

Chance-Constrained Multi-Terminal Network Design Problems

Yongjia Song ¹ and Minjiao Zhang ²

¹Department of Statistical Sciences and Operations Research, Virginia Commonwealth University
Richmond, VA 23284, ysong3@vcu.edu

²Department of Information Systems, Statistics, and Management Science, The University of Alabama
Tuscaloosa, AL 35487, mzhang@cba.ua.edu

April 15, 2015

Abstract

We consider a reliable network design problem under uncertain edge failures. Our goal is to select a minimum-cost subset of edges in the network to connect multiple terminals together with high probability. This problem can be seen as a stochastic variant of the Steiner tree problem. We propose two scenario-based Steiner cut formulations, study the strength of the proposed valid inequalities, and develop a branch-and-cut solution method. We also propose an LP-based separation for the scenario-based directed Steiner cut inequalities using Benders feasibility cuts, leveraging the success of the directed Steiner cuts for the deterministic Steiner tree problem. In our computational study, we test our branch-and-cut method on instances adapted from graphs in SteinLib Testdata Library with up to 100 nodes, 200 edges, and 17 terminals. The performance of our branch-and-cut method demonstrates the strength of the scenario-based formulations and the benefit from adding the additional valid inequalities that we propose.

Keywords: Stochastic network design, Steiner tree, Chance constraint, Benders decomposition

1 Introduction

Network design with connectivity requirements is an interesting and well-studied optimization problem. For example, the survivable network design problems (Grötschel et al. [16], Magnanti and Raghavan [26]) select a minimum-cost set of edges for the design of a robust network such that the connectivity requirements for a given set of nodes are satisfied. The robustness of the network is guaranteed by designing a number of edge-disjoint paths between specific pairs of nodes. The idea of using edge-disjoint paths is to hedge against the uncertain edge failure events.

Instead of accounting for the potential edge failure by imposing specific network structures like the edge-disjoint paths, stochastic network design incorporates random edge failure probability information in the design of the network. Song and Luedtke [28] study chance-constrained network design problems, where a single s - t pair of nodes or all pairs of nodes are required to be connected with high probability. It generalizes the shortest path problem and the minimum spanning tree problem in a chance-constrained setting. The case of connecting all pairs of nodes has also been studied in Beraldi et al. [4] using an extended mixed integer program (MIP) and a heuristic solution approach. Andreas et al. [2] combine the probability information and the specific network structure, by selecting a minimum-cost set of arc-disjoint paths between a pair of nodes to guarantee with high probability that there exists at least one path after a random failure.

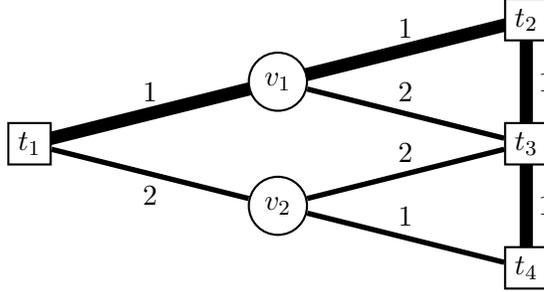


Figure 1: An example of the deterministic Steiner tree problem

In this paper, we study a chance-constrained multi-terminal connected network design problem on an undirected graph where edges may fail randomly. This problem is a generalization of the reliable s - t connected network design and fully connected network design problems studied in Song and Luedtke [28]. Reliably connected network design problems arise in many communication and transportation network applications. For example, the broadband wireless network, which uses microwave links (Claßen et al. [13]) to deliver data, has uncertain microwave capacity due to the channel conditions. To maintain a high connection probability between critical facilities, such as police stations and hospitals, we need to determine how to design the private network based on the available network. Another example appears in supply chain networks, in which all distribution centers and warehouses are needed to be connected with high probability to ensure the delivery of products to customers in local areas.

The deterministic version of our problem reduces to the *Steiner tree problem*, which is a well-known NP-hard problem. Consider an undirected graph $G = (V, E)$ with node set V and edge set E . Each edge $e \in E$ is associated with a nonnegative cost c_e . The goal of the deterministic Steiner tree problem is to select a minimum-cost subset of edges $Q \subseteq E$ such that a given subset of nodes $T \subseteq V$, $|T| \geq 2$, are connected in the graph $G(Q)$ induced by the selection Q . We refer to set T as the *terminal set* and nodes in set T as *terminal nodes*. The Steiner tree problem is defined as:

$$\min_{Q \subseteq E} \left\{ \sum_{e \in Q} c_e \mid \exists \text{ an } s\text{-}t \text{ path in graph } G(Q) \text{ for any } s, t \in T \right\}. \quad (1)$$

For example, we have a graph G as shown in Figure 1 with the terminal set $T = \{t_1, t_2, t_3, t_4\}$. The cost of each edge is given in the figure. The selection $Q = \{\{t_1, v_1\}, \{v_1, t_2\}, \{t_2, t_3\}, \{t_3, t_4\}\}$ gives a Steiner tree connecting all terminal nodes with a minimum cost of 4.

Taking into account that edges may fail randomly, several stochastic variants of the Steiner tree problem have been studied. A two-stage stochastic Steiner tree problem is proposed by Bomze et al. [5]. In their framework, some edges of the network are constructed in the first stage; after observing the realization of random edge failure, an additional set of edges is constructed in the second stage to ensure a connection among all terminal nodes. In this paper, we propose a chance-constrained model to directly control the probability of disconnectedness, rather than minimizing the expected cost of installing additional edges in the second stage due to random failure. To be more specific, we consider an undirected graph $\tilde{G} = (V, \tilde{E})$, where $\tilde{E} \subseteq E$ represents the set of available edges after the random edge failure. Our goal is to select a minimum-cost subset of edges $Q \subseteq E$ such that the terminal nodes in T are connected in the random graph $\tilde{G}(Q \cap \tilde{E})$ induced by the selection Q with probability at least $1 - \epsilon$, where $\epsilon \in [0, 1)$ is a given risk tolerance parameter. Our problem

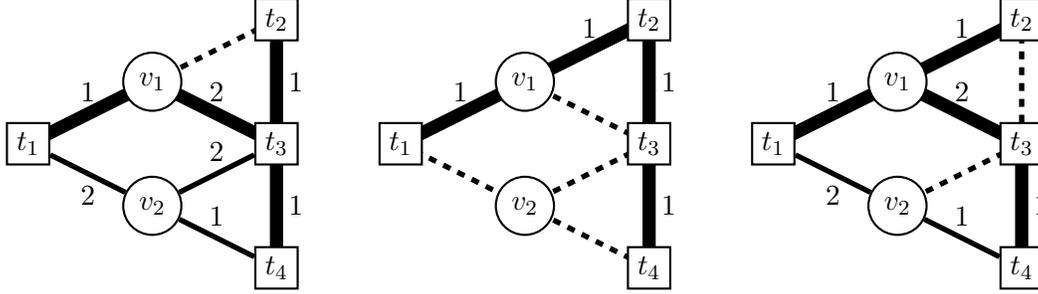


Figure 2: An example of the chance-constrained Steiner tree problem

is defined as

$$\min_{Q \subseteq E} \left\{ \sum_{e \in Q} c_e \mid \mathbb{P} \left\{ \exists \text{ an } s\text{-}t \text{ path in graph } \tilde{G}(Q \cap \tilde{E}) \text{ for any } s, t \in T \right\} \geq 1 - \epsilon \right\}. \quad (2)$$

We refer to this problem (2) as the *chance-constrained Steiner tree problem*. Note that our problem reduces to the case of reliably connecting a single pair of nodes when $|T| = 2$, and reduces to reliably connecting all pairs of nodes when $T = V$, both of which are studied in Song and Luedtke [28]. Figure 2 gives an example of a stochastic Steiner tree instance with three equally likely edge failure scenarios. In each scenario, the failed edges are represented by dotted links. Let the risk tolerance $\epsilon = 0.1$, i.e., all nodes in T must be connected in all three scenarios. The optimal selection for the deterministic Steiner tree problem, $Q = \{\{t_1, v_1\}, \{v_1, t_2\}, \{t_2, t_3\}, \{t_3, t_4\}\}$, fails to connect T in scenario 3, and therefore is infeasible to the chance-constrained Steiner tree problem (2). $Q' = \{\{t_1, v_1\}, \{v_1, t_2\}, \{t_2, t_3\}, \{t_3, t_4\}, \{v_1, t_3\}\}$ is a minimum-cost selection of edges that connects T in all three scenarios. Note that this selection Q' is not a Steiner tree, as “redundant” edges are introduced to ensure the connectivity in multiple edge failure scenarios.

The only constraint in model (2) is a chance constraint, which was first proposed and studied in Charnes et al. [6], Charnes and Cooper [7, 8]. Chance-constrained programs are difficult to solve because the feasible region of the corresponding formulation is in general nonconvex. In our problem (2), there are a finite number of edge failure scenarios, although this number could be very large in general (up to $2^{|E|}$). To make it computationally tractable, a sample average approximation (SAA) with a moderate number of scenarios is solved, where the set of scenarios is generated by drawing a sample from the collection of all possible scenarios (Luedtke and Ahmed [24]). Let K be the set of scenarios with $|K| = N$. We denote the probability that scenario k happens by p_k ; thus $\sum_{k \in K} p_k = 1$. Motivated by Monte Carlo sampling, we assume that each scenario k happens with the same probability, i.e., $p_k = 1/N$. Given a risk tolerance parameter ϵ , the maximum number of failed scenarios allowed is $q := \lfloor \epsilon N \rfloor$. We define $G_k = (V, E_k)$ as the graph in scenario $k \in K$, where $E_k = \{e \in E \mid \text{edge } e \text{ does not fail in scenario } k\}$. The SAA problem of a chance-constrained program could be formulated as an MIP, by introducing a binary decision variable for each scenario k to indicate whether or not constraints are satisfied in that scenario. Valid inequalities are proposed to strengthen the MIP formulation by exploiting the substructures of the problem, including mixing sets and continuous mixing sets. For example, Küçükyavuz [19] and Luedtke et al. [25] develop valid inequalities from a mixing set substructure. Zhang et al. [30] propose valid inequalities from a continuous mixing set substructure (van Vyve [29]) for a multistage chance-constrained stochastic program. Luedtke [23] combines the branch-and-cut method and Benders decomposition to solve a two-stage chance-constrained program with no recourse cost. Recently, Liu et al. [21] develop

a decomposition method with specialized optimality and feasibility cuts for a two-stage chance-constrained program with additional recourse cost.

In Section 2, we introduce two formulations for the chance-constrained Steiner tree problem (the scenario-based undirected Steiner cut formulation and the scenario-based directed Steiner cut formulation), and two sets of additional valid inequalities (probabilistic Steiner p -cut inequalities and mixing inequalities). In Section 3, we study the strengths of the scenario-based Steiner partition inequalities and the probabilistic Steiner p -cut inequalities. In Section 4, we present the computational results with different versions of the proposed branch-and-cut approach on our test instances. Computational results show that the scenario-based Steiner cut formulation with Steiner partition inequalities gives a tighter relaxation bound than the probabilistic Steiner p -cut formulation. Benders feasibility cuts and mixing inequalities are beneficial for strengthening the relaxation bound of the scenario-based Steiner cut formulation, at a price of significant extra computational time.

2 Problem formulations

2.1 Deterministic Steiner tree formulations

The Steiner tree problem (1) can be formulated by using the concept of *Steiner cuts* (Chopra and Rao [11, 12]). Given a partition $\{U, W\}$ of V such that $U \cap T \neq \emptyset$ and $W \cap T \neq \emptyset$, a Steiner cut can be defined as $\delta(W) = \{\{u, v\} \in E \mid u \in W, v \in U \text{ or } u \in U, v \in W\}$. In order to connect all nodes in T , we must ensure that at least one edge in each Steiner cut $\delta(W)$ has to be selected. Introducing a binary decision variable x_e to indicate whether or not an edge $e \in E$ is selected, the Steiner tree problem can be formulated based on the Steiner cuts:

$$\min \sum_{e \in E} c_e x_e \tag{3a}$$

$$\text{s.t. } \sum_{e \in \delta(W)} x_e \geq 1, \forall W \subseteq V, r \notin W, W \cap T \neq \emptyset, \tag{3b}$$

$$x_e \in \{0, 1\}, \forall e \in E, \tag{3c}$$

where $r \in T$ is an arbitrarily selected terminal node. We refer to model (3) as the *undirected Steiner cut formulation*.

It is known that the linear programming (LP) relaxation of the undirected Steiner cut formulation (3) is weak. Valid inequalities have been proposed to strengthen formulation (3). For example, given a partition (V_1, V_2, \dots, V_p) of the node set V with $p \geq 2$, i.e., $\cup_{i=1}^p V_i = V$ and $V_i \cap V_j = \emptyset$ for $i \neq j, \forall i, j = 1, 2, \dots, p$, suppose that $V_i \cap T \neq \emptyset$ for $i = 1, 2, \dots, p$, then a Steiner partition inequality is defined as:

$$\sum_{e \in E(V_1, V_2, \dots, V_p)} x_e \geq (p - 1), \tag{4}$$

where $E(V_1, V_2, \dots, V_p) := \{e = \{i, j\} \in E \mid \exists l, m = 1, 2, \dots, p, l \neq m, \text{ such that } i \in V_l, j \in V_m\}$. The Steiner partition inequalities are NP-hard to separate (Chopra and Rao [11]). There exists a $(2 - \frac{2}{p})$ approximation algorithm proposed by Chekuri et al. [9] based on Gomory-Hu trees. Other families of valid inequalities in the x space are also proposed and studied in Chopra and Rao [11], such as the odd-hole inequalities.

Furthermore, Chopra and Rao [11, 12] develop a direct Steiner cut formulation for the Steiner tree problem. They construct a directed graph $B = (V, A)$ by introducing two directed arcs (i, j)

and (j, i) for each edge $\{i, j\} \in E$. Note that, throughout this paper, we use $\{i, j\}$ to denote the undirected edge connecting nodes i and j , and use (i, j) to denote the directed arc from node i to node j . The costs of arcs (i, j) and (j, i) in the directed graph B are the same as the cost of the corresponding edge $\{i, j\}$ in the undirected graph G . A new binary decision variable y_a is introduced to indicate whether or not an arc $a \in A$ is selected. This leads to the following *directed Steiner cut formulation*:

$$\min \sum_{a \in A} c_a y_a \quad (5a)$$

$$\text{s.t.} \quad \sum_{a \in \delta^-(W)} y_a \geq 1, \quad \forall W \subseteq V, r \notin W, W \cap T \neq \emptyset, \quad (5b)$$

$$y_a \in \{0, 1\}, \quad \forall a \in A, \quad (5c)$$

where $\delta^-(W) = \{a = (i, j) \in A \mid i \notin W, j \in W\}$.

Compared to the undirected Steiner cut formulation (3), the number of variables in the directed Steiner cut formulation (5) is doubled. However, the LP relaxation of the directed Steiner cut formulation (5) is as least as strong as the LP relaxation of the undirected Steiner cut formulation (3) plus a whole set of valid inequalities including the Steiner partition inequalities, the odd-hole inequalities, etc. (Chopra and Rao [11, 12]).

The deterministic Steiner tree problem and its variants have also been well studied in computation. For example, Chopra et al. [10] solve the Steiner tree problem using a branch-and-cut framework. Lucena and Beasley [22] apply a branch-and-cut framework based on a strengthened minimum spanning tree formulation. Polzin and Daneshmand [27] compare several existing MIP formulations for the Steiner tree problem.

2.2 Scenario-based undirected Steiner cut formulation for the chance-constrained Steiner tree problem

Given a finite set K of the edge failure scenarios that may come from a Monte Carlo sample, we formulate the chance-constrained Steiner tree problem (2) as an MIP. We introduce a binary variable z_k for each scenario $k \in K$, where $z_k = 1$ indicates that the terminal nodes in T are connected in scenario k , and $z_k = 0$ otherwise. By adapting the undirected Steiner cut formulation (3) for the deterministic Steiner tree problem, we formulate the chance-constrained Steiner tree problem as:

$$\min \sum_{e \in E} c_e x_e \quad (6a)$$

$$\text{s.t.} \quad \sum_{e \in \delta_k(W)} x_e \geq z_k, \quad \forall k \in K, W \subseteq V, r \notin W, W \cap T \neq \emptyset, \quad (6b)$$

$$\sum_{k \in K} z_k \geq N - q, \quad (6c)$$

$$x_e \in \{0, 1\}, \quad \forall e \in E; z_k \in \{0, 1\}, \quad \forall k \in K, \quad (6d)$$

where $\delta_k(W) = \{\{i, j\} \in E_k \mid i \notin W, j \in W \text{ or } i \in W, j \notin W\}$. It is clear that when $z_k = 1$, inequalities (6b) ensure that, in scenario k , all the terminal nodes in T are connected, which therefore guarantees that formulation (6) is valid. We refer to inequalities (6b) as the *scenario-based undirected Steiner cut inequalities*, and we define the scenario-based feasible set $X_k = \{(x, z_k) \in \{0, 1\}^{|E|} \times \{0, 1\} \mid (x, z_k) \text{ satisfies (6b)}\}$. Furthermore, we observe that the integrality constraints on variables z can be relaxed.

Proposition 1. *It is valid to relax the integrality constraints on variables z in (6).*

Proof. Consider an optimal solution (\hat{x}, \hat{z}) , and $\hat{z} \in [0, 1]^N$. For all k such that $0 < \hat{z}_k < 1$, because $\sum_{e \in \delta_k(W)} \hat{x}_e$ is an integer, we can round up \hat{z}_k to 1, and inequalities (6b) and (6c) remain being satisfied. The objective value of formulation (6) does not change by this rounding-up procedure. \square

Although there are exponentially many scenario-based undirected Steiner cut inequalities (6b), we can solve formulation (6) using delayed constraint generation within a branch-and-cut framework. Iteratively, we solve a relaxed master problem with a subset of constraints (6b) within the branch-and-bound procedure, and obtain a node relaxation solution \hat{x} . When $\hat{x} \in \{0, 1\}^{|E|}$, we must check if there exists any valid inequality (6b) that is violated by \hat{x} in some scenario $k \in K$. When \hat{x} is fractional, this could be done optionally. We define $\bar{X}_k = \{(x, z_k) \in [0, 1]^{|E|} \times [0, 1] \mid (x, z_k) \text{ satisfies (6b)}\}$. Checking the existence of a violated valid inequality (6b) is referred to as the *separation* over \bar{X}_k . When a violated inequality (6b) is identified, we add it to the relaxed master problem as a feasibility cut, and then we continue the branch-and-cut process.

Suppose that we have a relaxation solution (\hat{x}, \hat{z}_k) , where $\hat{x} \in [0, 1]^{|E|}$, and $\hat{z}_k \in (0, 1)$. The separation problem over \bar{X}_k can be formulated as:

$$\hat{v}_k = \min \left\{ \sum_{e \in C_k} \hat{x}_e \mid C_k \text{ is a Steiner cut that separates node } r \text{ and some node } l \in T \setminus \{r\} \right\}. \quad (7)$$

If $\hat{v}_k < \hat{z}_k$, then we have found a Steiner cut C_k such that the scenario-based Steiner cut inequality $\sum_{e \in C_k} x_e \geq z_k$ is violated by (\hat{x}, \hat{z}_k) . The separation problem (7) can be solved by solving a maximum r - l flow problem separately for each terminal node $l \in T \setminus \{r\}$, which can be done in $O(|V|^3 \times |T|)$ time using an FIFO preflow-push algorithm (Ahuja et al. [1]). If $\hat{x} \in \{0, 1\}^{|E|}$, then (7) could be solved by a simple graph search from node r , using edges $e \in E_k$ such that $\hat{x}_e = 1$. Let $S_k(\hat{x})$ be the set of nodes that can be reached from node r , if there exists a terminal node $l \in T \setminus \{r\}$ such that $l \notin S_k(\hat{x})$, then cut $C_k = \delta_k(S_k(\hat{x}))$ separates r and l in scenario k and its cut value $\sum_{e \in C_k} \hat{x}_e$ equals to 0. Therefore, inequality (6b) with the constructed C_k cuts off the infeasible solution (\hat{x}, \hat{z}) . This simple graph search routine can also be used as a heuristic for the separation of a fractional \hat{x} .

The scenario-based undirected Steiner cut formulation (6) can be strengthened by the scenario-based Steiner partition inequalities. Given a partition (V_1, V_2, \dots, V_p) of the node set V with $p \geq 2$ and $V_i \cap T \neq \emptyset, \forall i = 1, 2, \dots, p$, a *scenario-based Steiner partition inequality* is defined as:

$$\sum_{e \in E_k(V_1, V_2, \dots, V_p)} x_e \geq (p-1)z_k, \quad (8)$$

where $E_k(V_1, V_2, \dots, V_p) := \{e = \{i, j\} \in E_k \mid \exists l, m = 1, 2, \dots, p, l \neq m, \text{ such that } i \in V_l, j \in V_m\}$. The undirected scenario-based Steiner cut (6b) can be seen as a special case of (8) with $p = 2$. It is clear that the scenario-based Steiner partition inequality is NP-hard to separate. The $(2 - \frac{2}{p})$ -approximation algorithm proposed by Chekuri et al. [9] for the deterministic Steiner tree problem can be adapted here as a heuristic separation for (8), with the same approximation guarantee. Notice that when $p = 2$, this algorithm is exact, which corresponds to the exact separation of inequality (6b). Again, if $\hat{x} \in \{0, 1\}^{|E|}$, then the exact separation of inequality (8) can be done by a simple graph search on graph G_k , using edges $e \in E_k$ such that $\hat{x}_e = 1$. Let $V_1(\hat{x}), V_2(\hat{x}), \dots, V_p(\hat{x})$ be the set of connected components of graph G_k as a result of the graph search with $p \geq 2$, then inequality (8) with $E_k(V_1(\hat{x}), V_2(\hat{x}), \dots, V_p(\hat{x}))$ is violated by (\hat{x}, \hat{z}_k) since $\sum_{e \in E_k(V_1(\hat{x}), V_2(\hat{x}), \dots, V_p(\hat{x}))} \hat{x}_e = 0$. This graph search routine contains the graph search from node

r for the separation of the scenario-based undirected Steiner cut inequality (6b) as a special case with $p = 2$. It is clear that this routine can also be used as a heuristic separation for a fractional \hat{x} .

2.3 Scenario-based directed Steiner cut formulations for the chance-constrained Steiner tree problem

In this subsection, we adapt the deterministic directed Steiner cut formulation (5) for the chance-constrained Steiner tree problem. Let $\delta_k^-(W) = \{(i, j) \in A_k \mid i \notin W, j \in W\}$, where $A_k = \{(i, j), (j, i) \mid \{i, j\} \in E_k\}$. We introduce a directed arc variable y_a^k for each scenario $k \in K$ and each arc $a \in A_k$ to indicate whether or not arc a is selected in scenario k . Then we propose the *scenario-based directed Steiner cut formulation* as:

$$\min \sum_{e \in E} c_e x_e \tag{9a}$$

$$\text{s.t.} \quad \sum_{a \in \delta_k^-(W)} y_a^k \geq z_k, \quad \forall k \in K, \forall W \subseteq V \text{ s.t. } r \notin W, W \cap T \neq \emptyset, \tag{9b}$$

$$y_{ij}^k + y_{ji}^k \leq x_{\{i,j\}}, \quad \forall \{i, j\} \in E_k, k \in K \tag{9c}$$

$$\sum_{k \in K} z_k \geq N - q, \tag{9d}$$

$$x_e \in \{0, 1\}, \quad \forall e \in E; y_a^k \in \{0, 1\}, \quad \forall a \in A_k, k \in K; z_k \in [0, 1], \quad \forall k \in K, \tag{9e}$$

where inequality (9c) indicates that if $x_{\{i,j\}} = 1$, at most one of y_{ij}^k and y_{ji}^k can be chosen.

Proposition 2. *It is valid to relax the integrality constraints (9) on variables y .*

Proof. Given a feasible solution $(\hat{x}, \hat{y}, \hat{z})$ to (9) with a fractional y_a^k , we can add up constraints $y_{ij}^k + y_{ji}^k \leq x_{\{i,j\}}$ for all $(i, j) \in A$, so that constraints $\sum_{e \in \delta_k^-(W)} x_e \geq z_k$ are satisfied for W with $r \notin W, W \cap T \neq \emptyset$. Thus (x, z) is feasible to (6). It is clear that the contrary is also true by appropriate path orientation in each scenario $k \in K$. \square

Recall that in the deterministic Steiner tree problem, the directed Steiner cut formulation (5) is at least as strong as the undirected Steiner cut formulation (3) with other known inequalities, including the Steiner partition inequalities and the odd-hole inequalities. It is clear that the scenario-based directed Steiner cut formulation (9) is at least as strong as the scenario-based undirected Steiner cut formulation (6) with the scenario-based Steiner partition inequalities (8). However, the scenario-based directed Steiner cut formulation (9) is significantly larger than the scenario-based undirected Steiner cut formulation (6), especially when the scenario set K is large. In the deterministic Steiner tree problem, this is not a concern since the number of variables is only doubled. It would be interesting if we only need to keep one copy of variables y for (9), i.e., $y^k = y^{k'}$, for any $k, k' \in K$. However, it has been pointed out in Song and Luedtke [28] that such a restriction is not valid.

In spite of its strength, formulation (9) is impractical to be solved directly when the number of scenarios is large due to the N copies of variables $y^k, k \in K$. On the other hand, formulation (6) with the scenario-based undirected Steiner cuts (6b) may yield a weak relaxation bound. In what follows, we propose a branch-and-cut algorithm to solve the scenario-based directed Steiner cut formulation (9), using the cuts projected from it onto the (x, z) space from the perspective of Benders decomposition in two-stage stochastic programs.

Benders feasibility cuts Instead of solving (9) directly, we treat model (9) as a two-stage formulation. Variables x and z are treated as the first-stage decision variables, and variables y^k are treated as the auxiliary second-stage variables, which are introduced to ensure the feasibility of (x, z_k) with respect to $\text{proj}_{(x, z_k)}(Y_k)$ in scenario $k \in K$, where

$$Y_k := \{x \in [0, 1]^{|E|}, y^k \in [0, 1]^{|A_k|}, z_k \in [0, 1] \mid (x, y^k, z_k) \text{ satisfies (9b) and (9c)}\}.$$

We next consider using the strength of formulation (9) by Benders decomposition (Benders [3]). Benders decomposition is developed based on the idea of partitioning critical decision variables and auxiliary decision variables, and delaying the generation of constraints associated with the auxiliary decision variables. Benders decomposition can be applied to project out our continuous auxiliary variables (y in (9)), by adding Benders feasibility cuts in the (x, z) space to cut off any $(\hat{x}, \hat{z}_k) \notin \text{proj}_{(x, z_k)}(Y_k)$. Benders decomposition has been widely applied for solving MIPs, especially on network design problems. For a detailed survey on various applications of Benders decomposition on network design problems, we refer readers to Costa [14].

We now consider generating a Benders feasibility cut by solving a feasibility variant of formulation (9). With a fixed scenario $k \in K$, the number of variables in (9) is small, however, it has exponentially many constraints. A Benders feasibility cut generation LP based on (9) for scenario k can be formulated as:

$$\min \xi \tag{10a}$$

$$\text{s.t. } \sum_{a \in \delta_k^-(W)} y_a^k \geq \hat{z}_k, \forall W \subseteq V, r \notin W, W \cap T \neq \emptyset, \tag{10b}$$

$$\xi - y_{ij}^k - y_{ji}^k \geq -\hat{x}_{\{i, j\}}, \forall \{i, j\} \in E_k, \tag{10c}$$

$$y_a^k \geq 0, \forall a \in A_k; \xi \geq 0. \tag{10d}$$

The dual of (10) is given by:

$$\max \sum_{C \in \mathcal{C}} \mu_C \hat{z}_k - \sum_{e \in E_k} \hat{x}_e \lambda_e \tag{11a}$$

$$\text{s.t. } \sum_{e \in E_k} \lambda_e \leq 1, \tag{11b}$$

$$-\lambda_e + \sum_{C \in \mathcal{C}, C \text{ contains } (i, j)} \mu_C \leq 0, \forall e = \{i, j\} \in E_k, \tag{11c}$$

$$-\lambda_e + \sum_{C \in \mathcal{C}, C \text{ contains } (j, i)} \mu_C \leq 0, \forall e = \{i, j\} \in E_k, \tag{11d}$$

$$\lambda_e \geq 0, \forall e \in E_k; \mu_a \geq 0, \forall a \in A_k, \tag{11e}$$

where \mathcal{C} is the set of cuts (10b) that has been added.

Formulation (10) needs to be solved by a cutting plane algorithm where the directed Steiner cut inequalities (10b) are iteratively generated. The separation of (10b) involves solving at most $|T| - 1$ maxflow problems. A Benders cut of the form $\sum_{e \in E_k} \lambda_e x_e \geq (\sum_{C \in \mathcal{C}} \mu_C) z_k$ will cut off the relaxation solution (\hat{x}, \hat{z}) if the optimal solution of (10) is strictly larger than 0.

The exact LP-based separation of a relaxation solution (\hat{x}, \hat{z}_k) from $\text{proj}_{(x, z_k)}(Y_k)$ is time-consuming. The separation routines proposed in Section 2.2 for the scenario-based undirected Steiner cut inequalities (6b) and the scenario-based Steiner partition inequalities (8) can be seen

as heuristic separations from $\text{proj}_{(x,z_k)}(Y_k)$. Since these heuristic separation routines can be implemented by strongly polynomial algorithms, in our computational experiments, we first apply these heuristic separations, and then solve the Benders feasibility LP only when these heuristic separation routines fail to return a valid inequality that is violated by the current relaxation solution. In particular, when the current relaxation solution is integral, the heuristic separation based on graph search is sufficient to check its feasibility and serve as an exact separation, and therefore there is no need to solve the Benders feasibility LP.

There are many alternative ways of generating Benders feasibility cuts based on different feasibility LPs. For example, one could introduce a flow variable f_a^k for each arc $a \in A_k$ in scenario $k \in K$, and establish a multicommodity flow extended formulation. According to our computational experiments, solving the feasibility LP (10) using a cutting plane method yields the best performance among all variants.

2.4 Additional valid inequalities

In addition to Benders feasibility cuts, we propose two additional sets of valid inequalities for the chance-constrained Steiner tree problem (2). The first set of valid inequalities is based on the concept of probabilistic Steiner p -cuts. The second set of valid inequalities is an adaptation of the mixing inequalities proposed by Küçükyavuz [19] and Luedtke [23].

2.4.1 Probabilistic Steiner p -cut inequalities

Given a set of edges C , and a number $p \geq 2$, we define $F(C, p) = \{k \in K \mid G_k(E \setminus C) \text{ has at least } p \text{ disconnected components } V_1, V_2, \dots, V_p \text{ such that } V_i \cap T \neq \emptyset, \forall i = 1, 2, \dots, p\}$ as a set of p -failure scenarios.

Definition 1. *A set of edges $C \subseteq E$ is a probabilistic Steiner p -cut if there exists a set of p -failure scenarios $F(C, p) \subseteq K$ with $|F(C, p)| \geq q + 1$.*

If C is a probabilistic Steiner p -cut, then C is a probabilistic Steiner p' -cut for all $2 \leq p' \leq p$. Therefore, for a probabilistic Steiner p -cut, we define $\theta(C) := \max\{p' : |F(C, p')| \geq q + 1\}$, and the following inequalities are valid for formulation (6):

$$\sum_{e \in C} x_e \geq \theta(C) - 1, \quad \forall \text{ probabilistic Steiner } p\text{-cut } C. \quad (12)$$

We refer to inequalities (12) as the *probabilistic Steiner p -cut inequalities*. It is clear that a formulation that includes all the probabilistic Steiner p -cut inequalities as well as the integrality constraints on x variables gives an exact formulation for the chance-constrained Steiner tree problem.

$$\min_{x \in \{0,1\}^{|E|}} \sum_{e \in E} c_e x_e \quad (13a)$$

$$\text{s.t. } \sum_{e \in C} x_e \geq \theta(C) - 1, \quad \forall \text{ probabilistic Steiner } p\text{-cut } C. \quad (13b)$$

The probabilistic Steiner p -cut inequalities (12) are NP-hard to separate, even for the special case when $p = 2$ and the terminal set consists of a single s - t pair (Song and Luedtke [28]). However, if a relaxation solution \hat{x} is integral, i.e., $\hat{x} \in \{0, 1\}^{|E|}$, then the exact separation can be done by a simple graph search on G_k for each scenario k using edges $e \in E_k$ that $\hat{x}_e = 1$. First, let

$F(\hat{x}) = \{k \in K \mid T \text{ is not connected in } G_k \text{ using edges } e \text{ that } \hat{x}_e = 1\}$. If $|F(\hat{x})| > q$, then a cut C_k with cut value $\sum_{e \in C_k} \hat{x}_e = 0$ could be found for each $k \in F(\hat{x})$, and $C = \bigcup_{k \in F(\hat{x})} C_k$ is a probabilistic Steiner p -cut with $\sum_{e \in C} \hat{x}_e = 0$. A *minimal* probabilistic Steiner p -cut for a particular $\theta(C)$ value can be obtained by sequentially removing edges $e \in C$ as long as $C \setminus \{e\}$ remains a probabilistic Steiner p -cut. This procedure is referred to as the “combine-and-reduce” procedure in Song and Luedtke [28]. For fractional relaxation solutions \hat{x} , this “combine-and-reduce” procedure could be applied as a heuristic separation routine. We first take the union of a set of scenario-based Steiner cuts C_k that may come from the separation of \hat{x} for each individual scenario $k \in F$, for a set F that $|F| > q$, and let $C = \bigcup_{k \in F} C_k$. We then sequentially remove the edges in C for a range of p values. However, the resulting probabilistic Steiner p -cut inequality (12) may not be violated by \hat{x} . We perform a sequential reduction and obtain a family of probabilistic Steiner p -cuts for all different $p \geq 2$.

Although the number of variables in formulation (13) is independent of the number of scenarios N , we observe in our computational experiments that this formulation does not yield a promising computational performance. Probabilistic Steiner p -cut inequalities (12) can also be added as valid inequalities to the scenario-based Steiner cut formulation (6).

2.4.2 Mixing inequalities

Benders feasibility cuts yield a strong continuous relaxation $\text{proj}_{(x, z_k)}(Y_k)$ for the feasible set X_k separately for each scenario $k \in K$. We next incorporate the mixing inequalities (Küçükyavuz [19] and Luedtke [23]) that combine multiple scenarios together, and exploit the chance constraint structure (6c). This structure is not considered in the cut generation for the scenario-based formulations discussed above. Given a general chance-constrained stochastic program with a finite number of scenarios:

$$\min f(x) \tag{14a}$$

$$\text{s.t. } z_k = 1 \Rightarrow x \in P_k, \forall k \in K, \tag{14b}$$

$$\sum_{k \in K} z_k \geq N - q, \tag{14c}$$

$$x \in X; z_k \in \{0, 1\}, \forall k \in K, \tag{14d}$$

where P_k is the feasible set defined by constraints in scenario k , and X is a deterministic feasible set. Given a coefficient vector α , a mixing inequality of the form $\alpha^\top x + \pi^\top z \geq \beta$ can be generated as follows. First, we solve a subproblem separately for each scenario k :

$$h_k(\alpha) = \min\{\alpha^\top x \mid x \in P_k \cap \bar{X}\}, \tag{15}$$

where \bar{X} is some relaxation of the original feasible set X , e.g., the LP relaxation. We then sort these $h_k(\alpha)$ values in non-increasing order: $h_{\sigma_1}(\alpha) \geq h_{\sigma_2}(\alpha) \geq \dots \geq h_{\sigma_N}(\alpha)$. Let $T = \{t_1, t_2, \dots, t_l\} \subseteq \{\sigma_1, \sigma_2, \dots, \sigma_q\}$, and let $t_{l+1} = \sigma_{q+1}$, the following mixing inequality is valid for (14):

$$\alpha^\top x - \sum_{i=1}^l [h_{t_i}(\alpha) - h_{t_{i+1}}(\alpha)] z_{t_i} \geq h_{\sigma_{q+1}}(\alpha). \tag{16}$$

The choice of α could be arbitrary. Luedtke [23] suggests using the coefficient vectors of the constraints that define the scenario-based feasible set P_k , or the strong valid inequalities for $\text{conv}(X \cap P_k)$. In our problem, we choose vector α from the coefficient vectors of the scenario-based Steiner partition inequalities (8). Therefore, calculating $h_k(\alpha)$ in (15) exactly with $P_k \cap X$

corresponds to solving the deterministic Steiner tree problem on graph G_k with α as the edge cost vector, which could be computationally intensive. Since a lower bound of $h_k(\alpha)$ gives a valid mixing inequality (16), we can solve (15) using a relaxation of $P_k \cap X$. For example, we can use the projection from the LP relaxation of the directed Steiner cut formulation (5) onto x space, which empirically has a very small integrality gap. Furthermore, since our choice of α is a 0-1 vector, a simple lower bound for $h_k(\alpha)$ can be calculated very efficiently by a graph search. In fact, let (V_1, V_2, \dots, V_t) be the partition of the node set V induced by the set of edges $\{e \in E_k \mid \alpha_e = 0\}$, and let $L = |\{i \in \{1, 2, \dots, t\} \mid V_i \cap T \neq \emptyset\}|$, then $L - 1$ is a valid lower bound for $h_k(\alpha)$, since we need at least $L - 1$ edges e such that $\alpha_e = 1$ to connect all terminal nodes. It is clear that there is a trade-off in the computation between using a stronger bound for $h_k(\alpha)$, which takes longer to compute, versus using a weaker bound, which can be obtained more efficiently. Our computational experiment indicates that the weaker lower bound obtained by the simple graph search works better for our set of instances. After $h_k(\alpha)$ values are calculated for all scenarios $k \in K$, we perform the exact separation of mixing inequalities using these $h_k(\alpha), k \in K$ values (exact or relaxation) following Küçükyavuz [19] and Luedtke [23].

3 Strength of valid inequalities

In Section 2.2, we generalized the deterministic Steiner partition inequalities (4) to the scenario-based Steiner partition inequalities (8). We denote the deterministic Steiner tree polytope as $P_{ST} := \{x \in \{0, 1\}^{|E|} \mid x \text{ is feasible to (3)}\}$, and denote its dominant as $P_{ST}^\uparrow := \{x \in \mathbb{Z}_+^{|E|} \mid \exists x' \in P_{ST} \text{ such that } x' \leq x\}$. Given a partition (V_1, V_2, \dots, V_p) , let $G(V_1, V_2, \dots, V_p)$ be the graph that is constructed by shrinking each subset V_i into one single node, and let $G(V_i)$ be the graph induced by node set V_i . Chopra and Rao [11] show that the Steiner partition inequality (4) is facet-defining for the convex hull of P_{ST}^\uparrow if and only if $G(V_1, V_2, \dots, V_p)$ is a 2-connected graph and $G(V_i)$ is connected for all $i = 1, 2, \dots, p$.

To study the strength of the scenario-based Steiner partition inequalities (8), we define the scenario-based Steiner tree polytope $Y := \{x \in \{0, 1\}^{|E|}, z \in \{0, 1\}^N \mid (x, z) \text{ is feasible to (6)}\}$, the dominant of the scenario-based Steiner tree polytope $Y^\uparrow := \{x \in \mathbb{Z}_+^{|E|}, z \in \mathbb{Z}_+^N \mid \exists (x', z') \in Y \text{ such that } x' \leq x \text{ and } z' \leq z\}$, and the partial dominant of the scenario-based Steiner tree polytope $\bar{Y}^\uparrow := \{x \in \mathbb{Z}_+^{|E|}, z \in \{0, 1\}^N \mid \exists x' \leq x \text{ such that } (x', z) \in Y\}$. Because of the nonnegative costs c_e for $e \in E$, we have $\min\{c^\top x \mid (x, z) \in Y\} = \min\{c^\top x \mid (x, z) \in Y^\uparrow\} = \min\{c^\top x \mid (x, z) \in \bar{Y}^\uparrow\}$. Therefore, it is interesting to study Y^\uparrow and \bar{Y}^\uparrow .

Similar to the deterministic case, given a partition (V_1, V_2, \dots, V_p) , let $G_k(V_1, V_2, \dots, V_p)$ be the graph constructed by shrinking each subset V_i into one single node in scenario k , and let $G_k(V_i)$ be the graph induced by node set V_i in scenario k . For each scenario k , we define $\mu_p(C, k) := \{C' \subset C, |C'| = p - 1 \mid G_k(E \setminus C \cup C') \text{ connects } T\}$. For two scenarios k_1 and k_2 , we define $\delta_p(C, k_1, k_2) := \mu_p(C, k_1) \cap \mu_p(C, k_2)$. In addition, for notational convenience, we use E_k^p to denote $E_k(V_1, V_2, \dots, V_p)$. Recall that $F(E_k^p) = \{k \in K \mid T \text{ is not connected in } G_k(E \setminus E_k^p)\}$.

Theorem 1. *Assume $q \geq 1$, let $k \in K$ and (V_1, V_2, \dots, V_p) be a partition of V with $p \geq 2$, the scenario-based Steiner partition inequality (8), $\sum_{e \in E_k^p} x_e \geq (p-1)z_k$, is facet-defining for $\text{conv}(\bar{Y}^\uparrow)$, if the following conditions are satisfied:*

1. $|F(E_k^p)| \leq q$.
2. *The shrunk graph $G_k(V_1, V_2, \dots, V_p)$ is a 2-connected graph, and $G_k(V_i)$ is connected for all $i = 1, 2, \dots, p$.*

3. $\delta_p(E_k^p, k, k') \neq \emptyset$ for every $k' \in F(E_k^p)$.

Proof. First, we show that $\text{conv}(\bar{Y}^\uparrow)$ is full-dimensional, i.e., there exist $|E| + N + 1$ affinely independent points in $\text{conv}(\bar{Y}^\uparrow)$. Let \vec{e}^i denote the unit vector where the i -th component is 1 and all other components are 0s, and define $\mathbf{1}_E := \sum_{i \in E} \vec{e}^i$ and $\mathbf{1}_K := \sum_{i \in K} \vec{e}^i$. Then point $(\mathbf{1}_E, \mathbf{1}_K) \in \text{conv}(\bar{Y}^\uparrow)$, and points $(\mathbf{1}_E, \mathbf{1}_{K \setminus \{k\}}) \in \text{conv}(\bar{Y}^\uparrow)$ for $k \in K$. Moreover, for each $e \in E$, point $(\mathbf{1}_E + \vec{e}^e, \mathbf{1}_K) \in \text{conv}(\bar{Y}^\uparrow)$ because $(\vec{e}^e, 0)$ is a feasible direction for $\text{conv}(\bar{Y}^\uparrow)$. It is not hard to see these $|E| + N + 1$ points are affinely independent. Therefore, the dimension of $\text{conv}(\bar{Y}^\uparrow)$ is $|E| + N$.

Next, we prove that conditions 1 to 3 are sufficient. According to condition 2, the scenario-based Steiner partition inequality $\sum_{e \in E_k^p} x_e \geq p - 1$ is facet-defining for the convex hull of the dominant of Steiner tree polytope in scenario k with $z_k = 1$. Therefore, there exist $|E_k|$ affinely independent points $\{\bar{x}^i\}_{i=1}^{|E_k|}$ in the $|E|$ -dimensional space of x that satisfy $\sum_{e \in E_k^p} \bar{x}_e^i = p - 1$, and $\bar{x}_e^i = 0$ for $e \in E \setminus E_k$ with $i = 1, 2, \dots, |E_k|$ (Chopra and Rao [11]). Let \bar{x}^{i_0} be one such point. For each scenario $k' \in F(E_k^p) \setminus \{k\}$, according to condition 3, let $C(k') \in \delta_p(E_k^p, k, k')$. We prove that the following $|E| + N$ points are feasible, satisfy (8) at equality, and are affinely independent:

- $\mathbf{x}^1 := (\mathbf{1}_{E \setminus E_k^p}, \mathbf{1}_{K \setminus F(E_k^p)})$, 1 point;
- $\mathbf{x}^2 := \{(\mathbf{1}_{E \setminus E_k^p} + \bar{x}^i, \vec{e}^k + \mathbf{1}_{K \setminus F(E_k^p)}) \mid i = 1, 2, \dots, |E_k|\}$, $|E_k|$ points;
- $\mathbf{x}^3 := \{(\mathbf{1}_{E \setminus E_k^p} + \bar{x}^{i_0} + \vec{e}^i, \vec{e}^k + \mathbf{1}_{K \setminus F(E_k^p)}) \mid i \in E \setminus E_k\}$, $|E| - |E_k|$ points;
- $\mathbf{x}^4 := \{(\mathbf{1}_{E \setminus E_k^p} + \bar{x}^{i_0}, \vec{e}^k + \mathbf{1}_{K \setminus F(E_k^p)} - \vec{e}^{k'}) \mid k' \in K \setminus F(E_k^p)\}$, $N - |F(E_k^p)|$ points;
- $\mathbf{x}^5 := \{(\mathbf{1}_{C(k')} + \mathbf{1}_{E \setminus E_k^p}, \vec{e}^k + \mathbf{1}_{K \setminus F(E_k^p)} + \vec{e}^{k'}) \mid k' \in F(E_k^p) \setminus \{k\}\}$, $|F(E_k^p)| - 1$ points.

Point \mathbf{x}^1 satisfies inequality (8) at equality since both sides are 0, points in $\{\mathbf{x}^i\}_{i=2}^4$ satisfy (8) at equality by the definitions of $\{\bar{x}^i\}_{i=1}^{|E_k|}$, and points in \mathbf{x}^5 satisfy (8) at equality by the definition of $C(k')$. For feasibility, \mathbf{x}^1 is feasible by condition 1 that $|F(E_k^p)| \leq q$; points in \mathbf{x}^2 are feasible by definition of $\{\bar{x}^i\}_{i=1}^{|E_k|}$; points in \mathbf{x}^3 and \mathbf{x}^4 are feasible by the choice of \bar{x}^{i_0} ; points in \mathbf{x}^5 are feasible by the choice of $C(k')$. To prove these points are affinely independent, we subtract point \mathbf{x}^1 from all other points to obtain $|E| + N - 1$ points that we show to be linearly independent. We place them in the following matrix, with the corresponding indices labeled above the columns:

$$\begin{pmatrix} z_k & x_e, e \in E_k & x_e, e \in E \setminus E_k & z_{k'}, k' \in K \setminus F(E_k^p) & z_{k'}, k' \in F(E_k^p) \setminus \{k\} \\ \mathbf{1} & \# & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & * & I_{|E \setminus E_k|} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & * & \mathbf{0} & -I_{|K \setminus F(E_k^p)|} & \mathbf{0} \\ \mathbf{1} & * & \mathbf{0} & \mathbf{0} & I_{|F(E_k^p)|-1} \end{pmatrix},$$

where I_r represents an $r \times r$ identity matrix, $\mathbf{0}$ and $\mathbf{1}$ represent appropriately sized matrices of zeros and ones, $*$ represents an appropriately sized unspecified matrix, and $\#$ represents an appropriately sized matrix whose rows are linearly independent. From the structure of the matrix, these points are linearly independent, which implies that the scenario-based Steiner partition inequalities (8) are facet-defining. \square

We next study the strength of the probabilistic Steiner p -cut inequalities (12), $\sum_{e \in E^p} x_e \geq \theta(E^p) - 1$, where E^p is a probabilistic Steiner p -cut. We let $p = \theta(E^p) = \max\{p' : |F(E^p, p')| \geq q + 1\}$. Recall $F(E^p, p) = \{k \in K \mid G_k(E \setminus E^p) \text{ has at least } p \text{ disconnected components } V_1, V_2, \dots, V_p \text{ such that } V_i \cap T \neq \emptyset, \forall i = 1, 2, \dots, p\}$.

Theorem 2. Assume $|F(E^p, p)| \geq q+1$, the probabilistic Steiner p -cut inequality, $\sum_{e \in E^p} x_e \geq p-1$, is facet-defining for $\text{conv}(Y^\uparrow)$ if:

1. $E^p = E_k^p$ for some scenario $k \in F(E^p)$, and the shrunk graph $G_k(V_1, V_2, \dots, V_p)$ is a 2-connected graph, and $G_k(V_i)$ is connected for all $i = 1, 2, \dots, p$.
2. According to condition 1, there exists $|E_k|$ affinely independent points $\{\bar{x}^i\}_{i=1}^{|E_k|}$ in the $|E|$ -dimensional space of x that satisfy $\sum_{e \in E_k^p} \bar{x}_e^i = p-1$. Let $C(\bar{x}^i) = \{e \in E \mid \bar{x}_e^i = 1\}$, and we assume $|F(E^p \setminus C(\bar{x}^i))| \leq q$, $\forall i = 1, 2, \dots, |E_k|$.

Proof. Notice that points $(\mathbf{1}_E, \mathbf{1}_K)$, $\{(\mathbf{1}_E + \vec{\mathbf{e}}^e, \mathbf{1}_K)\}_{e \in E}$, and $\{(\mathbf{1}_E, \mathbf{1}_K + \vec{\mathbf{e}}^e)\}_{e \in K}$ are affinely independent and feasible to $\text{conv}(Y^\uparrow)$. Hence the dimension of $\text{conv}(Y^\uparrow)$ is $|E| + N$.

To prove the sufficiency of conditions 1 and 2, we propose $|E| + N$ affinely independent points. First, the $|E_k|$ affinely independent points $\{\bar{x}^i\}_{i=1}^{|E_k|}$ in the $|E|$ -dimensional x space satisfy $\sum_{e \in E^p} \bar{x}_e^i = p-1$, and $\bar{x}_e^i = 0$ for $e \in E \setminus E_k$ with $i = 1, 2, \dots, |E_k|$ (Chopra and Rao[11]). Let \bar{x}^{i_0} be such a point. Now consider the following points:

- $\mathbf{x}^0 := (\mathbf{1}_{E \setminus E_k} + \bar{x}^{i_0}, \mathbf{1}_{F(E^p) \setminus F(E^p \setminus C(\bar{x}^{i_0}))} + \mathbf{1}_{K \setminus F(E^p)})$, 1 point;
- $\mathbf{x}^1 := \{(\mathbf{1}_{E \setminus E_k} + \bar{x}^i, \mathbf{1}_{F(E^p) \setminus F(E^p \setminus C(\bar{x}^i))} + \mathbf{1}_{K \setminus F(E^p)}) \mid i \in \{1, 2, \dots, |E_k|\} \setminus \{i_0\}\}$, $|E_k| - 1$ points;
- $\mathbf{x}^2 := \{(\mathbf{1}_{E \setminus E_k} + \vec{\mathbf{e}}^e + \bar{x}^{i_0}, \mathbf{1}_{F(E^p) \setminus F(E^p \setminus C(\bar{x}^{i_0}))} + \mathbf{1}_{K \setminus F(E^p)}) \mid e \in E \setminus E_k\}$, $|E| - |E_k|$ points;
- $\mathbf{x}^3 := \{(\mathbf{1}_{E \setminus E_k} + \bar{x}^{i_0}, \mathbf{1}_{F(E^p) \setminus F(E^p \setminus C(\bar{x}^{i_0}))} + \vec{\mathbf{e}}^{k'} + \mathbf{1}_{K \setminus F(E^p)}) \mid k' \in F(E^p)\}$, $|F(E^p)|$ points.
- $\mathbf{x}^4 := \{(\mathbf{1}_{E \setminus E_k} + \bar{x}^{i_0}, \mathbf{1}_{F(E^p) \setminus F(E^p \setminus C(\bar{x}^{i_0}))} + \mathbf{1}_{K \setminus F(E^p)} + \vec{\mathbf{e}}^{k'}) \mid k' \in N \setminus F(E^p)\}$, $N - |F(E^p)|$ points.

Points in $\{\mathbf{x}^i\}_{i=0}^4$ satisfy inequality $\sum_{e \in E^p} x_e \geq p-1$ at equality. Points in \mathbf{x}^0 to \mathbf{x}^4 are feasible by the definition of $\{\bar{x}^i\}_{i=1}^{|E_k|}$ and Y^\uparrow . We subtract point \mathbf{x}^0 from points in $\{\mathbf{x}^i\}_{i=1}^4$ to obtain $|E| + N - 1$ points. We show that they are linearly independent by placing them in the following matrix:

$$\begin{pmatrix} x_e, e \in E_k & x_e, e \in E \setminus E_k & z_{k'}, k' \in F(E^p) & z_{k'}, k' \in K \setminus F(E^p) \\ \# & \mathbf{0} & * & \mathbf{0} \\ \mathbf{0} & I_{|E|-|E_k|} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{|F(E^p)|} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{N-|F(E^p)|} \end{pmatrix},$$

where I_r represents an $r \times r$ identity matrix, $\mathbf{0}$ and $\mathbf{1}$ represent appropriately sized matrices of zeros and ones, $*$ represents an appropriately sized unspecified matrix, and $\#$ represents an appropriately sized matrix whose rows are affinely independent. From the structure of the matrix, these points are linearly independent, which implies that points in $\{\mathbf{x}^i\}_{i=0}^4$ are affinely independent. Therefore, the probabilistic p -cut inequalities, $\sum_{e \in E^p} x_e \geq p-1$, are facet-defining for $\text{conv}(Y^\uparrow)$. \square

4 Computational experiments

We conduct our computational experiments using the branch-and-cut algorithms described in Section 2 on the chance-constrained Steiner tree problem (2).

4.1 Computational settings

We implement the proposed branch-and-cut algorithms using commercial software CPLEX 12.5 with C++ language. We set the number of threads to be one. We also set the CPLEX MIPEmphasis parameter to “optimality”, since in our branch-and-cut framework with delayed constraint generation, CPLEX is not provided the actual feasible set in the first place. We add violated valid inequalities to CPLEX solver using cutcallback routines: for integral relaxation solutions, LazyConstraintCallback is used, and for fractional relaxation solutions, UserCutCallback is used. Because of these callback routines, CPLEX presolve is turned off. We iteratively generate cuts at the root node as long as there exists any, and for other nodes in the branch-and-bound tree, we limit the cut generation to be at most one round. All other CPLEX parameters are the same as the default setting. We use graph library LEMON [20] for solving the maxflow subproblems and constructing Gomory-Hu trees. All tests are performed in a Linux workstation with 3.00 GHz processors and 7.8 Gb memory.

We test our proposed algorithms on random graphs adapted from deterministic graph instances in SteinLib Testdata Library [18]. We create the random edge failure scenarios in a similar way as described in Song and Luedtke [28]. First, we generate edge failure probabilities according to an exponential distribution with mean $\lambda = 0.05$. Then based on these generated failure probabilities, we generate N failure scenarios by a Monte Carlo sample, with sample sizes $N = 100, 500$ and 1000 . For each instance with the same sample size, we test for five replications and report the average result of them. We use $\epsilon = 0.1$ as the risk tolerance parameter for all our test instances.

4.2 Implementation details

Preprocessing Duin and Volgenant [15], Koch and Martin [17] propose a variety of preprocessing tests for the deterministic Steiner tree problem. After reviewing these tests, we found a straightforward way to generalize the degree test [17] for our problem setting. Consider the following steps.

1. If a nonterminal node is of degree one, then we remove it from the graph.
2. If a nonterminal node v is of degree two, then the two incident edges $\{v, u\}$ and $\{v, w\}$ with $u \neq w$ can be replaced by an edge connecting u and w of cost $c_{\{u,w\}} = c_{\{v,u\}} + c_{\{v,w\}}$.
3. If edge e is incident to a terminal node v of degree one, then $x_e = 1$ in an optimal solution.
4. For a terminal node v , if edge $\{v, u\} = \operatorname{argmin}\{c_{\{v,w\}} | w \in V, \{v, w\} \in E_k \text{ for all } k \in K\}$ and $u \in T$, then $x_{\{v,u\}} = 1$ in an optimal solution.

We show the effect of preprocessing in Table 1. More sophisticated preprocessing techniques, e.g., the special distance test, bottleneck degree test, etc. [17], are worth investigating. We leave them as a future research direction.

In addition to the preprocessing procedure based on the network structure, we also perform a preprocessing step based on the connectivity of the network in each scenario. Specifically, we perform a simple graph search to check for each scenario k to see if all terminal nodes are connected when all edges are included in the network design, and set the scenario variable $z_k = 0$ otherwise.

Cut generation After the preprocessing, we use a branch-and-cut algorithm to solve the formulations that we proposed in Section 2. We use a cut violation threshold 10^{-3} for separating

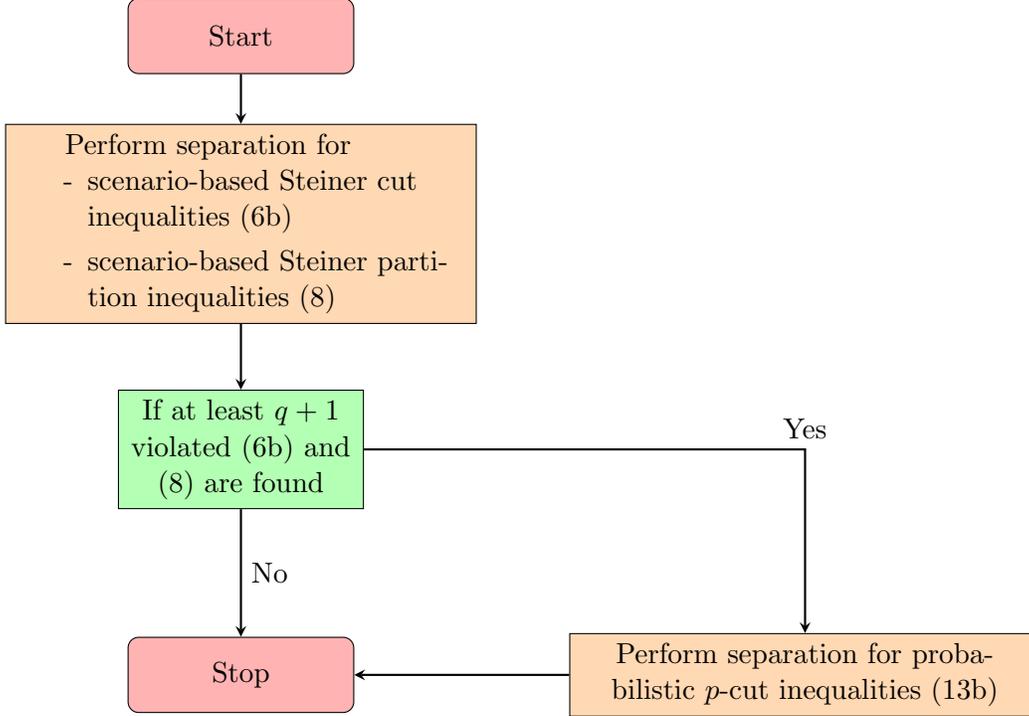


Figure 3: A flow chart for cut generation in option “basic”.

the scenario-based undirected Steiner cut inequalities (6b), the scenario-based Steiner partition inequalities (8), and Bender feasibility cuts. We use a cut violation threshold 5×10^{-4} for separating the probabilistic Steiner p -cut inequalities (12).

In all variants of the proposed branch-and-cut algorithm for solving the scenario-based undirected Steiner cut formulation (6), we use the following computational settings. Given a relaxation solution \hat{x} , we first conduct a separation for the scenario-based Steiner partition inequalities (8) using a simple graph search based on \hat{x} . If \hat{x} is not integral (otherwise the separation based on the graph search is exact), and in the current round of cut generation, we have found less than 50 scenario-based Steiner partition inequalities that are violated by the current solution \hat{x} by at least 1, we perform exact separation for the scenario-based undirected Steiner cuts (6b). The separation routine involves solving a maximum r - l flow problem for each $l \in T \setminus \{r\}$ on a directed graph, which is constructed by introducing a pair of arcs (i, j) and (j, i) for each edge $e = \{i, j\} \in E$, each having a weight \hat{x}_e . Following [17], we use the idea of “creep flows” when we construct the maximum flow problems. Specifically, we add a tiny capacity value 10^{-6} to all available arcs. In this way we can guarantee the minimality of the scenario-based undirected Steiner cuts. We perform this separation routine up to depth 8 in the branch-and-bound tree. If no violated cut is found after this separation, we conduct a heuristic separation for scenario-based Steiner partition inequalities (8), using the approximation algorithm by Chekuri et al. [9] based on Gomory-Hu trees. We only perform this Gomory-Hu tree-based routine at the root node. We refer to these computational settings as option “basic” in our experiments. When at least $q + 1$ violated scenario-based Steiner cut inequalities (6b), or scenario-based Steiner partition inequalities (8) are generated, we perform the “combine-and-reduce” procedure to generate the probabilistic Steiner p -cut inequalities (12). We summarize the cut generation procedure as a flow chart in Figure 3.

Next, we consider the branch-and-cut algorithm discussed in Section 2 for solving the scenario-based directed Steiner cut formulation. We generate Benders feasibility cuts using the LP-based separation based on (10). Suppose that, at the root node, \hat{x} is a fractional relaxation solution, and we have not found any violated cut by previously stated separation routines in option “basic”. We generate Benders feasibility cuts for scenarios k such that $\hat{z}_k \geq 0.8$. This parameter 0.8 is chosen according to our computational benchmark to balance between the time spent on cut separation and the strength of the relaxation. The LP-based separation is done by solving (10) via a cutting plane approach. We do not restart the cutting plane approach from scratch every time when we have a new \hat{x} solution. Instead, we keep up to 500 previously generated cutting planes (10b) in the LP formulation for each scenario k . We also implement an “early stop” criterion that we terminate the cutting plane generation if the cut violation value is less than 0.1, and the objective value of (10) does not increase for three consecutive iterations. This criterion helped to mitigate the “tailing off” effect of the cutting plane algorithm.

Heuristics We implement a heuristic algorithm that is very similar to the one presented in [28] to provide feasible solutions throughout the branch-and-bound tree. Given a relaxation solution \hat{x} , we initialize a candidate set of edges $Q = \{e \in E \mid \hat{x}_e = 1\}$. We then check for each scenario $k \in K$ to see if $G_k(Q)$ successfully connects all the terminal nodes. If the set of successful scenarios is large enough, we stop with this feasible solution and perform the sequential edge reduction to improve it. Otherwise, we go through scenarios $k \in K$ that fail to connect all the terminal nodes, and solve a Steiner tree problem using costs $\tilde{c}_e = 0$ if $e \in Q$ and $\tilde{c}_e = c_e$ otherwise. Optimally solving the Steiner tree problem is time-consuming, so we settle with a simple heuristic based on multiple shortest path problems. We then add the edges contained in the Steiner tree to the candidate set Q , thus ensuring that in scenario k , all terminals are connected. With the new set Q , we again check for the rest of the failed scenarios to see if any becomes successful, and then repeat this procedure until we find a feasible solution. Finally, we sequentially remove edges from Q to make it minimal. This heuristic is performed after the root node, once every 10 nodes for the first 500 nodes, and once every 50 nodes after that. We do not turn off the CPLEX default heuristics, so CPLEX may still look for good feasible solutions according to its own strategy.

4.3 Computational results

We use the following abbreviations throughout this section:

- AvT: Average computational time (in seconds).
- AvN: Average number of branch-and-bound nodes processed.
- AvR: Average root optimality gap. For each instance, the root optimality gap is calculated as $(S^* - S_R)/S^*$, where S^* is the optimal objective value, and S_R is the lower bound provided by CPLEX after the root node is processed.

Unless otherwise stated, we use a time limit of 3600 seconds for our experiments. If an instance is not solved to optimality within the time limit, we show in parentheses the number of instances out of five replications that are solved within the time limit, and we use the number of nodes that have been processed up to the time limit when calculating the average number of nodes, and use “>” to indicate the calculated average is a lower bound. The sizes of the graph instances that we test on are shown in Table 1.

First, we present the computational results for solving the scenario-based Steiner cut formulation (6) with option “basic” and the results for solving the probabilistic Steiner p -cut formulation (13).

Instance	Original size			After preprocess	
	$ V $	$ E $	$ T $	$ V $	$ E $
steinb6	50	100	13	44	94
steinb10	75	150	7	58	133
steinb11	75	150	10	62	137
steinb16	100	200	17	81	181

Table 1: Sizes of the test instances

We use a “combine-and-reduce” procedure to generate the probabilistic Steiner p -cut inequalities (12). In the “combine” phase of the procedure, we use the scenario-based Steiner cut inequalities and Steiner partition inequalities in a hierarchical manner following the implementation of option “basic”.

Instances		basic			Formulation (13)		
Instance	N	AvT	AvN	AvR	AvT	AvN	AvR
Steinb6	100	22.1	1019	10.7%	852.5	71051	32.9%
	500	126.0	955	12.0%	(2)	>82005	37.6%
	1000	235.9	769	12.0%	(2)	>54925	33.0%
Steinb10	100	10.2	175	14.0%	53.9	6669	27.5%
	500	59.4	209	13.8%	290.0	13197	32.4%
	1000	162.0	427	17.2%	749.1	15697	36.3%
Steinb11	100	16.4	259	11.3%	110.6	11241	27.8%
	500	100.0	425	14.0%	(4)	>41757	36.4%
	1000	363.3	1122	15.8%	(0)	>64218	32.8%
Steinb16	100	110.1	2828	9.8%	(0)	>94279	36.0%
	500	409.7	1129	8.5%	(0)	>37867	38.9%
	1000	1523.3	5045	10.2%	(0)	>14580	38.2%

Table 2: Average time, number of nodes, and root optimality gap for the scenario-based Steiner cut formulation solved by option “basic”, and the probabilistic Steiner p -cut formulation.

Table 2 shows that the scenario-based formulation spends much less computational time than the probabilistic Steiner p -cut formulation to solve the tested instances to optimality. The comparisons of “AvN” and “AvR” between these two options indicate that the probabilistic Steiner p -cut formulation is rather weak, which leads to a much larger root optimality gap and much more nodes to process than the scenario-based undirected Steiner cut formulation plus additional valid inequalities. Therefore, although the probabilistic Steiner p -cut formulation has a much more compact solution space, it is better to solve the scenario-based Steiner cut formulation, and add the probabilistic Steiner p -cut inequalities as valid inequalities in this formulation.

We next show the computational results for solving the scenario-based directed Steiner cut formulation with option “basic + Benders” and option “basic + Benders + mixing”. In option “basic + Benders + mixing”, we generate mixing inequalities at every node in the branch-and-bound tree. We calculate a lower bound of $h_k(\alpha)$ in (16) based on the simple graph search described in Section 2.4.2. We choose α as the coefficient vector of the most violated scenario-based undirected Steiner cut inequality (6b) or scenario-based partition inequality (8) that has been found for the current relaxation solution \hat{x} . In Table 3, abbreviation “AvLT” stands for the average time spent on

solving the Benders feasibility LP. We use a time limit of 7200 seconds for this set of experiments.

Instances		basic + Benders				basic + Benders + mixing			
Instance	N	AvT	AvLT	AvN	AvR	AvT	AvLT	AvN	AvR
Steinb6	100	25.3	4.4	594	8.5%	27.3	4.3	664	7.8%
	500	115.1	14.1	941	10.0%	130.4	13.3	781	9.6%
	1000	344.1	34.0	842	9.4%	282.0	32.8	782	9.1%
Steinb10	100	21.1	9.2	130	9.2%	21.2	8.4	125	8.8%
	500	74.2	15.7	197	11.4%	75.2	15.0	185	10.9%
	1000	227.8	32.5	408	14.8%	193.3	30.0	339	14.5%
Steinb11	100	36.5	17.3	186	8.1%	36.5	13.9	203	7.7%
	500	137.0	36.3	407	11.8%	140.7	31.3	385	11.3%
	1000	494.6	115.2	854	12.3%	525.3	105.7	872	12.1%
Steinb16	100	127.9	21.3	2079	7.6%	134.6	22.2	2029	7.4%
	500	439.6	45.8	1106	7.5%	405.5	43.7	1097	7.4%
	1000	1550.7	112.1	4679	8.7%	1456.8	115.7	3780	8.5%

Table 3: Average solution time, average time spent on generating Benders feasibility cuts, number of nodes, and root optimality gap for the scenario-based directed Steiner cut formulation with Benders feasibility cuts using option “basic + Benders” and option “basic + Benders + mixing”.

Comparing option “basic + Benders” in Table 3 with option “basic” in Table 2, we see that although the root optimality gap and the number of branch-and-bound nodes are significantly reduced, the effort of generating Benders feasibility cuts does not pay off for a reduction in computational time for option “basic + Benders” in all of our test instances. This motivates further investigation on how the Benders feasibility problem could be solved more efficiently. Comparing option “basic + Benders” and “basic + Benders + mixing” in Table 3, we see that, on top of option “basic + Benders”, the mixing inequalities further reduce the root optimality gap and lead to a significant reduction in the number of the branch-and-bound nodes processed. However, this reduction does not lead to less computational time in all of our test instances. On the other hand, the root optimality gap is still large even after we include both Benders feasibility cuts and the mixing inequalities. This motivates further exploration on additional valid inequalities for the chance-constrained Steiner tree problems.

Finally, we apply our proposed approach to solve the instances from [28], and compare the computational performance for cases when the entire graph must be connected, when only a subset of nodes (one quarter, one third, and one half of the nodes) must be connected, and when only an s - t pair needs to be connected. We use option “basic” for solving the chance-constrained Steiner tree problems when a subset of nodes are connected, and we use the computational setting that gives the best performance for the chance-constrained s - t connected and fully connected network design problems in [28].

From Table 4, we can see that when a single s - t pair is required to be connected, and when all nodes are required to be connected, the algorithms proposed in [28] could solve all instances within one minute. However, when we require only a subset of nodes to be connected, the computational time increases significantly. This is consistent with the fact that the deterministic version of the chance-constrained multi-terminal network design problem is an NP-hard problem, whereas the deterministic version of chance-constrained s - t connected or fully connected network design problem is polynomially solvable.

Instances		$s-t$		Full		1/4		1/3		1/2	
Instance	N	AvT	AvN	AvT	AvN	AvT	AvN	AvT	AvN	AvT	AvN
ins14	100	0.0	6	0.1	26	0.0	2	0.1	1	0.2	3
(15,22)	500	0.0	8	0.6	41	0.9	0	1.3	1	3.2	11
	1000	0.0	5	1.0	42	2.1	0	5.5	2	13.9	5
ins18	100	0.0	12	2.8	751	4.2	145	4.7	82	10.6	177
(37,57)	500	0.2	26	24.7	2316	56.8	114	74.1	210	125.0	504
	1000	0.2	15	40.6	1639	198.0	146	197.7	137	375.0	463

Table 4: Average time and number of nodes for chance-constrained $s-t$ connected network design problems, fully connected network design problems, and multi-terminal connected network design problems requiring the first $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ of the nodes to be connected.

5 Conclusion

In this paper, we studied a network design problem under uncertainty, in which a minimum-cost set of edges is selected to connect multiple terminals with probability no less than a given threshold. We modeled this problem as a chance-constrained Steiner tree problem. A branch-and-cut algorithm was used to solve the scenario-based undirected Steiner cut formulation, where several types of valid inequalities were applied, including the probabilistic Steiner p -cut inequalities, the scenario-based Steiner partition inequalities, and the mixing inequalities. In addition, we generated Benders feasibility cuts from the projection of the strong scenario-based directed Steiner cut formulation. Our computational results showed that the scenario-based formulation yields much better performance than the probabilistic Steiner p -cut formulation. The separation for scenario-based partition inequalities is effective. Benders feasibility cuts are useful for strengthening the relaxation bound, at a price of extra computational time performing LP-based separation. We also observed that the mixing inequalities are useful for strengthening the relaxation bound. Further exploration of additional valid inequalities that combine the chance-constrained structure and the Steiner tree problem structure is worthwhile, as we observed that the root optimality gap remains moderate even after Benders feasibility cuts and the mixing inequalities are included.

Acknowledgments

The authors thank James Luedtke for the beneficial discussion over this topic. The authors also thank the Editor-in-Chief, the Associate Editor, and two anonymous referees, whose insightful comments helped us to substantially improve the paper.

References

- [1] Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows*. Prentice Hall, New Jersey.
- [2] Andreas, A., Smith, J., and Küçükyavuz, S. (2008). Branch-and-price-and-cut algorithms for solving the reliable h -paths problem. *Journal of Global Optimization*, 42(4):443–466.
- [3] Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 2:238–252.

- [4] Beraldi, P., Bruni, M. E., and Guerriero, F. (2010). Network reliability design via joint probabilistic constraints. *IMA Journal of Management Mathematics*, 21(2):213–226.
- [5] Bomze, I., Chimani, M., Jnger, M., Ljubi, I., Mutzel, P., and Zey, B. (2010). Solving two-stage stochastic Steiner tree problems by two-stage branch-and-cut. In Cheong, O., Chwa, K.-Y., and Park, K., editors, *Algorithms and Computation*, volume 6506 of *Lecture Notes in Computer Science*, pages 427–439. Springer Berlin Heidelberg.
- [6] Charnes, A., Cooper, W., and Symonds, G. Y. (1958). Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263.
- [7] Charnes, A. and Cooper, W. W. (1959). Chance-constrained programming. *Management Science*, 6(1):73–79.
- [8] Charnes, A. and Cooper, W. W. (1963). Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39.
- [9] Chekuri, C., Guha, S., and Naor, J. (2006). The Steiner k -cut problem. *SIAM Journal of Discrete Mathematics*, 20(1):261–271.
- [10] Chopra, S., Gorres, E. R., and Rao, M. R. (1992). Solving the Steiner tree problem on a graph using branch and cut. *ORSA Journal on Computing*, 4(3):320–335.
- [11] Chopra, S. and Rao, M. R. (1994a). The Steiner tree problem I: formulations, compositions and extension of facets. *Mathematical Programming*, 64:204–229.
- [12] Chopra, S. and Rao, M. R. (1994b). The Steiner tree problem II: properties and classes of facets. *Mathematical Programming*, 64:231–246.
- [13] Claßen, G., Coudert, D., Koster, A., and Nepomuceno, N. (2011). A chance-constrained model and cutting planes for fixed broadband wireless networks. *Network Optimization*, 6701:37–42.
- [14] Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research*, 32:1429–1450.
- [15] Duin, C. W. and Volgenant, A. (1989). Reduction tests for the steiner problem in graph. *Networks*, 19(5):549–567.
- [16] Grötschel, M., Monma, C. L., and Stoer, M. (1995). Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Operations Research*, 43(6):1012–1024.
- [17] Koch, T. and Martin, A. (1998). Solving steiner tree problems in graphs to optimality. *Networks*, 32:207–232.
- [18] Koch, T., Martin, A., and Voß, S. (2000). SteinLib: An updated library on Steiner tree problems in graphs. Technical Report ZIB-Report 00-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, Berlin.
- [19] Küçükyavuz, S. (2012). On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1):31–56.
- [20] LEMON (2009). LEMON – Library for Efficient Modeling and Optimization in Networks. <http://lemon.cs.elte.hu/>.

- [21] Liu, X., Kkyavuz, S., and Luedtke, J. (2014). Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*.
- [22] Lucena, A. and Beasley, J. E. (1998). A branch and cut algorithm for the Steiner problem in graphs. *Networks*, 31:39–59.
- [23] Luedtke, J. (2013). A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146(1-2):219–244.
- [24] Luedtke, J. and Ahmed, S. (2008). A sample approximation approach for optimization with probabilistic constraints. 19:674–699.
- [25] Luedtke, J., Ahmed, S., and Nemhauser, G. (2010). An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272.
- [26] Magnanti, T. L. and Raghavan, S. (2005). Strong formulations for network design problems with connectivity requirements. *Networks*, 45(2):61–79.
- [27] Polzin, T. and Daneshmand, S. V. (2001). A comparison of Steiner tree relaxation. *Discrete Applied Mathematics*, 112:241–261.
- [28] Song, Y. and Luedtke, J. (2013). Branch-and-cut approaches for chance-constrained formulations of reliable network design problems. *Mathematical Programming Computation*, 5(4):397–432.
- [29] Van Vyve, M. (2005). The continuous mixing polyhedron. *Mathematics of Operations Research*, 30(2):441–452.
- [30] Zhang, M., Küçükyavuz, S., and Goel, S. (2014). A branch-and-cut method for dynamic decision under joint chance constraints. *Management Science*, 60(5):1317–1333.