof. Suppose Q satisfies $\mathbb{P}(S^Q) \geq 1 - \epsilon$, and, for contradiction, that $\chi^Q \notin X$. Then, for some $C \in \mathcal{C}^\epsilon$, $C \cap Q = \emptyset$, which implies that $Q \subseteq A \setminus C$. However, by definition of $\mathcal{C}^\epsilon$, $\mathbb{P}(F^C) > \epsilon$, and hence $\mathbb{P}(S^Q) \leq \mathbb{P}(S^{A\setminus C}) = 1 - \mathbb{P}(F^C) < 1 - \epsilon$, contradicting that $\mathbb{P}(S^Q) \geq 1 - \epsilon$. Thus, $\chi^Q \in X$.

Now consider $Q \subseteq A$ such that $\mathbb{P}(S^Q) < 1 - \epsilon$. We show $\chi^Q \notin X$. Let $C = A \setminus Q$. Then, $F^C = \{k \in N \mid G_k(Q) \text{ has no } s\text{-}t \text{ path}\}$, and hence $\mathbb{P}(F^C) > \epsilon$ and so $C \in \mathcal{C}^\epsilon$. But $\chi^Q_a = 0$ for all $a \in C$, and thus $\chi^Q \notin X$. □

This leads to the following IP formulation of (1):

$$\min \sum_{a\in A} c_a x_a \tag{15a}$$

$$\text{subject to} \sum_{a\in C} x_a \geq 1, \quad \forall C \in \mathcal{C}^\epsilon, \tag{15b}$$

$$x \in \{0, 1\}^{|A|}. \tag{15c}$$

The inequalities (15b) are exactly the probabilistic s-t cut inequalities (6).

We next analyze the strength of the probabilistic s-t cut inequalities (15b). As in Section 2.1, rather than study $\text{conv}(X)$, we instead study the convex hull of the dominant of X,

$$X^\uparrow := \{x \in \mathbb{Z}^{|A|}_+ \mid \exists x' \in X, \text{s.t. } x \geq x'\}.$$

This is justified for nonnegative cost vectors as $\min\{cx \mid x \in X^\uparrow\} = \min\{cx \mid x \in X\}$.

Theorem 10. *Let $C \in \mathcal{C}^\epsilon$. Then, the corresponding probabilistic s-t cut inequality* (15b), *is facet-defining for* $\text{conv}(X^\uparrow)$ *if and only if C is minimal.*

Proof. We use the same notation as introduced in the proof of Theorem 7. The "only if" part is immediate because if C is not minimal, then for some $C' \subsetneq C$, $C' \in \mathcal{C}^\epsilon$, the inequality (15b) corresponding to C' dominates the inequality corresponding to C.

Now, let C be a minimal probabilistic s-t cut. We provide $|A|$ affinely independent points in $X^\uparrow$ that satisfy $\sum_{a \in C} x_a = 1$. Let $a' \in C$ be fixed. Then, the $|A|$ points $e^a + \mathbf{1}_{A \setminus C}$, for all $a \in C$, and $e^a + \mathbf{1}_{A \setminus C} + e^{a'}$, for all $a \in A \setminus C$ satisfy $\sum_{a \in C} x_a = 1$ and are affinely independent. For each $a \in C$, the point $e^a + \mathbf{1}_{A \setminus C} \in X$, since otherwise C would not be minimal. For each $a \in A \setminus C$, the point $e^a + \mathbf{1}_{A \setminus C} + e^{a'} \in X^\uparrow$ since it dominates the point $e^{a'} + \mathbf{1}_{A \setminus C} \in X$. □

A branch-and-cut algorithm can be applied to solve the IP formulation (15), provided the probabilistic s-t cut inequalities (15b), are exactly separated for integral solutions. Our separation procedure for (15b) is based on the following relationship between minimal probabilistic s-t cuts of a random graph and minimal scenario-based s-t cuts.

Proposition 11. *If C is a minimal probabilistic s-t cut, then there exists a set of scenarios $F \subseteq N$ with $\mathbb{P}(F) > \epsilon$ such that $C = \bigcup_{k \in F} C_k$, where $C_k \subseteq C$ is a minimal scenario-based s-t cut in $G_k, k \in F$.*

Proof. Let C be a minimal probabilistic s-t cut and let $F = F^C$. Because C is a probabilistic s-t cut, $\mathbb{P}(F) > \epsilon$. By definition C is a scenario-based s-t cut in G_k for each $k \in F$. Now, for each $k \in F$, let $C_k \subseteq C$ be a minimal scenario-based s-t cut in G_k. We claim that $C = \bigcup_{k \in F} C_k$. Trivially, $C \supseteq \bigcup_{k \in F} C_k$. To show that $C \subseteq \bigcup_{k \in F} C_k$, assume for contradiction that $\exists a \in C, a \notin \bigcup_{k \in F} C_k$. We claim that $C \setminus \{a\}$ is a probabilistic s-t cut, which contradicts the minimality of C. Since $a \notin \bigcup_{k \in F} C_k$, $a \notin C_k$ for all $k \in F$, and hence $C_k \subseteq C \setminus \{a\}$ for all $k \in F$. Thus, $C \setminus \{a\}$ is a scenario-based s-t cut in G_k for all $k \in F$ which implies $C \setminus \{a\}$ is a probabilistic s-t cut. □

The converse of the Proposition 11 does not hold, because if C_k is a minimal scenario-based s-t cut in G_k, for all $k \in F$, then $C = \bigcup_{k \in F} C_k$ may not be a *minimal* probabilistic s-t cut ($C \in \mathcal{C}^\epsilon$ is trivially true if $\mathbb{P}(F) > \epsilon$). However, any $C \in \mathcal{C}^\epsilon$ can be reduced to be minimal simply by removing arcs from C as long as it remains a probabilistic s-t cut.

2.3 Separation

We now describe how the valid inequalities described in the previous subsections can be separated within a branch-and-cut algorithm for solving the formulations (4) and (15). Of particular importance, we show that if $\hat{x}$ is integral, then (4c) and (6) can efficiently be separated exactly, which ensures correctness of a branch-and-cut algorithm for formulations (4) and (15), respectively.

Benders inequalities. Let $(\hat{x}, \hat{z}) \in [0,1]^{|A|+|N|}$. To determine if there is a violated inequality in (4c) we must solve the following problem:

$$\hat{v}_k := \min\Big\{\sum_{a \in C} \hat{x}_a \mid C \in \mathcal{C}^k\Big\}, \tag{16}$$

for each $k \in N$. If $\hat{v}_k \geq \hat{z}_k$ for all $k \in N$, then $(\hat{x}, \hat{z})$ satisfies (4c). If, furthermore, $(\hat{x}, \hat{z})$ is integral and satisfies (4b), then it is feasible to (4). Otherwise, we identify a k' and $C' \in \mathcal{C}^{k'}$ with $\sum_{a \in C'} \hat{x}_a < \hat{z}_{k'}$, so that the corresponding inequality (4c) is violated.

Suppose that $\hat{x} \in \{0,1\}^{|A|}$. Then, solving (16) for a fixed k can be accomplished by conducting a graph search from node s in the graph $G_k(A^{\hat{x}})$ to determine the set of nodes S that can be reached from s using arcs in $A^{\hat{x}} \cap A_k$. If $t \in S$, then a path from s to t exists, and so no cut C with $\sum_{a \in C} \hat{x}_a = 0$ can be found. Thus, $\hat{v}_k \geq 1 \geq \hat{z}_k$. If $t \notin S$, then we can take $C = \delta^+_k(S)$, where $\delta^+_k(S) = \{(i,j) \in A_k \mid i \in S, j \notin S\}$. By construction, $\hat{x}_a = 0$ for all $a \in C$, and hence (4c) for this C and k cuts off the infeasible solution. The overall complexity of exact separation of (4c) when $\hat{x}$ is integral is therefore $O(|A||N|)$.

Now suppose $\hat{x} \in [0,1]^{|A|}$ is not necessarily integral. For each k, (16) is the problem of finding a minimum weight scenario-based s-t cut in the graph G_k, where the weight on each arc $a \in A_k$ is given by $\hat{x}_a$. A minimum scenario-based s-t cut in G_k can be found by finding the maximum flow from s to t, which can be done, for example with complexity $O(|V|^3)$ using a FIFO preflow-push algorithm [1]. A graph search can also be used as a heuristic for separating (4c) for fractional $\hat{x}$. For a scenario $k \in N$, we conduct a graph search on the graph induced by arcs with positive $\hat{x}$ values $\{a \in A_k \mid \hat{x}_a > 0\}$. If t is not connected from s in this graph, a cut C with weight zero is found in G_k. In our implementation, we use these separation routines in a hierarchical manner. For each scenario $k \in N$, we first use this graph search procedure to identify a cut with weight zero if one exists. If this fails, we then optionally solve the maximum flow problem to exactly separate (4c).

Probabilistic s-t cut inequalities. Proposition 11 shows that a probabilistic s-t cut can be obtained by combining scenario-based s-t cuts. For $\hat{x} \in \{0,1\}^{|A|}$, we first check whether $\hat{x}$ is feasible (i.e., $\hat{x} \in X$) by checking if there exists an s-t path in $G_k(A^{\hat{x}})$ for each $k \in N$. We then obtain a set $F \subseteq N$ of scenarios that do not contain an s-t path if $A^{\hat{x}}$ is the set of selected arcs. If $\mathbb{P}(F) \leq \epsilon$, then $\hat{x}$ defines a feasible solution. Otherwise, for each $k \in F$, we obtain a scenario-based s-t cut, C_k, in G_k such that $\hat{x}_a = 0$ for all $a \in C_k$. Thus, if we let $C = \bigcup_{k \in F} C_k$, then C is a probabilistic s-t cut, and $\sum_{a \in C} \hat{x}_a = 0$, so we have found a probabilistic s-t cut inequality that cuts off $\hat{x}$.

Now suppose $\hat{x} \in [0,1]^{|A|}$ is not necessarily integral. To heuristically separate (6), we use any of the methods described above for identifying low weight scenario-based s-t cuts in each scenario $k \in N$, and then attempt to combine these to obtain a violated probabilistic s-t cut inequality. Let F be the set of scenarios $k \in N$ for which we have found a scenario-based s-t cut C_k with $\hat{u}_k := \sum_{a \in C_k} \hat{x}_a < 1$. If $\mathbb{P}(F) > \epsilon$, then for any $F' \subseteq F$ such that $\mathbb{P}(F') > \epsilon$, the set $C' = \bigcup_{k \in F'} C_k$ is a probabilistic s-t cut. A simple strategy for choosing $F' \subseteq F$ is to start with $F' = \emptyset$ and greedily add scenarios from F to F', in increasing order of $\hat{u}_k$, until $\mathbb{P}(F') > \epsilon$. After using this combination procedure to identify a candidate probabilistic s-t cut C, we sequentially remove arcs from C as long as $C \setminus \{a\}$ remains a probabilistic s-t cut. This ensures that we only add minimal probabilistic s-t cuts. A naive implementation of this reduction procedure would be to conduct a new graph search in each scenario in F^C to determine the set $F^{C \setminus \{a\}}$ for each candidate arc $a \in C$, yielding a complexity of $O(|A||C||F^C|)$. Algorithm 1 describes an implementation of this reduction procedure that limits the work required to check $\mathbb{P}(F^{C \setminus \{a\}}) > \epsilon$ by using the connectivity information from the current arc set, reducing the complexity to a total of $O(|A||F^C|)$. We refer to this separation approach as "combine-and-reduce."

Partial cut inequalities. Given a relaxation solution $(\hat{x}, \hat{z})$, we heuristically search for violated partial cut inequalities by sequentially removing arcs from a probabilistic s-t cut, one at a time. For each set of arcs C encountered in this reduction, with $\mathbb{P}(F^C) \leq \epsilon$, the separation problem for

Algorithm 1 A simple reduction algorithm to obtain a minimial probabilistic s-t cut.

Input: $C \subseteq A$ and $F^C = \{k \in N \mid G_k(A \setminus C)$ does not contain any s-t path $\}$, with $\mathbb{P}(F^C) > \epsilon$
for k in F^C **do**
  Obtain $S_k = \{v \in V \mid v$ is reachable from s in $G_k(A \setminus C)\}$
end for
for a in C (in lexicographic order, or in decreasing order of $\hat{x}_a$) **do**
  for k in F^C **do**
    Calculate $\overline{S}_k = \{v \in V \mid v$ is reachable from s in $G_k(A \setminus C \cup \{a\}\}$ by a graph search starting from S_k.
  end for
  Set $F^{C \setminus \{a\}} = \{k \in F^C \mid t \notin \overline{S}_k\}$
  if $\mathbb{P}(F^{C\setminus\{a\}}) > \epsilon$ **then**
    $C \leftarrow C \setminus \{a\},\ S_k \leftarrow \overline{S}_k, \forall k \in F^C$
  end if
end for
return C

(7) can be solved by finding:

$$\hat{\eta} := \min\{\sum_{k \in T} \hat{z}_k \mid T \subseteq N \setminus F^C, \mathbb{P}(T) > \epsilon - \mathbb{P}(F^C)\}. \tag{17}$$

If $\hat{\eta} < 1 - \sum_{a \in C} \hat{x}_a$, then a violated inequality is found (defined by any T that attains the minimum in (17)). The problem (17) is a knapsack problem, which can be solved heuristically using a greedy algorithm, by including in T just enough of the elements $k \in N \setminus F^C$ that have the smallest values of $\hat{z}_k/p_k$ to obtain that $\mathbb{P}(T) = \sum_{k \in T} p_k > \epsilon - \mathbb{P}(F^C)$. In the important special case in which all scenarios are equally likely ($p_k = 1/|N|$), (17) can be solved exactly with this greedy approach. In this case, the condition $\mathbb{P}(T) > \epsilon - \mathbb{P}(F^C)$ becomes $|T| \geq q - |F^C| + 1$, where $q = \lfloor \epsilon |N| \rfloor$. We then sort the values $\{\hat{z}_k \mid k \in N \setminus F^C\}$ to obtain $t_1, \ldots, t_{|N \setminus F^C|}$ such that $\hat{z}_{t_1} \leq \hat{z}_{t_2} \leq \cdots \leq \hat{z}_{t_{|N \setminus F^C|}}$, and then set $T = \{t_1, \ldots, t_{q - |F^C| + 1}\}$. In our implementation, we continue removing arcs from our candidate set until $\hat{\eta} \geq 1$, and add all violated partial cut inequalities found during the sequence.

3 Chance-constrained fully connected network design

In this section, we show how the methods developed in the last section for the chance-constrained s-t connected network design problem can be applied for the chance-constrained fully connected network design problem. First, note that, although our development for the s-t connected case was for a directed graph, it extends in a natural way to undirected graphs. Following previous related work, e.g., [9, 20], we consider undirected graphs in this section.

Given an undirected graph $G = (V, E)$ with edges that may fail at random, we seek the minimum cost subset of edges such that all nodes are simultaneously connected with probability at least $1 - \epsilon$. Assume we have a set N of edge failure scenarios, where each scenario $k \in N$ happens with probability p_k, where $\sum_{k \in N} p_k = 1$. The set of available edges in senario $k \in N$ is $E_k \subseteq E$. Given a set of edges Q, we let the graph $G_k(Q)$ be the graph with node set V and edge set $E_k \cap Q$. For $Q \subseteq E$, let $S^Q = \{k \in N \mid G_k(Q)$ is fully connected$\}$. The chance-constrained fully connected network design problem is:

$$\min_{Q \subseteq E} \{c(Q) \mid \mathbb{P}(S^Q) \geq 1 - \epsilon\}. \tag{18}$$

Theorem 12. *The chance-constrained fully connected network design problem* (18) *is $\mathcal{NP}$-hard in the special case in which $p_k = 1/|N|$ for all $k \in N$.*

Proof. The proof is nearly identical to the proof for Theorem 1 for s-t connected case, so we only sketch the differences. The construction of the graph is the same, except that in this case the graph is undirected and and the nodes s and t are not included. The other modification is that, given a "Yes" instance of DPCLP, a solution verifying that the decision version of the constructed instance is "Yes" is obtained by defining x by:

- For $j = 1, 2, ..., m$: If $\max_{k\in I}\{\xi_{kj}\} = 0$, let $x_{(r_j,r_{j+1})} = x_{(v_j,r_{j+1})} = 1$ and $x_{(r_j,v_j)} = 0$; If $\max_{k\in I}\{\xi_{kj}\} = 1$, let $x_{(r_j,r_{j+1})} = 0$ and $x_{(r_j,v_j)} = x_{(v_j,r_{j+1})} = 1$.

This modification ensures that the constructed solution corresponds to a spanning tree, as opposed to an s-t path, in every required scenario. □

3.1 A formulation based on scenario-based graph cuts

Introducing variables z_k for $k \in N$ to model whether or not the graph obtained in scenario k is connected, we obtain the following formulation of (18):

$$\min \sum_{e\in E} c_e x_e \tag{19a}$$

$$\text{subject to} \sum_{k\in N} p_k z_k \geq 1 - \epsilon \tag{19b}$$

$$z_k = 1 \Rightarrow \; G_k(E^x) \text{ is fully connected}, \forall k \in N \tag{19c}$$

$$x \in \{0,1\}^{|E|}, z \in \{0,1\}^{|N|}, \tag{19d}$$

where $E^x := \{e \in E \mid x_e = 1\}$.

For each individual scenario $k \in N$, a natural way of modeling the full connectivity of the graph $G_k(E^x)$ is given by the following undirected cut formulation (see [4]),

$$\sum_{e\in\delta_k(S)} x_e \geq 1, \; \forall S \subsetneq V, S \neq \emptyset, \tag{20a}$$

$$x \in \{0,1\}^{|E|}, \tag{20b}$$

where $\delta_k(S)$ is the set of edges in G_k with exactly one endpoint in S, and (20a) are undirected cut inequalities. This motivates the following formulation, analogous to formulation (4) in the s-t connectivity case:

$$\min \sum_{e\in E} c_e x_e \tag{21a}$$

$$\text{subject to} \sum_{k\in N} p_k z_k \geq 1 - \epsilon, \tag{21b}$$

$$\sum_{e\in\delta_k(S)} x_e \geq z_k, \; \forall S \subsetneq V, S \neq \emptyset, \; k \in N, \tag{21c}$$

$$x \in \{0,1\}^{|E|}, z \in \{0,1\}^{|N|}. \tag{21d}$$

For each scenario $k \in N$, the undirected cut formulation (20) can be strengthened using the partition inequalities [30, 42]:

$$\sum_{e \in \delta_k(V_1,\dots,V_p)} x_e \geq p-1, \;\forall \text{ partitions } (V_1,\dots,V_p) \text{ of } V, \tag{22}$$

where $(V_1,\dots,V_p)$ denotes a partition of the node set V, and $\delta_k(V_1,\dots,V_p)$ is the set of edges $e \in E_k$ that have endpoints in two different sets of the partition. Chopra [16] showed that the partition inequalities (22) and $x \geq 0$ fully describe the dominant of spanning tree polytope. The cut inequalities (20a) are a special case of the partition inequalities (22) with partition $(V_1, V_2) = (S, V \setminus S)$. Thus, the Benders formulation (21) is strengthened by replacing (21c) with:

$$\sum_{e \in \delta_k(V_1,\dots,V_p)} x_e \geq (p-1)z_k, \;\forall \text{ partitions } (V_1,\dots,V_p) \text{ of } V, \;\forall k \in N. \tag{23}$$

For any $k \in N$ and partition $(V_1,\dots,V_p)$ we call $\delta_k(V_1,\dots,V_p)$ a *scenario-based graph cut*, and we call (23) the *scenario-based partition inequalities*. We mention that, using an argument similar to the s-t connectivity case in Section 2.1, integrality on z variables in (21d) can be relaxed.

Another alternative to strengthen the undirected cut formulation (20) is to use a directed cut formulation with an additional set of variables (see, e.g., [29]). Following [29], we introduce a pair of binary variables y^k_{ij} and y^k_{ji} for each edge $e = \{i,j\} \in E_k$ to represent the two directions edge $\{i,j\}$ could be traversed. This yields the following directed cut formulation:

$$\sum_{a \in \delta^-_k(S)} y^k_a \geq 1, \;\forall S \subsetneq V, S \neq \emptyset, r_k \notin S, \tag{24a}$$

$$x_e \geq y^k_{ij} + y^k_{ji}, \;\forall e = \{i,j\} \in E, \tag{24b}$$

$$x_e, y^k_{ij}, y^k_{ji} \in \{0,1\}, \;\forall e = \{i,j\} \in E, \tag{24c}$$

where $r_k \in V$ is a root node chosen arbitrarily. Constraints (24a) ensure there exists a directed path from r_k to every node $v \in V \setminus \{r_k\}$ using the directed arcs chosen by y^k. For deterministic connectivity problems, this formulation yields tighter relaxation bounds than (20), and is advantageous because the number of variables is only doubled (after the redundant x_e variables are removed) and the inequalities (24a) can be separated efficiently using minimum cut algorithms. Using (24) for the chance-constrained fully connected network design problem, the constraints in formulation (21) can be replaced by:

$$\sum_{a \in \delta^-_k(S)} y^k_a \geq z_k, \;\forall S \subsetneq V, S \neq \emptyset, r_k \notin S, \; k \in N, \tag{25a}$$

$$x_e \geq y^k_{ij} + y^k_{ji}, \;\forall e = \{i,j\} \in E, \forall k \in N, \tag{25b}$$

$$x \in \{0,1\}^{|E|}, y^k \in \{0,1\}^{2|E|} \;\forall k \in N. \tag{25c}$$

Formulation (25) requires an additional set of y variables for each scenario $k \in N$. Thus, in contrast to the case of deterministic graphs, the formulation based on (24) is significantly larger than the undirected cut formulation (21).

A formulation based on (25) would be very attractive if we could fix $y^k = y^{k'}$ for all scenarios $k, k' \in N$, so that only one set of y variables would be needed. Unfortunately, this is not the case, as the example in Figure 3 shows. This example has two scenarios, and we require that both scenarios be satisfied. Edge $\{1,3\}$ fails in the first scenario and edge $\{0,2\}$ fails in the second scenario.

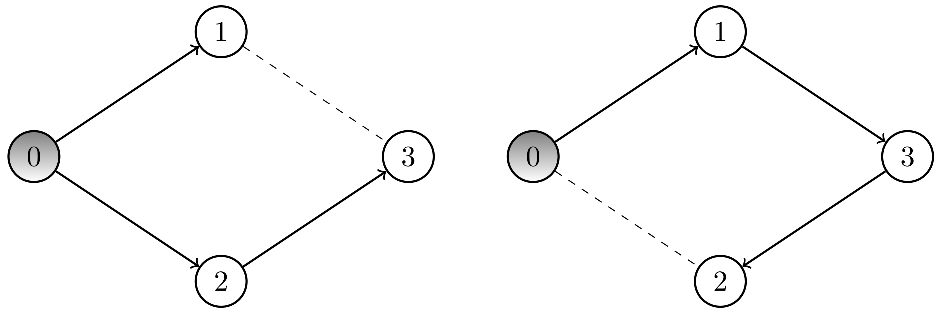

Figure 3: An example graph with two different failure scenarios.

Consider the solution that selects all four edges. This solution is feasible because the graph is fully connected in both scenarios. However, if node 0 is chosen as the root in both scenarios, then we must set $y^1_{23} = 1$ in the first scenario, and $y^2_{32} = 1$ in the second scenario. Because of this restriction that formulation (25) introduces variables y^k for all scenarios $k \in N$, we choose to instead use the undirected cut formulation in which (21c) are strengthened by using the scenario-based partition inequalities (23).

We can also extend the idea of probabilistic *s*-*t* cuts to the fully connected network design case. Given $C \subseteq E$, and an integer $p \geq 2$, we let

$$F_p^C = \{k \in N \mid G_k(E \setminus C) \text{ has at least } p \text{ connected components}\}$$

and we call C an $\epsilon-$*probabilistic graph cut* if $\mathbb{P}(F_2^C) > \epsilon$, or simply a probabilistic graph cut when ϵ is understood. Also, for $C \subseteq E$, let

$$p(C) := \max\{p : \mathbb{P}(F_p^C) > \epsilon\}.$$

For any $p \geq 2$, we call C a *probabilistic p-cut* if $p(C) = p$. Let $\mathcal{C}$ be the family of all probabilistic graph cuts, and for all $p \geq 2$ let $\mathcal{C}_p$ be the family of all probabilistic p-cuts. Then, $\mathcal{C} = \bigcup_{p \geq 2} \mathcal{C}_p$. The following probabilistic graph cut inequalities

$$\sum_{e \in C} x_e \geq p(C) - 1, \quad \forall C \in \mathcal{C} \tag{26}$$

are valid for (19).

3.2 A formulation based on probabilistic graph cuts

Using similar arguments in the *s*-*t* connectivity case, we can formulate the chance-constrained fully connected network design problem (19) using only probabilistic graph cuts:

$$\min \sum_{e \in E} c_e x_e \tag{27a}$$

$$\text{subject to} \sum_{e \in C} x_e \geq p(C) - 1, \quad \forall C \in \mathcal{C} \tag{27b}$$

$$x \in \{0,1\}^{|E|}. \tag{27c}$$

3.3 Separation

We now describe separation routines for scenario-based partition inequalities (23) and probabilistic graph cut inequalities (26).

We first describe a graph search based procedure for separation of scenario-based partition inequalities (23), which is exact if $\hat{x}$ is integral and is a heuristic otherwise. Let $(\hat{x}, \hat{z})$ be a given relaxation solution, and consider a fixed scenario k with $\hat{z}_k > 0$. We conduct a graph search in the graph induced by edges $\{e \in E_k \mid \hat{x}_e > 0\}$ to identify the connected components, $V_1, \ldots, V_p$, in the graph. If $p > 1$, then this yields a scenario-based graph cut $\delta_k(V_1, \ldots, V_p)$, and we obtain a violated scenario-based partition inequality (23) since, by definition $\sum_{e \in \delta_k(V_1,\ldots,V_p)} \hat{x}_e = 0$. This is an exact separation procedure if $\hat{x}$ is integral, which ensures the correctness of a branch-and-cut algorithm. Separation of scenario-based partition inequalities (23) having $p = 2$, can be done by finding a minimum weight cut in the graph G_k, where the weight of each edge $e \in E_k$ is given by $\hat{x}_e$. In our implementation, we use the minimum cut algorithm proposed by Hao and Orlin in [22], and implemented in [24]. Exact separation of scenario-based partition inequalities (23) for any $p(C) \geq 2$, can also be done. Given $\bar{x} \in \mathbb{R}_+^{|E|}$, Baïou and Barahona [6] give a procedure for exact separation of partition inequalities of the form $\sum_{e \in \delta(V_1,\ldots,V_p)} x_e \geq p - 1$, based on solving a sequence of minimum cut problems. If $\hat{z}_k > 0$, by defining $\bar{x}_e = \hat{x}_e / \hat{z}_k$, the scenario-based partition inequality (23) will be violated if and only if $\bar{x}$ violates a partition inequality in G_k. We use these separation routines hierarchically, first attempting the simple graph search, then attempting to find a violated scenario-based partition inequality with $p(C) = 2$ if the simple search fails, and finally attempting exact separation when both of these separation routines fail.

To separate probabilistic graph cut inequalities (26), we extend the combine-and-reduce framework for generating probabilistic s-t cut inequalities (6). Let $F \subseteq N$ be a set of scenarios with $\mathbb{P}(F) > \epsilon$, and let $\delta_k(V_1, ..., V_p)$ be a scenario-based graph cut in scenario $k, \forall k \in F$. Then $C = \bigcup_{k \in F} \delta_k(V_1, ..., V_p)$ is a probabilistic graph cut, and we obtain a valid inequality $\sum_{e \in C} x_e \geq p(C) - 1$. To strengthen this inequality, we follow the idea of Algorithm 1 to conduct a sequential reduction on C to make C a minimal probabilistic graph cut. Note that in this case, we are not limited to a specific $p(C)$ value in (26). Instead, we obtain a whole family of minimal probabilistic $p(C)$-cuts with different $p(C)$ values as long as $p(C) \geq 2$ by sequential reduction.

For fomulation (27), which uses only probabilistic graph cut inequalities, we found it helpful to also use a simple greedy heuristic to attempt to find more violated inequalities. The heuristic greedily includes edges in a set C sequentially, in increasing order of $\hat{x}_e$ values as long as $\sum_{e \in C} \hat{x}_e \leq 6$ and $\hat{x}_e \leq 0.9$. We then check if the obtained set C is a probabilistic graph cut, and if so we conduct a full sequential reduction on it, as described in the previous paragraph.

4 Computational Experiments

We now present the results of computational experiments using our branch-and-cut algorithms to solve the formulations introduced in Section 2 and 3.

4.1 Implementation Details

Basic settings. We implement the branch-and-cut algorithms within the commercial integer programming solver IBM Ilog CPLEX, version 12.2. We turn off the CPLEX Presolve and set the number of threads to one. We set the CPLEX MIPEmphasis parameter to "optimality" and set the CPLEX variable selection strategy to "pseudo reduced cost". Separation of valid inequalities is done within a cut callback that is called by CPLEX throughout the branch-and-bound tree, and in particular is called at every integral solution, so that a cut will be added if the solution does not satisfy all constraints of the formulation. We use the graph library LEMON [24] for solving graph-related subroutines. All tests are conducted on a Linux workstation with eight 2.93GHz processors and 2.9Gb memory. We use a time limit of 3600 seconds for s-t connected instances and 14400 seconds for fully connected instances.

Heuristics. We implement two heuristics for finding feasible solutions, represented by $Q \subseteq A$. The first heuristic is a very simple greedy strategy that is called at every node in the branch-and-bound tree. Given a relaxation solution $\hat{x}$, we sort the values of $\hat{x}_a$ so that $\hat{x}_{t_1} \geq \hat{x}_{t_2} \geq \cdots \geq \hat{x}_{t_{|A|}}$. We then find $\hat{r} = \max\{r \mid \sum_{i=1}^{r} c_{t_i} \leq UB\}$, where UB is the objective value of the best feasible solution found so far, and set $Q = \{t_1, \ldots, t_r\}$. If $\mathbb{P}(S^Q) \geq 1 - \epsilon$, then Q is a feasible solution that, by construction, has cost no larger than UB. We then attempt to improve this solution by sequentially removing arcs $a \in Q$ from Q as long as $\mathbb{P}(S^{Q \setminus \{a\}}) \geq 1 - \epsilon$. We also occasionally call a more intensive greedy heuristic. In this heuristic, given a relaxation solution $\hat{x}$, we begin with a candidate set of arcs $Q = \{a \in A \mid \hat{x}_a = 1\}$. We then find the set $S^Q \subseteq N$, and if $\mathbb{P}(S^Q) \geq 1 - \epsilon$ we stop with this feasible solution and perform the sequential arc reduction to improve it. Otherwise, we identify a scenario $k \in N \setminus S^Q$ that has the minimum cost s-t path, using costs $\tilde{c}_a = 0$ if $a \in Q$ and $\tilde{c}_a = c_a$ if $a \notin Q$. We then add the arcs in the optimal s-t path of this scenario to the candidate set Q, thus ensuring that an s-t path now exists in scenario k. With the new set Q, we again find the set S^Q and repeat this procedure until either $\sum_{a \in Q} c_a > UB$ (if a feasible solution with cost UB is known) or $\mathbb{P}(S^Q) \geq 1 - \epsilon$. In the latter case, we sequentially remove arcs from $a \in Q$ from Q as long as $Q \setminus \{a\}$ remains feasible. This heuristic is run after solving the initial LP relaxation, once every node for the first 100 nodes, and once every 5 nodes after 100 nodes and before 1000 nodes. We also call this heuristic before our branch-and-cut algorithm starts, in which case we initialize $Q = \emptyset$. This heuristic is guaranteed to find a feasible solution if one exists, so this ensures that the branch-and-cut algorithm begins with a feasible solution. These heuristic routines are extended in an obvious way to the chance-constrained fully connected network design problem.

Dominance. Ruszczyński [37] showed how a notion of dominance between scenarios can be used to strengthen a MIP formulation of a chance-constrained optimization problem. We adapt these ideas to show how dominance can be used to fix variables in formulation (4) and to save work in searching for violated inequalities in formulations (4) and (15). We describe the idea for the s-t connected problem, but it extends in an obvious way to the fully connected problem. Given two scenarios $j, k \in N$, we say k *dominates* j, denoted $k \succeq j$, if the set of available arcs in scenario j is a subset of available arcs in scenario k, i.e., $A_j \subseteq A_k$. If $k \succeq j$, then for any $Q \subseteq A$, if $G_k(Q)$ is not s-t connected, then also $G_j(Q)$ is not s-t connected. For each $k \in N$, let $D(k) = \{j \in N \mid k \succeq j\}$. If $\mathbb{P}(D(k)) > \epsilon$, then for any feasible set of arcs Q, $G_k(Q)$ must be s-t connected, since otherwise for all $j \in D(k)$, $G_j(Q)$ would not be s-t connected, and so $\mathbb{P}(S^Q) < 1 - \epsilon$. Let $\mathcal{D} := \{k \in N \mid \mathbb{P}(D(k)) > \epsilon\}$. Then, for each $k \in \mathcal{D}$, we can fix $z_k = 1$ in formulation (4), and hence any scenario-based s-t cut in G_k defines a probabilistic s-t cut. For each scenario $k \in \mathcal{D}$, define $I(k) := D(k) \setminus \mathcal{D}$, and let $\mathcal{I} := \{k \in \mathcal{D} \mid \mathbb{P}(I(k)) > \epsilon\}$. Our interest in the set $\mathcal{I}$ is that, Benders inequalities (4c) for scenarios $k \in \mathcal{I}$ are not needed in formulation (4), and thus scenarios in $k \in \mathcal{I}$ can be ignored after fixing $z_k = 1$.

Proposition 13. *Excluding inequalities* (4c) *for all* $k \in \mathcal{I}$ *does not change the set of feasible solutions in formulation* (4).

Proof. Let $k \in \mathcal{I}$, so that $\mathbb{P}(I(k)) > \epsilon$, and let $C \in \mathcal{C}^k$. We show that the Benders inequality $\sum_{a \in C} x_a \geq z_k$ is implied by the remaining constraints in (4). Because C is a scenario-based s-t cut in G_k, C is also a scenario-based s-t cut in G_i for all $i \in I(k)$. By definition $I(k) \cap \mathcal{D} = \emptyset$, and hence $I(k) \cap \mathcal{I} = \emptyset$, so the scenario-based cuts derived from scenarios $i \in I(k)$ are not excluded in (4). Thus, the inequalities $\sum_{a \in C} x_a \geq z_i$, are included for all $i \in I(k)$. Adding these inequalities, with weight p_i applied to each, yields $\sum_{i \in I(k)} p_i \sum_{a \in C} x_a \geq \sum_{i \in I(k)} p_i z_i$. We claim that $\sum_{i \in I(k)} p_i z_i > 0$, since otherwise, $\sum_{i \in N} p_i z_i = \sum_{i \in N \setminus I(k)} p_i z_i \leq \sum_{i \in N \setminus I(k)} p_i < 1 - \epsilon$, which violates the reliability constraint (4b). Consequently, $\sum_{a \in C} x_a > 0$, and by integrality of x, this implies that $\sum_{a \in C} x_a \geq 1 \geq z_k$. □

Similarly, the following proposition shows that if a set of arcs C is a probabilistic s-t cut, i.e., $\mathbb{P}(F^C) > \epsilon$, then also $\mathbb{P}(F^C \setminus \mathcal{I}) > \epsilon$. Thus, when searching for violated probabilistic s-t cut inequalities to check if a candidate solution is feasible to (15), it is not necessary to check whether an s-t path exists in scenarios $k \in \mathcal{I}$.

Proposition 14. *If C is a probabilistic s-t cut, then $\mathbb{P}(F^C \setminus \mathcal{I}) > \epsilon$.*

Proof. Let C be a probabilistic s-t cut, so that $\mathbb{P}(F^C) > \epsilon$. The result is trivial if $F^C \cap \mathcal{I} = \emptyset$. So assume $F^C \cap \mathcal{I} \neq \emptyset$, and choose any $k \in F^C \cap \mathcal{I}$. Then $\mathbb{P}(I(k)) > \epsilon$, and $I(k) \subseteq F^C \setminus \mathcal{I}$ since each scenario in $I(k)$ is dominated by scenario k and $G_k(A \setminus C)$ does not have an s-t path because $k \in F^C$. Thus $\mathbb{P}(F^C \setminus \mathcal{I}) \geq \mathbb{P}(I(k)) > \epsilon$. □

In our implementation, we first solve the root LP relaxation considering only scenarios in the set $\mathcal{I}$. Given a relaxation solution $(\hat{x}, \hat{z})$ for formulation (4) or $\hat{x}$ for formulation (15), we obtain probabilistic s-t cuts by just solving minimum cut problems for scenarios in $\mathcal{I}$, and add violated probabilistic s-t cut inequalities to the root LP relaxation. We resolve the root LP relaxation and repeat until no violated valid inequalities can be found. The inequalities found during this procedure are then included in the initial formulation given to CPLEX. We then begin the branch-and-cut algorithm. From then on, we ignore scenarios in the set $\mathcal{I}$, which is justified by Proposition 13 and Proposition 14.

Dominance is likely to occur more often when the arcs in the random network are more reliable, since then fewer arcs are likely to fail in each scenario. In our experiments, we found that dominance rarely occurred for the s-t connectivity instances, and hence we do not apply these ideas for solving them. The edges in the full connectivity instances are more reliable by our construction, so that dominance frequently occurs for these. (See Table 6 for the average size of $\mathcal{I}$ in our full connectivity test instances.) Applying the dominance ideas to these instances improved the computational results.

Cut selection. Adding many cuts will slow down the solution of the LP relaxation, especially when the cuts are approximately parallel. To mitigate this effect, we apply a cut selection strategy motivated by [3]. We apply the strategy within each round of cut generation. Among all cuts that are generated in the current round, we aim to select a subset of cuts that are not too parallel to each other. Consider two generic cuts $\alpha_i y \geq \beta_i$ and $\alpha_j y \geq \beta_j$, where y represents (x, z) in Benders inequalities (4c) and scenario-based partition inequalities (23), and x in probabilistic s-t cut inequalities (6) and probabilistic graph cuts (26). A measure of parallelism $\theta(i,j)$ between these two cuts is defined as follows:

$$\theta(i,j) := \frac{|\alpha_i \cdot \alpha_j|}{\|\alpha_i\|_2 \|\alpha_j\|_2}.$$

Let C_{can} be the set of candidate cuts, and C_{added} be the set of cuts that have been added to the LP relaxation in the current round. Initially, we add the deepest cut with respect to the current relaxation solution $\hat{y}$, i.e., the cut $\alpha y \geq \beta$ that has the largest normalized violation value $(\beta - \alpha\hat{y})/\|\alpha\|_1$, and include it in C_{added}. Then, in decreasing order of normalized cut violation, we add cut $i \in C_{can}$ if $\theta(i,j) < \theta_0, \forall j \in C_{added}$, where θ_0 is a parameter that controls the acceptable level of parallelism. In our experiments, we use $\theta_0 = 0.7$ for the s-t connected case and $\theta_0 = 0.5$ in the full connected case. We check parallelism separately for Benders inequalities (4c) and probabilistic s-t cut inequalities (6), and similarly for the fully-connected formulations. We also set a cut violation threshold $\delta = 0.2$ for adding Benders inequalities (4c) and scenario-based partition inequalities (23), and cut violation threshold $\delta' = 0.1$ for adding probabilistic s-t cut inequalities (6), partial cut inequalities (7), and probabilistic graph cut inequalities (26).

4.2 Chance-constrained s-t connected network design

In this section we compare the branch-and-cut methods for solving formulation (4), the alternative formulation (15), and the extended formulation that directly uses the MIP formulation based on (3) in Section 2. We test the proposed algorithms on random graphs constructed from deterministic graphs in the OR Library [7]. The set of arcs, A, and their corresponding costs, c_a for $a \in A$, are based on these deterministic instances. The instances used and their sizes are shown in Table 2. For a given graph $G = (V, A)$, we create a random graph as follows. First, for simplicity, we assume the arcs $a \in A$ fail independently with probability ρ_a. We emphasize, however, that independence is not required by our methods. These failure probabilities are randomly generated independently according to the exponential distribution with mean $\lambda = 0.1$. After the failure probabilities, ρ_a, have been generated, we generate a set N of failure scenarios by taking a Monte Carlo sample with these failure probabilities fixed, and set $p_k = 1/|N|$ for all $k \in N$. We check if there is any set of identical scenarios $S \subset N$, and if so, we choose $k \in S$ and set $p_k = |S|/|N|$ and delete the scenarios in $S \setminus \{k\}$. For each graph instance and sample size $|N|$ we repeat this procedure five times to obtain five different instances. (The failure probabilities are re-sampled each time, so the underlying distribution of arc failures is different in the different instances, not just the sample.) We present the average performance over the five instances. The instances and full computational results with respect to individual instances are available from the authors upon request. Unless otherwise stated, we use $\epsilon = 0.05$ as the risk tolerance.

Instance	$\|V\|$	$\|A\|$
rcsp1	100	955
rcsp3	100	990
rcsp5	200	2040
rcsp7	200	2080

Table 2: Graphs used to construct test instances for s-t connected network design and their sizes.

We investigate three options for our branch-and-cut methods:

Benders: Solve formulation (4) using only (4c). At every relaxation solution $(\hat{x}, \hat{z})$, for each scenario $k \in N$ with $\hat{z}_k > \delta$, we first use the simple graph search procedure to search for a violated Benders inequalities (4c). In addition, if this fails, we optionally solve a maximum flow problem to exactly separate Benders inequalities (4c). Exact separation is done in this case for every relaxation solution at the root node, at every node in the branch-and-bound tree up to depth $d = 8$, and beyond that, when the depth of the node is a multiple of five. Exact separation is done at most once for non-root nodes. Furthermore, we limit the number of Benders inequalities that we add per round to be no more than $M = 20$.

Benders+: Solve formulation (4) using (4c), probabilistic s-t cut inequalities (6), and partial cut inequalities (7). The separation approach for (4c) is identical to option Benders. We use the combine-and-reduce procedure that we described in Section 2.3. After combining scenario-based cuts to obtain a probabilistic s-t cut C, we reduce C to be minimal following two different sequences: lexicographic and decreasing order of relaxation values $\hat{x}_a$, potentially yielding two different minimal cuts. We further reduce these minimal probabilistic s-t cuts to search for violated partial cut inequalities (7).

Prob s-t: Solve formulation (15), using the same combine-and-reduce strategy in Benders+ to separate probabilistic s-t cut inequalities (6).

Table 3 presents the results of the extended formulation and the Benders version of the branch-and-cut algorithm on a set of small instances. We use the following notation in this and other tables:

- #: Number of instances solved to optimality within the time limit.
- #Mem: Number of instances that hit the memory limit during the solution process.
- AvgT: Average computational time, in seconds, for the instances solved to optimality.
- AvgG: Average optimality gap remaining when the time limit is reached, for the instances that are not solved to optimality. Optimality gap is calculated as $(UB - LB)/UB$, where UB and LB are the values of the best upper and lower bounds found by the method. "-" means all five instances were solved to optimality.
- AvgN: Average number of nodes processed for the instances solved to optimality.
- AvgR: Average root gap. Root gap is calculated as $(U^* - RLB)/U^*$ where U^* is the true optimal value and RLB is the lower bound provided at the root node after it is processed.
- AvgR-T: Average processing time at the root node.
- AvgNC: Average number of cuts added, for the instances solved to optimality.

Instances		Ext-form				Benders	
Graph	$\|N\|$	#	#Mem	AvgT	AvgG	#	AvgT
rcsp1	100	5	0	142.2	-	5	1.7
	200	1	0	1609.6	13.7%	5	6.6
	300	1	3	282.5	17.0%	5	8.6
rcsp3	100	5	0	376.6	-	5	1.3
	200	5	0	629.0	-	5	0.8
	300	3	1	1307.5	27.6%	5	7.9
rcsp5	100	1	0	2374.5	22.5%	5	23.9
	200	2	3	2242.9	-	5	7.0
	300	0	5	-	-	5	45.3
rcsp7	100	3	0	1099.5	28.8%	5	2.7
	200	0	3	-	37.9%	5	7.4
	300	0	4	-	99.9%	5	35.4

'-': not applicable.

Table 3: Comparison of the extended formulation based on (3) and branch-and-cut method Benders.

Table 3 demonstrates that the extended formulation based on (3) is not practical. This is because the formulation is so large that even solving the LP relaxation becomes too time-consuming. Moreover, as the number of scenarios increases, for example, for instances with more than 200 scenarios, the extended formulation frequently hits the memory limit. In contrast, the Benders formulation solves all of these small instances in less than one minute.

Instances		Benders			Benders+			Prob s-t	
Graph	$\|N\|$	#	AvgT	AvgN	#	AvgT	AvgN	AvgT	AvgN
rcsp1	100	5	1.7	26.0	5	0.9	8.8	0.9	22.6
	1000	5	22.6	35.0	5	15.6	32.6	11.0	64.8
	2000	5	39.3	42.6	5	18.4	11.2	8.6	17.4
	5000	5	427.0	341.8	5	340.4	219.6	163.2	1085.4
rcsp3	100	5	1.3	40.0	5	0.8	12.8	0.7	21.2
	1000	5	13.7	49.8	5	11.2	30.6	8.5	85.2
	2000	5	29.9	35.0	5	17.6	16.4	14.7	72.0
	5000	5	258.7	971.2	5	62.8	14.8	21.8	19.0
rcsp5	100	5	23.9	70.4	5	8.4	60.2	8.0	170.6
	1000	4	>851.5	145.2	5	90.5	120.0	61.4	550.0
	2000	4	>1051.9	992.8	5	329.0	2410.2	152.9	1340.8
	5000	4	>1935.1	252.8	5	1022.1	292.2	678.7	2628.8
rcsp7	100	5	2.7	46.4	5	5.6	18.8	4.1	49.6
	1000	5	99.2	1875.2	5	37.9	65.0	26.6	152.8
	2000	5	271.6	378.4	5	210.5	72.4	90.5	369.6
	5000	4	>1203.1	131.2	4	>1035.2	72.0	234.4	577.8

'>': we use 3600s for the instance that is not solved to optimality.

Table 4: Average computational time and number of nodes processed for the three branch-and-cut algorithms.

Table 4 presents results comparing the different variants of the branch-and-cut algorithms on larger instances, by showing the number of instances that can be solved, average computational times and average numbers of processed nodes. This table demonstrates that option Prob s-t, based on formulation (15), generally yields the shortest computation time, and is the only option to solve all instances within the time limit. Comparing the computational time and number of processed nodes for options Benders and Benders+ in Table 4, we see that adding probabilistic s-t cut inequalities and partial cut inequalities to formulation (4) reduces the number of nodes and leads to many more instances being solved within the time limit. Comparing options Benders+ and Prob s-t, we see that Benders+ benefits in terms of number of nodes from using Benders inequalities, but takes more time at each node on average, leading to longer total solution times.

Table 5 presents the average root optimality gaps and average time to process the root node for all options, and the average total number of cuts added for options Benders+ and Prob s-t. Option Benders+, which uses probabilistic s-t cut inequalities and partial cut inequalities, has significantly smaller gap than option Benders in most cases, and also tends to require less time processing the root node. Benders+ also yields smaller root gaps than Prob s-t. In spite of this, we saw from Table 4 that Prob s-t yielded the best overall performance. As Table 5 demonstrates, Prob s-t spends less time processing the root node, and adds fewer cuts in total, leading to significantly shorter time to process each node on average. In addition, despite the poor root gaps of Prob s-t, Table 4 indicates that the number of nodes processed using this option is not significantly higher than

Instances		Benders		Benders+			Prob s-t		
Graph	$\|N\|$	AvgR	AvgR-T	AvgR	AvgR-T	AvgNC	AvgR	AvgR-T	AvgNC
rcsp1	100	11.8%	1.2	4.4%	0.7	288.0	9.0%	0.7	119.6
	1000	9.2%	16.1	7.8%	9.5	1242.4	13.0%	7.8	231.8
	2000	14.1%	28.7	7.6%	13.4	1151.8	10.9%	6.9	112.4
	5000	22.7%	144.9	18.3%	97.8	4184.0	24.4%	35.9	983.2
rcsp3	100	14.1%	0.8	11.6%	0.6	272.0	14.8%	0.5	117.8
	1000	14.3%	8.9	33.6%	6.8	943.4	37.2%	4.9	192.4
	2000	14.6%	22.5	9.5%	13.3	1132.6	13.1%	9.5	174.8
	5000	21.6%	108.7	11.5%	54.1	2027.0	15.2%	17.0	123.2
rcsp5	100	15.9%	13.7	15.2%	3.4	746.4	21.9%	2.2	448.6
	1000	18.0%	92.8	16.0%	45.1	2247.2	25.7%	23.4	748.6
	2000	20.2%	153.7	20.3%	70.2	3079.8	27.6%	25.6	1036.4
	5000	16.4%	820.5	16.6%	421.8	5660.2	29.1%	66.8	2264.4
rcsp7	100	14.7%	1.8	7.8%	4.0	629.0	12.5%	2.5	277.0
	1000	21.3%	26.8	16.3%	17.3	1525.4	19.3%	15.3	376.6
	2000	24.0%	127.3	20.1%	106.9	2904.2	25.7%	26.1	676.6
	5000	21.5%	447.5	18.6%	323.9	3895.8	23.3%	97.2	763.4

Table 5: Average root optimality gaps, average time to precess the root node, and average number of cuts added for the three branch-and-cut algorithms.

the number of nodes processed by option Benders+. A possible explanation for why option Prob s-t has relatively large root gap but yet does not process too many nodes is that, the probabilistic s-t cut inequalities (15b) provide a tight relaxation bound, but our methods are most effective at separating them at integral solutions which are mostly found throughout the branch-and-bound tree, as opposed to at the root node. This suggests there is room for improvement in devising effective and computationally efficient strategies for separating probabilistic s-t cut inequalities at fractional solutions.

4.3 Chance-constrained fully connected network design

We now present computational results of our branch-and-cut methods for solving formulations (21) and (27) for chance-constrained fully-connected network design problems. We test the proposed algorithms on random graphs constructed from deterministic graphs in the Survivable Network Design Data Library (SNDLIB) [34]. The instances used and their sizes are shown in Table 6. We construct the random graphs in the same way as the s-t connectivity case, except that in this case the edge failure probability follows an exponential distribution with mean $\lambda = 0.01$. For each graph we create five different sampled instances, for sample sizes $N \in \{100, 500, 1000\}$. We present the average performance measures over the five instances. We choose a small λ value in this case because the underlying graph instances are sparse, and so a larger failure probability leads to many scenarios k in which G_k is disconnected, so that the connectivity requirement cannot be met even if all edges are selected. Such scenarios can trivially be discarded, making the instance easier to solve, and possibly even leading to a trivially infeasible instance. On the other hand, when the edges are more reliable, dominance between scenarios is more common, so that the set of scenarios, $\mathcal{I}$, for which we can fix $z_k = 1$ in (21) and that can be ignored when searching for violated probabilistic s-t cut inequalities, is larger. This is demonstrated in Table 6, which shows the average size of $\mathcal{I}$ for each instance and sample size.

Instances	$\|V\|$	$\|E\|$	Average $\|\mathcal{I}\|$ $\|N\|$=100	$\|N\|$=500	$\|N\|$=1000
Atlanta	15	22	35.4	198.8	434.4
Di-yuan	11	42	11.8	69.2	180.6
Ta-1	24	55	10.8	38.6	74.8
Cost266	37	57	31.6	173.2	318.0
Pioro40	40	89	17.0	92.4	142.4

Table 6: Graphs used to construct test instances for fully connected network design and their sizes, and also the average size of the set of scenarios that can be excluded using the dominance condition in our sampled instances.

We investigate three options for our branch-and-cut methods:

BendersFull: Solve formulation (21) using only (23). At every relaxation solution $(\hat{x}, \hat{z})$, for each scenario $k \in N$ with $\hat{z}_k > \delta$, use the graph search procedure to search for violated scenario-based partition inequalities (23). If this fails, optionally separate scenario-based partition inequalities having $p = 2$ exactly using a minimum cut algorithm. If this also fails, optionally call the exact separation routine for scenario-based partition inequalities (23). We call all the subroutines at the root node, and every node in the branch-and-bound tree up to depth $d = 4$. Beyond that, we call the minimum cut routine if the depth of the node is a multiple of five, and the exact scenario-based partition inequality separation routine if the depth is a multiple of 15. The minimum cut and exact separation routines are done at most once for non-root nodes.

BendersFull+: Solve formulation (21) using (23) and also probabilistic graph cut inequalities (26) derived from scenario-based partition inequalities using the combine-and-reduce procedure. Given a set of scenarios F with $\mathbb{P}(F) > \epsilon$, we obtain a probabilistic graph cut C by combining scenario-based graph cuts derived from the scenarios in F, and then conduct a full sequential reduction on the obtained probabilistic $p(C)$-cut.

ProbFull: Solve formulation (27). Given a relaxation solution $\hat{x}$, we use the same combine-and-reduce strategy as in option BendersFull+ for separation of probabilistic graph cut inequalities, except that we additionally use the greedy heuristic described in Section 3.3.

Table 7 presents the average computational time and average number of processed nodes for the three options, and the average root optimality gap for options BendersFull+ and ProbFull. These results indicate that option BendersFull+ has the best computational performance. Also, option ProbFull, which uses the probabilistic graph cut formulation (27), performs relatively poorly, which is in contrast to the results for the s-t connected case. The root optimality gaps are much larger for option ProbFull, and in this case this translates to requiring significantly more nodes in the branch-and-bound tree than option BendersFull+. Comparing option BendersFull and BendersFull+, we see some improvement in the computational time, and number of nodes processed, suggesting that the probabilistic graph cut inequalities do help this formulation. On the other hand, the average root gaps for option BendersFull (not shown in the table) are very similar to those obtained with option BendersFull+. We also observe that in general the fully connected network design problem appears much more difficult than the s-t connected design problem, as the number of nodes required to solve these instances can increase dramatically as the number of scenarios increases, especially for instance Pioro40.

As a point of comparison, in [9] random versions of the deterministic instances Cost266 and Pioro40 are also used to test their methods (heuristic and exact) for solving the chance-constrained fully-connected network design problem, although we do not have the details on how they create

Instances		BendersFull		BendersFull+			ProbFull		
Graph	$\|N\|$	AvgT	AvgN	AvgT	AvgN	AvgR	AvgT	AvgN	AvgR
Atlanta	100	0.2	32.4	0.2	23.6	4.5%	0.2	21.6	5.0%
	500	0.7	45.2	0.6	37.0	3.8%	0.6	26.0	4.8%
	1000	1.3	74.4	1.1	51.6	5.0%	0.7	38.4	6.5%
Di-yuan	100	16.3	4144.4	5.1	1578.2	6.3%	2.2	352.2	6.8%
	500	68.0	4127.6	25.7	1762.6	6.8%	7.2	751.8	7.7%
	1000	156.6	6139.4	87.1	3226.6	6.3%	12.9	761.2	6.9%
Ta-1	100	2.0	126.6	1.7	97.4	4.9%	6.8	911.6	15.1%
	500	11.3	246.8	10.3	215.8	6.5%	199.2	9461.0	23.4%
	1000	32.5	466.0	35.2	443.2	7.8%	378.0	12274.4	22.1%
Cost266	100	6.6	1042.4	5.2	758.4	4.0%	33.3	2454.6	8.7%
	500	60.9	2684.8	45.6	2316.4	6.3%	625.4	14625.2	12.3%
	1000	92.8	2114.4	77.8	1931.2	6.2%	669.7	14457.0	12.6%
Pioro40	100	76.9	4886.8	45.5	2945.2	5.2%	1684.2	31890.6	11.4%
	500	3257.0	37604.8	2468.8	33220.6	7.7%	$*$	$*$	13.5%
	1000	6332.4†	26562.2†	5493.7	36182.0	8.9%	$*$	$*$	15.7%

'$*$': No instances solved within time limit.

'†': One instance is not in time limit. We use 14400s and # of nodes up to the time limit for that instance.

Table 7: Average optimality gap, solution time, number of nodes, and root gap results for chance-constrained fully connected network design instances.

random versions. (The instances Atlanta and Di-yuan are also tested, but results of their exact method are not reported.) In [9], they report results for graphs with a maximum of 70 scenarios. For random graphs based on the Cost266 instance, their exact method takes on average more than 6000 seconds. In contrast, we are able to solve instances with up to 1000 scenarios in a little more than one minute on average. For the random graphs based on the Pioro40 instance, their exact method takes on average more than 1500 hours to solve, and our algorithm is able to solve them within 2 hours on average for instances with 1000 scenarios. Unfortunately, we cannot draw a final conclusion from this comparison, as the instances are not identical (because the randomization schemes are different) and the methods are run on different machines.

4.4 Solution Quality

In this section, we illustrate the potential of the chance-constrained model to provide a rich variety of solutions, by comparing the solutions obtained from the chance-constrained model with the optimal solutions of the classic survivable network design (SND) model [20] in terms of reliability and cost. We take the deterministic graph instance Di-yuan, and consider two cases for the reliability of the edges: high reliability, with edge failure probabilities generated according to the exponential distribution with mean $\lambda = 0.01$, and low reliability, where $\lambda = 0.1$ is used. For each case (high and low reliability) we obtain five different sample approximations, each with 500 scenarios based on the fixed edge failure probabilities that we generated. We solve each instance with various values of risk tolerance parameter ϵ. (For the high edge reliability case ϵ is ranged from 0 to 0.1, incremented by 0.01, and for the low edge-reliability case ϵ is ranged from 0 to 0.4, incremented by 0.05.) After solving those instances, we estimate the failure probability of the solutions by checking them with a large sample of 10^6 scenarios generated using the same edge failure probabilities. We show the scatter plots of the objective (cost) and the estimated failure probability (risk) of all solutions in

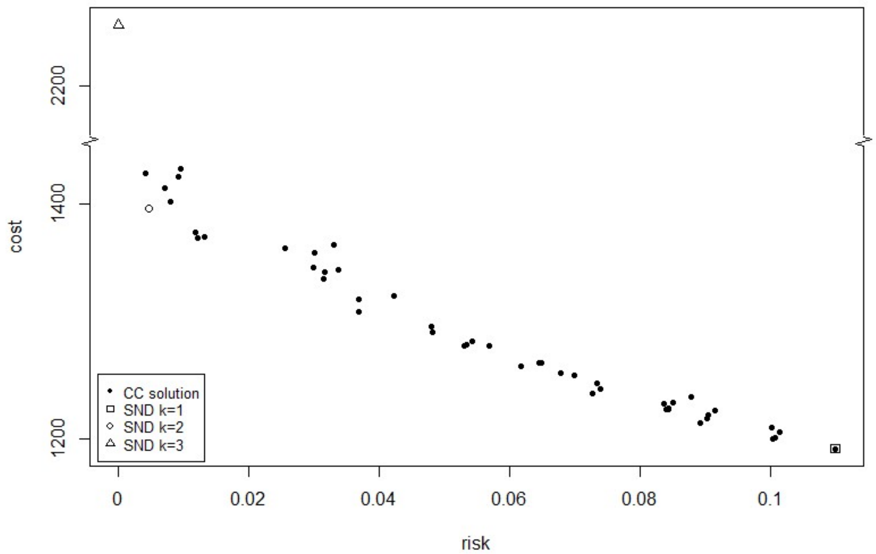

Figure 4: Scatter plot of solutions obtained using the chance-constrained formulation using varying ϵ values and optimal SND solutions obtained using $k = 1, 2, 3$, for a network with low average edge failure rate ($\lambda = 0.01$).

Figures 4 and 5. For comparison with the SND model, we also show the optimal SND solutions on the plot, using different edge connectivity requirements k, $k = 1, 2, 3$ for the high-edge reliability case, and $k = 2, 3, 4$ for low edge-reliability case.

We observe from Figure 4 that, in the case of high edge-reliability, the optimal SND solutions lie on the efficient frontier of cost and risk identified by the chance-constrained solutions. However, when the edges are less reliable, Figure 5 demonstrates that the optimal SND solutions obtained with $k = 2$ and $k = 3$ are dominated by chance-constrained solutions, in the sense that they have both higher risk and higher cost than some chance-constrained solutions. Furthermore, the discrete nature of the SND reliability requirement (minimum connectivity k must be an integer) leads to wide gaps: increasing k by just one simultaneously induces a large reduction in risk and increase in cost. In contrast, this example demonstrates that the chance-constrained model has the potential to yield a wide variety of solutions on the efficient frontier of cost and risk.

5 Concluding Remarks

We studied solution methods of a chance-constrained reliable network design model. We first studied the problem of finding a minimum cost set of arcs in a directed graph such that node s is connected to node t with high probability. After showing this problem is $\mathcal{NP}$-hard, we proposed two formulations and corresponding branch-and-cut algorithms to solve them. The first formulation is based on the max-flow min-cut theorem applied to scenarios of the random graph, thus enforcing the existence of an s-t path in selected scenarios by adding inequalities derived from s-t cuts in scenarios of the random graph. The second formulation uses an extension of the s-t cut concept to random graphs to define the set of feasible solutions. We analyzed the strength of valid inequalities for both formulations. Computational experiments indicate that while a simple extended formulation has difficulty solving this problem for even a moderate number of scenarios, the branch-and-cut algorithms we proposed can solve instances having up to 200 nodes, 2000 arcs, and 5000 scenarios.

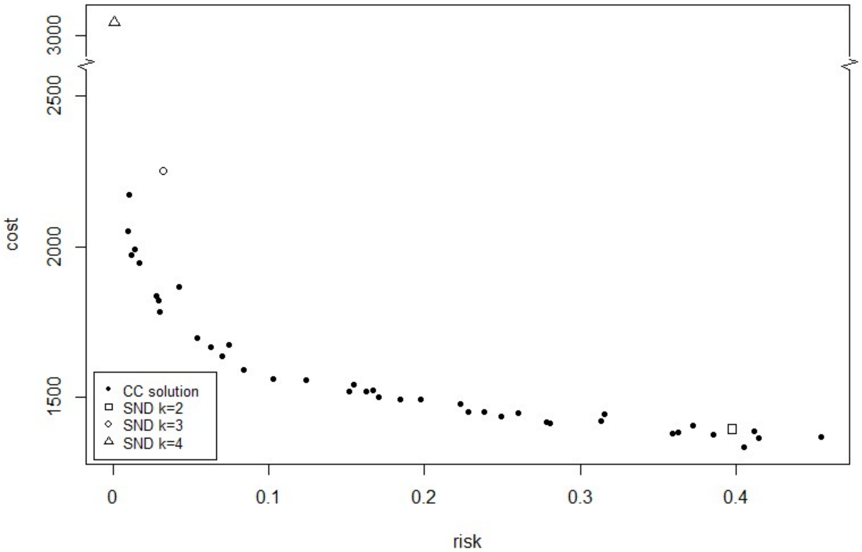

Figure 5: Scatter plot of solutions obtained using the chance-constrained formulation using varying ϵ values and optimal SND solutions obtained using $k = 2, 3, 4$, for a network with high average edge failure rate ($\lambda = 0.1$)

We also demonstrated how the same ideas can be used to design undirected networks in which all nodes must be connected simultaneously with high probability. We extended both formulations for the chance-constrained s-t connected network design problem to this case, with the simple modification that we use probabilistic graph cuts in place of probabilistic s-t cuts. However, in this case an additional class of valid inequalities – the partition inequalities – is important. Once again, we found that a probabilistic extension of these inequalities can be obtained, and is useful for computational performance.

Although the methods we have presented are promising, because the root gaps obtained by these formulations were often large, there appears to be room for improvement in deriving methods for separating strong valid inequalities at fractional solutions. This appears to be more important for the full connectivity network design instances, which we found more challenging than the s-t connected instances.

Another possible direction for future work is to explore connections between the formulations presented in this paper with the *stochastic network interdiction problem* (SNIP) with binary interdiction [17]. In particular, it is straightforward to show that the set of all minimal probabilistic s-t cuts in a graph and the set of all minimal feasible solutions to (1) form a *blocking pair of clutters* (see [31]). Using the theory of blocking clutters, this immediately yields a formulation analogous to (15) for the problem of finding a minimum weight probabilistic s-t cut in a random graph, where feasible solutions of (1) yield inequalities that define the set of probabilistic s-t cuts, analogous to (15b). The only difference between this problem and SNIP is that in SNIP the role of objective and constraint are swapped: the goal is to choose a set of arcs that maximizes probability of disconnecting s from t, with a budget constraint on the sum of the weights in the selected arcs. It would be interesting to investigate whether the approach used in this paper may be useful for this class of problems.

Acknowledgments.

This research has been supported in part by the National Science Foundation under grant CMMI-0952907. The authors are grateful to the reviewers and a technical editor for comments and suggestions that significantly improved the presentation and results of this paper.

References

[1] Ahuja, R., Magnanti, T., Orlin, J.: Network Flows. Prentice Hall, New Jersey (1993)

[2] Andreas, A.K., Smith, J.C.: Mathematical programming algorithms for two-path routing problems with reliability considerations. INFORMS J. Comput. **20**, 553–564 (2008)

[3] Andreello, G., Caprara, A., Fischetti, M.: Embedding cuts in a branch and cut framework: a computational study with 0,1/2-cuts. INFORMS J. Comput. **19(2)**, 229–238 (2007)

[4] Aneja, Y.: An integer linear programming approach to the steiner problem in graphs. Networks **10**, 167–178 (1980)

[5] Atamtürk, A., Nemhauser, G.L., Savelsbergh, M.W.P.: The mixed vertex packing problem. Math. Program. **89**, 35–53 (2000)

[6] Baïou, M., Barahona, F., A.R.Mahjoub: Separation of partition inequalities. Mathematics of Operations Research **25**, 243–254 (2000)

[7] Beasley, J.: OR-library: Distributing test problems by electronic mail. Journal of the Operational Research Society **41**, 1069–1072 (1990)

[8] Beraldi, P., Bruni, M.: An exact approach for solving integer problems under probabilistic constraints with random technology matrix. Ann. Oper. Res. **177**, 127–137 (2010)

[9] Beraldi, P., Bruni, M.E., Guerriero, F.: Network reliablity design via joint probabilistic constraints. IMA Journal of Management Mathematics **21**, 213–226 (2010)

[10] Beraldi, P., Ruszczyński, A.: The probabilistic set-covering problem. Oper. Res. **50**, 956–967 (2002)

[11] Calafiore, G., Campi, M.: Uncertain convex programs: randomized solutions and confidence levels. Math. Program. **102**, 25–46 (2005)

[12] Calafiore, G., Campi, M.: The scenario approach to robust control design. IEEE Trans. Automat. Control **51**, 742–753 (2006)

[13] Campi, M., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. J. Optim. Theory Appl. **148**, 257–280 (2011)

[14] Charnes, A., Cooper, W.W.: Deterministic equivalents for optimizing and satisficing under chance constraints. Oper. Res. **11**, 18–39 (1963)

[15] Charnes, A., Cooper, W.W., Symonds, G.H.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. Manage. Sci. **4**, 235–263 (1958)

[16] Chopra, S.: On the spanning tree polyhedron. Operation Research Letters **8**, 25–29 (1989)

[17] Cormican, K., Morton, D., Wood, R.: Stochastic network interdiction. Oper. Res. **46**, 184–197 (1998)

[18] Dentcheva, D., Prékopa, A., Ruszczyński, A.: Concavity and efficient points of discrete distributions in probabilistic programming. Math. Program. **89**, 55–77 (2000)

[19] Fischetti, M., Monaci, M.: Robustness by cutting planes and the uncertain set covering problem (2009). Preliminary draft

[20] Grötschel, M., Monma, C., Stoer, M.: Polyhedral and computational investigations for designing communication networks with high survivability requirements. Oper. Res. **43**, 1012–1024 (1995)

[21] Günlük, O., Pochet, Y.: Mixing mixed-integer inequalities. Math. Program. **90**, 429–457 (2001)

[22] Hao, J., Orlin, J.: A faster algorithm for finding the minimum cut in a directed graph. J. of Algorithms **17**, 424–446 (1994)

[23] Küçükyavuz, S.: On mixing sets arising in chance-constrained programming. Math. Program. **132**, 31–56 (2012)

[24] LEMON – Library for Efficient Modeling and Optimization in Networks. `http://lemon.cs.elte.hu/` (2009)

[25] Luedtke, J.: An integer programming and decomposition approach to general chance-constrained mathematical programs. In: F. Eisenbrand, F. Shepherd (eds.) IPCO 2010, Lecture Notes in Comput. Sci., pp. 271–284. Springer-Verlag, Berlin (2010)

[26] Luedtke, J.: An branch-and-cut decomposition algorithm for solving general chance-constrained mathematical programs (2011). Preprint available at: www.optimization-online.org/DB_HTML/2011/05/3033.html

[27] Luedtke, J., Ahmed, S.: A sample approximation approach for optimization with probabilistic constraints. SIAM J. Optim. **19**, 674–699 (2008)

[28] Luedtke, J., Ahmed, S., Nemhauser, G.L.: An integer programming approach for linear programs with probabilistic constraints. Math. Program. **12**, 247–272 (2010)

[29] Magnanti, T.L., Raghavan, S.: Strong formulations for network design problems with connectivity requirements. Networks **45**, 61–79 (2005)

[30] Nash-Williams, C.: Edge-disjoint spanning trees of finite graphs. J. London Math. Soc. **36**, 445–450 (1961)

[31] Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley (1988)

[32] Nemirovski, A., Shapiro, A.: Scenario approximation of chance constraints. In: G. Calafiore, F. Dabbene (eds.) Probabilistic and Randomized Methods for Design Under Uncertainty, pp. 3–48. Springer, London (2005)

[33] Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. SIAM J. Optim. **17**, 969–996 (2006)

[34] Orlowski, S., Wessäly, R., Pioro, M., Tomaszewski, A.: Survivable network design data library (2012). `http://sndlib.zib.de`

[35] Prékopa, A.: Dual method for the solution of a one-stage stochastic programming problem with random rhs obeying a discrete probability distribution. ZOR - Methods and Models of Operations Research **34**, 441–461 (1990)

[36] Prékopa, A.: Probabilistic programming. In: A. Ruszczyński, A. Shapiro (eds.) Stochastic Programming, *Handbooks Oper. Res. Management Sci.*, vol. 10, pp. 267–351. Elsevier (2003)

[37] Ruszczyński, A.: Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. Math. Program. **93**, 195–215 (2002)

[38] Saxena, A., Goyal, V., Lejeune, M.: MIP reformulations of the probabilistic set covering problem. Math. Program. **121**, 1–31 (2009)

[39] Schrijver, A.: Combinatorial Optimization. Springer (2003)

[40] Sorokin, A., Boginski, V., Nahapetyan, A., Pardalos, P.M.: Computational risk management techniques for fixed charge network flow problems with multiple uncertain arc failures. J. of Combinatorial Optimization (2011). Accepted

[41] Tanner, M., Ntaimo, L.: IIS branch-and-cut for joint chance-constrained programs and application to optimal vaccine allocation. European J. Oper. Res. **207**, 290–296 (2010)

[42] Tutte, W.: On the problem of decomposing a graph into n connected factors. J. London Math. Soc. **36**, 221–230 (1961)