Design of survivable networks with vulnerability constraints

Luis Gouveia[1] and Markus Leitner[2]

[1]Universidade de Lisboa, Faculdade de Ciências, Departamento de Estatística e Investigação Operacional, Lisbon, Portugal.
legouveia@fc.ul.pt

[2]Department of Statistics and Operations Research, Faculty of Business, Economics and Statistics, University of Vienna, Vienna, Austria.
markus.leitner@univie.ac.at

September 4, 2016

Abstract

We consider the Network Design Problem with Vulnerability Constraints (NDPVC) which simultaneously addresses resilience against failures (network survivability) and bounds on the lengths of each communication path (hop constraints). Solutions to the NDPVC are subgraphs containing a path of length at most H_{st} for each commodity $\{s,t\}$ and a path of length at most H'_{st} between s and t after at most $k-1$ edge failures. We first show that a related and well known problem from the literature, the Hop-Constrained Survivable Network Design Problem (kHSNDP), that addresses the same two measures produces solutions that are too conservative in the sense that they might be too expensive in practice or may even fail to provide feasible solutions. We also explain that the reason for this difference is that Mengerian-like theorems not hold in general when considering hop-constraints. Three graph theoretical characterizations of feasible solutions to the NDPVC are derived and used to propose integer linear programming formulations. In a computational study we compare these alternatives with respect to the lower bounds obtained from the corresponding linear programming relaxations and their capability of solving instances to proven optimality. In addition, we show that in many cases, the solutions produced by solving the NDPVC are cheaper than those obtained by the related kHSNDP.

Keywords: Networks, Integer programming, Survivable Network Design, Hop-constraints, OR in telecommunications

1 Introduction

For Internet service providers, it is essential to provide stable and reliable communications between any two points in their supporting telecommunication networks. However, it is not easy to provide a precise definition of reliability. A vast body of literature, both in the engineering and operations research community, suggests various concepts for network reliability. This is because the reliability of a network depends on several factors. On the one side, it depends on the technical equipment installed along the links and nodes of the network. On the other side, even with the best available equipment, reliability may be easily destroyed, if the underlying network *topology* is vulnerable to failures. Therefore, resistance to network failures (also known as network *survivability*) has been used in the network optimization literature as one of the main criteria for designing reliable communication networks (see, e.g., [9]). A network is said to be survivable, if communications between nodes can be established, even after failures of a pre-defined number of nodes or links. Starting with the seminal work by Grötschel et al. [8], a large body of mathematical models and algorithmic approaches for designing survivable networks has been proposed.

Another important issue for Internet service providers is quality of service, see e.g., Klincewicz [10]. Each packet of a data flow traveling through a path from its source node to its destination node suffers a total delay that is given by the propagation delay on each link and the queuing and transmission

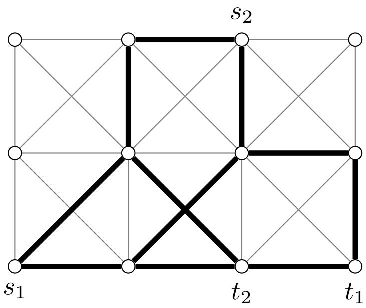


Figure 1: Example of an instance of the NDPVC with $\mathcal{R} = \{\{s_1, t_1\}, \{s_2, t_2\}\}$, $H_{st} = 3$, $H'_{st} = 4$, $\forall \{s, t\} \in \mathcal{R}$, and a feasible solution to this instance (bold edges).

delays on each intermediate node. Jitter, defined as the time difference between the maximum delay and the minimum delay among all packets of a data flow, is an important quality of service parameter that should be bounded to guarantee a given quality of service [16]. This parameter is of particular importance for multimedia services (see, Roychoudhuri et al. [14]) but also for data service running over mobile networks (see, Scharf et al. [15]). Note that the dominant factor on jitter is the queuing delay since propagation introduces a constant delay on each packet and transmission delay is only dependent on packet size statistics. A simple way of bounding jitter is to bound the number of packet queues, which is equivalent to bound the number of hops of each routing path. Hence, the quality of service can be ensured by imposing so-called hop-constraints.

Recent literature suggests to combine survivability and quality of service by additionally imposing hop-constraints when designing survivable networks. Thus, one guarantees that for every distinct pair of nodes, there exists a pre-defined number of edge/node disjoint paths, such that each such path does not exceed the given hop limit [3]. In this article we show that solutions to this problem variant are too conservative and too expensive, from the perspective of a network provider. We therefore propose to study a new (and related) problem, that ensures both survivability, and the maintenance of the hop-limits after a failure of a pre-defined number of nodes or links, but with significantly less restrictions on the underlying network topology. We call the problem the *Network Design Problem with Vulnerability Constraints* (NDPVC). The term "vulnerability" is coined from graph theory (see, e.g., Bermond et al. [2]) and is usually associated to the study of changes in the distance between pairs of nodes due to graph alterations (e.g., edge or node removals). We will have more to say about this in Section 2. As in many related problems, one may consider protection against edge or node failures. We focus on the case of edge-connectivity for which the formal problem definition is given below. The node-connectivity case will be briefly addressed in Section 5.

Problem definition and motivation We are given an undirected graph $G = (V, E)$, with nonnegative edge costs $c_e \geq 0$, for all $e \in E$, and a parameter $k \in \mathbb{N}$ specifying the network survivability. In addition, we are given a set of commodities $\mathcal{R} \subseteq V \times V \setminus \{(v, v) \mid v \in V\}$ and two hop limits, $H_{st} \leq H'_{st} \in \mathbb{N}$, for each pair $\{s, t\} \in \mathcal{R}$. The goal is to find a minimum cost subgraph of G, such that for each pair $\{s, t\} \in \mathcal{R}$, it contains a path of length at most H_{st}, and after removal of any $k - 1$ edges from it, the resulting graph contains a path of length at most H'_{st}.

Figure 1 illustrates an input graph G with commodities $\mathcal{R} = \{\{s_1, t_1\}, \{s_2, t_2\}\}$, $k = 2$, and hop limits $H_{st} = 3$, $H'_{st} = 4$ for all $\{s, t\} \in \mathcal{R}$, together with a feasible solution. Notice, that the well known, NP-hard, survivable network design problem (see, e.g., [8]) is a special case of the NDPVC when the hop-limits are redundant. Thus the NDPVC is NP-hard as well.

A related problem that has already been studied in the literature is the hop-constrained survivable network design problem (kHSNDP). In this problem, one searches for a minimum-cost subgraph, such that between each pair of commodities there exist k edge- (node-) disjoint paths, each containing at most H edges. Various integer programming formulations and solution algorithms have been proposed recently for the kHSNDP, see, e.g., [3, 6, 12]. At first one may assume that the NDPVC and the kHSNDP are equivalent, at least when $H = H'$. The reason for this is that if one would ignore the hop constraints,

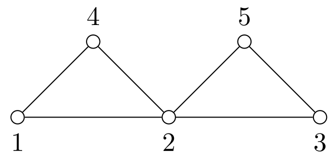


Figure 2: A feasible solution for the edge disjoint variant of NDPVC with $k = 2$, $\mathcal{R} = \{\{1, 3\}\}$, $H_{13} = 2$, $H'_{13} = 3$, that does not contain two edge disjoint paths of length 2 and 3 between nodes 1 and 3, cf. Exoo [4].

the two problems, kHSNDP and NDPVC, become equivalent. This equivalence follows immediately from Menger's theorem [13]. As a matter of fact, quite often in the literature, mathematical formulations for modeling survivable networks are derived using the results of the Menger's theorem (see, e.g, Grötschel et al. [8], Kerivin and Mahjoub [9], Ljubić [11]).

Unfortunately, once the hop-constraints are imposed, the two problems are no longer equivalent, since hop-constrained *Mengerian-like theorems* (see Section 2 for a more formal definition and discussion) are valid only for small or large hop-limits. To see that the two problems, NDPVC and kHSNDP, are different in general case, consider the example given in Figure 2. Assume that a network provider wants to make this network survivable against single edge failures (i.e., $k = 2$), and to protect the vulnerability of the network by assuming that for a commodity pair $\mathcal{R} = \{\{1, 3\}\}$, the distance between 1 and 3 should be at most $H_{13} = 2$ and, after a single edge failure, this distance should not be greater than $H'_{13} = 3$. The path $P = (1, 2, 3)$ is the unique $(1, 3)$-path of length less than or equal to H_{13}. If an arbitrary edge in this graph fails, the solution will still contain a feasible $(1, 3)$-path of length at most three. Hence, the graph depicted in Figure 2 is a feasible NDPVC solution. On the other hand, if the network provider would try solve the related kHSNDP on this graph, instead, then one can easily observe that no feasible solution exist. A feasible kHSNDP solution needs to contain two *edge-disjoint* paths between 1 and 3, such that one path contains at most H_{13} edges and the other at most H'_{13} edges. This is, however impossible, since the only $(1, 3)$-path P' that is edge disjoint to P, is given by $P' = (1, 4, 2, 5, 3)$ and has length four. This example illustrates that for network providers it could be more attractive to consider the NDPVC to protect vulnerability of a network, rather than the kHSNDP.

This example can be easily generalized for $k \geq 3$ and for other values of H_{st} and H'_{st}, $\{s, t\} \in \mathcal{R}$. The optimal value of kHSNDP always gives an upper bound on the optimal value of the NDPVC. It is easy to find examples where solutions to both problems exist but the optimal solution to the kHSNDP is more expensive than of the NDPVC. Observe that not only the gap between the cost of an optimal solution of NDPVC and kHSDNP can be arbitrarily large, but, as demonstrated above, there exist networks which are feasible for NDPVC but infeasible for kHSNDP. Since these relations motivate this work we summarize them in Observation 1.

Observation 1. *Let I be an arbitrary, feasible instance of the NDPVC and $v(I)$ be its optimal cost. Then, exactly one of the following holds:*

(i) There does not exist a feasible solution of the kHSNDP for I.

(ii) $v(I) \leq v'(I)$, where $v'(I)$ is the optimal cost of the kHSNDP for I.

Furthermore, there exist instances such that $v(I) < v'(I)$ and $\frac{v'(I)}{v(I)}$ can be arbitrary large.

Note also that, cases in which $\mathcal{R}$ is induced by all node pairs $\{s, t\}$ from a given set of "terminals" T, i.e., $\mathcal{R} = \{\{s, t\} \mid s, t \in T\}$, give rise to several interesting diameter variants of the problem. As one example, we mention the case when $H_{st} = D$ and $H'_{st} = D'$ for all $\{s, t\} \in \mathcal{R}$, in which case we aim to identify a minimum cost Steiner subgraph with diameter at most D such that after the removal of k edges (nodes), the graph is connected and has diameter at most D'.

The remainder of this article in which we mainly focus on the case of single edge failures (i.e. $k = 2$) is organized as follows. We point out some graph-theoretical properties concerning the validity of Mengerian-like theorems involving distance constraints in Section 2. In Section 3 we present the integer linear programming (ILP) formulations for the NDPVC that are based on three graph-theoretical characterizations of solutions. The results of our computational study are given in Section 4 where we compare the linear programming (LP) relaxation bounds of all proposed formulations, examine their efficiency in computing optimal integer solutions and make a comparative study of the optimal solutions of the NDPVC and the kHSNDP. Finally, we briefly address the node failure case in Section 5 before we draw final conclusions in Section 6 where we also comment on the case of multiple edge- or node-failures and point out interesting directions for future work.

2 Mengerian-like results

The problems dealing with network vulnerability and preservation of diameters (or hop-constraints between a given subset of node pairs) have been very popular in the graph theory since the 60's. Most of these articles, however, study special graph properties related with the concept of edge (or node) *persistence* of a graph, which is the minimum number of edges (nodes) whose removal increases the diameter. Apparently, very few edge cost minimization problems addressing vulnerability in the way considered in this paper, do exist in the literature. One such example is the problem of finding graphs G with a given number of nodes and with a minimum number of edges, so that the diameter of G is D, and such that after a removal of any $k-1$ edges (nodes) from G, the remaining graph has diameter at most D', see [2]. This problem might be viewed as a cost minimization problem with uniform edge costs equal to one.

In this section, we describe in more detail Mengerian results with hop constraints. One of the two results is well known from the literature, see for instance [2] and [4], and the other is a variant where the hop limit is not the same for all paths. Although Result 2 is new, we only provide a proof sketch below since a complete proof would be similar to the one of Result 1.

Result 1 ("Mengerian-like result for edge-disjoint hop-constrained paths", Exoo [4], Bermond et al. [2]). *Let i and j be two distinct nodes of a given graph $G = (V, E)$, such that the length of the shortest (i, j)-path is H, $H \leq 3$ or $H \geq |V| - 1$. Then, the minimal number of edges that need to be removed from G in order to increase the length of the shortest (i, j)-path, is equal to the maximum number of pairwise edge-disjoint paths from i to j of length at most H.*

Result 2 ("Mengerian-like result for edge-disjoint hop-constrained paths with $H < H'$"). *Let i and j be two distinct nodes of a given graph $G = (V, E)$ that contains at least one path from i to j of length at most H. Then, minimal number of edges k after whose removal G does not contain a path from i to j of length at most H', $H' > H$, is equal to the maximum number of pairwise edge-disjoint paths from i to j of length at most H' (where one of them is of length at most H) if $H \geq |V| - 1$ or $H = 1$ and either $k = 2$ or $H' \leq 3$ or $H' \geq |V| - 1$.*

Observe that when $H \geq |V| - 1$ in Result 1 we obtain the unconstrained case mentioned above. The NDPVC and the kHSNDP are equivalent if we consider H and H' satisfying the conditions of any of the two results. Also, in such cases we can use the methods already devised for the equivalent kHSNDP and do not need to look for alternative models.

It is, however, not difficult to see that Result 1 does not hold for the other cases, see Exoo [4] for counterexamples. Notice, that Result 2 obviously holds for $H = 1$ and $k = 2$ since every backup path of some commodity $\{s, t\}$ must be edge-disjoint from the only feasible primary path defined by the edge $\{s, t\}$. The example illustrated in Figure 2 shows that Result 2 does not hold for $H \geq 2$. The result follows from observing that the case $k \geq 3$ and $H = 1$ can be reduced to the case of Result 1 by decreasing k by one, removing the edge $\{s, t\}$ and setting $H = H'$.

3 ILP Formulations

In this section, we will first introduce a generic formulation for the NDPVC, see Section 3.1. For any $\{s, t\} \in \mathcal{R}$, the required path of length at most H_{st} will be called *primary path* while the edges necessary to ensure the existence of a path of length at most H'_{st} after at most $k-1$ edge failures will be called *backup edges*. We will consider two building blocks, one corresponding to the design of the primary path, and the other to the set of the backup edges. While models for designing the primary path can be straightforwardly taken from the literature (see, e.g., [5, 6] and [17] for a more recent article), this is not true for models used to determine the optimal set of backup edges. In Section 3.2, we will provide three characterizations of valid sets of backup edges.

Each characterization will be used to derive hop-indexed multicommodity flow based formulations in Section 3.3. For the latter, we will assume $k = 2$, i.e., we ensure survivability against single edge failures.

Notation In the following, we will use $T = \{s \in V \mid \exists\{s,t\} \in \mathcal{R}, s < t\}$ to denote the set of commodity origins (sources) and $T(s) = \{t \in V \mid \{s,t\} \in \mathcal{R},\ s < t\}$ to denote the set of commodity destinations (targets) of origin $s \in T$. Let $A = \{(i,j) \mid \{i,j\} \in E\}$ denote the arc set obtained from bi-directing edge set E. For a subset $W \subset V'$ of nodes of some graph $G' = (V', E')$, we will use $\delta(W) = \{\{i,j\} \in E' \mid i \in W, j \notin W\}$ to denote the cutset of W with respect to G'. Let, furthermore $d_{ij} \in \mathbb{N}$ denote the minimum distance between nodes i and j in G (measured in the number of hops). Then, for each commodity pair $\{s,t\} \in \mathcal{R}$, $A_{st} = \{(i,j) \in A \mid d_{si} + d_{jt} + 1 \leq H_{st}\}$ and $A'_{st} = \{(i,j) \in A \mid d_{si} + d_{jt} + 1 \leq H'_{st}\}$ are the sets of arcs feasible for establishing a primary or secondary connection of commodity $\{s,t\}$, respectively. Similarly, $E_{st} = \{\{i,j\} \in E \mid (i,j) \in A_{st} \vee (j,i) \in A_{st}\}$ and $E'_{st} = \{\{i,j\} \in E \mid (i,j) \in A'_{st} \vee (j,i) \in A'_{st}\}$ are the sets of eligible primary and secondary edges for commodity $\{s,t\}$ while $V_{st} = \{i \in V \mid \exists\{i,j\} \in E_{st}\}$ and $V'_{st} = \{i \in V \mid \exists\{i,j\} \in E'_{st}\}$ are the eligible nodes within a primary or secondary connection, respectively. Finally, for $e = \{i,j\} \in E$ and a set of arcs A' let $A'[e] = A' \setminus \{(i,j),(j,i)\}$.

3.1 Generic Formulation.

Let $x_e \in \{0,1\}$, for all $e \in E$, be decision variables indicating whether or not an edge $e \in E$ is used in a solution, and let $E(\mathbf{x})$ denote the set of edges such that $x_e = 1$. Let furthermore,

$$\mathcal{F}_{st} = \{\mathbf{x} \in \{0,1\}^{|E|} \mid \exists\ (s,t)\text{-path } P \text{ in } E(\mathbf{x}) \text{ s.t. } |P| \leq H_{st}\}$$

be the set of feasible incidence vectors $\mathbf{x}$ containing a path of length at most H_{st} for each commodity pair $\{s,t\} \in \mathcal{R}$. Similarly, for the set of backup edges, we will denote by

$$\mathcal{B}_{st} = \{\mathbf{x} \in \{0,1\}^{|E|} \mid \forall F \subset E,\ |F| = k-1,\ \exists\ (s,t)\text{-path } P \text{ in } E(\mathbf{x}) \setminus F \text{ s.t. } |P| \leq H'_{st}\}$$

the set of feasible incidence vectors of $\mathbf{x}$ ensuring the required redundancy, i.e., the existence of a path of length at most H'_{st} for each commodity pair $\{s,t\} \in \mathcal{R}$ after removing $k-1$ edges. To be more precise $\mathcal{B}_{st}$ depends on the set of chosen primary edges. To keep notation simple, we will, however, maintain the notation used above.

A generic ILP formulation for the NDPVC is given by (1)–(4).

$$\min \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{F}_{st} \quad \{s,t\} \in \mathcal{R}, \tag{2}$$

$$\mathbf{x} \in \mathcal{B}_{st} \quad \{s,t\} \in \mathcal{R}, \tag{3}$$

$$\mathbf{x} \in \{0,1\}^{|E|}. \tag{4}$$

As pointed out before, modeling $\mathcal{F}_{st}$ is well known from the literature. In the next section we provide three characterizations for the set of backup edges that will permit us to provide models for $\mathcal{B}_{st}$.

3.2 Graph theoretical characterization of $\mathcal{B}_{st}$

In this subsection we provide three characterizations of the set $\mathcal{B}_{st}$. The first characterization is straightforward, while the remaining two are based on less obvious and, as far as we know, new observations that motivate more efficient models. In fact, we can even say that these new characterizations also motivate the current study.

Characterization 1 (CH1)**.** *Let $P \subset E_{st}$ be the primary (s,t)-path of length $\leq H_{st}$ for a given commodity $\{s,t\} \in \mathcal{R}$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that:*

$$\forall e \in P, \quad \exists\ (s,t)\text{-path } P'[e] \subset E'_{st} \setminus \{e\},\ s.t.\ |P'[e]| \leq H'_{st}.$$

Thereby, $H_{st} \leq H'_{st}$ and $\hat{E} = \left(\bigcup_{e \in P} P'[e]\right) \setminus P$.

For the remaining two observations, we observe that for each primary path of length $l \leq H_{st}$ associated with some commodity $\{s,t\} \in \mathcal{R}$, it suffices to establish only l additional secondary (s,t)-paths, each of length at most H'_{st}, provided that we additionally ensure that no edge of the primary path is contained in all of them. Thus, each potentially failing edge (from the primary path) is covered by at least one backup path.

Characterization 2 (CH2)**.** *Let $P \subset E_{st}$ be the primary (s,t)-path of length l, $l \leq H_{st}$, for a given commodity $\{s,t\} \in \mathcal{R}$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that there exist l additional (s,t)-paths $P'_i \subset E'_{st}$, $i = 1,2,\ldots,l$, of length at most H'_{st}, $H'_{st} \geq H_{st}$, such that at most $l-1$ of them contain the same edge from P, i.e.:*

$$\exists\ (s,t)\text{-paths } P'_i \subset E'_{st},\ i = 1,2,\ldots,l,\ s.t.\ |P'_i| \leq H'_{st}$$
$$\text{and } \forall e \in P : \sum_{i=1}^{l} |P'_i \cap \{e\}| \leq l-1.$$

Thereby, $\hat{E} = \left(\bigcup_{i=1}^{l} P'_i\right) \setminus P$.

The last characterization is more informative than the previous one. It will permit us to define models that are disaggregations of the ones based on Characterization 2. To describe the necessary backup edges for a primary path of commodity $\{s,t\} \in \mathcal{R}$ of length $l \leq H_{st}$, we ensure the existence of l additional (s,t)-path of length at most H'_{st} such that the lth such path does not use the lth edge of the primary path.

Characterization 3 (CH3)**.** *Let $P = \{e_1, e_2, \ldots, e_l\} \subset E_{st}$ be the primary (s,t)-path of length l, $l \leq H_{st}$ for a given commodity $\{s,t\} \in \mathcal{R}$, such that $e_i = \{u_{i-1}, u_i\}$, $u_i \in V_{st}$, $i = 1,2,\ldots,l$, $u_0 = s$, and $u_l = t$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that:*

$$\exists\ (s,t)\text{-paths } P'_i \subset E'_{st}, i = 1,2,\ldots,l,\ s.t.\ |P'_i| \leq H'_{st} \text{ and } P'_i \cap \{e_i\} = \emptyset.$$

Thereby, $H_{st} \leq H'_{st}$ and $\hat{E} = \left(\bigcup_{i=1}^{l} P'_i\right) \setminus P$.

3.3 Hop-indexed Formulations

In this subsection we present, as noted before, hop-indexed multicommodity flow based formulations for $\mathcal{F}_{st}$ and $\mathcal{B}_{st}$.

3.3.1 Formulation for $\mathcal{F}_{st}$

For a given $\{s,t\} \in \mathcal{R}$, a formulation for $\mathcal{F}_{st}$ can be derived by using variables $y_{ij}^{st,h} \in \{0,1\}$, indicating whether or not arc $(i,j) \in A_{st}$ is used at position h, in the path from source s to target t, for all $(i,j) \in A_{st}$, $h \in \{1,2,\ldots,H_{st}\}$. The following system (5)–(9) can be used to model (2):

$$\sum_{(s,j) \in A_{st}} y_{sj}^{st,1} = 1 \tag{5}$$

$$\sum_{(i,j) \in A_{st}} y_{ij}^{st,h} = \sum_{(j,i) \in A_{st}} y_{ji}^{st,h+1} \qquad i \in V_{st} \setminus \{s,t\},\ h \in \{1,\ldots,H_{st}-1\} \tag{6}$$

$$\sum_{h=1}^{H_{st}} \sum_{(i,t) \in A_{st}} y_{it}^{st,h} = 1 \tag{7}$$

$$\sum_{h=1}^{H_{st}}(y_{ij}^{st,h} + y_{ji}^{st,h}) \leq x_e \qquad e = \{i,j\} \in E_{st} \qquad (8)$$

$$y_{ij}^{st,h} \in \{0,1\} \qquad (i,j) \in A_{st},\ h \in \{1,\ldots,H_{st}\} \qquad (9)$$

Equations (5)–(7) establish the flow system in the space of disaggregated variables while inequalities (8) are the linking constraints between variables $\mathbf{y}$ and $\mathbf{x}$. Notice, that constraints (8) can be strengthened in a similar manner as previously considered inequalities by the so-called bidirectional commodity-pair forcing constraints, cf. Balakrishnan et al. [1]. Similar ways for strengthening the linking constraints for all models introduced in the remainder of this article can also be proposed. For simplicity, we will omit references to them in the appropriate places.

3.3.2 Formulation for $\mathcal{B}_{st}$ based on Characterization 1

We can easily obtain a valid formulation for $\mathcal{B}_{st}$ based on Characterization 1. We will need to provide a set of variables and a system for each commodity $\{s,t\} \in \mathcal{R}$ and each possible edge failure. Thus consider variables $z_{ij}^{ste,h} \in \{0,1\}$, $\forall s \in T$, $\forall t \in T(s)$, $\forall e \in E_{st}$, $\forall (i,j) \in A'_{st}[e]$, $\forall h \in \{1,2,\ldots,H'_{st}\}$, indicating whether arc (i,j) is used at position h of the backup path of commodity $\{s,t\}$ not using edge e. The corresponding formulation is given by (10)–(14).

$$\sum_{(s,j)\in A'_{st}[e]} z_{sj}^{ste,1} = \sum_{h=1}^{H_{st}} \left(y_{uv}^{st,h} + y_{vu}^{st,h}\right) \qquad e = \{u,v\} \in E_{st} \qquad (10)$$

$$\sum_{(i,j)\in A'_{st}[e]} z_{ij}^{ste,h} = \sum_{(j,i)\in A'_{st}[e]} z_{ji}^{ste,h+1} \qquad e \in E_{st}, i \in V'_{st} \setminus \{s,t\}, \; h \in \{1,\ldots,H'_{st}-1\} \qquad (11)$$

$$\sum_{h=1}^{H'_{st}} \sum_{(i,t)\in A'_{st}[e]} z_{it}^{ste,h} = \sum_{h=1}^{H_{st}} \left(y_{uv}^{st,h} + y_{vu}^{st,h}\right) \qquad e = \{u,v\} \in E_{st} \qquad (12)$$

$$\sum_{h=1}^{H'_{st}}(z_{ij}^{stb,h} + z_{ji}^{stb,h}) \leq x_e \qquad b \in E_{st},\ e = \{i,j\} \in E'_{st} \setminus \{b\} \qquad (13)$$

$$z_{ij}^{ste,h} \in \{0,1\} \qquad e \in E_{st},\ (i,j) \in A'_{st}[e],\ h \in \{1,\ldots,H'_{st}\} \qquad (14)$$

For each commodity $\{s,t\} \in \mathcal{R}$, equations (10)–(12) establish that one unit of backup flow (not using edge e) must be sent between s and t if edge e is contained in the primary path between s and t. This condition is indicated by the right hand side of equations (10) and (12). Inequalities (13) are the straightforward linking constraints.

3.3.3 Formulations for $\mathcal{B}_{st}$ based on Characterization 2

Aggregated formulation Characterization 2 can be enforced by using a straightforward hop-indexed model with an additional set of constraints guaranteeing the condition described in the characterization. More precisely, for each $\{s,t\} \in \mathcal{R}$, constraint (3) of the generic model is replaced by constraints (15)–(20). This formulation uses integer variables $\tilde{z}_{ij}^{st,h} \in \{0,1,\ldots,H_{st}\}$, $\forall \{s,t\} \in \mathcal{R}$, $\forall (i,j) \in A'_{st}$, $\forall h \in \{1,2,\ldots,H'_{st}\}$, indicating the number of backup paths for commodity $\{s,t\}$ that use arc (i,j) at position h.

$$\sum_{(s,j)\in A'_{st}} \tilde{z}_{sj}^{st,1} = \sum_{h=1}^{H_{st}} \sum_{(i,t)\in A_{st}} h y_{it}^{st,h} \qquad (15)$$

$$\sum_{(i,j)\in A'_{st}} \tilde{z}_{ij}^{st,h} = \sum_{(j,i)\in A'_{st}} \tilde{z}_{ji}^{st,h+1} \qquad i \in V'_{st} \setminus \{s,t\},\ \forall h \in \{1,\ldots,H'_{st}-1\} \qquad (16)$$

$$\sum_{h=1}^{H'_{st}} \sum_{(i,t)\in A'_{st}} \tilde{z}_{it}^{st,h} = \sum_{h=1}^{H_{st}} \sum_{(i,t)\in A_{st}} h y_{it}^{st,h} \tag{17}$$

$$\sum_{h=1}^{H_{st}} \left(y_{ij}^{st,h} + y_{ji}^{st,h}\right) + \sum_{h=1}^{H'_{st}} \left(\tilde{z}_{ij}^{st,h} + \tilde{z}_{ji}^{st,h}\right) \leq \sum_{h=1}^{H_{st}} \sum_{(i,t)\in A_{st}} h y_{it}^{st,h} \qquad \{i,j\} \in E_{st} \tag{18}$$

$$\sum_{h=1}^{H'_{st}} \left(\tilde{z}_{ij}^{st,h} + \tilde{z}_{ji}^{st,h}\right) \leq H_{st} x_e \qquad e = \{i,j\} \in E'_{st} \tag{19}$$

$$\tilde{z}_{ij}^{st,h} \in \{0,1,\ldots,H_{st}\} \qquad s \in T,\ t \in T(s), (i,j) \in A'_{st} \tag{20}$$

Equations (15)–(17) ensure that for each commodity pair $\{s,t\} \in \mathcal{R}$ the amount of flow $\tilde{\mathbf{z}}^{st}$ sent from s to t matches the length of the primary path between s and t. Inequalities (18) prevent that all units of backup flow are routed along a single edge that is used in the primary path and therefore assure the existence of an (s,t)-path of length at most H'_{st} after an edge failure, cf. Characterization 2. Finally, constraints (19) are the aggregated linking constraints.

The linear programming relaxation of the previous formulation can be improved by observing that at most all but one of the backup paths of each commodity can be routed via each edge of the *primary path*. Thus, for the edges $e \in E_{st}$, that is edges that may be contained in the primary path, we can strengthen the linking constraints (19) by adding to the left hand side information indicating whether or not the edge is used in the primary path. That is, for $e \in E_{st}$, we use instead the following set of linking constraints.

$$\sum_{h=1}^{H_{st}} \left(y_{ij}^{st,h} + y_{ji}^{st,h}\right) + \sum_{h=1}^{H'_{st}} \left(\tilde{z}_{ij}^{st,h} + \tilde{z}_{ji}^{st,h}\right) \leq H_{st} x_e \qquad e = \{i,j\} \in E_{st} \tag{21}$$

Observe that we maintain (19) for edges $e \in E'_{st} \setminus E_{st}$. Computational results show that with the stronger set of linking constraints the bounds of the aggregated models improve substantially on certain instances.

Disaggregated model It is easy to see that in order to model Characterization 3 we need information from each of the paths in each commodity. For that we need a model that can be viewed as a disaggregation of the previous model. However, we can view this disaggregated model still in terms of Characterization 2 and as a means to improve the performance of the original "aggregated model".

Thus, for each commodity $\{s,t\} \in \mathcal{R}$, we consider binary variables $\hat{z}_{ij}^{stl,h} \in \{0,1\}$, for all $(i,j) \in A'_{st}$, $l \in \{1,2,\ldots,H_{st}\}$, and $h \in \{1,2,\ldots,H'_{st}\}$. Variable $\hat{z}_{ij}^{stl,h}$ is set to one if arc (i,j) is used at position h in the backup path with index l, and to zero, otherwise. In this model, for each $\{s,t\} \in \mathcal{R}$, constraint (3) is replaced by (22)–(27).

$$\sum_{(s,j)\in A'_{st}} \hat{z}_{sj}^{stl,1} = \sum_{h=l}^{H_{st}} \sum_{(i,t)\in A_{st}} y_{it}^{st,h} \qquad l \in \{1,\ldots,H_{st}\} \tag{22}$$

$$\sum_{(i,j)\in A'_{st}} \hat{z}_{ij}^{stl,h} = \sum_{(j,i)\in A'_{st}} \hat{z}_{ji}^{stl,h+1} \qquad i \in V'_{st} \setminus \{s,t\}, l \in \{1,\ldots,H_{st}\}, \; h \in \{1,\ldots,H'_{st}-1\} \tag{23}$$

$$\sum_{h=1}^{H'_{st}} \sum_{(i,t)\in A'_{st}} \hat{z}_{it}^{stl,h} = \sum_{h=l}^{H_{st}} \sum_{(i,t)\in A_{st}} y_{it}^{st,h} \qquad l \in \{1,\ldots,H_{st}\} \tag{24}$$

$$\sum_{h=1}^{H_{st}} \left(y_{ij}^{st,h} + y_{ji}^{st,h}\right) + \sum_{h=1}^{H'_{st}} \sum_{l=1}^{H_{st}} \left(\hat{z}_{ij}^{stl,h} + \hat{z}_{ji}^{stl,h}\right) \leq \sum_{h=1}^{H_{st}} \sum_{(i,t)\in A_{st}} h y_{it}^{st,h} \qquad \{i,j\} \in E_{st} \tag{25}$$

$$\sum_{h=1}^{H'_{st}} \left(\hat{z}_{ij}^{stl,h} + \hat{z}_{ji}^{stl,h} \right) \leq x_e \qquad l \in \{1, \dots, H_{st}\}, e = \{i,j\} \in E'_{st} \tag{26}$$

$$\hat{z}_{ij}^{stl,h} \in \{0,1\} \qquad (i,j) \in A'_{st},\ l \in \{1,\dots,H_{st}\},\ h \in \{1,\dots,H'_{st}\} \tag{27}$$

For each $l \in \{1, \dots, H_{st}\}$, constraints (22)–(24) establish that one unit of backup flow must be sent between s and t if the primary path between s and t has at least l arcs. Inequalities (25) ensure the extra condition of Characterization 2, namely that no edge of the primary path will be used by all established backup paths of one commodity. Constraints (26) are the linking constraints but as in the case of the aggregated model they can be tightened into

$$\sum_{h=1}^{H_{st}} \left(y_{ij}^{st,h} + y_{ji}^{st,h} \right) + \sum_{l=1}^{H_{st}} \sum_{h=1}^{H'_{st}} \left(\hat{z}_{ij}^{stl,h} + \hat{z}_{ji}^{stl,h} \right) \leq H_{st} x_e \qquad e = \{i,j\} \in E_{st} \tag{28}$$

for edges $e \in E_{st}$.

It is worth to compare in size, this disaggregated model with the formulation used to model Characterization 1. The latter includes one set of variables and one set of constraints for each commodity and each possible edge failure. The disaggregated formulation described in this subsection considers one set of variables and constraints for each commodity and each possible edge failure "of the edges in the primary path associated to the considered commodity" whose number is bounded from above by H_{st}. Thus, we see a substantially gain, at least in terms of size for formulations used for Characterization 2.

Relating the LP bounds of the two formulations We start by pointing out that in several situations, the aggregated model produces the same LP bound as the disaggregated model, see for instance [6], in which case it is clearly preferable to the disaggregated one. In fact, it is also with such an aggregated model that the good results reported in Botton et al. [3] are obtained. However, in many other situations the models are not equivalent (see, for instance Gouveia et al. [7]) due to extra constraints, although in most of such cases the aggregated model is more efficient for computing the optimal solutions. In fact, here we are in a similar situation as we see next.

The following constraints link the two sets of variables, from the disaggregated and the aggregated model:

$$z_{ij}^{st,h} = \sum_{l=1}^{H_{st}} \hat{z}_{ij}^{stl,h}.$$

This relation permits us to relate the LP relaxation of the two models. In fact it is easy to see that the disaggregated model is at least as strong as the aggregated model. This follows from the facts that constraints (18) are constraints (25) rewritten with the aggregated variables and that (19) (or (21)) result from adding (26) (or (28)) for all $l \in \{1, \dots, H_{st}\}$. Our computational results show that for some instances the disaggregated model provides slightly better bounds. It also turns out (see Section 4) that the aggregated model typically outperforms the disaggregated one when attempting to obtain optimal integer solutions.

3.3.4 Formulation for $\mathcal{B}_{st}$ based on Characterization 3

Characterization 3 can be easily written in terms of the disaggregated model of the previous subsection, because the variables associated to the primary path systems have information on the position of each arc in the path and the disaggregated model has a separate hop-indexed system for each backup path l with $l \in \{1, \dots, H_{st}\}$. Recall that Characterization 3 assumes a one-to-one correspondence between the lth edge in the primary path, and the lth backup path, and requires that the corresponding backup path does not contain the lth primary edge. This property can be ensured by replacing inequalities (25) by the following inequalities.

$$y_{ij}^{st,l} + y_{ji}^{st,l} + \sum_{h=1}^{H'_{st}} \left(\hat{z}_{ij}^{stl,h} + \hat{z}_{ji}^{stl,h} \right) \leq x_e \qquad e = \{i,j\} \in E'_{st},\ l \in \{1, \dots, H_{st}\} \tag{29}$$

We observe that the new constraints (29) are a disaggregation of constraints (28). However, the aggregated constraints (28) do not even guarantee the extra condition of Characterization 2, namely that no edge of the primary path will be used by all established backup paths of one commodity. That is why we need constraints (25) to obtain a valid model based on Characterization 2. On the other hand, constraints (29) guarantee the condition of Characterization 3 and we can obtain a valid model for $\mathcal{B}_{st}$ by replacing (25) by (29) in (22)–(27). In oder to compare the linear programming relaxation of the two disaggregated models, the one guaranteeing Characterization 2 and the one just presented that guarantees Characterization 3, we first observe that by summing constraints (29) over all $l \in \{1, \ldots, H_{st}\}$ shows that they imply (28). Observe also that the same argument also shows that they imply (26). However, computational experiments prove that constraints (25) are not implied by (29) in terms of linear programming relaxations. Preliminary computational experiments showed that the gain (in terms of LP bounds) from considering both (25) and (29) simultaneously is usually very small and does not compensate the additional time needed for solving the resulting, even larger model. Thus, in our computational study we consider this disaggregated formulation without (25) but conclude this section by emphasizing that from a theoretical perspective the LP relaxation of the last model does not imply the LP relaxation of the previous one.

4 Computational Study

The purpose of our computational study is threefold: (i) to empirically compare the quality of the linear programming relaxations of the proposed models obtained from the three different characterizations; (ii) to analyze the performance of these models within a general-purpose ILP solver; (iii) to compare the solutions obtained by considering the NDPVC to those of the kHSNDP. Notice that (i), (ii) and (iii) focus on the relative potential of the different formulations. Thus, we do not use a problem specific (initial or primal) heuristic but rely on the heuristics included in the used general-purpose solver. Clearly, the absolute performance could be improved (for each of the considered formulations) by augmenting the approach with such heuristic components.

For this purpose, all formulations described in the previous sections have been implemented in C++ using IBM CPLEX 12.6, see Table 1 for a summary. Thereby, standard settings of CPLEX have been used and each experiment has been performed on a single core within a cluster of computers, each consisting of 20 cores (2.3GHz). A time limit of 7 200 CPU-seconds and a memory limit of 3GB has been applied to each experiment. We used identical (commodity independent) primary and secondary hop limits H and H' for all commodities, i.e., $H_{st} = H$ and $H'_{st} = H'$, for all $\{s,t\} \in \mathcal{R}$. In our computational study we have considered uniform hop-limits over all commodities, with the following options for each instance: $H \in \{H_{\min}, H_{\min}+1, H_{\min}+2\}$ and $H' \in \{H, H+1, H+2\}$ with $H_{\min} = \max_{\{s,t\}\in\mathcal{R}} d_{st}$ indicating the smallest commodity independent hop limit that may enable a feasible solution of the instance.

Table 1: Overview on all considered formulations. Formulation name (Name), constraints used for $\mathcal{B}_{st}$ (Cons), used characterization (Char), model variant (Variant), and numbers of variables ($\#_{\text{var}}$) per commodity $\{s,t\} \in \mathcal{R}$.

Name	Cons	Char	Variant	$\#_{\text{var}}$
H_1	(5)–(9)	1	-	$\mathcal{O}(A_{st}H_{st} + A'_{st}A_{st}H'_{st})$
H_2	(22)–(27)	2	disaggregated	$\mathcal{O}(A_{st}H_{st} + A'_{st}H_{st}H'_{st})$
H_2^{S}	(22)–(27), (28)	2	disaggregated, strong linking	$\mathcal{O}(A_{st}H_{st} + A'_{st}H_{st}H'_{st})$
H_2^{A}	(15)–(20)	2	aggregated	$\mathcal{O}(A_{st}H_{st} + A'_{st}H'_{st})$
H_2^{AS}	(15)–(20), (21)	2	aggregated, strong linking	$\mathcal{O}(A_{st}H_{st} + A'_{st}H'_{st})$
H_3	(22)–(24), (26), (27), (29)	3	-	$\mathcal{O}(A_{st}H_{st} + A'_{st}H_{st}H'_{st})$

4.1 Benchmark Instances

We have created two classes of benchmark instances that are described in detail in the following two paragraphs. One main difference between the two instances classes is their graph topology. The first class is based on grid graphs with chords while the second one is generated by keeping only the cheapest edges from a randomly generated complete graph.

Table 2: Summary of parameters used for creating grid instance sets and resulting minimum, average, and maximum values of $H_{\min}$.

Set	X	Y	$\|\mathcal{R}\|$	#	$\|V\|$	$\|E\|$	H_{L}	H_{U}	$C_{\min}$	$C_{\max}$	$C^{\mathrm{D}}_{\min}$	$C^{\mathrm{D}}_{\max}$	$H_{\min}$ min	$H_{\min}$ avg	$H_{\min}$ max
C	10	10	5	20	100	342	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	3	4.7	7
C	10	10	10	20	100	342	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.4	7
C	20	20	5	20	400	1482	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	3	4.7	7
C	20	20	10	20	400	1482	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.1	7
C	20	20	20	20	400	1482	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.8	7
C	30	30	5	20	900	3422	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	3	4.7	7
C	30	30	10	20	900	3422	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.4	7
C	30	30	20	20	900	3422	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.6	7
C	30	30	30	20	900	3422	2	$\in \{5,7\}$	1	10	10	$\in \{20,50\}$	4	5.9	7
D	5	5	10	10	25	72	-	-	1	10	10	$\in \{20,50\}$	3	3.8	4
D	7	7	10	10	49	156	-	-	1	10	10	$\in \{20,50\}$	4	5.2	6
D	10	10	10	10	100	342	-	-	1	10	10	$\in \{20,50\}$	7	8.2	9
D	10	10	45	10	100	342	-	-	1	10	10	$\in \{20,50\}$	6	8.1	9
D	20	20	10	10	400	1482	-	-	1	10	10	$\in \{20,50\}$	12	14.6	18

Grid instances Within this class, we have created two subsets of benchmark instances which are based on grid-graphs additionally including two chords for each 4-cycle. Their structure is motivated from street networks (which may be represented by grid graphs) in which diagonal connections (chords in the resulting 4-cycles) are possible but usually more expensive. Each generated instance consists of $X \cdot Y$ nodes, where X and Y are the numbers of columns and rows in the resulting graph, see Figure 3 for one example. Costs of horizontal and vertical edges (that correspond to lay cables along streets) are integer values chosen randomly from the interval $[C_{\min}, C_{\max}]$ while costs of the remaining "diagonal" edges are random integers from $[C^{\mathrm{D}}_{\min}, C^{\mathrm{D}}_{\max}]$.

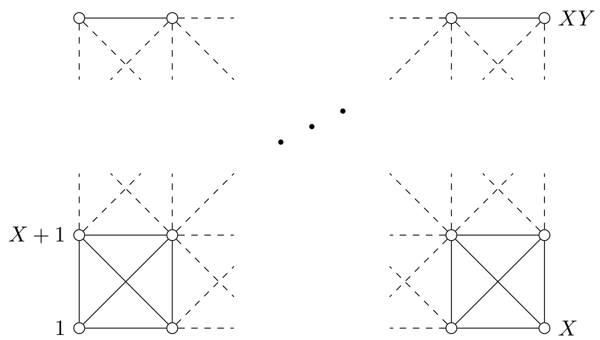


Figure 3: Structure of the created input graphs for grid instances.

The two subclasses differ with respect to the selection of commodities. For set C, we first chose $|\mathcal{R}|$ nodes as commodity sources uniformly at random. For each source s, we then pick the corresponding target node at random among all nodes $t \in V$ such that $d_{st} \in [H_{\mathrm{L}}, H_{\mathrm{U}}]$, i.e., the distance between s and t (measured in hops) is at least H_{L} and at most H_{U}. Thereby, we also ensure that no commodity is created twice. Instances from set D correspond to the diameter variant of the problem mentioned in the introduction. Thus, $|T|$ nodes are chosen uniformly at random as terminals and one commodity is created for each pair of selected nodes, i.e., we have $|\mathcal{R}| = \binom{|T|}{2}$ and $\mathcal{R} = \{\{s,t\} \mid s,t \in T, s < t\}$. In either case (i.e., both for instances of set C and D) we created five instances for each considered combination of parameters. Table 2 provides a summary of all considered parameters for grid-graph based instances in which we also report minimum, average and maximum values of $H_{\min}$ for the instances of each set.

Random (Euclidean) instances This class consists of two subclasses (*EU* and *RE*) of benchmark instances that have been created as follows: First a complete graph (V, E) is created that consists of

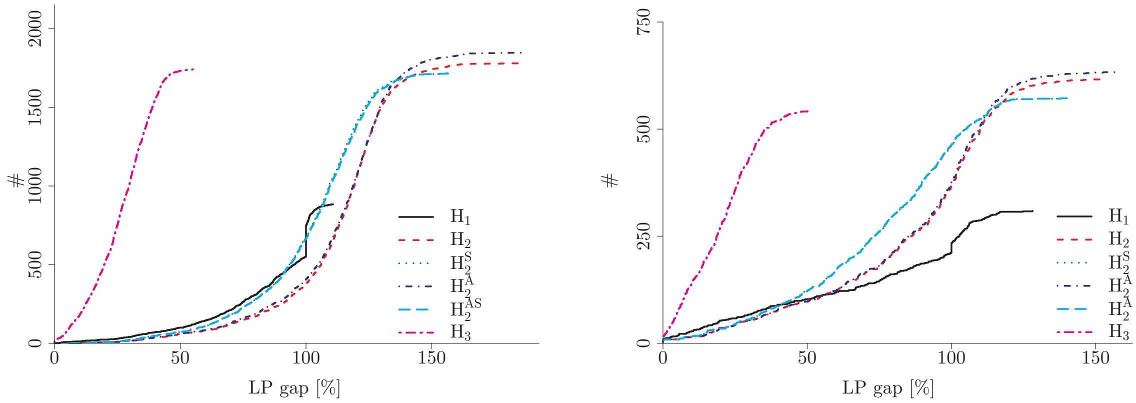
Table 3: Summary of parameters used for creating (random) Euclidean instance sets and resulting minimum, average, and maximum values of $H_{\min}$.

Set	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{R}\rvert$	#	α	β	$H_{\min}$ min	avg	max
E	50	122	10	5	1	0.1	6	7.6	9
E	50	122	45	5	1	0.1	7	8.6	11
E	50	245	10	5	1	0.2	4	4.6	6
E	50	245	45	5	1	0.2	4	4.8	6
E	75	277	10	5	1	0.1	5	5.6	6
E	75	277	45	5	1	0.1	6	8.6	12
E	75	555	10	5	1	0.2	3	4.4	6
E	75	555	45	5	1	0.2	4	4.6	5
E	100	495	10	5	1	0.1	5	5.2	6
E	100	495	45	5	1	0.1	6	7.2	9
E	100	990	10	5	1	0.2	3	4.0	5
E	100	990	45	5	1	0.2	3	4.8	6
RE	50	122	10	5	[1,10)	0.1	3	4.6	6
RE	50	122	45	5	[1,10)	0.1	4	5.2	6
RE	50	245	10	5	[1,10)	0.2	2	2.8	3
RE	50	245	45	5	[1,10)	0.2	3	3.0	3
RE	75	277	10	5	[1,10)	0.1	3	3.8	4
RE	75	277	45	5	[1,10)	0.1	4	4.2	5
RE	75	555	10	5	[1,10)	0.2	2	2.6	3
RE	75	555	45	5	[1,10)	0.2	2	2.8	3
RE	100	495	10	5	[1,10)	0.1	3	3.6	5
RE	100	495	45	5	[1,10)	0.1	4	4.0	4
RE	100	990	10	5	[1,10)	0.2	2	2.0	2
RE	100	990	45	5	[1,10)	0.2	3	3.0	3

$|V|$ nodes which are placed at random integer coordinates $(x, y) \in [0, 100) \times [0, 100)$. As for instance set D, $|T|$ of these nodes are chosen uniformly at random as terminals and one commodity is created for each pair of terminals. Let d_{uv} denote the Euclidean distance between nodes u and v. The costs of each edge $(u, v) \in E$ are set to $\lceil \alpha d_{uv} \rceil$. Thereby, parameter α is set to one for instance family EU (Euclidean instances) and to a random value in $[1, 10)$ that is drawn uniformly at random individually for each edge for instance family RE (random Euclidean instances). Parameter β, $0 \leq \beta \leq 1$ defines the desired density of the resulting graph, i.e., all but $\lfloor \beta \binom{|V|}{2} \rfloor$ edges of the previously created complete graph are removed. To this end, let $\mathcal{T}$ be a minimum spanning tree of graph (V, E) and $\mathcal{T}'$ be a minimum spanning tree of $V, E \setminus \mathcal{T})$, i.e., a tree of (V, E) that is edge disjoint from $\mathcal{T}$ ensuring that there are at least two edge disjoint paths between every pair of nodes in $(V, \mathcal{T} \cup \mathcal{T}')$. Let $E' \subseteq E \setminus (\mathcal{T} \cup \mathcal{T}')$ be the $\max\{0, \lfloor \beta \binom{|V|}{2} \rfloor - 2(|V|-1)\}$ cheapest edges of E not contained in any of these two spanning trees. Then, the final graph defining an instance is given by $(V, E' \cup \mathcal{T} \cup \mathcal{T}')$. Five instances have been created for each subclass and each considered combination of $|V| \in \{50, 75, 100\}$, $|T| \in \{5, 10\}$, and $\beta \in \{0.1, 0.2\}$. Table 3 provides a summary of all considered parameters for these instances including minimum, average and maximum values of $H_{\min}$ for the instances of each set.

4.2 Quality of the LP Relaxation Values

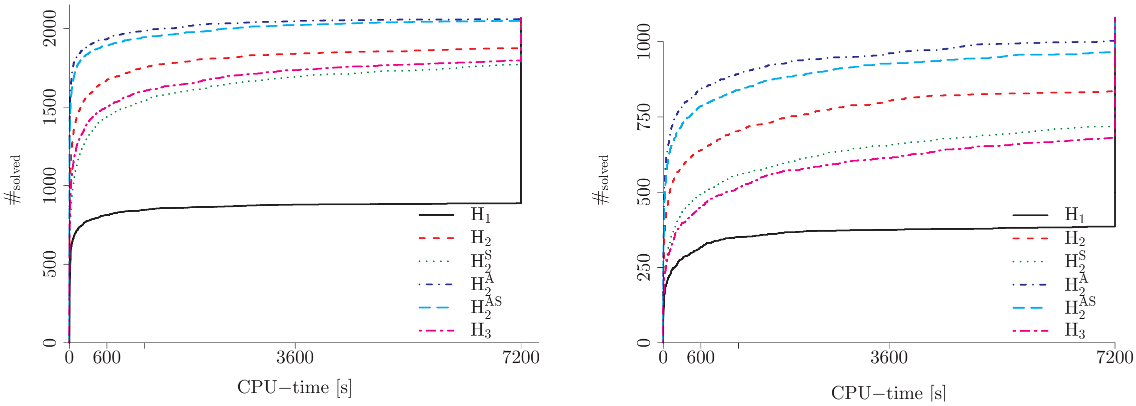
We first compare the quality of lower bounds (LP relaxation) of the proposed formulations and the time needed to compute them. For each considered formulation $M \in \{\mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_2^{\mathrm{A}}, \mathrm{H}_2^{\mathrm{S}}, \mathrm{H}_2^{\mathrm{AS}}, \mathrm{H}_3\}$, Figure 4 shows gaps between the LP relaxation ($\mathrm{LP}(M)$) and the best upper bound (UB) obtained for that particular instance (by any of the tested formulations) in our computational study which are computed as $\frac{UB - \mathrm{LP}(M)}{\mathrm{LP}(M)}$. Thereby, we only consider those instances for which the optimal solution value of either the NDPVC or the kHCNDP is known in order to ensure that the value of UB is close to the real optimum. Corresponding CPU-times for solving the LP relaxations are shown in Figure 5. In addition, Tables 4 and 5 summarize numbers of instances for which the LP relaxation could be solved within the given time and memory limits and corresponding average CPU-times grouped by the different instance



(a) Grid instances

(b) (Random) Euclidean instances

Figure 4: Cumulative numbers of instances (#) for which gap of LP relaxation to the best known solution (LP gap [%]) is within a certain value. (a) Results for instance sets *C* and *D*. (b) Results for instance sets *EU* and *RE*.



(a) Grid instances

(b) (Random) Euclidean instances

Figure 5: Cumulative numbers of instances ($\#_{\text{solved}}$) for which LP relaxation could be solved within a certain time (CPU-time [s]). (a) Results for instance sets *C* and *D*. (b) Results for instance sets *EU* and *RE*.

sets considered in this study.

From Figure 4 we first observe that formulation H_3 based on Characterization 3 is significantly stronger than all other variants on the considered instances. The gaps of its LP relaxation to the best known solution can still be quite significant but are usually below 50%. In contrast, the gaps of all other formulations are substantially larger in almost all cases. Formulation H_1 seems to be stronger (in practice) than the formulations based on Characterization 2 on grid based instances (see, Figure 4(a)). On the contrary, this does not seem to be the case on (random) Euclidean) instances. The results on the latter instances do not indicate a clear trend regarding the comparison of the empirically observed LP gaps between H_1 and the formulations based on Characterization 2. In both cases, solving the LP relaxation of H_1 was often impossible given the available computational resources due to its large number of variables.

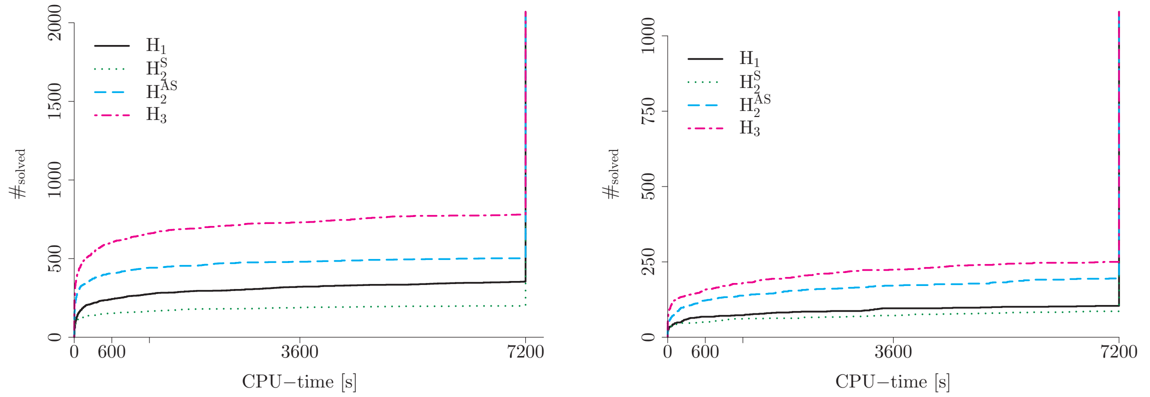
We also observe that two strengthened formulations based on Characterization 2 (H_2^S, H_2^{AS}) are significantly stronger than their simpler counterparts H_2 and H_2^A, respectively. The bounds obtained from the aggregated and disaggregated variants of the latter formulations (both for the weak and strong alternatives) are almost identical. Here, we note that one cannot draw conclusions regarding theoretical strength relations between the different models as the considered instances are different for the various

Table 4: Number of cases in which the LP relaxation could be solved ($\#_{\text{solved}}$) and corresponding average CPU-times in seconds (t_{avg}) for instance sets C and D. Best values are marked in bold.

			$\#_{\text{solved}}$						t_{avg} [s]					
Set	$\lvert\mathcal{R}\rvert$	#	H_1	H_2	$\mathrm{H}_2^{\mathrm{S}}$	$\mathrm{H}_2^{\mathrm{A}}$	$\mathrm{H}_2^{\mathrm{AS}}$	H_3	H_1	H_2	$\mathrm{H}_2^{\mathrm{S}}$	$\mathrm{H}_2^{\mathrm{A}}$	$\mathrm{H}_2^{\mathrm{AS}}$	H_3
C10x10	5	180	**180**	**180**	**180**	**180**	**180**	**180**	156	10	57	**1**	**1**	34
C10x10	10	180	141	**180**	177	**180**	**180**	**180**	2074	145	654	**8**	14	455
C20x20	5	180	143	**180**	**180**	**180**	**180**	**180**	1490	11	62	**1**	**1**	34
C20x20	10	180	80	**180**	179	**180**	**180**	**180**	4007	50	289	**3**	4	160
C20x20	20	180	13	170	146	**180**	**180**	148	6681	980	2318	**63**	130	2157
C30x30	5	180	102	**180**	**180**	**180**	**180**	**180**	3125	7	24	**1**	**1**	15
C30x30	10	180	21	**180**	**180**	**180**	**180**	**180**	6362	69	350	**3**	5	213
C30x30	20	180	4	176	160	**180**	**180**	161	7040	628	1605	**30**	56	1427
C30x30	30	180	0	144	130	**180**	**180**	134	7200	1918	3162	**142**	309	2813
D5x5	10	90	**90**	**90**	**90**	**90**	**90**	**90**	28	2	8	**0**	1	4
D7x7	10	90	**90**	**90**	**90**	**90**	**90**	**90**	400	33	167	**2**	3	81
D10x10	10	90	24	88	69	**90**	**90**	77	5498	1063	3221	**29**	61	2322
D10x10	45	90	0	30	10	**90**	86	15	7200	5841	6806	**1053**	2081	6499
D20x20	10	90	0	7	1	**81**	73	2	7200	6874	7120	**2249**	2920	7098
-	-	2070	888	1875	1772	**2061**	2049	1797	4200	933	1494	**167**	266	1331

Table 5: Number of cases in which the LP relaxation could be solved ($\#_{\text{solved}}$) and corresponding average CPU-times in seconds (t_{avg}) for instance sets EU and RE. Best values are marked in bold.

					$\#_{\text{solved}}$						t_{avg} [s]					
Set	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{R}\rvert$	#	H_1	H_2	$\mathrm{H}_2^{\mathrm{S}}$	$\mathrm{H}_2^{\mathrm{A}}$	$\mathrm{H}_2^{\mathrm{AS}}$	H_3	H_1	H_2	$\mathrm{H}_2^{\mathrm{S}}$	$\mathrm{H}_2^{\mathrm{A}}$	$\mathrm{H}_2^{\mathrm{AS}}$	H_3
EU	50	122	10	45	**45**	**45**	**45**	**45**	**45**	**45**	157	29	101	**2**	**2**	260
EU	50	122	45	45	14	**45**	39	**45**	**45**	31	4997	929	2407	**33**	45	3068
EU	50	245	10	45	35	**45**	44	**45**	**45**	**45**	2049	47	414	**6**	8	565
EU	50	245	45	45	2	29	20	**45**	**45**	15	6880	3463	5070	**372**	651	5413
EU	75	277	10	45	32	**45**	44	**45**	**45**	**45**	2504	106	642	**9**	11	771
EU	75	277	45	45	0	16	9	**45**	**45**	9	7200	5317	6122	**1006**	1371	6235
EU	75	555	10	45	7	**45**	38	**45**	**45**	29	6109	597	2246	**58**	125	3378
EU	75	555	45	45	0	16	6	**35**	29	7	7200	5189	6490	**3085**	4155	6642
EU	100	495	10	45	16	**45**	42	**45**	**45**	38	4741	351	1467	**25**	47	2119
EU	100	495	45	45	0	10	6	**33**	28	5	7200	5836	6369	**3269**	4230	6496
EU	100	990	10	45	2	37	23	**45**	**45**	20	6880	2239	4376	**278**	661	4569
EU	100	990	45	45	0	7	5	**16**	12	5	7200	6222	6441	**5319**	5785	6446
RE	50	122	10	45	**45**	**45**	**45**	**45**	**45**	**45**	234	14	88	**2**	3	134
RE	50	122	45	45	20	43	36	**45**	**45**	33	4857	1081	2649	**57**	84	3107
RE	50	245	10	45	37	**45**	**45**	**45**	**45**	**45**	1505	18	124	**3**	6	155
RE	50	245	45	45	15	40	30	**45**	**45**	29	5169	1815	3543	**212**	529	3389
RE	75	277	10	45	36	**45**	**45**	**45**	**45**	**45**	1805	29	173	**5**	8	386
RE	75	277	45	45	8	36	24	**45**	**45**	25	6126	2593	4246	**330**	648	4399
RE	75	555	10	45	21	**45**	43	**45**	**45**	41	3879	128	823	**22**	46	1403
RE	75	555	45	45	8	24	19	**43**	35	19	5990	3754	4601	**1705**	2862	4575
RE	100	495	10	45	21	**45**	44	**45**	**45**	40	3906	212	1103	**21**	39	1635
RE	100	495	45	45	1	21	12	**41**	32	13	7040	4667	5757	**2182**	3136	5796
RE	100	990	10	45	21	**45**	43	**45**	**45**	39	3850	71	1064	**26**	81	1505
RE	100	990	45	45	0	18	12	**25**	20	13	7200	4976	5825	**4151**	4578	5721
-	-	-	-	1080	386	837	719	**1003**	966	681	4778	2070	3006	**924**	1213	3257



(a) Grid instances

(b) (Random) Euclidean instances

Figure 6: Cumulative numbers of instances ($\#_{\text{solved}}$) solved within a certain time (CPU-time [s]). (a) Results for instance sets C and D. (b) Results for instance sets EU and RE.

variants (only those instances where the LP relaxation was solved are considered). From our detailed results, we observe that the disaggregated variants are sometimes slightly stronger than their aggregated counterparts. This difference was, however, extremely small and the obtained bounds were identical in many cases.

Considering the CPU-times given in Figure 5 we observe that the LP relaxations of the two aggregated variants H_2^{A} and H_2^{AS} can be solved much faster than those of all others. Solving the LP relaxations of formulation H_3 takes approximately the same time as for the strong variant of the disaggregated model based on Characterization 2, i.e., H_2^{S}. Model H_1 clearly is too slow to be of much use in practice. Overall, formulations H_2^{AS} and H_3 seem to provide the best compromises between strength of their LP relaxations and the time needed to solve them.

These observations are also supported by the more detailed results reported in Tables 4 and 5.

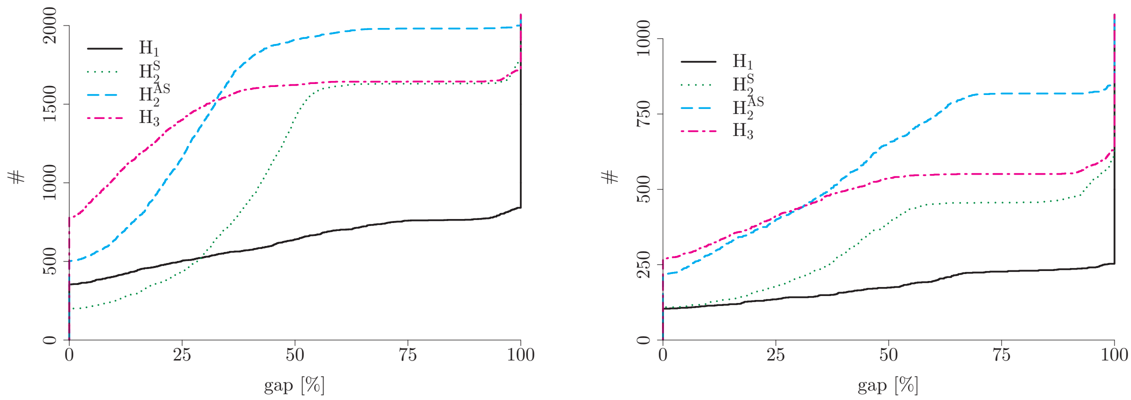
4.3 Capability of Computing Optimal Solutions

Our next goal was to compare the performance of the proposed models, when it comes to finding optimal or near-optimal solutions. As discussed above, the bounds of formulations H_2 and H_2^{A} are extremely weak. We therefore do not consider these two variants in the remainder of the study but focus on their stronger counterparts H_2^{S} and H_2^{AS}, respectively. Results are visualized in Figures 6 and 7 which show cumulative numbers of instances solved within a given time and for which the remaining optimality gap is within a given value, respectively. Since no feasible solutions could be identified in some cases, optimality gaps have been computed as $\frac{UB(M)-LB(M)}{UB(M)}$ for each model $M \in \{\text{H}_1, \text{H}_2^{\text{S}}, \text{H}_2^{\text{AS}}, \text{H}_3\}$. Thereby, $UB(M)$ and $LB(M)$ denote the upper and lower bound, respectively, obtained by model M.

From Figure 6 we conclude that formulation H_3 is able to solve more instances to proven optimality than any of the other models considered. Despite its significantly weaker LP bounds, formulation H_2^{AS} comes relatively close possibly due to its capability to enumerate much more branch-and-bound nodes in the same amount of time. Figure 7 indicates that H_2^{AS} seems to be able to provide smaller optimality gaps than H_3 in some of the most difficult test instances considered. The other two options (H_1, H_2^{S}) are clearly dominated by H_3 and H_2^{AS}. All these observations hold for instance sets C and D (grid instances) as well as for instance sets EU and RE (random Euclidean instances).

Additional insights can be obtained from the more fine grained information provided in Tables 6-9 which detail numbers of solved instances, average CPU-times and average optimality gaps grouped by instance set and considered combination of (H, H'), respectively.

First consider the results for grid instances that are summarized in Tables 6 and 7. From Table 6 we conclude that H_3 outperforms H_2^{AS} with respect to numbers of solved instances and average CPU-times in particular on instances of set C, i.e., those where commodities are generated independently. On the contrary, they achieve a comparable performance on set D, i.e., the "diameter cases" where commodities



(a) Grid instances

(b) (Random) Euclidean instances

Figure 7: Cumulative numbers of instances (#) for which remaining optimality gap (gap [%]) is within a certain value. (a) Results for instance sets *C* and *D*. (b) Results for instance sets *EU* and *RE*.

Table 6: Numbers of instances from instance sets *C* and *D* solved to optimality ($\#_{\text{solved}}$), average CPU-times in seconds (t_{avg}), and remaining average optimality gaps (avg. gap [%]). Best values are marked in bold.

			$\#_{\text{solved}}$				t_{avg} [s]				avg. gap [%]			
Set	$\|\mathcal{R}\|$	#	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3
C10x10	5	180	63	28	74	**124**	5030	6188	4463	**2624**	29.1	27.8	11.1	**5.2**
C10x10	10	180	17	4	19	**42**	6646	7073	6556	**5710**	62.2	45.2	**22.5**	24.7
C20x20	5	180	80	42	93	**134**	4305	5595	3665	**2224**	30.9	25.4	9.4	**3.9**
C20x20	10	180	18	4	24	**66**	6635	7050	6381	**4824**	68.6	38.4	19.7	**12.3**
C20x20	20	180	1	0	1	**6**	7162	7200	7161	**6999**	97.4	61.9	**31.4**	44.8
C30x30	5	180	71	37	100	**142**	4577	5797	3344	**1811**	53.3	26.0	8.9	**3.2**
C30x30	10	180	11	7	30	**71**	6770	6925	6115	**4677**	92.3	40.6	19.6	**12.0**
C30x30	20	180	1	1	9	**29**	7160	7160	6883	**6208**	98.9	52.9	**24.9**	29.6
C30x30	30	180	0	1	2	**8**	7200	7162	7128	**6945**	100.0	70.6	**35.0**	52.8
D5x5	10	90	66	60	**88**	**88**	2660	2932	**437**	518	10.0	8.8	1.2	**1.1**
D7x7	10	90	25	15	53	**62**	5822	6191	3404	**3087**	44.9	29.6	**9.0**	9.2
D10x10	10	90	1	0	**9**	**9**	7184	7200	6591	**6544**	92.5	71.5	**26.4**	66.5
D10x10	45	90	0	0	0	0	7200	7200	7200	7200	100.0	98.3	**67.9**	96.6
D20x20	10	90	0	0	0	0	7200	7200	7200	7200	100.0	100.0	**70.3**	100.0
-	-	2070	354	199	502	**781**	6132	6566	5575	**4721**	70.1	47.2	**23.5**	28.3

Table 7: Numbers of instances from instance sets C and D solved to optimality ($\#_{\text{solved}}$), average CPU-times in seconds (t_{avg}), and remaining average optimality gaps (avg. gap [%]) for each considered combination of $(H, H') = (H_{\min} + \Delta_H, H_{\min} + \Delta_H + \Delta_{H'})$. Best values are marked in bold.

			$\#_{\text{solved}}$				t_{avg} [s]				avg. gap [%]			
Set	$(\Delta_H, \Delta_{H'})$	#	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3
C	(0,0)	180	65	50	84	**123**	4748	5284	3989	**2569**	45.6	20.6	8.8	**3.7**
	(0,1)	180	57	28	66	**109**	5085	6151	4669	**3128**	53.9	28.3	12.1	**8.0**
	(0,2)	180	46	19	56	**99**	5570	6511	5102	**3574**	59.8	34.0	13.8	**12.3**
	(1,0)	180	32	13	44	**70**	6108	6711	5568	**4613**	67.4	40.4	18.7	**13.5**
	(1,1)	180	22	7	34	**61**	6442	6944	5980	**4899**	72.9	44.7	**20.5**	**20.5**
	(1,2)	180	15	5	24	**58**	6736	7028	6366	**5160**	78.6	49.5	**22.6**	26.7
	(2,0)	180	12	2	20	**41**	6815	7122	6493	**5726**	81.3	52.0	**26.5**	27.3
	(2,1)	180	7	0	12	**37**	6951	7200	6739	**5969**	85.0	56.7	**29.1**	33.2
	(2,2)	180	6	0	12	**24**	7032	7200	6790	**6385**	88.0	62.7	**30.5**	43.3
D	(0,0)	50	15	14	**20**	19	5183	5206	**4352**	4477	60.1	49.2	**25.4**	45.0
	(0,1)	50	17	14	23	**24**	5007	5264	3942	**3797**	58.6	51.0	**23.5**	46.4
	(0,2)	50	16	13	21	**22**	5172	5415	4338	**4084**	61.5	54.6	**25.4**	47.9
	(1,0)	50	13	10	16	**20**	5959	6141	5094	**4724**	64.8	59.6	**36.3**	53.7
	(1,1)	50	10	8	17	**18**	6021	6343	4959	**4797**	69.4	62.6	**34.5**	55.8
	(1,2)	50	10	7	16	**18**	6212	6292	4963	**4937**	71.8	64.8	**36.4**	58.3
	(2,0)	50	4	3	**13**	**13**	6809	6838	**5622**	5684	77.5	70.2	**42.6**	59.4
	(2,1)	50	4	3	**13**	**13**	6838	6958	5685	**5843**	77.6	70.8	**43.3**	59.8
	(2,2)	50	3	3	11	**12**	6917	6843	**5742**	5844	84.0	72.0	**47.3**	65.9
	-	2070	354	199	502	**781**	6132	6566	5575	**4721**	70.1	47.2	**23.5**	28.3

Table 8: Numbers of instances from instance sets EU and RE solved to optimality ($\#_{\text{solved}}$), average CPU-times in seconds (t_{avg}), and remaining average optimality gaps (avg. gap [%]). Best values are marked in bold.

					$\#_{\text{solved}}$				t_{avg} [s]				avg. gap [%]			
Set	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert \mathcal{R} \rvert$	#	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3
EU	50	122	10	45	8	2	**19**	15	6179	6971	**4725**	5262	63.8	48.4	**13.4**	27.4
	50	122	45	45	0	0	**1**	0	7200	7200	**7065**	7200	100.0	79.1	**47.0**	82.2
	50	245	10	45	5	4	10	**15**	6641	6789	5775	**5257**	72.8	44.4	25.5	**25.4**
	50	245	45	45	0	0	0	0	7200	7200	7200	7200	100.0	94.4	**59.5**	90.7
	75	277	10	45	1	0	7	**8**	7052	7200	6329	**6016**	84.4	50.9	**26.8**	32.4
	75	277	45	45	0	0	0	0	7200	7200	7200	7200	100.0	95.6	**73.7**	96.5
	75	555	10	45	1	1	6	**10**	7050	7059	6541	**5835**	92.4	69.8	**35.5**	48.6
	75	555	45	45	0	0	0	0	7200	7200	7200	7200	100.0	97.9	**78.7**	94.1
	100	495	10	45	2	1	4	**7**	6918	7086	6691	**6147**	89.9	67.2	**37.9**	47.5
	100	495	45	45	0	0	0	0	7200	7200	7200	7200	100.0	99.8	**89.7**	99.8
	100	990	10	45	1	2	3	**6**	7045	7068	6724	**6257**	97.8	80.4	**45.8**	64.3
	100	990	45	45	0	0	0	**1**	7200	7200	7200	**7093**	100.0	98.5	**94.3**	96.0
RE	50	122	10	45	16	13	27	**29**	4915	5625	3331	**3036**	40.3	28.9	**8.4**	13.6
	50	122	45	45	3	2	**8**	6	6825	6902	**6217**	6474	79.3	72.1	**34.2**	60.3
	50	245	10	45	16	16	26	**32**	4738	4926	3845	**2735**	49.7	30.1	14.3	**9.6**
	50	245	45	45	0	0	7	**10**	7200	7200	6465	**6161**	87.2	73.8	**41.4**	56.8
	75	277	10	45	12	8	20	**25**	5774	6070	4352	**3615**	57.4	38.2	19.9	**14.9**
	75	277	45	45	0	0	**1**	**1**	7200	7200	7168	**7105**	97.1	85.8	**51.1**	76.1
	75	555	10	45	12	13	16	**22**	5732	5731	4852	**3976**	66.5	47.4	**24.7**	24.9
	75	555	45	45	2	2	2	**7**	6881	6880	6880	**6318**	91.4	79.9	**58.4**	64.4
	100	495	10	45	10	7	16	**21**	5948	6187	5099	**4241**	72.4	51.6	**27.2**	30.4
	100	495	45	45	0	0	0	0	7200	7200	7200	7200	100.0	96.6	**68.7**	88.2
	100	990	10	45	15	15	22	**32**	4898	4981	4215	**2818**	62.2	37.7	21.5	**13.1**
	100	990	45	45	0	0	0	**3**	7200	7200	7200	**6941**	100.0	89.9	**72.5**	74.8
	-	-	-	1080	104	86	195	**250**	6608	6728	6111	**5770**	83.5	69.1	**44.6**	55.5

Table 9: Numbers of instances from instance sets *EU* and *RE* solved to optimality ($\#_{\text{solved}}$), average CPU-times in seconds (t_{avg}), and remaining average optimality gaps (avg. gap [%]) for instance sets *EU* and *RE* and each considered combination of $(H, H') = (H_{\min} + \Delta_H, H_{\min} + \Delta_H + \Delta_{H'})$. Best values are marked in bold.

			$\#_{\text{solved}}$				t_{avg} [s]				avg. gap [%]			
Set	$(\Delta_H, \Delta_{H'})$	#	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3	H_1	H_2^S	H_2^{AS}	H_3
EU	(0,0)	60	3	4	6	**8**	6987	6939	6547	**6256**	93.5	76.5	**71.1**	72.0
	(0,1)	60	6	4	16	**20**	6527	6874	5634	**4973**	78.1	56.9	**29.5**	45.2
	(0,2)	60	7	1	13	**18**	6521	7089	5779	**5216**	79.9	64.9	**31.6**	51.6
	(1,0)	60	1	1	4	**5**	7122	7129	6819	**6645**	91.3	71.2	**45.9**	54.6
	(1,1)	60	1	0	3	**5**	7104	7200	6897	**6768**	91.7	77.2	**50.8**	65.7
	(1,2)	60	0	0	4	**5**	7200	7200	**6891**	6980	96.0	83.4	**53.6**	71.8
	(2,0)	60	0	0	**2**	1	7200	7200	**7029**	7161	97.0	82.0	**60.3**	71.4
	(2,1)	60	0	0	**1**	0	7200	7200	**7175**	7200	99.0	86.8	**63.6**	81.8
	(2,2)	60	0	0	**1**	0	7200	7200	**7117**	7200	99.3	95.9	**64.3**	89.7
RE	(0,0)	60	15	14	16	**25**	5528	5676	5392	**4543**	66.8	58.0	52.3	**49.6**
	(0,1)	60	27	24	34	**39**	4332	4536	3238	**3028**	39.4	26.6	**7.2**	10.6
	(0,2)	60	25	23	**38**	36	4478	4660	**3108**	3155	50.8	36.8	**8.4**	20.6
	(1,0)	60	9	7	19	**29**	6529	6811	5408	**4089**	66.7	51.5	**27.2**	31.1
	(1,1)	60	4	3	14	**25**	6839	6967	6064	**4827**	80.3	62.3	**33.0**	41.8
	(1,2)	60	3	2	17	**19**	6889	7058	5991	**5472**	89.0	67.8	**36.6**	52.4
	(2,0)	60	1	1	3	**7**	7084	7092	6967	**6585**	93.4	75.6	54.2	**54.0**
	(2,1)	60	1	1	2	**4**	7103	7135	6975	**6851**	94.5	81.6	**55.5**	63.6
	(2,2)	60	1	1	2	**4**	7102	7141	6975	**6915**	96.7	89.0	**57.4**	71.7
	-	1080	104	86	195	**250**	6608	6728	6111	**5770**	83.5	69.1	**44.6**	55.5

are present between each pair of a chosen set of terminals. Table 6 also confirms that the remaining gaps of H_2^{AS} after reaching the time- or memory limit tend to be slightly smaller than those of H_3 on particularly hard subsets (e.g., C30x30 with $\mathcal{R} \in \{20, 30\}$ or D10x10 and D20x20). A clear correlation between difficulty of an instance and increasing value of (H, H') can be observed from Table 7 for all formulations both for instance sets C and D. When H and / or H' increases, less instances can be solved to optimality, average CPU-time and remaining optimality gaps increase. We suppose that besides the additionally considered variables in each formulation (due to a larger number of eligible primary and / or secondary edges) this stems from typically weaker LP relaxation values (in particular for H_2^S and H_2^{AS} due to the coefficient H in the linking constraints), and the fact that the used values of H and H' are typically still rather restrictive (recall that we start with the smallest values that may allow for a feasible solution).

Similar trends can be observed for Euclidean and random Euclidean instances, i.e., from Tables 8 and 9. In addition, we conclude that the Euclidean instance set *EU* seems significantly harder to solve than set *RE* in which the Euclidean distances are multiplied by random values in order to obtain the edge costs. Many more instances from set *RE* could be solved to proven optimality and the remaining gaps are significantly smaller. The results from Tables 8 also clearly indicate that the performance of all considered formulations heavily suffers from an increasing number of commodities.

Overall, both H_2^{AS} and H_3 can be recommended for solving not too large instances of the NDPVC. While, the quite large number of variables prevents their direct application to large scale instances in particular H_3 may be a good starting point for a (Benders) decomposition approach in which most of these variables can be projected out due to its stronger LP bounds.

4.4 Comparison to the kHSNDP

Finally, we compare the solutions obtained from solving the NDPVC to those of the related kHSNDP. Table 10-13 provide numbers of cases in which the optimal solution to the NDPVC are proven to be better or equal than those of the kHSNDP (which has been solved by a simple flow formulation and using a time limit of 10 800 seconds) grouped by instance sets and considered hop limits, respectively. As above, results are separately shown for grid instances (Tables 10 and 11) and (random) Euclidean instances (Tables 12 and 13). A feasible solution found by any of the formulations considered for the

Table 10: Overall number of cases from instance sets C and D in which solution to NDPVC is provable better (bt) and provably equal (eq) compared to the kHSNDP and corresponding values obtained for the different formulations considered grouped for each considered instance set. Remaining cases could not be decided.

			Overall		H_1		H_2^S		H_2^{AS}		H_3	
Set	$\lvert\mathcal{R}\rvert$	#	bt	eq	bt	eq	bt	eq	bt	eq	bt	eq
C10x10	5	180	48	91	20	45	27	18	45	55	37	91
C10x10	10	180	27	33	4	13	6	4	25	14	15	33
C20x20	5	180	25	115	13	68	19	36	24	82	21	115
C20x20	10	180	52	43	13	12	22	3	47	15	41	43
C20x20	20	180	36	4	2	1	7	0	31	1	22	4
C30x30	5	180	24	122	16	56	23	30	24	84	24	122
C30x30	10	180	31	50	6	5	15	5	28	22	25	50
C30x30	20	180	37	15	0	1	10	1	34	6	28	15
C30x30	30	180	16	6	0	0	1	1	14	2	10	6
D5x5	10	90	44	45	37	30	37	29	44	44	44	44
D7x7	10	90	51	19	23	6	31	6	51	17	47	18
D10x10	10	90	41	4	0	1	11	0	40	4	12	3
D10x10	45	90	4	0	0	0	0	0	4	0	0	0
D20x20	10	90	14	0	0	0	0	0	14	0	0	0
-	-	2070	450	547	134	238	209	133	425	346	326	544

NDPVC is better than any feasible solution to the kHSNDP (on the same instance) if its costs are smaller than the lower bound obtained from solving the kHSNDP. Equality can, clearly, be only shown if optimal solutions to both variants could be computed. Thus, a quite significant part of all considered test instances remain undecided.

From Table 10 we conclude that the solutions of the NDPVC than the corresponding ones of the kHSNDP in a significant number of cases for grid instances from both considered sets C and D. Clearly this trend is more pronounced for those sets for which more instances could be solved to proven optimality. This is particularly true for instance set D where more interdependencies between the various commodities are present and one therefore expect more complex solutions graphs. The new problem considered in this article allows to find better solutions on more than 50% of the comparably small instances from sets D5x5, D7x7, and D10x10 with only ten commodities (and equality could be proven for only very few of those instances). Thus one can safely expect that this effect will be increased for larger instances with more commodities. Clearly, the times needed for solving instances of the more complex NDPVC are significantly higher than those for the kHSNDP. To analyze this in more detail, however, further problem formulations capable of solving larger instances or decomposition based algorithms for the present formulations seem necessary. However, the potential savings in solving the NDPVC instead of the kHSDNP may justify this additional effort.

For instances of set C, the results in Table 11 indicate that the solutions of the NDPVC are often cheaper than the corresponding ones of the kHSNDP in particular when H' is strictly larger than H. A slightly different behavior can be observed for instance set D where – with the exception of $H = 0$ – this effect is not so pronounced.

The results summarized in Table 12 show that solving the NDPVC instead of solving the kHSNDP may yield cheaper solutions on (random) Euclidean instances as well. These results also indicate that the number of cases where the optimal solutions of the two problems differ seems slightly larger for the subclass of Euclidean instances. In addition, the results from Table 13 indicate that the difference between the two problems is pronounced when the hop limits are tight.

Overall, the large number cheaper solutions clearly justifies studying the NDPVC also from a practical perspective.

Table 11: Overall number of cases from instance sets C and D in which solution to NDPVC is provable better (bt) and provably equal (eq) compared to the kHSNDP and corresponding values obtained for the different formulations considered grouped by the considered values of $(H, H') = (H_{\min} + \Delta_H, H_{\min} + \Delta_H + \Delta_{H'})$. Remaining cases could not be decided.

			Overall		H_1		H_2^S		H_2^{AS}		H_3	
Set	$(\Delta_H, \Delta_{H'})$	#	bt	eq	bt	eq	bt	eq	bt	eq	bt	eq
C	(0,0)	180	14	115	3	65	9	50	12	83	13	115
	(0,1)	180	86	53	37	26	51	12	82	29	78	53
	(0,2)	180	89	42	28	21	48	11	85	28	78	42
	(1,0)	180	5	70	0	32	0	13	5	44	3	70
	(1,1)	180	32	58	1	21	4	7	27	33	16	58
	(1,2)	180	42	44	5	11	13	3	38	20	22	44
	(2,0)	180	3	40	0	12	0	2	2	20	2	40
	(2,1)	180	12	35	0	7	1	0	11	12	6	35
	(2,2)	180	13	22	0	6	4	0	10	12	5	22
D	(0,0)	50	6	16	3	14	4	13	5	16	5	15
	(0,1)	50	34	0	19	0	26	0	34	0	25	0
	(0,2)	50	30	2	14	2	17	2	30	2	22	2
	(1,0)	50	18	9	6	7	9	7	18	9	13	9
	(1,1)	50	22	4	9	2	12	2	22	4	15	4
	(1,2)	50	21	5	7	3	7	3	21	5	14	5
	(2,0)	50	5	12	1	3	2	3	5	11	3	11
	(2,1)	50	11	11	0	4	0	3	11	10	3	10
	(2,2)	50	7	9	1	2	2	2	7	8	3	9
-	-	2070	450	547	134	238	209	133	425	346	326	544

Table 12: Overall number of cases from instance sets EU and RE in which solution to NDPVC is provable better (bt) and provably equal (eq) compared to the kHSNDP and corresponding values obtained for the different formulations considered grouped for each considered instance set. Remaining cases could not be decided.

					Overall		H_1		H_2^S		H_2^{AS}		H_3	
Set	$\|V\|$	$\|E\|$	$\|\mathcal{R}\|$	#	bt	eq	bt	eq	bt	eq	bt	eq	bt	eq
EU	50	122	10	45	7	12	6	2	5	1	7	12	7	8
	50	122	45	45	2	0	0	0	0	0	2	0	0	0
	50	245	10	45	17	3	6	1	7	1	16	2	15	3
	50	245	45	45	3	0	0	0	0	0	3	0	0	0
	75	277	10	45	12	2	3	0	3	0	11	2	11	1
	75	277	45	45	2	0	0	0	0	0	2	0	1	0
	75	555	10	45	6	4	2	1	3	1	6	2	6	4
	75	555	45	45	0	0	0	0	0	0	0	0	0	0
	100	495	10	45	11	1	2	1	4	1	10	1	9	1
	100	495	45	45	1	0	0	0	0	0	1	0	0	0
	100	990	10	45	6	4	1	0	3	1	3	1	4	4
	100	990	45	45	1	0	0	0	0	0	1	0	1	0
RE	50	122	10	45	8	22	4	12	3	10	7	21	8	21
	50	122	45	45	6	3	3	1	3	1	6	3	5	2
	50	245	10	45	6	27	3	13	3	13	5	22	6	27
	50	245	45	45	6	6	0	0	2	0	6	2	4	6
	75	277	10	45	8	18	5	8	4	6	8	14	7	18
	75	277	45	45	2	2	0	0	0	0	2	1	1	1
	75	555	10	45	6	18	1	11	3	12	6	13	4	18
	75	555	45	45	3	6	0	2	0	2	3	2	1	6
	100	495	10	45	4	17	3	7	4	4	4	12	4	17
	100	495	45	45	3	0	0	0	0	0	3	0	0	0
	100	990	10	45	3	29	1	14	3	14	3	19	3	29
	100	990	45	45	1	3	0	0	0	0	0	0	1	3
-	-	-	-	1080	124	177	40	73	50	67	115	129	98	169

Table 13: Overall number of cases from instance sets EU and RE in which solution to NDPVC is provable better (bt) and provably equal (eq) compared to the kHSNDP and corresponding values obtained for the different formulations considered grouped by the considered values of $(H, H') = (H_{\min} + \Delta_H, H_{\min} + \Delta_H + \Delta_{H'})$. Remaining cases could not be decided.

			Overall		H_1		H_2^S		H_2^{AS}		H_3	
Set	$(\Delta_H, \Delta_{H'})$	#	bt	eq	bt	eq	bt	eq	bt	eq	bt	eq
EU	(0,0)	60	2	8	0	3	1	4	2	6	2	8
	(0,1)	60	25	3	10	0	12	0	23	2	21	2
	(0,2)	60	23	2	9	1	9	0	22	1	18	2
	(1,0)	60	7	2	0	1	1	1	7	1	6	2
	(1,1)	60	6	3	1	0	2	0	4	2	4	3
	(1,2)	60	5	4	0	0	0	0	4	4	3	3
	(2,0)	60	0	2	0	0	0	0	0	2	0	1
	(2,1)	60	0	1	0	0	0	0	0	1	0	0
	(2,2)	60	0	1	0	0	0	0	0	1	0	0
RE	(0,0)	60	0	25	0	15	0	14	0	16	0	25
	(0,1)	60	17	26	9	18	10	18	16	22	15	26
	(0,2)	60	18	26	8	18	8	17	18	25	12	25
	(1,0)	60	2	27	1	9	0	7	1	18	2	27
	(1,1)	60	8	18	1	3	3	2	8	12	7	18
	(1,2)	60	6	16	0	3	3	2	6	11	5	14
	(2,0)	60	2	6	1	0	1	0	2	2	1	6
	(2,1)	60	2	4	0	1	0	1	1	2	1	4
	(2,2)	60	1	3	0	1	0	1	1	1	1	3
-	-	1080	124	177	40	73	50	67	115	129	98	169

5 Node-disjoint case

In this section, we focus on the node-disjoint variant of the NDPVC whose definition is obtained from the NDPVC by ensuring that each solution must contain a path of length at most H'_{st} after removing a subset of nodes W of cardinality $k-1$ and all edges incident to nodes in W for each pair $\{s,t\} \in \mathcal{R}$ such that $s, t \notin W$. The following Mengerian results for node disjoint paths with hop constraints show that the resulting problem is different from the node-disjoint variant of the related kHSNDP. As for the edge disjoint case, the first result is known from the literature [2, 4] and the second, although new can be proved in a similar manner.

Result 3 ("Mengerian-like result for node-disjoint hop-constrained paths", Exoo [4], Bermond et al. [2]). *Let i and j be two distinct nodes of a given graph $G = (V, E)$, such that the length of the shortest (i, j)-path is H, $H \leq 4$ or $H \geq |V| - 1$. Then, the minimal number of nodes (other than i and j) that need to be removed from G in order to increase the length of the shortest (i, j)-path, is equal to the maximum number of pairwise node-disjoint paths from i to j of length at most H.*

Result 4 ("Mengerian-like result for node-disjoint hop-constrained paths with $H < H'$"). *Let i and j be two distinct nodes of a given graph $G = (V, E)$ that contains at least one path from i to j of length at most H. Then, the minimal number k of nodes (other than i and j) after whose removal G does not contain a path from i to j of length at most H', $H' > H$, is equal to the maximum number of pairwise node-disjoint paths from i to j of length at most H' (where one of them is of length at most H) if $H \geq |V| - 1$ or $H \leq 2$ and either $k = 2$ or $H' \leq 4$ or $H' \geq |V| - 1$).*

It is easy to find examples showing that the cost differences between the node-disjoint variants of solutions to the NDPVC and the kHSNDP can be arbitrary large and that there exist instances that are feasible for the former but infeasible for the latter problem. One such example is given in Figure 8 which provides a feasible solution of the node-disjoint variant of NDPVC for $\mathcal{R} = \{\{1, 4\}\}$, $H_{14} = 3$, and $H'_{14} = 4$. Since there does not exist a path that is node-disjoint from $P = (1, 2, 3, 4)$ which is the unique $(1, 4)$-path of length $\leq H_{14}$, the graph does not contain a feasible solution for the node-disjoint variant of the kHSNDP.

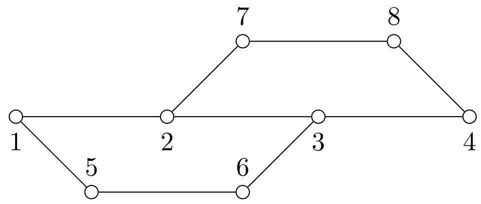


Figure 8: A feasible solution for the node disjoint variant of NDPVC with $k = 2$, $\mathcal{R} = \{\{1, 4\}\}$, $H_{14} = 3$, $H'_{14} = 4$, that does not contain two node disjoint paths of length 3 and 4 between nodes 1 and 4, respectively.

We observe that the generic formulation given in Section 3.1 can be adapted to the node-disjoint case by considering an appropriately modified set of feasible backup edges. Formally these sets are described by incidence vectors $\mathbf{x}$ in

$$\mathcal{B}_{st} = \{\mathbf{x} \in \{0,1\}^{|E|} \mid \forall W \subset V,\ |W| = k-1,\ W \cap \{s,t\} = \emptyset, \\ \exists\ (s,t)\text{-path } P \text{ in } E(\mathbf{x}) \setminus \delta(W) \text{ s.t. } |P| \le H'_{st}\}.$$

Thus, the following three characterizations that correspond to those of the edge-disjoint case can be used to describe set $\mathcal{B}_{st}$.

Characterization 4. *Let $P = \{e_1, e_2, \ldots, e_l\} \subset E_{st}$ be the primary path of length l, $l \le H_{st}$, for a given commodity $\{s,t\} \in \mathcal{R}$ such that $e_i = \{u_{i-1}, u_i\}$, $u_i \in V_{st}$, $1 \le i \le l$, $u_0 = s$, and $u_l = t$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that:*

$$\forall u_i \in \{u_1, \ldots, u_{l-1}\}, \quad \exists\ (s,t)\text{-path } P'[u_i] \subset E'_{st} \setminus \delta(u_i),\ \text{s.t. } |P'[u_i]| \le H'_{st}.$$

Thereby, $H_{st} \le H_{st}$ and $\hat{E} = \left(\bigcup_{i=1}^{l-1} P'[u_i]\right) \setminus P$.

Characterization 5. *Let $P = \{e_1, e_2, \ldots, e_l\} \subset E_{st}$ be the primary path of length l, $l \le H_{st}$, for a given commodity $\{s,t\} \in \mathcal{R}$ such that $e_i = \{u_{i-1}, u_i\}$, $u_i \in V_{st}$, $1 \le i \le l$, $u_0 = s$, and $u_l = t$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that there exist $l-1$ additional (s,t)-paths $P'_i \subset E'_{st}$, $i = 1, 2, \ldots, l-1$ of length at most H'_{st}, $H'_{st} \ge H_{st}$, such that at most $l-2$ of them contain the same node from $P \setminus \{s,t\}$, i.e.:*

$$\exists\ (s,t)\text{-paths } P'_i \subset E'_{st},\ i = 1, 2, \ldots, l-1,\ \text{s.t. } \forall v \in V_{st} : \sum_{i=1}^{l-1} |P'_i \cap \delta(v)| \le 2(l-2).$$

Thereby, $H_{st} \le H_{st}$ and $\hat{E} = \left(\bigcup_{i=1}^{l-1} P'_i\right) \setminus P$.

Notice that the validity of the latter claim follows from the fact that degree of any internal node v in an (s,t)-path is two.

Characterization 6. *Let $P = \{e_1, e_2, \ldots, e_l\} \subset E_{st}$ be the primary path of length l, $l \le H_{st}$, for a given commodity $\{s,t\} \in \mathcal{R}$ such that $e_i = \{u_{i-1}, u_i\}$, $u_i \in V_{st}$, $1 \le i \le l$, $u_0 = s$, and $u_l = t$. Then, a valid set of backup edges $\hat{E}$ is established by ensuring that:*

$$\exists\ (s,t)\text{-paths } P'_i \subset E'_{st},\ i = 1, 2, \ldots, l-1,\ \text{s.t. } |P'_i| \le H'_{st} \text{ and } P'_i \cap \delta(u_i) = \emptyset.$$

Thereby, $\hat{E} = \left(\bigcup_{i=1}^{l-1} P'_i\right) \setminus P$.

Based on these characterizations it is not too difficult to adapt the formulations given in Section 3.3 to the node-disjoint case and we therefore skip the details. Notice, however, that in the node-disjoint case a formulation based on Characterization 4 only needs to establish $O(|V|)$ additional backup flows for each commodity while $O(|E|)$ backup flows are necessary in the edge-disjoint case. Thus, a formulation based on the straightforward Characterization 4 would be computationally more attractive to use than the one for the edge-disjoint case based on Characterization 1. Even though, our preliminary computational experiments did, however, indicate that formulations based on the less obvious Characterizations 5 and 6 outperform the former also for the node-disjoint case.

6 Conclusions

This article deals with the design of survivable networks in which hop constraints ensure a certain maximum distance between each commodity pair before and after each considered set of failures. Motivated from previous studies related to vulnerability of graphs we show that a previously considered optimization problem (the kHSNDP) that aims to cover the same kind of scenarios is in fact too conservative in the sense that it is overly constrained. Thus, we introduce the Network Design Problem with Vulnerability Constraints (NDPVC) which overcomes this shortage. We show that optimal solutions to the NDPVC are at least as good than those of the kHSNDP, can be better for almost all relevant hop limits, and that such solutions may exist even for instances that are infeasible for the kHSNDP. We have introduced three graph theoretical characterizations of feasible backup systems and proposed integer programming models for each of them. In a computational study, we have shown that two formulations based on the the two non-obvious characterizations clearly outperform the one based on the first, and obvious characterization. Our computational results also confirm that the solutions of the NDPVC are cheaper than those of the kHSNDP in many cases.

This study also suggests multiple directions for future research. Observe that the three graph theoretic characterizations provided in Section 3.2 can in principle be extended to the case of multiple edge failures by appropriately considering all subsets of relevant primary edges of the appropriate size. Such straightforward extensions will, however, lead to huge formulations at least when following the ideas presented in the current work. Thus, the study of further characterizations for the case of multiple edge failures that lead to reasonably sized formulations would be a worthwhile goal for future research. For the case of single edge failures it would be interesting to derive more sophisticated solution techniques such as Benders decomposition algorithms based on the formulations introduced in this article in order to tackle larger problem instances. The linear programming relaxation gaps reported in our computational study also show the need to identify further valid inequalities in order to reduce the number of branch-and-bound nodes that need to be enumerated for identifying optimal solutions. The performance of such approaches can be further improved by (primal) heuristics. The development of heuristics for the NDPVC seems, however, significantly harder than for the kHSNDP and might be an interesting topic for future research. Finally, from a practical perspective it would be relevant to study even more complex variants that are defined on directed input graphs or additionally consider edge capacities and demands associated to commodities.

7 Acknowledgments

We thank Amaro de Sousa for helpful discussions on the motivation for this paper, in particular with respect to applications of hop constraints in current telecommunications networks. We would also like to thank the referees whose suggestions have helped to considerably improve the paper.

This work is supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the project UID/MAT/04561/2013, by the Austrian Science Fund (FWF), under grant I892-N23, and by the Vienna Science and Technology Fund (WWTF) through project ICT15-014. Part of this research has been performed while M. Leitner was a research fellow at the Department of Computer Science, Université Libre de Bruxelles (Brussels, Belgium) where he was supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. These supports are greatly acknowledged.

References

[1] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40(7):846–867, 1994.

[2] J.C. Bermond, J. Bond, M. Paoli, and C. Peyrat. Graphs and interconnection networks: diameter and vulnerability. In *Surveys in combinatorics*, volume 82, pages 1–30. Cambridge Univ. Press, 1983.

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

[4] G. Exoo. On a measure of communication network vulnerability. *Networks*, 12:405–409, 1982.

[5] L. Gouveia. Using Variable Redefinition for Computing Lower Bounds for Minimum Spanning and Steiner Trees with Hop Constraints. *INFORMS Journal on Computing*, 10(2):180–188, 1998.

[6] L. Gouveia, P. Patricio, and A. de Sousa. Compact models for hop-constrained node survivable network design: An application to MPLS. In *Telecommunications Planning: Innovations in Pricing, Network Design and Management*, volume 33 of *Operations Research/Computer Science Interfaces Series*, pages 167–180. Springer, 2006.

[7] L. Gouveia, P. Patricio, and A. de Sousa. Lexicographical minimization of routing hops in hop-constrained node survivable networks. *Telecommunication Systems*, 62:417–434, 2016.

[8] M. Grötschel, C.L. Monma, and M. Stoer. Design of survivable networks. In M.O. Ball et al., editors, *Handbooks in OR & MS*, volume 7, chapter 10, pages 617–672. Elsevier, 1995.

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

[10] J. Klincewicz. Optimization issues in quality of service. In M.G.C. Resende and P.M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, pages 435–458. Springer, 2006.

[11] I. Ljubić. A branch-and-cut-and-price algorithm for vertex biconnectivity augmentation. *Networks*, 56(3):169–182, 2010.

[12] A. R. Mahjoub, L. Simonetti, and E. Uchoa. Hop-level flow formulation for the survivable network design with hop constraints problem. *Networks*, 61:171–179, 2013.

[13] K. Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[14] L. Roychoudhuri, E. Al-Shaer, and G. Brewster. On the impact of loss and delay variation on internet packet audio transmission. *Computer Communications*, 29:1578–1589, 2006.

[15] M. Scharf, M. Necker, and B Gloss. The sensitivity of TCP to sudden delay variations in mobile networks. In N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, and L. Merakos, editors, *Networking 2004*, volume 3042 of *LNCS*, pages 76–87, 2004.

[16] W. Sheikh and A. Ghafoor. Jitter-minimized reliability-maximized management of networks. *International Journal of Network Management*, 21:185–222, 2011.

[17] B. Thiongane, J. Cordeau, and B. Gendron. Formulations for the nonbifurcated hop-constrained multicommodity capacitated fixed-charge network design problem. *Computers & Operations Research*, 53:1–8, 2015.