Stochastic Survivable Network Design Problems: Theory and Practice

Ivana Ljubić[a], Petra Mutzel[b], Bernd Zey[b]

[a]*ESSEC Business School of Paris*
[b]*Department of Computer Science, TU Dortmund*

Abstract

We study survivable network design problems with edge-connectivity requirements under a two-stage stochastic model with recourse and finitely many scenarios. For the formulation in the natural space of edge variables we show that facet defining inequalities of the underlying polytope can be derived from the deterministic counterparts. Moreover, by using graph orientation properties we introduce stronger cut-based formulations. For solving the proposed mixed integer programming models, we suggest a two-stage branch&cut algorithm based on a decomposed model. In order to accelerate the computations, we suggest a new technique for strengthening the decomposed L-shaped optimality cuts which is computationally fast and easy to implement. A computational study shows the benefit of the decomposition and the cut strengthening—which significantly reduces the number of master iterations and the computational running time. Moreover, we evaluate the stability of the scenario generation method and analyze the value of the stochastic solution.

Keywords: stochastic network design problems, stochastic integer programming, branch&cut, Benders decomposition, integer L-shaped method

1. Introduction

Motivation. Survivable network design problems with edge-connectivity requirements (SNDPs) are among the most fundamental problems in the field of network optimization. Many classical network design problems including the shortest path problem, the minimum spanning tree problem, the Steiner tree problem, the minimum-weight edge-connected subgraph problem, and edge-connectivity augmentation problems are special cases of the survivable network design problem. Applications of the (special cases of the) SNDP can be found in many different fields, e.g., in the design of supply chain and distribution networks or in the chip layout design [5, 20, 22, 38]. The field of telecommunications belongs

Email addresses: ivana.ljubic@essec.edu (Ivana Ljubić), petra.mutzel@tu-dortmund.de (Petra Mutzel), bernd.zey@tu-dortmund.de (Bernd Zey)

to the most important applications that request building cost-effective networks with higher connectivity requirements.

In a typical telecommunication network [11, 12], an edge weighted graph is given with three *types of nodes*: "special" offices (which are nodes of type 2 that correspond to important hubs, business customers, or private households with high-demanding service packages), "ordinary" offices (nodes of type 1 corresponding to regular customers, like single households), and "optional" offices (nodes of type 0 representing street junctions). The goal is to find a cost-minimal subnetwork which ensures that all special offices are connected by two paths, all ordinary offices are at least simply connected, and optional offices can be used to establish connections, if that would lead to a cheaper solution. This problem is known as the $\{0,1,2\}$-SNDP. Here, we consider the general SNDP with arbitrary connectivity requirements between each pair of nodes.

In practice, however, from the moment that the information concerning the type of node is gathered until the moment in which the solution has to be implemented, some of the data might change with respect to the initial setting. In the present article we define a mathematical model that helps decision makers to deal with the following two types of uncertainty:

- Uncertainty with respect to node types: Node types may change over time, subject to many external conditions. For example, socio-economic factors like customer's purchase power, recession, or inflation may influence expected customer's demand. Furthermore, changes in urban city planning or political factors can lead to changes in the demand of a whole neighborhood, or availability of a given location to host an office. Finally, multiple service providers compete for the customers such that customer demand highly depends on the competing service offers available at the market.

  For all these reasons, during the strategic planning of a telecommunication network, it remains unclear which potential customers may be willing to subscribe to the service and to which particular service package.

- Uncertainty with respect to investment costs: The costs of establishing links (installing new cables, pipes, etc.) may be subject to inflations and price deviations. Price deviations are common due to the frequent changes in the underlying technology and (un)availability of the corresponding equipment.

Hence, a solution obtained using a classical deterministic model might become suboptimal or even infeasible once the network is deployed in which case a new solution might have to be redefined from scratch.

Despite the great importance of the SNDP and the relevance of the uncertainty for practical applications, to our knowledge, no publications are available that investigate the SNDP under these two particular sources of data uncertainty. In this article we attempt to close this gap by considering *two-stage stochastic* versions of survivable network design problems with edge-connectivity requirements (for an introduction to stochastic programming see, e.g., [3]).

Thereby, the uncertain data is modeled using random variables with a set of scenarios defining their possible outcomes. Typically, a solution is comprised by first- and second-stage decisions such that a partial subnetwork is built in the first stage which is then completed once the uncertain data becomes available in the second stage.

More precisely, in *the two-stage stochastic survivable network design problem* (SSNDP), network planners want to establish profitable connections now (in the *first stage*) while taking all possible outcomes—the *scenarios*—into account. In the future (in the *second stage*) the actual scenario with its requirements and connection costs is revealed and additional connections can be purchased (through so-called *recourse* actions) to satisfy the now known requirements. The objective is to minimize the *expected costs* of the solution, i.e., the sum of the first-stage costs plus the expected costs of the second stage. Thereby, all connectivity requirements for all scenarios have to be satisfied. The formal definition of the SSNDP is given in Section 3.

Previous work. There exists a large body of work on different variants of the deterministic survivable network design problem. We refer to [20, 22] for a comprehensive literature overview on the SNDP. Many polyhedral studies were done in the 90's, see, e.g., [14, 22]. A decade later the question of deriving stronger mixed integer programming (MIP) formulations by orienting the k-connected subgraphs has been considered by e.g. [1, 30]. Among the approximation algorithms for the SNDP, we point to the work of Jain [18] whose approximation factor of two remains the best one up to date.

Regarding the stochastic variants there are significantly less results published so far. To the best of our knowledge, the only results about the SSNDP with general edge/node-connectivity requirements are contained in our short paper [27]; cf. the next paragraph. One of the investigated special cases of the SSNDP is the *two-stage stochastic Steiner tree problem* in which node types are either zero or one. For this problem approximation algorithms (see, e.g., [15, 42]), MIP approaches (see [4]), and heuristics (see [17]) were developed. For the SSNDP involving node types ≥ 2, up to our knowledge, there only exists an $O(1)$ approximation algorithm (see [16]) for the following special case of the $\{0, k\}$-SSNDP: For each pair of distinct nodes i and j a single scenario, whose probability is p_{ij}, is given in which nodes i and j need to be k-edge-connected. But in general, however, it follows by [37] that the SSNDP is as hard to approximate as label cover—which is $\Omega(\log^{2-\epsilon} n)$ hard. In fact, the hardness-proof already works for the stochastic shortest path problem.

Besides the design of survivable networks a lot of research has been done concerning the design of reliable networks (e.g., recent articles can be found in the special issue [36]). Design of reliable networks under network uncertainty using the approach of *chance-constrained programming* (see, e.g., [35]), has been considered in [43, 44]. In chance-constrained programming, there is usually one decision horizon (i.e., no recourse) and a feasible solution has to satisfy the constraints with a given probability. In [43, 44] s-t-paths and the Steiner tree problem, respectively, have been considered under possible network failure scenarios. In contrast to the SSNDP studied in this article, these problems assume the set of customers remains the same across all scenarios but a whole subnetwork can be subject to failure. Each failure scenario happens with a certain probability and the goal is to find a reliable network that ensures given connectivity requirements with a certain probability. The authors introduce several (M)IP formulations, facet-defining inequalities, and provide computational studies.

Our contribution. Our contribution is twofold, it concerns theoretical models as well as practical algorithms.

Theory: In the past, the seminal result of Nash-Williams [32] has been used to develop stronger MIP models for the deterministic SNDP by exploiting graph orientations [7, 8, 30]. Here, we discuss that graph orientation properties cannot be used in a straight-forward fashion to develop similar models for the SSNDP. As an alternative, we propose two general ways to develop *semi-directed* MIP models in which only the second-stage solutions are oriented. We develop two novel cut-based MIP models of the deterministic equivalent for solving the SSNDP on undirected graphs based on these orientation properties. We prove that the new models are stronger than the original one based on standard undirected cuts. Moreover, when considering the undirected formulation of the SSNDP we show that facet defining inequalities can be easily derived from their deterministic counterparts.

Computational study: The SSNDP belongs to a broader class of two-stage integer stochastic programs with binary first-stage solutions and binary recourse. These NP-hard problems are known to be notoriously difficult to solve [39]. In this paper, we use a recently introduced decomposition approach called *two-stage branch&cut* [4]. This approach uses a Benders decomposition and two nested branch&cut algorithms and is similar to the integer L-shaped method [24]. In the *subproblems*, violated directed cuts are separated, while the *master problem* is expanded by L-shaped and integer optimality cuts. To enhance the algorithmic performance, we propose a new computationally inexpensive procedure that strengthens the inserted L-shaped optimality cuts by simple modifications of the dual solutions of the subproblems. To illustrate the effectiveness of the strengthening procedure, we compare our approach with the classical method by Magnanti and Wong [31] for generating Pareto-optimal L-shaped cuts. Using a large set of realistic instances, we analyze in detail the characteristics of the proposed models and the obtained solutions as well as the performance (e.g., on denser graphs), behavior, and limitations of the designed algorithmic approach. The computational study is completed by an evaluation of the value of the stochastic solution and an analysis of the stability of the scenario generation method.

A small portion of results presented in this paper appeared in the conference proceedings [27]. This article offers a significant extension of the theoretical and computational results which can be summarized as follows: First, two additional MIP models are presented and theoretically compared with the one given in [27]. Second, for the undirected cut-based model given in this paper, we prove that facet defining inequalities can be derived from their deterministic counterpart (under some mild conditions). Third, this article contains detailed descriptions of the two-stage branch&cut algorithm and our cut strengthening procedure (including proofs of its correctness). Last but not least, we present a comprehensive computational study which shows the performance of the decomposition and the cut strengthening; the latter method is additionally compared to the method by Magnanti and Wong [31]. The study also contains an analysis of the value of the stochastic solution and evaluation of the used scenario generation methods.

Organization of the paper. We start by recalling the basic MIP formulation for the deterministic SNDP in Section 2. Moreover, we summarize the ideas of Magnanti and Raghavan [30] for strengthening the undirected formulation by orienting the solution and describe the corresponding MIP model. This model is the starting point for our models concerning the stochastic SNDP in Section 3. Here, the undirected formulation (Section 3.1) and two stronger semi-directed formulations (Sections 3.2 and 3.3, respectively) are described. Furthermore, structural results for the associated polyhedron with the undirected formulation are provided. Section 4 is dedicated to the two-stage branch&cut algorithm with descriptions of the algorithm and the decomposition. Afterwards, in Section 5, we describe the procedure for strengthening the generated L-shaped optimality cuts. The benefit of these cuts—and the decomposition itself—is presented in the results of the experimental study in Section 6.

2. The deterministic survivable network design problem

Definition. Formally, the deterministic version of the SNDP is defined as follows: We are given a simple undirected graph $G = (V, E)$ with edge costs $c_e \geq 0$, $\forall e \in E$, and a symmetric $|V| \times |V|$ *connectivity requirement* matrix $\mathbf{r} = [r_{uv}]$. Thereby, $r_{uv} \in \mathbb{N} \cup \{0\}$ represents the minimal required number of edge-disjoint paths between two distinct nodes $u, v \in V$. The goal consists of finding a subset of edges $E' \subseteq E$ satisfying all connectivity requirements and minimizing the overall solution costs which are defined as $\sum_{e \in E'} c_e$.

While the given definition of the SNDP is as general as possible, one additional assumption is made in this paper which is commonly considered in the literature: it is assumed that the connectivity requirements imply that each optimal solution comprises a single connected component. In this case the SNDP—as well as the connectivity matrix—is called *unitary*. One example for a *non-unitary* SNDP is the Steiner forest problem where optimal solutions might be disconnected.

A closely related problem to the SNDP is the *node-connectivity SNDP* where requirements have to be satisfied by node-disjoint paths. A small example depicting optimal solutions for both problems is illustrated in Figure 1. For the ease of presentation, this article focuses on edge-connectivity but the ideas and algorithms are transferable to the related node-connectivity problem as well.

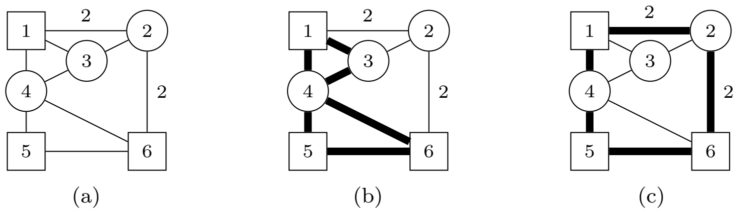


Figure 1: (a) An example network where rectangles have connectivity requirement two w.r.t. each other and all other requirements are zero. Unlabeled edges are assigned cost 1 and labeled edges cost 2. The bold edges depict the optimal solution for (b) edge-connectivity with costs 6 and (c) node-connectivity with costs 7, respectively.

We note that in some applications (including our example mentioned in the introduction), edge- or node-connectivity requirements can be specified using node types $\rho_u \in \mathbb{N} \cup \{0\}$, for all $u \in V$. In that case, the required connectivity between a pair of distinct nodes $u, v \in V$ is defined as $r_{uv} = \min\{\rho_u, \rho_v\}$. In presenting the main results of this paper we will stick to the more general definition of connectivity requirements using the connectivity matrix $\mathbf{r}$. For illustrating examples, however, we will use the more intuitive concept of node types.

(M)ILP models. The classical cut-based integer linear programming (ILP) formulation for the deterministic SNDP (see, e.g., [14, 13]) uses binary decision variables x_e for each edge $e \in E$. We use notations $x(E') := \sum_{e \in E'} x_e, \forall E' \subseteq E$, and $\delta(W) := \{e = \{i,j\} \in E \mid |\{i,j\} \cap W| = 1\}$, for $W \subset V$. Moreover, let us denote by

$$f(W) := \max\{r_{uv} \mid u \in W, v \notin W\}, \quad \forall W \subset V,$$

the *connectivity function* on G. The SNDP based on undirected cuts then reads as follows:

$$(SNDP_{ucut}) \ \min \sum_{e \in E} c_e x_e$$

$$s.t. \quad x(\delta(W)) \geq f(W) \qquad \forall \emptyset \neq W \subset V \qquad (ucut{:}1)$$

$$\boldsymbol{x} \in \{0,1\}^{|E|} \qquad (ucut{:}2)$$

This model is one of the most famous models in the literature and has been used in polyhedral studies (see, e.g., [14]) or to estimate the quality of approximative solutions (see, e.g., [18]). Magnanti and Raghavan [30] showed how to strengthen this formulation by using a famous theorem by Nash-Williams [32] about graph orientations that we restate here:

Theorem 1 (Nash-Williams [32]). *Let $G = (V, E)$ be an undirected graph and let κ_{uv} be the maximum number of edge-disjoint paths from u to v, where $u, v \in V$, $u \neq v$. Then G has an orientation such that for every pair of nodes u and v in G, the number of pairwise edge-disjoint directed paths from u to v in the resulting directed graph is at least $\lfloor \frac{\kappa_{uv}}{2} \rfloor$.*

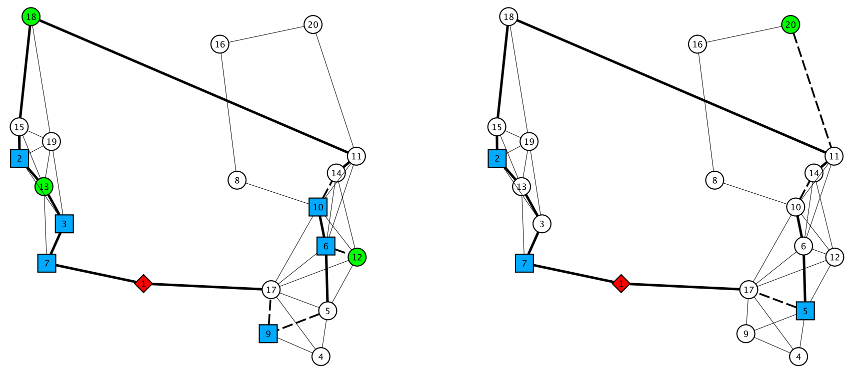
If connectivity requirements are in $\{0, 1\} \cup \{2k \mid k \in \mathbb{N}\}$ then it is possible to orient any optimal SNDP solution as follows: Since we are dealing with the unitary SNDP, any optimal SNDP solution consists of edge-biconnected components connected with each other by cut nodes or bridges. Using the result of Theorem 1, each of those edge-biconnected components can be oriented such that for each pair of distinct nodes u and v from the same component there exist $r_{uv}/2$ edge-disjoint directed paths from u to v and $r_{uv}/2$ edge-disjoint directed paths from v to u. To orient possible bridges, a node v_r is chosen for which we know that it is a part of an edge-biconnected component and each bridge is oriented away from this component. To this end, the edge-biconnected components are oriented, shrunk into single nodes, and the obtained tree is oriented away from the "root" v_r. These orientation properties can be used to derive an MIP model that uses binary arc variables d_{ij} associated to the orientation. By projecting the arc variables into the space of undirected edges as $x_e = d_{ij} + d_{ji}$, for all $e = \{i, j\} \in E$, it is not difficult to see that the obtained directed model is stronger than the undirected one given above. In fact, the directed model is strictly stronger if and only if there exists a pair of distinct nodes $u, v \in V$, such that $r_{uv} = 1$ [30].

To model the general SNDP—i.e., the SNDP with arbitrary connectivity requirements $r_{uv} \in \mathbb{N} \cup \{0\}$—Magnanti and Raghavan [30] present an *extended MIP formulation* which is similar to the one described above with the only difference that the binary arc variables d_{ij} are relaxed to be continuous. This small change makes the model valid for arbitrary values of r_{uv} and provably stronger than its undirected counterpart. For describing this model, we will need the following notation: let A be the arc set of the *bidirection* of G containing two directed arcs for each undirected edge, i.e., $\forall\{i, j\} \in E : (i, j), (j, i) \in A$. For a vertex set $W \subset V$ let $\delta^-(W) = \{(i, j) \in A \mid i \notin W, j \in W\}$ and analogously $\delta^+(W) = \{(i, j) \in A \mid i \in W, j \notin W\}$ be the set of ingoing and outgoing arcs, respectively. By using fractional arc variables $d_{ij}, \forall(i, j) \in A$, the resulting model by [30] reads as follows:

$$(SNDP_{dcut}) \ \min \sum_{e \in E} c_e x_e$$

$$\begin{aligned}
s.t.\quad d(\delta^-(W)) &\geq f(W)/2 && \forall \emptyset \neq W \subset V, f(W) \geq 2 && (dcut{:}1)\\
d(\delta^-(W)) &\geq 1 && \forall \emptyset \neq W \subset V, f(W) = 1, v_r \notin W && (dcut{:}2)\\
x_e &\geq d_{ij} + d_{ji} && \forall e = \{i, j\} \in E && (dcut{:}3)\\
d_{ij} &\geq 0 && \forall (i, j) \in A && (dcut{:}4)\\
\boldsymbol{x} &\in \{0, 1\}^{|E|} && && (dcut{:}5)
\end{aligned}$$

Constraints ($dcut$:2) are classical directed Steiner cuts implying connectivity of the solution. The directed cuts ($dcut$:1) ensure that for each two distinct nodes $u, v \in V$ such that $u \in W, v \notin W$, $r_{uv}/2$ directed paths are selected from u to v and from v to u, respectively. The capacity constraints ($dcut$:3) enforce that each selected directed arc is paid for in the objective function.



(a) (b)

Figure 2: An example SSNDP instance with 20 nodes, 40 edges, and 2 scenarios. Nodes depicted as blue rectangles imply connectivity requirement 2 (together with the red diamond which is a special root node) and green circles imply connectivity 1. (a) and (b) show the optimal solution for the first and second scenario, respectively, with bold edges being selected first- and dashed edges being selected second-stage edges.

The main results concerning the (strength of the) two presented formulations are summarized in the following theorem. Let $\mathcal{P}_{SNDP_{ucut}}, \mathcal{P}_{SNDP_{dcut}}$ denote the polyhedra defined by the linear relaxation of $(SNDP_{ucut})$ and $(SNDP_{dcut})$, respectively. Moreover, let $\mathrm{Proj}_x(\mathcal{P}_{SNDP_{dcut}})$ denote the projection of $\mathcal{P}_{SNDP_{dcut}}$ onto the space of undirected x-variables.

Theorem 2 (Magnanti and Raghavan [30]). *$(SNDP_{dcut})$ is a valid formulation for the SNDP and $(SNDP_{dcut})$ is strictly stronger than $(SNDP_{ucut})$, i.e., $Proj_x(\mathcal{P}_{SNDP_{dcut}}) \subseteq \mathcal{P}_{SNDP_{ucut}}$ and there exist instances for which the relationship is "$\subset$".*

3. The stochastic survivable network design problem

Definition. Let $G = (V, E)$ denote the undirected input graph with known first-stage edge costs $c_e^0 \geq 0$ for all $e \in E$. The actual connectivity requirements as well as the future edge costs are only known in the second stage. These values together form a random variable ξ for which we assume that it has finite support. It can therefore be modeled using a finite set of scenarios $\mathcal{K} = \{1, \ldots, K\}$, $K \geq 1$. The realization probability of each scenario is given by $p^k > 0$, $k \in \mathcal{K}$, with $\sum_{k \in \mathcal{K}} p^k = 1$. Edge costs in the second stage under scenario $k \in \mathcal{K}$ are denoted by $c_e^k \geq 0$ for all $e \in E$. Furthermore, let $\mathbf{r}^k$ be the matrix of *unitary* connectivity requirements in the k-th scenario.

The *two-stage stochastic survivable network design problem* (SSNDP) is defined as follows: Determine the subset of edges $E^0 \subseteq E$ to be purchased in the first stage and the sets $E^k \subseteq E$ of additional (*recourse*) edges to be purchased in each scenario $k \in \mathcal{K}$, such that the overall expected costs defined as

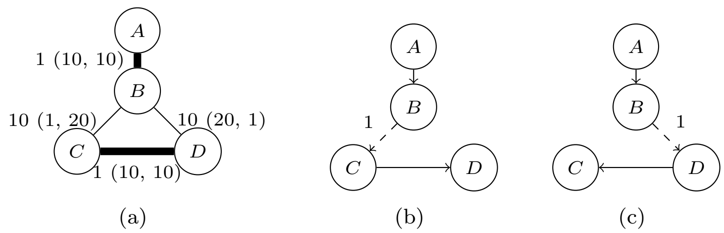


Figure 3: (a) An instance for the SSNDP with a highlighted optimal first-stage solution being a forest. Figures (b) and (c) depict the optimal solution for scenarios 1 and 2, respectively, with selected first-stage edges being drawn as solid lines and second-stage edges as dashed lines. For the orientations in the semi-directed models we assume vertex A being the root node.

$\sum_{e\in E^0} c_e^0 + \sum_{k\in\mathcal{K}} p^k \sum_{e\in E^k} c_e^k$ are minimized. Thereby, $E^0 \cup E^k$ has to satisfy all connectivity requirements between each pair of nodes defined by $\mathbf{r}^k$ for all $k \in \mathcal{K}$. W.l.o.g. we assume that $\sum_{k\in\mathcal{K}} p^k c_e^k \geq c_e^0$, for all $e \in E$; Otherwise, one would never install such an edge in the first stage. We also assume that the connectivity requirement $\mathbf{r}^k$ is *unitary* for each scenario $k \in \mathcal{K}$, as in the deterministic setting. Figure 2 shows an example of an SSNDP instance with $k = 2$ scenarios and its optimal solution.

Solution Topology. Observe that, despite the fact the connectivity requirements are unitary, the optimal first-stage solution E^0 of the SSNDP is not necessarily connected. This holds even if the connectivity requirements are from the set $\{0,1\}$ only. The optimal first-stage solution may contain several disjoint components depending on the values r_{uv}^k throughout different scenarios—or depending on the second-stage cost structure. A small example is shown in Figure 3: This instance consists of $K = 2$ scenarios with equal probability and with the set of nodes $\{A, C, D\}$ being of "type one" in both scenarios; Node B is of "type zero"in both scenarios. The costs are given next to the edges in Figure 3(a) by "$c_e^0(c_e^1, c_e^2)$", for all edges e. The optimal solution consists of the two edges $\{A, B\}$ and $\{C, D\}$ purchased in the first stage expanded by one edge in scenario 1 (edge $\{B, C\}$, Figure 3(b)) and in scenario 2 (edge $\{B, D\}$, Figure 3(c)). Hence, the overall expected costs are 3.

Moreover, notice that the connectivity requirements in the scenarios do not necessarily imply connectivity requirements of the first stage solution. Imposing particular connectivity requirements (between some important nodes, for example) is not prohibited by our model and can be easily handled by additional constraints regarding the topology of the first stage solution (similar to (*ucut*:1), see, e.g., [12]). However, such additional constraints imply additional costs and provide sub-optimal solutions, as demonstrated in a simple example given by Figure 4: Even though all connectivity requirements are identical over all scenarios, the optimal first stage solution is empty.

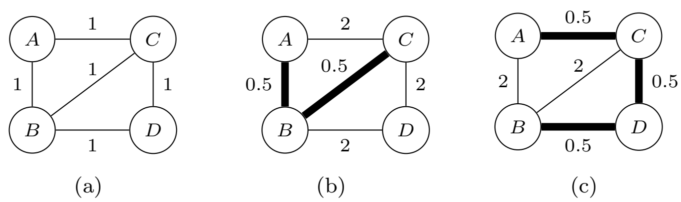


Figure 4: An instance for the SSNDP with two equally probable scenarios which have the same connectivity requirements: vertices $\{A, B, C\}$ need to be simply connected. However, in the optimal solution, no edge is selected in the first stage, cf. Figure (a). Figures (b) and (c) depict the two scenarios with the selected edges of the optimal solution being drawn as bold lines. Overall solution costs are 1.25 whereas the selection of any first stage edge would lead to larger costs.

3.1. Undirected model

We first present the deterministic equivalent—in extensive form—of the SSNDP in the natural space of undirected edge variables. Later on we show how to derive stronger extended formulations using the orientation properties presented in Section 2 by assigning a unique direction to each edge of a feasible *second-stage solution*.

Let binary variables x_e^0 indicate whether an edge $e \in E$ belongs to E^0, and binary second-stage variables x_e^k indicate whether e belongs to E^k, for all scenarios $k \in \mathcal{K}$. For $E' \subseteq E$ let $(x^0 + x^k)(E') := \sum_{e \in E'}(x_e^0 + x_e^k)$. Moreover, we expand the connectivity function to f^k, for each scenario $k \in \mathcal{K}$ and $W \subset V$:

$$f^k(W) := \max\{r_{uv}^k \mid u \notin W, v \in W\}.$$

A deterministic equivalent of the SSNDP can then be modeled easily using undirected cuts as follows:

$$(UD)\ \min \sum_{e \in E} c_e^0 x_e^0 + \sum_{k \in \mathcal{K}} p^k \sum_{e \in E} c_e^k x_e^k$$

$$\begin{aligned} s.t. \quad & (x^0 + x^k)(\delta(W)) \geq f^k(W) && \forall \emptyset \neq W \subset V, \forall k \in \mathcal{K} && (UD{:}1)\\ & x_e^0 + x_e^k \leq 1 && \forall e \in E, \forall k \in \mathcal{K} && (UD{:}2)\\ & (\boldsymbol{x^0}, \boldsymbol{x^1}, \ldots, \boldsymbol{x^K}) \in \{0,1\}^{|E|(K+1)} && && (UD{:}3) \end{aligned}$$

This model is a direct extension of the model from Section 2. Constraints $(UD{:}1)$ ensure edge-connectivity between each pair of nodes in each scenario realization while first- and second-stage edges can be used. The additional constraints $(UD{:}2)$ simply forbid the installation of the same edge in the first stage and in any scenario.

Polyhedral properties. Let $\mathcal{S}^k$ be the convex hull of all integer points that define feasible SNDP solutions w.r.t. connectivity requirements $\mathbf{r}^k$, i.e.,

$$\mathcal{S}^k = \text{conv}\{\boldsymbol{x^k} \in \{0,1\}^{|E|} \mid x^k(\delta(W)) \geq f^k(W), \forall \emptyset \neq W \subset V\}.$$

$$\begin{bmatrix} \boldsymbol{A_0} & \boldsymbol{I_m} & \boldsymbol{I_m} & \boldsymbol{I_m} & \boldsymbol{I_m} \\ \mathbf{0} & \boldsymbol{A_0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{A_0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \boldsymbol{A_0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \boldsymbol{A_0} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{A^\ell} & \mathbf{1}-\boldsymbol{A^\ell} & \mathbf{1}-\boldsymbol{A^\ell} & \mathbf{0} & \mathbf{1}-\boldsymbol{A^\ell} \\ \mathbf{0} & \boldsymbol{A_0} & \mathbf{1} & \boldsymbol{A^\ell} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{A_0} & \boldsymbol{A^\ell} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \boldsymbol{A^\ell} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \boldsymbol{A^\ell} & \boldsymbol{A_0} \end{bmatrix}$$

Figure 5: Structures of the constructed matrices (with $K = 4$) from Lemma 1 (*left*) and Theorem 3 (*right*); for the latter we have $\ell = 3$.

Similarly, let $\mathcal{S}$ be the convex hull of all integer points that define feasible SSNDP solutions, i.e.,

$$\mathcal{S} = \text{conv}\{\boldsymbol{x} = (\boldsymbol{x^0}, \boldsymbol{x^1}, \ldots, \boldsymbol{x^K}) \in \{0,1\}^{|E|(K+1)} \mid \boldsymbol{x} \text{ satisfies } (UD{:}1),(UD{:}2)\}.$$

In the following, we will study some properties of the polytope $\mathcal{S}$.

Lemma 1. *If the polytopes $\mathcal{S}^k$ for all $k \in \mathcal{K}$ are full-dimensional, then the polytope $\mathcal{S}$ is full-dimensional as well.*

Proof. Let $m := |E|$. We need to show that $\dim(\mathcal{S}) = m(K+1)$. To this end, we now construct a matrix that contains $mK + m$ linearly independent feasible solutions to SSNDP. In the last step we extend it by one more solution with the whole collection of solutions being affinely independent.

The matrix is constructed in $(K+1) \cdot (K+1)$ blocks of size $m \times m$ and each row of the matrix represents one feasible solution in $\mathcal{S}$. Each block column corresponds to a binary variable of the vector $(\boldsymbol{x^0}, \boldsymbol{x^1}, \ldots, \boldsymbol{x^K})$; the first m rows represent feasible independent solutions involving the $\boldsymbol{x^0}$ variables, the next Km solutions are linearly independent with respect to the $\boldsymbol{x^k}$ variables, for all $k \in \mathcal{K}$.

For each edge $e \in E$ let the solution s_e contain all edges except e. Then, we observe that for each scenario $k \in \mathcal{K}$ the collection of the m solutions $\mathcal{E} = \cup_{e \in E} s_e$ represents a set of m linearly independent points of the polytope $\mathcal{S}^k$. Let $\boldsymbol{A_0}$ denote the $m \times m$ matrix obtained by row-wise concatenation of the characteristic vectors of these solutions, i.e., $\boldsymbol{A_0} = \mathbf{1} - \boldsymbol{I_m}$.

1. Initialize the first $m \times m$ block with $\boldsymbol{A_0}$. Fill out the remaining K blocks at position $[0, k], k \in \mathcal{K}$, with the $m \times m$ identity matrix $\boldsymbol{I_m}$.
2. For $k \in \mathcal{K}$: set up the block at the position $[k, 0]$ to $\mathbf{0}$, and the block at the position $[k, k]$ to $\boldsymbol{A_0}$. The remaining blocks at positions $[\ell, k]$ are set to $\mathbf{1}$, for all $\ell \in \mathcal{K}, \ell \neq k$.

It is not difficult to see that the obtained matrix (cf. Figure 5) has full rank $mK + m$. In the last step, we add the vector that is obtained by concatenating the $\mathbf{0}$ vector solution for $\boldsymbol{x^0}$ and $\mathbf{1}$'s for the remaining coordinates $\boldsymbol{x^1}$ to $\boldsymbol{x^K}$. Subtracting all solutions contained in the matrix from the latter solution gives a new matrix—with full rank, too. Hence, all solutions are affinely independent. $\square$

Theorem 3. *If for all $k \in \mathcal{K}$, the polytopes $\mathcal{S}^k$ are full-dimensional and the inequality $\boldsymbol{\pi x^\ell} \geq \pi_0$, with coefficients $\pi_e \in \mathbb{N} \cup \{0\}, \forall e \in E$, and $\pi_0 \geq 1$, defines a facet of the polytope $\mathcal{S}^\ell$ for some $\ell \in \mathcal{K}$, then the inequality $\boldsymbol{\pi x^0} + \boldsymbol{\pi x^\ell} \geq \pi_0$ is facet defining for the polytope $\mathcal{S}$.*

Proof. We denote the affine independent solutions of the polytope $\mathcal{S}^\ell$ that satisfy $\boldsymbol{\pi x^\ell} = \pi_0$ by $T_1^\ell, \dots, T_m^\ell$. Since $\pi_0 \geq 1$, these points are also linearly independent (the origin does not belong to the set of feasible points). Let $\boldsymbol{A^\ell}$ be the matrix obtained by the row-wise insertion of these solutions, and let $\mathbf{1} - \boldsymbol{A^\ell}$ be the complementary matrix of $\boldsymbol{A^\ell}$. Construction is done by a row-wise insertion of blocks, similarly as above.

1. Initialize the first $m \times m$ block with $\boldsymbol{A^\ell}$ (meaning, set first-stage solutions to be equal to the solutions of $\mathcal{S}^\ell$). Fill out the remaining blocks at the position $[0, k]$ with $\mathbf{1} - \boldsymbol{A^\ell}$, for all $k \in \mathcal{K}$, $k \neq \ell$. The block at the position $[0, \ell]$ is set to $\mathbf{0}$.

2. For $k \in \mathcal{K}$, $k \neq \ell$: set up the block at the position $[k, 0]$ to $\mathbf{0}$, at the position $[k, \ell]$ to $\boldsymbol{A^\ell}$ and the block at the position $[k, k]$ to $\boldsymbol{A_0}$ (defined above). The remaining blocks at $[k, i]$ are set to $\mathbf{1}$, for all $i \in \mathcal{K}, i \neq k, \ell$.

3. Set up the block at the position $[\ell, 0]$ to $\mathbf{0}$, and the block at $[\ell, \ell]$ to $\boldsymbol{A^\ell}$. The remaining blocks at $[\ell, i]$ are set to $\mathbf{1}$, for all $i \in \mathcal{K}, i \neq \ell$.

It is not difficult to see that the obtained matrix (cf. Figure 5) has full rank, i.e., $(K+1)m$, and each row satisfies $\boldsymbol{\pi x^0} + \boldsymbol{\pi x^\ell} = \pi_0$ which concludes the proof. □

Many facet-defining inequalities (see, e.g., [46]), known for the deterministic case, can therefore be easily translated into facets of the SSNDP. There is also a large body of work on polyhedral studies for many variants of the SNDP (see, e.g., surveys in [20, 22]).

Let F be an extended MIP formulation for the SSNDP. We will denote the polyhedron defined by the LP-relaxation of F by $\mathcal{P}_F$ and the natural projection of $\mathcal{P}_F$ onto the space of undirected $(\boldsymbol{x^0}, \boldsymbol{x^1}, \dots, \boldsymbol{x^K})$ variables by $\mathrm{Proj}_{(x^0,\dots,x^K)}(\mathcal{P}_F)$. Furthermore, for two (extended) formulations F_1 and F_2, we will say that F_1 is *strictly stronger* than F_2 if and only if $\mathrm{Proj}_{(x^0,\dots,x^K)}(\mathcal{P}_{F_1}) \subseteq \mathrm{Proj}_{(x^0,\dots,x^K)}(\mathcal{P}_{F_2})$ and there exists an instance for which the relationship is "$\subset$", i.e., for which the bound of the LP-relaxation of F_1 is tighter than the one of the LP-relaxation of F_2.

3.2. Semi-directed model

It is known that MIP models on bidirected graphs provide better LP-based lower bounds for many types of network design problems, cf. Section 2. In the following we present a way for strengthening the model (UD) by bi-directing the given graph G and replacing edge- by arc-variables in the same model. The main difficulty with the SSNDP arises from the fact that the first-stage solution may be disconnected (cf. Figure 3), and as such it cannot be oriented, even though the deterministic counterpart admits an orientation. In the following, we will introduce two extended formulations (semi-directed models) to overcome these obstacles and provide two MIP models that are strictly stronger than the model (*UD*).

The example given in Figure 3 illustrates an instance in which it is impossible to uniquely orient the edges from E^0 since there exists an edge from E^0 that cannot be used in exactly the same direction over all scenarios. More precisely, the edge $\{C, D\}$ is selected in the first stage and during the orientation process it is used in the direction (C, D) in scenario 1 and in the direction (D, C) in scenario 2, respectively. Therefore, requiring a fixed orientation for an edge in the first stage would conflict with optimal scenario solutions and would in total lead to more expensive, suboptimal solutions.

Hence, the first-stage decision variables need to remain associated with undirected edges. However, one can provide a directed formulation once the solution gets completed in the second stage, i.e., one can orient the edges of $E^0 \cup E^k$ independently for each scenario, as depicted in Figures 3(b) and 3(c). We set the root v_r^k for each scenario $k \in \mathcal{K}$ to be one of the nodes with the highest connectivity requirement and search for individual orientations of the scenario solutions $E^0 \cup E^k$, for each $k \in \mathcal{K}$.

By borrowing the notation from [1], let

$$\begin{aligned}
\mathcal{W}_1^k :=& \{W \mid W \subset V, f^k(W) = 1, v_r^k \notin W\} \\
\mathcal{W}_{\geq 2}^k :=& \{W \mid W \subset V, f^k(W) \geq 2\}
\end{aligned}$$

be the set of *regular cutsets* and *critical cutsets*, respectively.

Given the installation of undirected edges from the first stage, the following model constructs oriented second-stage solutions. As above, we use variables $\boldsymbol{x^0}, \ldots, \boldsymbol{x^K}$ to model the solution edges. In addition, we introduce continuous variables $\boldsymbol{d^1}, \ldots, \boldsymbol{d^K}$ associated to directed arcs to "orient" the second-stage solutions. The first semi-directed model is called (*SD₁*):

$$(SD_1)\ \min \sum_{e \in E} c_e^0 x_e^0 + \sum_{k \in \mathcal{K}} p^k \sum_{e \in E} c_e^k x_e^k$$

$$\begin{aligned}
s.t.\quad (x^0 + x^k)(\delta(W)) &\geq f^k(W) && \forall W \in \mathcal{W}_{\geq 2}^k, \forall k \in \mathcal{K} && (SD_1{:}1) \\
x^0(\delta(W)) + d^k(\delta^-(W)) &\geq 1 && \forall W \in \mathcal{W}_1^k, \forall k \in \mathcal{K} && (SD_1{:}2) \\
x_e^k &\geq d_{ij}^k + d_{ji}^k && \forall e = \{i, j\} \in E, \forall k \in \mathcal{K} && (SD_1{:}3) \\
x_e^0 + x_e^k &\leq 1 && \forall e \in E, \forall k \in \mathcal{K} && (SD_1{:}4) \\
d_{ij}^k &\geq 0 && \forall (i, j) \in A, \forall k \in \mathcal{K} && (SD_1{:}5) \\
(\boldsymbol{x^0}, \boldsymbol{x^1}, \ldots, \boldsymbol{x^K}) &\in \{0, 1\}^{|E|(K+1)} && && (SD_1{:}6)
\end{aligned}$$

Constraints (SD_1:1) ensure that in each scenario k there are at least r_{uv}^k edge-disjoint paths between u and v, $u \in W, v \notin W$, consisting of first- and second-stage edges. Due to constraints (SD_1:2) there is at least one path from the root node v_r^k to each vertex u whenever $r_{v_r^k u} = 1$. If an edge is purchased in the second stage, then constraints (SD_1:2) associated to bridges will force the orientation of those bridges away from the root node v_r^k. Furthermore, since variables d_{ij}^k are fractional, by using the same arguments as in Theorem 2 the model is valid for any $r_{uv}^k \in \mathbb{N} \cup \{0\}$. Hence, we have the following lemma.

Lemma 2. *Formulation (SD_1) models the deterministic equivalent of the two-stage stochastic survivable network design problem correctly.*

Proof. Consider an optimal solution to an SSNDP instance with its characteristic binary vectors being described by $\tilde{\boldsymbol{x}} := (\tilde{\boldsymbol{x}}^{\boldsymbol{0}}, \tilde{\boldsymbol{x}}^{\boldsymbol{1}}, \dots, \tilde{\boldsymbol{x}}^{\boldsymbol{K}})$. Using $\tilde{\boldsymbol{x}}$ as solution to (SD_1), all undirected cuts (constraints (SD_1:1)) are obviously satisfied. Moreover, due to Theorem 1 and 2 it is possible to find an orientation in each scenario such that constraints (SD_1:2) are satisfied. Following this orientation the values for the directed variables d^k can be set accordingly. Non-negativity and all other constraints follow directly. Hence, there is a feasible solution to (SD_1) with the same objective value.

On the other hand, each optimal solution to (SD_1) obviously satisfies all connectivity requirements and induces a solution to the SSNDP with the same costs. □

Note that constraints (SD_1:1) can also be expressed as $x^0(\delta(W)) + d^k(\delta^-(W)) + d^k(\delta^+(W)) \geq f^k(W)$ (which better explains the original intention of this model). However, one easily observes that this is just an equivalent way of rewriting (SD_1:1), without any influence on the lower bounds of the given model.

Theorem 4. *The semi-directed formulation (SD_1) is strictly stronger than the undirected formulation (UD).*

Proof. It follows from the proof of Lemma 2 that any solution $(\tilde{\boldsymbol{x}}^{\boldsymbol{0}}, \tilde{\boldsymbol{x}}^{\boldsymbol{1}}, \dots, \tilde{\boldsymbol{x}}^{\boldsymbol{K}}) \in \mathrm{Proj}_{(x^0,\dots,x^K)}(\mathcal{P}_{SD_1})$ is a valid solution to (UD) with the same objective value.

The strict inequality concerning the strength of the formulations is shown by the example given in Figure 6(a). Assume there are 2 scenarios with equal probability and the non-zero connectivity requirements $r_{01}^1 = r_{02}^1 = r_{12}^1 = 1$, $r_{03}^2 = 1$, and $c_e^0 = 10, c_e^k = 12, \forall e \in E, k \in \{1,2\}$. In the optimal solution of (UD), only edges in the second stage are purchased with a total objective value of 15: $x_{01}^1 = x_{12}^1 = x_{02}^1 = 0.5$ and $x_{03}^2 = 1$. On the other hand, this solution is infeasible for the relaxed model of (SD_1), i.e., there is no solution to (SD_1) with the same objective value. □

3.3. Stronger semi-directed formulation

In the following model, which represents an alternative model to (SD_1), binary edge variables $\boldsymbol{y}^{\boldsymbol{k}}$ are used to model the second-stage solution. These variables additionally include the edges that are already bought in the first stage, i.e., we have $y_e^k = 1$ if $e \in E^0 \cup E^k$, and $y_e^k = 0$, otherwise. Moreover,

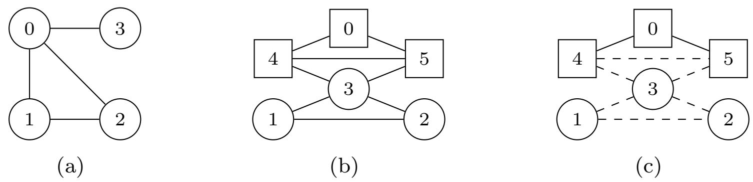


Figure 6: Two counterexamples that prove the strength of the new formulations. (a) Instance with $LP(SD_1) > LP(UD)$, (b) instance with $LP(SD_2) > LP(SD_1)$, and (c) the optimal LP-solution of (SD_1): A solid line represents an LP value of 1, a dashed line a value of 0.5. Rectangles have connectivity requirement two, all other nodes connectivity requirement one.

continuous variables z_{ij}^k are used to orient the edges from $E^0 \cup E^k$. The model will be called (SD_2):

$$(SD_2) \ \min \sum_{e \in E} c_e^0 x_e^0 + \sum_{k \in \mathcal{K}} p^k \sum_{e \in E} c_e^k (y_e^k - x_e^0)$$

$$\begin{aligned}
s.t. \quad & z^k(\delta^-(W)) \geq f^k(W)/2 && \forall W \in \mathcal{W}_{\geq 2}^k, \forall k \in \mathcal{K} && (SD_2{:}1)\\
& z^k(\delta^-(W)) \geq 1 && \forall W \in \mathcal{W}_1^k, \forall k \in \mathcal{K} && (SD_2{:}2)\\
& z_{ij}^k + z_{ji}^k \geq x_e^0 && \forall e = \{i,j\} \in E, \forall k \in \mathcal{K} && (SD_2{:}3)\\
& y_e^k \geq z_{ij}^k + z_{ji}^k && \forall e = \{i,j\} \in E, \forall k \in \mathcal{K} && (SD_2{:}4)\\
& z_{ij}^k \geq 0 && \forall (i,j) \in A, \forall k \in \mathcal{K} && (SD_2{:}5)\\
& (\boldsymbol{x^0}, \boldsymbol{y^1}, \ldots, \boldsymbol{y^K}) \in \{0,1\}^{|E|(K+1)} && && (SD_2{:}6)
\end{aligned}$$

The directed cuts $(SD_2{:}1)$ and $(SD_2{:}2)$ model the orientation of the solution and ensure the required connectivities independently for each scenario. Notice that due to the symmetry, if $W \in \mathcal{W}_{\geq 2}^k$ it follows that $V \setminus W \in \mathcal{W}_{\geq 2}^k$, too. Hence, for each $W \in \mathcal{W}_{\geq 2}^k$ the ingoing and outgoing cut, i.e., $z^k(\delta^-(W)) \geq f^k(W)/2$ and $z^k(\delta^+(W)) \geq f^k(W)/2$, is contained in (SD_2).

Constraints $(SD_2{:}3)$ and $(SD_2{:}4)$ ensure that variables z_{ij}^k can be used only along the edges that are either purchased in the first stage or added in the second stage. In particular, $(SD_2{:}3)$ forces the orientation of selected first-stage edges in each scenario. Therefore, these constraints strengthen the model as they impose restrictions on the first-stage solutions: only first-stage solutions allowing for feasible orientations are valid. Moreover, it holds $y_e^k \geq x_e^0$ which explains the corrective term in the objective function. Since the variables z_{ij}^k are fractional, the model is valid for any $r_{uv}^k \in \mathbb{N} \cup \{0\}$; The straightforward proof is omitted here.

Lemma 3. *Formulation (SD_2) models the deterministic equivalent of the two-stage stochastic survivable network design problem correctly.*

The following result shows that the new way of orienting the second stage solutions provides strictly stronger lower bounds than the model shown in the previous section. Note that this result applies to two-stage survivable network design models only, and its proof cannot be derived from Theorem 2.

Theorem 5. *The semi-directed formulation* (SD_2) *is strictly stronger than the semi-directed formulation* (SD_1).

PROOF. Let $(\hat{\boldsymbol{x}}^{\mathbf{0}}, \hat{\boldsymbol{y}}^{\mathbf{1}}, \ldots, \hat{\boldsymbol{y}}^{\boldsymbol{K}}, \hat{\boldsymbol{z}}^{\mathbf{1}}, \ldots, \hat{\boldsymbol{z}}^{\boldsymbol{K}}) \in \mathcal{P}_{SD_2}$. For $k \in \mathcal{K}$ and $(i,j) \in A$ set $\lambda_{ij}^k := 0$ if $z_{ij}^k + z_{ji}^k = 0$ and $\lambda_{ij}^k := z_{ij}^k/(z_{ij}^k + z_{ji}^k)$, otherwise. Hence, $\lambda_{ij}^k + \lambda_{ji}^k = 1, \forall \{i,j\} \in E$ and $k \in \mathcal{K}$ with $\hat{z}_{ij}^k + \hat{z}_{ji}^k > 0$. Moreover, set $\bar{\boldsymbol{x}}^{\mathbf{0}} := \hat{\boldsymbol{x}}^{\mathbf{0}}, \bar{\boldsymbol{x}}^{\boldsymbol{k}} := \hat{\boldsymbol{y}}^{\boldsymbol{k}} - \hat{\boldsymbol{x}}^{\mathbf{0}}$, and $\forall (i,j) \in A$ with $e = \{i,j\} : \bar{d}_{ij}^k := \hat{z}_{ij}^k - \lambda_{ij}^k \hat{x}_e^0$, for all $k \in \mathcal{K}$.

Obviously, interpreting $(\bar{\boldsymbol{x}}^{\mathbf{0}}, \bar{\boldsymbol{x}}^{\mathbf{1}} \ldots, \bar{\boldsymbol{x}}^{\boldsymbol{K}}, \bar{\boldsymbol{d}}^{\mathbf{1}}, \ldots, \bar{\boldsymbol{d}}^{\boldsymbol{K}})$ as (SD_1)-solution gives the same objective value. This solution is also feasible due to the following arguments.

The connectivity constraints (SD_1:1) are satisfied since for each $W \in \mathcal{W}_{\geq 2}^k$, $k \in \mathcal{K}$, we have:

$$\begin{aligned} & (\bar{x}^0 + \bar{x}^k)(\delta(W)) = \hat{x}^0(\delta(W)) + \hat{y}^k(\delta(W)) - \hat{x}^0(\delta(W)) \\ \geq & \sum_{e=\{i,j\} \in \delta(W)} (\hat{z}_{ij}^k + \hat{z}_{ji}^k) = \hat{z}^k(\delta^-(W)) + \hat{z}^k(\delta^+(W)) \geq f^k(W) \end{aligned}$$

The 1-connectivity constraints (SD_1:2) are also fulfilled since for each $W \in \mathcal{W}_1^k$, $k \in \mathcal{K}$, we have:

$$\begin{aligned} \bar{x}^0(\delta(W)) + \bar{d}^k(\delta^-(W)) &= \hat{x}^0(\delta(W)) + \sum_{(i,j) \in \delta^-(W), e=\{i,j\}} (\hat{z}_{ij}^k - \lambda_{ij}^k \hat{x}_e^0) \\ &\geq \hat{z}^k(\delta^-(W)) \geq 1 \end{aligned}$$

The remaining constraints are also satisfied: (SD_1:3): $\bar{d}_{ij}^k + \bar{d}_{ji}^k = \hat{z}_{ij}^k + \hat{z}_{ji}^k - \hat{x}_e^0 \leq \hat{y}_e^k - \hat{x}_e^0 = \bar{x}_e^k$, ($SD_1$:4): $\bar{x}_e^0 + \bar{x}_e^k = \hat{x}_e^0 + \hat{y}_e^k - \hat{x}_e^0 \leq 1$. Moreover, d_{ij}^k-variables are non-negative: $\bar{d}_{ij}^k = \hat{z}_{ij}^k - (\hat{z}_{ij}^k/(\hat{z}_{ij}^k + \hat{z}_{ji}^k))\hat{x}_e^0 \geq \hat{z}_{ij}^k - (\hat{z}_{ij}^k/(\hat{z}_{ij}^k + \hat{z}_{ji}^k))(\hat{z}_{ij}^k + \hat{z}_{ji}^k) = 0$.

Last but not least, it trivially holds $\bar{\boldsymbol{x}}^{\mathbf{0}} \in [0,1]^{|E|}$ and $\bar{\boldsymbol{x}}^{\boldsymbol{k}} \in [0,1]^{|E|}, \forall k \in \mathcal{K}$, since $\bar{x}_e^k = \hat{y}_e^k - \hat{x}_e^0$ and $\hat{y}_e^k \geq \hat{x}_e^0$. Hence, $(\bar{\boldsymbol{x}}^{\mathbf{0}}, \bar{\boldsymbol{x}}^{\mathbf{1}} \ldots, \bar{\boldsymbol{x}}^{\boldsymbol{K}}, \bar{\boldsymbol{d}}^{\mathbf{1}}, \ldots, \bar{\boldsymbol{d}}^{\boldsymbol{K}})$ is a feasible solution for (SD_1) with the same objective value.

To show that there exists an instance for which the strict inequality holds, consider the graph shown in Figure 6(b). We assume that the input consists of a single scenario in which the gray nodes require two-connectivity and the remaining ones only one-connectivity. Furthermore, all edge costs are 1 in the first stage and 10 in the second stage. The LP solution shown in Figure 6(c) shows the first-stage solution—nothing needs to be purchased in the second stage—with a total objective value of 5. This solution is valid for the model (SD_1) but it is impossible to "orient" this solution such that it becomes feasible for the model (SD_2). □

Intuitively, formulation (SD_2) gives stronger lower bounds than (SD_1) for instances where the (fractional) undirected first-stage edges allow no valid orientation. Here, the first stage covers cuts in (SD_1) but (SD_2) has to purchase additional arcs in the second stage to ensure feasibility.

4. Decomposition

Notice that, even for a constant number of scenarios, our deterministic equivalent models contain an exponential number of constraints associated with directed cuts. Although these cuts can be separated in a cutting-plane fashion leading to a branch&cut approach, the main drawback of such a *single-stage* branch&cut approach is that we still have to deal with a large set of variables; e.g., formulation (SD_2) contains $|E|(3K+1)$ variables.

Due to the sparsity and the block-angular structure of the constraint matrix, decomposition methods (see, e.g., [6, 40, 39]) proved to be an effective way for solving stochastic optimization problems. Our method of choice in the present article is Benders decomposition, originally introduced in [2] and also known as the L-shaped method for linear stochastic programs, cf., [3, 24].

Nowadays most MIP solvers provide branch&cut frameworks such that Benders decomposition can be implemented as a pure branch&cut approach by the use of callbacks. In the stochastic programming context, Benders cuts are added to the master problem to model valid lower bounds on the expected second-stage costs. This idea has been also exploited in various other applications (including some deterministic problems) where "complicated variables" are projected out and replaced by Benders cuts [5, 28]. Typically, finding a single violated Benders cut requires solving several *compact MIP or LP models.*

When applying the Benders decomposition concept to the proposed MIP models for the SSNDP, attention should be given to the following two nonstandard aspects: a) First of all, one has to deal with the integer recourse. For that purpose, we integrate a separation of the integer L-shaped cuts within a branch&cut framework. b) The second main difficulty arises with the fact that the associated subproblems contain an exponential number of constraints, and can therefore be solved only by means of a cutting plane approach (for finding optimal LP solutions), or branch&cut (for finding optimal integer solutions). Therefore, in order to apply a Benders-like decomposition, one needs to *nest branch&cut algorithms*: a branch&cut is employed for solving the master problem and violated Benders cuts are detected by solving the K subproblems with a dedicated branch&cut algorithm. In [4], this approach has been called *two-stage branch&cut algorithm (2-stage b&c).*

More precisely, in the 2-stage b&c, the variables of the first stage are kept in the master problem, and the second-stage variables are projected out and replaced by a single variable per scenario, i.e., $\Theta^k, \forall k \in \mathcal{K}$. The objective function of the decomposed model becomes $\min \sum_{e \in E} c_e^0 x_e^0 + \Theta$ with the expected second-stage costs $\Theta = \sum_{k \in \mathcal{K}} p^k \Theta^k$ and with Θ^k representing a lower bound on the value of the second-stage subproblem in scenario $k \in \mathcal{K}$. For a fixed first-stage decision $\tilde{\boldsymbol{x}}^{\mathbf{0}}$, the problem decomposes into K smaller subproblems, each of which can be independently solved using a branch&cut approach. Dual variables of the LP-relaxations of these subproblems impose L-shaped cuts that are added to the master while the exact solutions of the subproblems impose integer optimality cuts [24, 47].

The key results that allow us to apply the 2-stage b&c to the SSNDP and detect violated L-shaped cuts in polynomial time are 1) the fact that all considered cut inequalities can be separated in polynomial time (see, e.g., [46]), and 2) the famous result by Grötschel, Lovász, and Schrijver (cf., e.g, [33]) that shows the equivalence of optimization and separation. Consequently, in each of the subproblems, the number of tight inequalities of an arbitrary LP optimal solution is polynomial, and therefore, only a polynomial number of dual multipliers will be non-zero in the associated L-shaped constraint. Therefore, we are able to apply the 2-stage b&c to any of the three formulations considered in this paper. For the ease of presentation, we will demonstrate how to solve the SSNDP using the 2-stage b&c applied to the strongest of the three models, namely (SD_2). The main details of this algorithm are provided in Section 4.2.

4.1. Decomposition of the (SD_2) model

For each fixed—and possibly fractional—first-stage solution $\tilde{\boldsymbol{x}}^{\boldsymbol{0}}$, the second-stage problem decomposes into K independent subproblems, which we will refer to as *restricted deterministic SNDP's*. They are special cases of the deterministic SNDP due to the capacity constraints (P:2) (see below). To simplify the notation, we define $\mathcal{W}^k := \mathcal{W}_1^k \cup \mathcal{W}_{\geq 2}^k$ and merge the constraints (SD_2:1) and (SD_2:2) by using functions $\Phi^k : 2^V \mapsto \mathbb{N} \cup \{0\}$, for all $k \in \mathcal{K}$, that give the correct right-hand side of the directed cuts:

$$\Phi^k(W) := \begin{cases} \frac{1}{2} f^k(W), & W \in \mathcal{W}_{\geq 2}^k \\ 1, & W \in \mathcal{W}_1^k \end{cases}$$

For a given first-stage solution $\tilde{\boldsymbol{x}}^{\boldsymbol{0}}$, and for each $k \in \mathcal{K}$, these subproblems—already transformed into standard form—are given as follows:

$$(P{:}SD_2) \ \min \sum_{e \in E} c_e^k y_e^k - \sum_{e \in E} c_e^k \tilde{x}_e^0$$

$$\begin{aligned}
s.t. \quad z^k(\delta^-(W)) &\geq \Phi^k(W) && \forall W \in \mathcal{W}^k && (P{:}1)\\
z_{ij}^k + z_{ji}^k &\geq \tilde{x}_e^0 && \forall e = \{i,j\} \in E && (P{:}2)\\
y_e^k - z_{ij}^k - z_{ji}^k &\geq 0 && \forall e = \{i,j\} \in E && (P{:}3)\\
-y_e^k &\geq -1 && \forall e \in E && (P{:}4)\\
z_{ij}^k &\geq 0 && \forall (i,j) \in A && (P{:}5)\\
\boldsymbol{y^k} &\in \{0,1\}^{|E|} && && (P{:}6)
\end{aligned}$$

To improve readability we move the corrective constant term w.r.t. $\tilde{\boldsymbol{x}}^{\boldsymbol{0}}$ in the objective function into a second sum, i.e., $-\sum_{e \in E} c_e^k \tilde{x}_e^0$.

By relaxing the integrality constraints (P:6) and using dual variables α_W, β_e, γ_e, and τ_e associated to constraints (P:1), (P:2), (P:3), and (P:4), respectively, we obtain the following dual problem, for each scenario $k \in \mathcal{K}$ and fixed first-stage solution $\tilde{\boldsymbol{x}}^{\boldsymbol{0}}$:

$$(D{:}SD_2) \ \max \sum_{W \in \mathcal{W}^k} \Phi^k(W)\alpha_W + \sum_{e \in E} (\tilde{x}_e^0 \beta_e - \tau_e) - \sum_{e \in E} c_e^k \tilde{x}_e^0$$

$$\gamma_e - \tau_e \le c_e^k \quad \forall e \in E \tag{D:1}$$

$$\sum_{\substack{W \in \mathcal{W}^k: \\ (i,j) \in \delta^-(W)}} \alpha_W + \beta_e - \gamma_e \le 0 \quad \forall (i,j) \in A, e = \{i,j\} \tag{D:2}$$

$$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\tau} \ge \mathbf{0} \tag{D:3}$$

Let $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\tau}})$ describe an optimal solution to (D:SD_2). A (decomposed) *L-shaped optimality cut* is then defined as follows:

$$\Theta^k + \sum_{e \in E} (c_e^k - \tilde{\beta}_e) x_e^0 \ge \sum_{W \in \mathcal{W}^k} \Phi^k(W) \tilde{\alpha}_W - \sum_{e \in E} \tilde{\tau}_e \tag{LS}$$

In the algorithm such a constraint is added if $\Theta^k < \sum_{W \in \mathcal{W}^k} \Phi^k(W) \tilde{\alpha}_W + \sum_{e \in E} (\tilde{x}_e^0 \tilde{\beta}_e - \tilde{\tau}_e - c_e^k \tilde{x}_e^0)$.

Notice that the right hand side of (LS) is a constant. Depending on its value, the obtained cut can be strengthened by rounding: the coefficients next to each x_e^0 can be replaced by $\min\{c_e^k - \tilde{\beta}_e, \sum_{W \in \mathcal{W}^k} \Phi^k(W) \tilde{\alpha}_W - \sum_{e \in E} \tilde{\tau}_e\}$, for each $e \in E$ and $k \in \mathcal{K}$. Validity of this rounding procedure can be easily shown by case distinction concerning the sign of coefficients next to x_e^0 and the sign of the right-hand side.

Integer optimality cuts. Let $(\tilde{\boldsymbol{x}}^{\mathbf{0}}, \tilde{\Theta})$ be a first-stage solution with $\tilde{\boldsymbol{x}}^{\mathbf{0}}$ being binary. Moreover, let $Q(\tilde{\boldsymbol{x}}^{\mathbf{0}}, k)$ denote the optimal binary solution for the kth scenario with the second-stage value $Q(\tilde{\boldsymbol{x}}^{\mathbf{0}}_{\boldsymbol{\nu}}) := \sum_{k \in \mathcal{K}} p^k Q(\tilde{\boldsymbol{x}}^{\mathbf{0}}_{\boldsymbol{\nu}}, k)$. Finally, let $\mathcal{I}_\nu := \{e \in E \mid \tilde{x}_{\nu,e}^0 = 1\}$ be the index set of the edge variables chosen in the first stage, and the constant L be a known lower bound of the recourse function—for the SSNDP a feasible value is $L = 0$. To explicitly cut off the solution $(\tilde{\boldsymbol{x}}^{\mathbf{0}}, \tilde{\Theta})$ we use the general integer optimality cuts of the L-shaped scheme [24]:

$$\Theta \ge (Q(\tilde{\boldsymbol{x}}^{\mathbf{0}}) - L) \left(\sum_{e \in \mathcal{I}} x_e^0 - \sum_{e \in E \setminus \mathcal{I}} x_e^0 - |\mathcal{I}| + 1 \right) + L. \tag{int-LS}$$

These cuts are quite weak since they almost only cut off the current first-stage solution. However, these cuts are necessary for closing the integrality gap, cf. [24] (recall that we are dealing with an NP-hard second-stage problem with binary variables).

A second (even simpler) type of optimality cuts looks like follows:

$$\sum_{e \in \mathcal{I}_\nu} x_e^0 - \sum_{e \in E \setminus \mathcal{I}_\nu} x_e^0 \le |\mathcal{I}_\nu| - 1 \tag{i-LS}$$

These cuts (coined *combinatorial Benders cuts* in [9]) cut off the current first-stage solution $\tilde{\boldsymbol{x}}^{\mathbf{0}}$ but do not contain the explicit bound on the Θ variable. Since the coefficients of these cuts are all binary they are numerically more stable. In our experiments, these cuts turn out being very important for avoiding numerical problems and tailing off effects, cf. Section 6.

4.2. Two-stage branch&cut algorithm

To describe the algorithm we use a slightly more general and compact notation: Let $\boldsymbol{x^0}$ and $\boldsymbol{x^k}$ be variable vectors for the first stage and scenario $k \in \mathcal{K}$, respectively. Moreover, let $\boldsymbol{c^0}$ be the objective coefficient vector in the first stage. With $\mathcal{P}^0_\nu$ being the first-stage polyhedron in iteration ν defined by the separated L-shaped and integer optimality cuts— and no other constraints—let *RMP* denote the relaxed master problem, i.e.,

$$\min\{\boldsymbol{c^0 x^0} + \Theta \;|(\boldsymbol{x^0}, \Theta, \Theta^1, \ldots, \Theta^K) \in \mathcal{P}^0_\nu, \Theta = \sum_{k\in\mathcal{K}} p^k \Theta^k,$$
$$(\Theta^1, \ldots, \Theta^K) \geq \mathbf{0}, \boldsymbol{x^0} \in [0,1]^{|E|}\}. \tag{RMP}$$

Furthermore, let $(R)SP^k$ denote the (relaxed) subproblem, i.e., the restricted deterministic SNDP of scenario $k \in \mathcal{K}$. A brief description of the algorithm is given as follows.

Step 0: Initialization. $UB := +\infty$ (global upper bound, corresponding to a feasible solution), $\nu := 0$. Create the first pendant node. In the initial *RMP* the set of (integer) L-shaped cuts is empty.

Step 1: Selection. Select a pendant node from the branch&bound tree, if such a node exists. Otherwise STOP.

Step 2: Separation. Solve the *RMP* at the current node. $\nu := \nu + 1$. Let $(\boldsymbol{\tilde{x}^0_\nu}, \tilde{\Theta}_\nu, \tilde{\Theta}^1_\nu, \ldots, \tilde{\Theta}^K_\nu)$ be the current optimal solution.
(2.1) If $\boldsymbol{c^0 \tilde{x}^0_\nu} + \tilde{\Theta}_\nu > UB$ fathom the current node and goto Step 1.
(2.2) Search for **violated L-shaped cuts**:
For all $k \in \mathcal{K}$, compute the LP-relaxation value $R(\boldsymbol{\tilde{x}^0_\nu}, k)$ of RSP^k. If $R(\boldsymbol{\tilde{x}^0_\nu}, k) > \tilde{\Theta}^k_\nu$: insert the rounded L-shaped cut (LS) into *RMP*.
If at least one L-shaped cut was inserted goto Step 2.
(2.3) If $\boldsymbol{\tilde{x}^0_\nu}$ is binary, search for **violated integer optimality cuts**:
(2.3.1) For all $k \in \mathcal{K}$ s.t. $\boldsymbol{\tilde{x}^k_\nu}$ was not binary in the previously computed LP-relaxation, solve SP^k to integer optimality: Let $Q(\boldsymbol{\tilde{x}^0_\nu}, k)$ be the optimal solution value.
(2.3.2) $UB := \min\{UB, \boldsymbol{c^0 \tilde{x}^0_\nu} + \sum_{k\in\mathcal{K}} p^k Q(\boldsymbol{\tilde{x}^0_\nu}, k)\}$.
(2.3.3) If $\sum_{k\in\mathcal{K}} p^k Q(\boldsymbol{\tilde{x}^0_\nu}, k) > \tilde{\Theta}_\nu$ insert integer optimality cut (int-LS) (and optionally (i-LS)) into *RMP*. Goto Step 2.
(2.3.4) Fathom current node. Goto Step 1.

Step 3: Branching. Using a branching criterion create two branch&bound (b&b) nodes and append them to the list of pendant nodes. Goto Step 1.

The types of generated cuts are L-shaped optimality cuts (LS), and integer optimality cuts (int-LS) and (i-LS), respectively. Notice that we do not need to add any *feasibility cuts* since we are dealing with a problem with *complete recourse*, i.e., every first-stage solution is feasible and can be augmented—and oriented—to a feasible scenario solution.

5. Deriving stronger L-shaped cuts

Since the relaxed master problem mainly consists of L-shaped optimality cuts, the number of master iterations of the 2-stage b&c approach—and hence, the overall running time—is highly influenced by the strength of the generated L-shaped cuts. In this paper we propose a new and fast way of strengthening the generated L-shaped cuts.

Most of the previously proposed strengthening approaches (cf. [10, 31, 34, 41, 48]) require solving an auxiliary LP in order to generate a stronger L-shaped cut. With our new approach, this is not the case; the procedure is very efficient and we are able to find a stronger L-shaped cut in linear time (with respect to the number of variables). Similar approaches were used in stabilization methods for column generation, cf., e.g., [26, 25].

Instead of solving an additional optimization problem, the L-shaped cuts for the formulation (SD_2) of the SSNDP can be strengthened as follows: Notice that if for an edge $e \in E$ the current first-stage solution satisfies $\tilde{x}_e^0 = 0$, then the corresponding dual variable β_e does not appear in the objective function of the dual (D:SD_2). Furthermore, the variables γ_e do not appear in the objective function neither. Hence, it is not difficult to see that we deal with a highly degenerate LP and one can expect that the optimal solutions to the dual subproblem (D:SD_2) usually produce positive slacks in the constraints (D:2) (typically, if possible, dual variables with zero coefficients in the objective function will be fixed to zero by an LP solver). The idea is now to produce another LP optimal solution of the dual subproblem such that the corresponding slacks are reduced to zero. Therefore, the values of the dual multipliers (β_e) in the associated L-shaped cut will be increased as follows:

Let $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\tau}})$ be an optimal solution to (D:SD_2) as before. For all edges $e = \{i, j\} \in E$ set

$$\hat{\beta}_e := \begin{cases} \tilde{\gamma}_e - \max\limits_{a \in \{(i,j),(j,i)\}} \left\{ \sum\limits_{W \in \mathcal{W}^k : a \in \delta^-(W)} \tilde{\alpha}_W \right\} & \text{if } \tilde{x}_e^0 = 0 \\ \tilde{\beta}_e & \text{otherwise.} \end{cases}$$

If $\hat{\beta}_e > \tilde{\beta}_e$ holds for at least one edge $e \in E$ the strengthened L-shaped cut is given as:

$$\Theta^k + \sum_{e \in E} (c_e^k - \hat{\beta}_e) x_e^0 \geq \sum_{W \in \mathcal{W}^k} \Phi^k(W) \tilde{\alpha}_W - \sum_{e \in E} \tilde{\tau}_e. \qquad \text{(l-LS)}$$

Theorem 6. *The strengthened L-shaped cuts* (l-LS) *are valid and strictly stronger than the standard L-shaped cuts* (LS)*.*

Proof. Consider two L-shaped cuts: the standard one implied by the dual solution $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\tau}})$ and the strengthened one $(\tilde{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\tau}})$ with $\hat{\boldsymbol{\beta}}$ being set as described above. Obviously, $(\tilde{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\tau}})$ is a feasible (and LP-optimal) solution to the dual subproblem (D:SD_2) since $\hat{\boldsymbol{\beta}}$ is set without violating any dual constraints.

Furthermore, notice that $\hat{\beta}_e \geq \tilde{\beta}_e$, for all $e \in E$, and that the right-hand-side of both cuts is identical. Since there exists $e_1 \in E$ such that $\hat{\beta}_{e_1} > \tilde{\beta}_{e_1}$, the coefficient of $x^0_{e_1}$ is strictly smaller for the strengthened L-shaped cut than for the standard one which concludes the proof. $\square$

It is well known that there is a trade-off between the invested running time for finding a Pareto-optimal L-shaped cut and its strength. One can easily construct an example where the strengthened L-shaped cuts are not Pareto-optimal, i.e., they can be dominated by other L-shaped cuts with the same LP-value for $\tilde{\boldsymbol{x}}^{\mathbf{0}}$. Nonetheless, as it is demonstrated in the next section, our procedure is a good heuristic alternative for strengthening L-shaped cuts without sacrificing the overall running time.

6. Computational study

Benchmark instances. To evaluate the performance of the 2-stage b&c in practice we focus on the restricted version of the SSNDP where connectivity requirements in each scenario $k \in \mathcal{K}$ are defined by nodes of type two (subset $\mathcal{R}^k_2 \subseteq V$), type one (subset $\mathcal{R}^k_1 \subseteq V$) and type zero ($V \setminus (\mathcal{R}^k_2 \cup \mathcal{R}^k_1)$). The main motivation for this choice is the application in the design of telecommunication networks where nodes of type two are important infrastructure nodes, or business customers, nodes of type one are single households, and nodes of type zero are, e.g., street intersections. For two distinct nodes u and v and each scenario $k \in \mathcal{K}$, the connectivity requirements are therefore: $r^k_{uv} = 2$ if both u and v are in $\mathcal{R}^k_2$, $r^k_{uv} = 1$ if one of them is in $\mathcal{R}^k_1$ and the other in $\mathcal{R}^k_1 \cup \mathcal{R}^k_2$, and $r^k_{uv} = 0$, otherwise.

Deterministic instances were generated by adopting the idea of Johnson, Minkoff, and Phillips [19], which is frequently used as benchmark in the network design community. After randomly distributing $n \in \{30, 50, 75\}$ points in the unit square, a minimum spanning tree is computed using the points as nodes and the Euclidean distances between all vertex pairs as edge costs. To generate only feasible instances we augmented this tree by inserting edges between leaves which are adjacent in the planar embedding. The resulting biconnected graph is extended by adding all edges for which the Euclidean distance is less than or equal to $1.6\alpha/\sqrt{n}$. We have introduced α in order to control the density (i.e. $|E|/n$) of the graph[1]. In our experiments we use $\alpha = 0.9$ which led to graphs with average density 2.07, e.g., for $n = 30$ and $n = 75$ the density is 1.97 and 2.12, respectively. The edge-connectivity requirements are set as follows. We have randomly drawn ρ% of the nodes as base sets of $\mathcal{R}_1$ and $\mathcal{R}_2$ customers. Here, we use $\rho = 40$ and we additionally introduce a random root node that is contained in $\mathcal{R}_2$. An example is given in Figure 2.

To transform these instances into stochastic ones we randomly and independently generate $\bar{K} = 1000$ scenarios. The probabilities are set by distributing

[1]The original parameter used by [19] was 1.6 and corresponds to $\alpha = 1$ in our setting.

10,000 points over all $\bar{K}$ scenarios. We start by assigning 1 point to each scenario (1 point corresponds to a probability of 0.01%). Then, we distribute the remaining 10,000 - $\bar{K}$ points by selecting one of the $\bar{K}$ scenarios uniformly at random and increasing its number of points by 1. This procedure continues until all 10,000 points are distributed. Hence, at the end, each scenario has a probability ≥ 0.0001 and all probabilities sum up to 1 (since 10,000 points are distributed).

Edge costs $\boldsymbol{c^0}$ in the first stage are Euclidean distances and in the second stage for each edge e and scenario $k \in \mathcal{K}$ randomly drawn from $[1.1c_e^0, 1.3c_e^0]$. Edge-connectivity requirements are generated by randomly drawing ρ^k% from the vertex sets $\mathcal{R}_1$ and $\mathcal{R}_2$ each as $\mathcal{R}_1^k$ and $\mathcal{R}_2^k$ customers, respectively, for scenario k. Here, we use $\rho^k = 30$ for all scenarios k. The special root node was set to be an $\mathcal{R}_2^k$ node in each scenario k.

For each deterministic instance we generated a stochastic instance with $\bar{K} = 1000$ scenarios and took the first K to obtain an SSNDP instance with K scenarios[2]. Probabilities for the scenarios of the instances with $K < 1000$ are scaled appropriately. Overall, we generated 20 graphs for each $n \in \{30, 50, 75\}$ and k leading to 840 instances[3]. Due to the high computational effort we used 580 instances in the experiments: for $n = 30$ all 280 instances are used, for $n = 50$ instances with at most 250 scenarios (180 instances), and for $n = 75$ we used instances with at most 100 scenarios (120 instances).

Computational settings. We implemented the (single-stage) b&c and the 2-stage b&c for the strongest of the three presented models, namely (SD_2), considering the following settings:

- `EF`: single, direct b&c applied to the extended formulation without decomposition. Recall, the model is defined by using binary variables $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and constraints (SD_2:1)-(SD_2:5). To separate connectivity constraints (SD_2:1), arc capacities are defined as z_{ij}^k, for all $(i, j) \in A$, and all $k \in \mathcal{K}$. Then, a flow-based separation procedure known for the deterministic SNDP, cf., e.g., [46], is applied for each scenario $k \in \mathcal{K}$. We have also implemented separation enhancements proposed in [7].

- `2bc`: 2-stage b&c with the separation and strengthening of L-shaped cuts and integer L-shaped cuts. Each subproblem is an instance of a restricted deterministic SNDP, and the separation of associated cut inequalities (P:1) is performed in the same fashion as above, i.e., using the maximum flow-based procedures from [7, 46].

- `2bc-n`: the same as `2bc`, but without strengthening the L-shaped cuts.

We used Abacus 3.0 as a generic branch&cut framework with IBM CPLEX (version 12.1) as LP solver via the interface COIN-Osi 0.102. All experiments

[2] We use 14 values for K: $K \in \{5, 10, 20, 50, 75, 100, 150, 200, 250, 300, 400, 500, 750, 1000\}$.
[3] These instances can be downloaded from our SSNDP webpage, see [45].

were performed on an Intel Xeon 2.5 GHz machine with six cores and 64 GB RAM under Ubuntu 12.04. Each run was performed on a single core and the time limit was set to 2 hours. In all three settings, the maximum flow between the root and the terminals is calculated in a random order of terminals which may result in different running times for the same setting. We therefore perform 5 independent runs and report averaged values over these runs.

For `2bc` and `2bc-n`, integer optimality cuts (i-LS) were included by default. These cuts are numerically more stable and turned out to be necessary in practice, in order to avoid numerical difficulties with some of the instances.

Value of stochastic solution. Since this work presents the first experimental study concerning the SSNDP we start by analyzing the *value of the stochastic solution (VSS)* in order to asses the actual need for formulating the considered deterministic problem as a stochastic problem (for a detailed definition of the VSS and related values we refer the reader to, e.g., [3, 29]).

To calculate the VSS, we first need to find the optimal solution for a deterministic problem in which all random variables are replaced by their expected values (also known as the *expected value problem*, EV). Let $\bar{\boldsymbol{x}}^{\mathbf{0}}$ denote an optimal first-stage solution to this problem. We then evaluate this first-stage solution by considering the EEV (*expected result of the EV solution*): this is the optimal solution value to the original stochastic problem in which the first-stage variables are fixed to $\bar{\boldsymbol{x}}^{\mathbf{0}}$. Finally, the VSS is obtained as VSS = EEV - opt, where opt denotes the optimal SSNDP solution. Hence, the VSS measures the quality of the stochastic solution compared to the solution of the problem using the expected values—which is obviously much easier to compute. The larger the gap between the VSS and the EEV, the more risky/costly it is to replace the uncertain input parameters with their expected values.

For the graphs with 50 vertices Table 1 shows the VSS results, grouped by the number of scenarios K. We report the relative cost increase of the EEV solution compared to opt, the number of edges installed in the first stage (for opt and EEV, respectively). Not surprisingly, the solution costs increase drastically when the EEV is used: on average, EEV solutions are between 20% and 26% costlier than the optimal SSNDP solutions. The gap between opt and EEV also increases with an increasing number of scenarios. Looking at the structure of optimal EEV solutions, we observe that they typically consist of significantly more edges installed in the first stage: on average, between 76% and 90% of the overall solution cost is induced by the first-stage solution, whereas for opt the corresponding values range between 35% and 50%. Finally, for both opt and EEV we notice that with increasing number of scenarios the number of edges installed in the first stage decreases.

Sample stability. Most stochastic programs cannot be solved (directly) because of, e.g., a continuous distribution of the random variables ξ or due to a huge number of possible scenarios K. To create a deterministic equivalent of a reasonable size, one typically samples a set of scenarios which then can be solved

K	value	sol. value increase (%)	# edges 1st stage opt	EV sol.	% 1st stage costs opt	EV sol.
5	avg	20.25	15.95	40.70	50.13	89.08
	min	7.50	6.00	36.00	26.26	73.54
	max	35.71	29.00	44.00	77.61	98.69
	dev	7.29	5.01	2.18	12.09	6.80
10	avg	24.06	14.50	43.65	42.67	87.96
	min	8.75	7.00	38.00	18.12	71.66
	max	39.04	29.00	46.00	72.02	99.04
	dev	7.73	6.09	2.21	15.82	6.68
20	avg	24.96	12.40	43.15	38.04	79.79
	min	15.39	3.00	39.00	9.98	67.37
	max	38.16	28.00	48.00	72.69	91.73
	dev	6.16	6.60	2.37	17.00	7.30
50	avg	26.95	11.63	42.44	35.44	76.60
	min	19.39	2.00	39.00	10.97	67.29
	max	34.88	29.00	47.00	73.50	87.90
	dev	5.16	6.88	2.31	17.70	5.80

Table 1: Results concerning the VSS for instances with 50 vertices and $K \in \{5, 10, 20, 50\}$ scenarios. Used abbreviations: average (avg), minimum (min), maximum (max), standard deviation (dev), solution value (sol), optimum stochastic solution value (opt).

to optimality. It is therefore important to evaluate the underlying scenario generation procedure and to estimate the required number of scenarios needed to achieve good and stable solutions. Two related quality measures are *in-sample stability* and *out-of-sample stability* whose definitions we briefly recall here (cf. [21, 23] for in-depth discussions).

Consider a two-stage stochastic program (in a simplified notation): $\min_{x^0 \in X} f(\boldsymbol{x^0}, \xi)$ where $\boldsymbol{x^0}$ are the first-stage variables, X is the feasible set, f is the objective function, and ξ the random variables vector[4]. Moreover, let $\min_{x^0 \in X} f(\boldsymbol{x^0}, \bar{s})$ denote the stochastic program restricted to a (sampled) scenario set $\bar{s}$. Now, let $\bar{s}, \hat{s}$ denote two scenario sets of the same size and let $\bar{\boldsymbol{x}}^{\boldsymbol{0}}, \hat{\boldsymbol{x}}^{\boldsymbol{0}}$ denote optimal solutions to $\min_{x^0 \in X} f(\boldsymbol{x^0}, \bar{s})$ and $\min_{x^0 \in X} f(\boldsymbol{x^0}, \hat{s})$, respectively. A scenario generation method is called *in-sample stable* if the optimal solution values of two independently sampled scenario sets are similar, i.e., if $f(\bar{\boldsymbol{x}}^{\boldsymbol{0}}, \bar{s}) \approx f(\hat{\boldsymbol{x}}^{\boldsymbol{0}}, \hat{s})$. If $f(\bar{\boldsymbol{x}}^{\boldsymbol{0}}, \xi) \approx f(\hat{\boldsymbol{x}}^{\boldsymbol{0}}, \xi)$ holds, the method is called *out-of-sample stable*.

For this analysis we consider instances with 50 vertices that could be solved to optimality for 1000 scenarios. We then compute the in- and out-of-sample stability values by sampling K out of these 1000 scenarios; we let the sample size K vary between 5 and 500. For each fixed value of K, we create 20 instances by sampling K scenarios out of 1000. Figure 7 shows the results for one representative instance from this set with $|V| = 50$ and $|E| = 100$; the results for the other instances look very similar. Given a fixed value of K, blue crosses and red circles show the distribution of 20 solution values concerning the in-sample and

[4]We use here a simplified notation for a stochastic (mixed-integer) linear program with fixed recourse, i.e., $\min_{x^0 \in X} f(\boldsymbol{x^0}, \xi)$ is $\min_{x^0 \in X} c^T x^0 + Q(x^0, \xi)$, where $Q(x^0, \xi) = \min\{q(\xi)^T y \mid Wy = h(\xi) - T(\xi)x^0,\ y \in Y\}$ with second-stage variables y and feasible set Y, cf. e.g., [3].

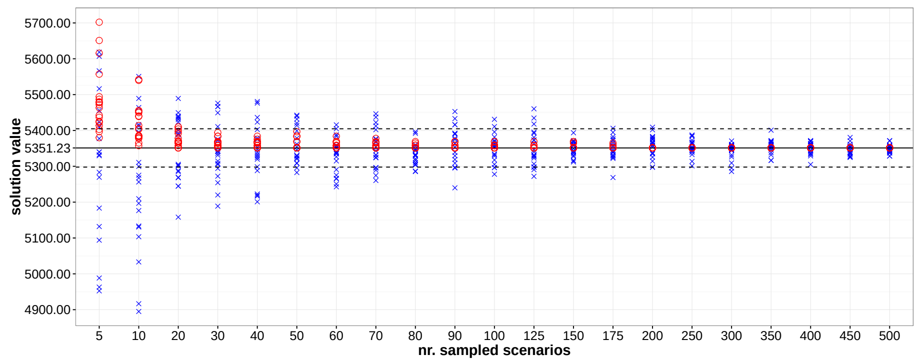


Figure 7: In- and out-of-sample stability for an instance with 50 vertices and 100 edges. The horizontal axis gives the sample size and the vertical axis the objective value of in-sample (blue crosses) and out-of-sample stability (red circles), respectively. Each data point represents one sample. The solid horizontal line is the true optimum solution value opt=5351.23. The dashed horizontal lines indicate the interval $[0.99\,opt, 1.01\,opt]$.

out-of-sample stability, respectively. The solid horizontal line shows the value of the optimal solution i.e., the solution obtained by taking all 1000 scenarios into account. As one can see, already for $K \geq 30$ scenarios, out-of-sample stability can be reached (i.e., optimal first-stage solutions, evaluated on the whole set of 1000 scenarios, fall within a 1% confidence interval). Similarly, in-sample stability can be reached for $K \geq 150$: here, optimal solution values for K-scenario solutions lie within a 1% confidence interval.

Computational benefits of the decomposition approach. Figures 8 and 9 show the comparison of the running times of the three considered settings. The average running times of `EF`, `2bc` and `2bc-n` for the instances with 30 nodes are given in Figure 8. We report that all instances with $n = 30$ could be solved to optimality by the decompositions (`2bc` and `2bc-n`) but there are 115 instances that could not be solved by `EF` within the time limit of 2 hours. Moreover, we observe that for up to 50 scenarios, `EF` is faster, but for instances with 75 or more scenarios, both decomposition approaches clearly outperform `EF`. For example, on the instances with 1000 scenarios, `2bc` is on average about 22 times faster than `EF`. For a lower number of scenarios, `EF` is superior due to the set-up overhead needed for the decomposition. With an increasing number of scenarios, the 2-stage b&c pays off and significantly outperforms the `EF` approach.

A similar behavior can also be observed on the sets of larger instances: The upper and lower part of Figure 9 show the distribution of the running times for instances with 50 and 75 nodes, respectively. Instances are grouped according to the number of scenarios, with 20 instances per group. For each group of 20 instances, the boxplot representation shows the range between the first and third quartiles of the corresponding running times—they are represented as the bottom and top of each box, respectively. Median running times are indicated by a horizontal line and outlier points by small circles.

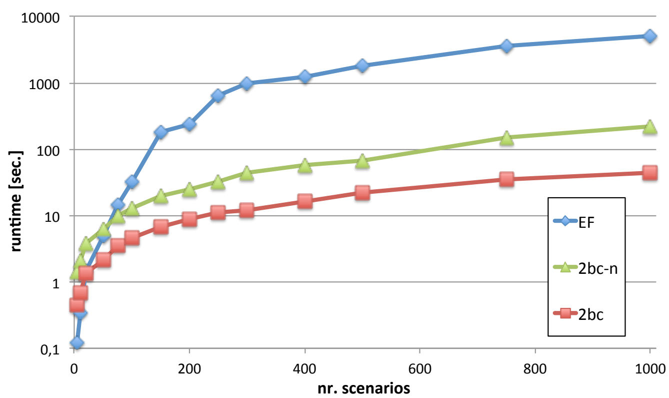


Figure 8: Average running time for graphs with 30 nodes grouped by the number of scenarios. `EF`: direct approach (extended formulation), `2bc-n`: 2-stage branch&cut without cut strengthening, `2bc`: 2-stage branch&cut with cut strengthening

	# b&b nodes			# L-shaped cuts			# Int. opt. cuts		
K	med	avg	dev	med	avg	dev	med	avg	dev
5	3	945.68	3986.69	242.5	7012.41	28426.94	0	260.43	1099.86
10	5	406.50	1678.75	430.0	4553.05	17001.77	0	96.55	403.01
20	10	26.50	75.24	731.5	1112.45	1566.20	1	4.80	14.23
50	11	55.66	161.90	1188.5	8955.32	32562.01	2	12.33	35.76
75	7	59.92	157.72	1621.0	12421.19	42500.32	2	15.48	45.97
100	5	47.08	138.67	2108.0	10504.61	33817.88	2	14.71	48.04
150	6	46.04	121.47	2849.5	12486.56	33197.29	2	13.10	33.98
200	7	45.46	121.14	3700.0	13188.84	30695.46	1.5	11.27	28.38
250	6	50.22	152.83	5015.5	15921.07	33907.50	1.5	13.47	37.38

Table 2: The median, average, and standard deviation for the number of b&b nodes, L-shaped cuts and integer optimality cuts for `2bc` on instances with 50 vertices and 5 to 250 scenarios.

We observe that already for 20 scenarios `2bc` outperforms `EF` and with an increasing input size we need less and less scenarios to draw the immediate advantage of the decomposition. Furthermore, the performance of `2bc` remains relatively stable, whereas high dispersion and skewness of the running times for `EF` can be observed. More precisely, out of 20 instances with 50 nodes and 250 scenarios, 18 of them are solved within the time limit of 2 hours using `2bc` (the average running time of `2bc` is 13 min.), whereas only 5 of them could be solved with the `EF` approach (with average running time of 101 min.). Similarly, for instances with 75 nodes and 100 scenarios, `2bc` solves 15 out of 20 to optimality, and within the same time limit `EF` solves only 3. Outlier points of the decomposition are due to numerical issues; for all of these instances the optimal solution is known early but needs to be verified (many b&b nodes).

Table 2 reports the number of b&b nodes, L-shaped cuts, and integer optimality cuts for `2bc` on instances with 50 vertices. In general, the number of L-shaped cuts increases with the number of scenarios and the number of b&b nodes and integer optimality cuts remains quite low. However, as mentioned

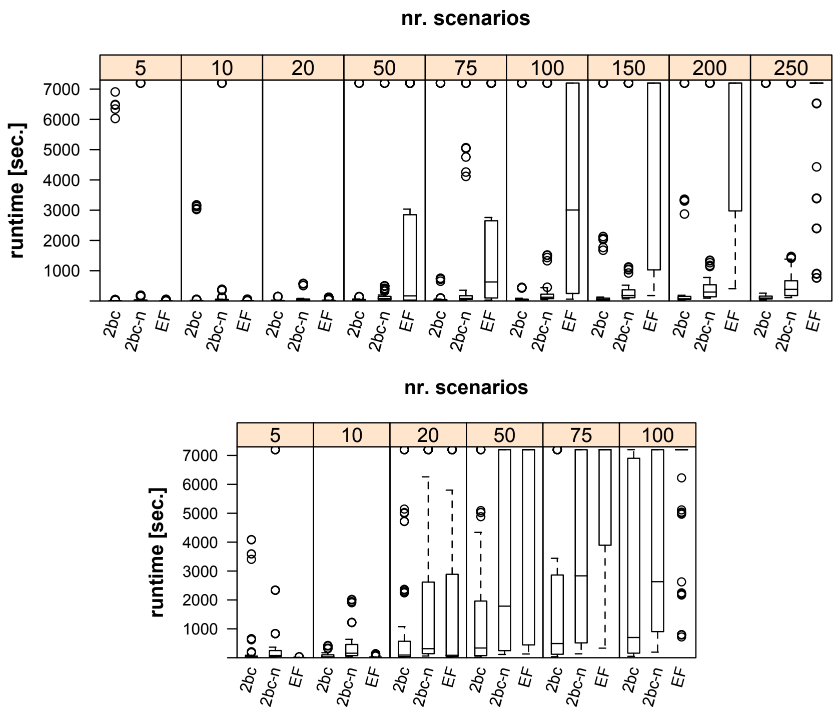


Figure 9: Boxplots showing the running time for graphs with 50 nodes (top) and 75 nodes (below), respectively, grouped by number of scenarios.

before and as one can see from Figure 9, there are outlier points that highly influence the average values and the standard deviation.

Strengthening and comparison to Pareto-optimal approach. Figures 8–9 also highlight the benefits of our strengthening procedure proposed in Section 5. The running times of `2bc` are always smaller and less dispersed when compared to the running times of `2bc-n`. Over all instances, the average speedup obtained through the strengthening of L-shaped cuts is about four. The average speed-up increases with an increasing graph size, and the most significant speedup of 16 is achieved when solving the largest instances, i.e., graphs with 75 nodes.

A well-known and frequently used approach for strengthening L-shaped cuts is the method for finding Pareto-optimal cuts by Magnanti and Wong [31]. In the following, we compare the performance of three decomposition approaches: `2bc` and `2bc-n` described above, and `2bc-MW`, which is 2-stage b&c with Pareto-optimal L-shaped cuts added at each iteration (i.e., after solving the subproblem an additional LP is solved to obtain stronger dual multipliers corresponding to a Pareto-optimal L-shaped cut).

In Figure 10 we compare the three approaches considering the following two performance indicators: 1) the average running times (shown by solid lines),

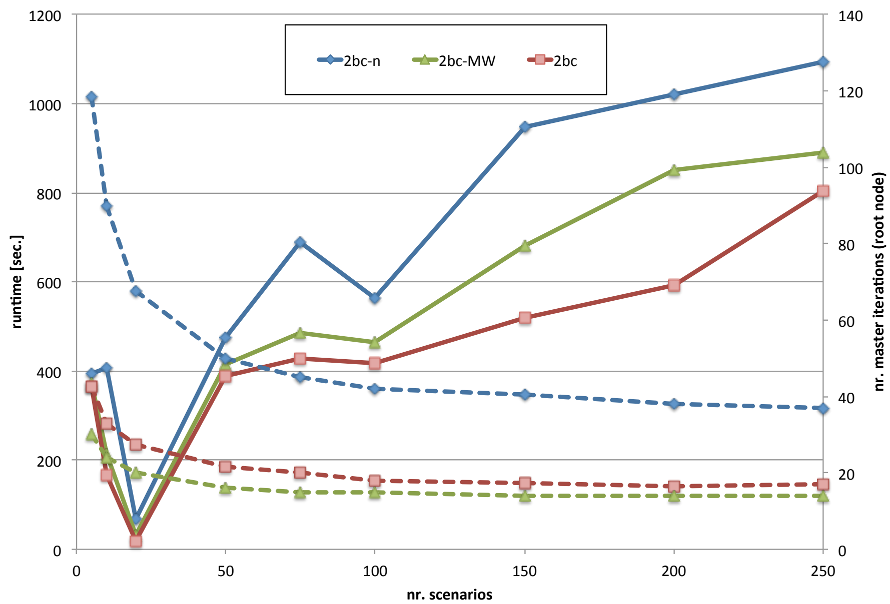


Figure 10: Average running time (solid lines) and median number of master iterations (dashed lines) of the 2-stage b&c with cut strengthening (`2bc`, red lines), without cut strengthening (`2bc-n`, blue lines), and with the Magnanti-Wong method (`2bc-MW`, green lines) for graphs with 50 nodes.

and 2) the average number of master iterations (shown by dashed lines). The figure reports results for graphs with 50 nodes, but a similar behavior can be observed for the remaining instances. Comparing the number of master iterations, we note that the `2bc-n` requires the largest number of iterations, and that the average reduction obtained by our cut strengthening is about 9. The approach `2bc-MW` requires even less master iterations than `2bc`; However, there is a computational overhead associated with `2bc-MW`, induced by solving additional LPs for finding Pareto-optimal cuts. This results in the overall running time of `2bc-MW` which is worse when compared to the running time of `2bc`. We also notice that `2bc-MW` outperforms `2bc-n` in terms of the running time, which underlines the importance of the strengthening procedures in the generation of L-shaped cuts. Finally, we also tried to hybridize `2bc-MW` with `2bc`, but it turned out that this method does not improve the running time of `2bc`.

Graph density. To show the robustness of our decomposition method we evaluate its performance when the graph density $|E|/|V|$ is increased. For this set of experiments we consider the instances with 50 vertices and 50 scenarios and insert new edges to obtain an instance with a higher density. Edge costs of the new edges are generated in the same way as before and edge connectivity requirements remain unchanged. Overall, for each of the 20 instances with $|V| = 50$ and $K = 50$ we generate graphs with density 3, 4, 5, 6, 8, 10, 12,

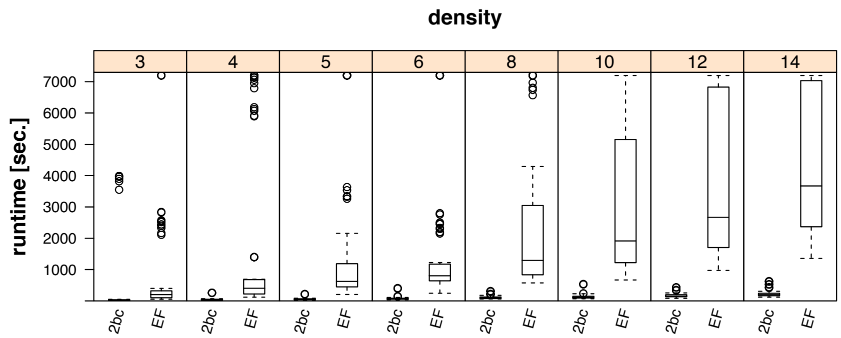


Figure 11: Boxplot showing the running time of `2bc` and `EF` when the density of the graphs increases. Here we use instances with $|V| = 50$ and $K = 50$.

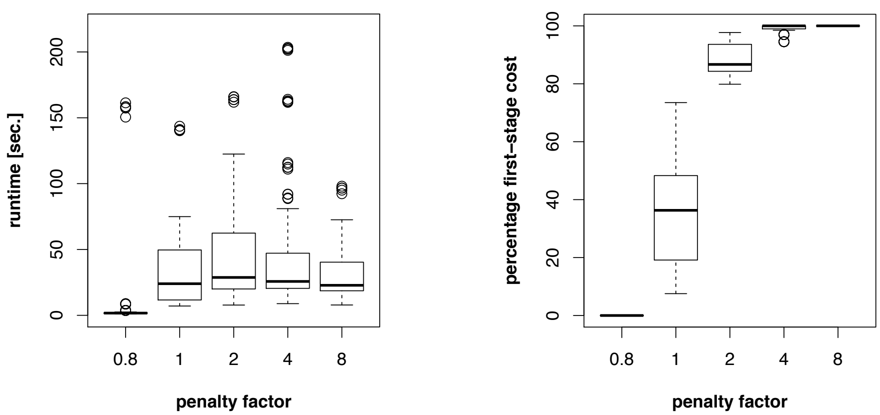


Figure 12: Boxplots showing the impact of a penalty factor for the second-stage costs on instances with $|V| = 50$ and $K = 50$: (*left*) running time of `2bc` and (*right*) percentage of first-stage cost w.r.t. optimal solution value.

and 14, respectively; 14 implies that the graph contains more than half of all possible edges.

Figure 11 shows the running times of `2bc` and `EF` grouped by the graph density. As one can see the running time of `2bc` increases only moderately; the median running time for density 3 is 28.09 sec. and for density 14 it is 197.86 sec. Since there are also only a few outlier points we conclude that the performance of `2bc` is robust and remains stable even when the graph density increases. On the contrary, the direct approach `EF` performs much worse on denser graphs: the median running time increases from 200.86 sec. to 3670.13 sec. (density 3 and 14, respectively). Moreover, `EF` could not solve 25% of the densest instances within the time limit of two hours.

Second stage as penalty. Last but not least, we consider the impact of the relative second-stage costs w.r.t. the first-stage costs. Here, we take all instances with 50 vertices and simply multiply all second-stage costs with a penalty factor from the set $\{0.8, 2, 4, 8, 12\}$. Figure 12 presents the results for the instances with 50 scenarios—the results for the other instances look similar. Clearly, a more expensive second stage leads to the installation of more edges in the first stage as shown by Figure 12 (right). Moreover, our initial setting of second-stage costs seems to be reasonable as the optimal solutions consist of both first- and second-stage edges. With an increasing (or decreasing) factor this changes drastically and one stage dominates: for a factor of 0.8 or less the optimal first stage solution is always empty and for a factor of ≥ 8 this holds for the second stage. Moreover, we conclude that our algorithm performs well even when more edges are bought in the first stage, cf. Figure 12 (left). This is important since the decomposed model initially contains no constraints in the master problem.[5]

7. Discussion, conclusions, and future work

In this paper we introduced the two-stage stochastic version of the survivable network design problem with a finite number of discrete scenarios and complete recourse. We presented three ways to model this problem: an undirected and two semi-directed formulations. For the undirected model we showed that facet defining inequalities of the deterministic counterparts can be lifted and yield facet defining inequalities of the stochastic model. This is the first result concerning the polyhedra of stochastic network design problems. The semi-directed formulations rely on orientation properties of edge-κ-connected graphs, and we prove that they are strictly stronger than the undirected formulation. We also discussed a 2-stage branch&cut algorithm using a decomposition approach. Moreover, a strengthening procedure is introduced, representing an easy way for strengthening the L-shaped optimality cuts. Our computational study showed the benefits of using the decomposition for stochastic network design problems. Furthermore, the cut strengthening technique highly reduces the number of master iterations and the computational running time. Compared to the generation of Pareto-optimal L-shaped cuts with the Magnanti-Wong method, our new procedure requires slightly more master iterations, but outperforms the Magnanti-Wong method with respect to running time (due to the overhead of solving additional LPs for the latter approach).

Our computational experiments show that our two-stage branch&cut approach is more than an order of magnitude faster than the direct single-stage b&c approach. For the SSNDP, we are able to solve instances with up to 75 nodes, or 1000 scenarios, respectively, to provable optimality. We are convinced that our simple and fast cut strengthening technique will be useful for general network optimization problems in the two-stage (stochastic optimization) setting.

[5] More detailed results regarding our computational experiments (including optimal solution values, running times, etc.) can be found at [45].

An open problem remains how to generate Pareto-optimal L-shaped cuts by a combinatorial algorithm. Our cut strengthening does not ensure the Pareto optimality—counter examples can be derived easily.

We remark that similar results can be obtained for the stochastic version of the node-connectivity $\{0, 1, 2\}$-SNDP, cf. Section 2. Only recently, in [7], new graph orientation properties have been given that allow to derive stronger directed MIP formulations for the deterministic $\{0, 1, 2\}$-SNDP with node-connectivity requirements. Using this result and analogously to formulation (SD_2), it is possible to formulate a stronger semi-directed model for the stochastic counterpart. L-shaped cuts and their strengthening can be derived from this model in a similar way, to be used within the 2-stage b&c.

Last but not least, we like to mention that several interesting problems are still open for the deterministic SNDP. For example, stronger directed formulations for higher node-connectivity requirements, i.e., > 2, or orienting solutions where both node- and edge-connectivity is required are still open problems. However, by using the same techniques presented in this article, any improvements in the deterministic context should be transferable to the related stochastic problems, formulations, and algorithms.

Acknowledgements

We thank the four anonymous referees for their useful comments and suggestions that helped to improve the article. Part of this research has been done while visiting TU Dortmund, when I. Ljubić has been supported by the APART Fellowship of the Austrian Academy of Sciences (OEAW). This support is greatly acknowledged. The Dortmund part of this work has been partially supported by the DFG Research Training Group 1855.

References

[1] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. Connectivity-splitting models for survivable network design. *Networks*, 43(1):10–27, 2004.

[2] J. F. Benders. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4:238–252, 1962.

[3] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, second edition, 2011.

[4] I. M. Bomze, M. Chimani, M. Jünger, I. Ljubić, P. Mutzel, and B. Zey. Solving two-stage stochastic Steiner tree problems by two-stage branch-and-cut. In *ISAAC (1)*, LNCS, pages 427–439. Springer Berlin, 2010.

[5] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, 25(1):13–26, 2013.

[6] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(3):451–464, 1998.

[7] M. Chimani, M. Kandyba, I. Ljubić, and P. Mutzel. Orientation-based models for {0, 1, 2}-survivable network design: theory and practice. *Mathematical Programming*, 124(1–2):413–439, 2010.

[8] S. Chopra. The k-edge-connected spanning subgraph polyhedron. *SIAM Journal on Discrete Mathematics*, 7(2):245–259, 1994.

[9] G. Codato and M. Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

[10] M. Fischetti, D. Salvagnin, and A. Zanette. A note on the selection of Benders' cuts. *Mathematical Programming*, 124:175–182, 2010.

[11] B. Fortz, A. R. Mahjoub, S. T. McCormick, and P. Pesneau. Two-edge connected subgraphs with bounded rings: Polyhedral results and branch-and-cut. *Mathematical Programming*, 105(1):85–111, 2006.

[12] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 4(2):309–330, 1992.

[13] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Operations Research*, 43(6):1012–1024, 1995.

[14] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 617–672. Elsevier, 1995.

[15] A. Gupta, R. Ravi, and A. Sinha. LP rounding approximation algorithms for stochastic network design. *Mathematics of Operations Research*, 32(2): 345–364, 2007.

[16] A. Gupta, R. Krishnaswamy, and R. Ravi. Online and stochastic survivable network design. In *ACM-SIAM STOC*, pages 685–694. ACM, 2009.

[17] P. Hokama, M. C. San Felice, E. C. Bracht, and F. L. Usberti. A heuristic approach for the stochastic Steiner tree problem. In *11th DIMACS Implementation Challenge*, 2014.

[18] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[19] D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting Steiner tree problem: Theory and practice. In *ACM-SIAM SODA*, pages 760–769. SIAM, 2000.

[20] M. Kandyba. *Exact Algorithms for Network Design Problems using Graph Orientations*. PhD thesis, Technische Universität Dortmund, 2011.

[21] M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271,2007.

[22] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[23] A. J. King and S. W. Wallace. *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering, 2012.

[24] G. Laporte and F. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13: 133–142, 1993.

[25] M. Leitner. *Solving Two Network Design Problems by Mixed Integer Programming and Hybrid Optimization Methods*. PhD thesis, Technische Universität Wien, 2010.

[26] M. Leitner, M. Ruthmair, and G. R. Raidl. Stabilizing branch-and-price for constrained tree problems. *Networks*, 61(2):150–170, 2013.

[27] I. Ljubić, P. Mutzel, and B. Zey. Stochastic survivable network design problems. *INOC*, Electronic Notes in Discrete Mathematics, 41(0):245–252, 2013.

[28] I. Ljubić, P. Putz, and J. J. S. González. Exact approaches to the single-source network loading problem. *Networks*, 59(1):89–106, 2012.

[29] F. Maggioni and S. W. Wallace. Analyzing the quality of the expected value solution in stochastic programming. *Annals of Operations Research*, 200(1):37–54, 2010.

[30] T. L. Magnanti and S. Raghavan. Strong formulations for network design problems with connectivity requirements. *Networks*, 45(2):61–79, 2005.

[31] T. L. Magnanti and R. T. Wong. Accelerating Benders' decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[32] C. S. J. A. Nash-Williams. On orientations, connectivity, and odd vertex pairings in finite graphs. *Canadian Journal of Mathematics*, 12:555–567, 1960.

[33] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1988.

[34] N. Papadakos. Practical enhancements to the Magnanti-Wong method. *Operations Research Letters*, 36(4):444–449, 2008.

[35] A. Prékopa. Probabilistic Programming. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, 267–351, Elsevier, 2003.

[36] J. Rak and J. Sterbenz. Preface: Optimization issues in resilient network design and modeling. *Networks*, 66(4): 251–252, 2015.

[37] R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. *Mathematical Programming*, 108(1): 97–114, 2006.

[38] R. Rodríguez-Martín and J.-J. Salazar-González and H. Yaman. A branch-and-cut algorithm for two-level survivable network design problems. *Computers & Operations Research*, 67:102–112, 2016.

[39] R. Schultz. Stochastic programming with integer variables. *Mathematical Programming*, 97(1-2):285–309, 2003.

[40] S. Sen and H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[41] H. Sherali and B. Lunday. On generating maximal nondominated Benders cuts. *Annals of Operations Research*, pages 1–16, 2011.

[42] D. B. Shmoys and C. Swamy. Algorithms column: Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1): 33–46, 2006.

[43] Y. Song and J. R. Luedtke. Branch-and-cut approaches for chance-constrained formulations of reliable network design problems. *Mathematical Programming Computation*, 5(4):397–423, 2013.

[44] Y. Song and M. Zhang. Chance-constrained multi-terminal network design problems. *Naval Research Logistics*, 62(4):321–334, 2015.

[45] SSNDPLib. SSNDPLib: `http://ls11-www.cs.tu-dortmund.de/staff/zey/ssndp`, 2014.

[46] M. Stoer. Design of survivable networks, volume 1531 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1992.

[47] R. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, 17(4):638–663, 1967.

[48] P. Wentges. Accelerating Benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, 44: 267–290, 1996.