Strong Formulations for 2-Node-Connected Steiner Network Problems (TR)

Markus Chimani[1], Maria Kandyba[1⋆], Ivana Ljubić[2⋆⋆], and Petra Mutzel[1]

[1] Faculty of Computer Science, Technical University of Dortmund
{markus.chimani,maria.kandyba,petra.mutzel}@cs.uni-dortmund.de
[2] Faculty of Business, Economics and Statistics, University of Vienna
ivana.ljubic@univie.ac.at

Technical Report TR07-1-008
November 2007

Chair XI – Algorithm Engineering
Faculty of Computer Science
Technical University of Dortmund

Abstract. We consider a survivable network design problem known as the *2-Node-Connected Steiner Network Problem* (2NCON): we are given a weighted undirected graph with a node partition into two sets of customer nodes and one set of Steiner nodes. We ask for the minimum weight connected subgraph containing all customer nodes, in which the nodes of the second customer set are nodewise 2-connected. This problem class has received lively attention in the past, especially with regard to exact ILP formulations and their polyhedral properties.
In this paper, we present a transformation of this problem into a related problem considering directed graphs and use this to establish two novel ILP formulations to solve 2NCON, based on multi-commodity flow and on directed cuts, respectively. We prove the advantages of our formulations and compare both approaches theoretically as well as experimentally. Thereby we solve instances with up to 1600 nodes to provable optimality.

1 Introduction

Various survivable network design problems occur prominently in many real-world fiber-optic networks and telecommunication applications, see, e.g., [11, 21] for general surveys. We concentrate on the following NP-hard problem class:

Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}^+$ and a vector of connectivity requirements $\varrho \in \{0, 1, \ldots, k\}^{|V|}$ for some constant $k > 0$. A solution of the *k-Node-Connected Steiner Network* problem (kNCON)[3] [19] is a subgraph $N =$

⋆ Supported by the German Research Foundation (DFG) through the Collaborative Research Center "Computational Intelligence" (SFB 531)
⋆⋆ Supported by the Hertha-Firnberg Fellowship of the Austrian Science Foundation (FWF)
[3] In the literature there are various names for this problem, with sometimes slightly differing definitions. kNCON is also known as {0,...,k}-(N)SND (Node Survivable Network Design) and Generalized Steiner Network problem.

(V_N, E_N) of G which contains all nodes $v \in V$ with $\varrho_v > 0$, minimizes $\sum_{e \in E_N} c_e$ and satisfies the following connectivity property: for every pair of nodes $s, t \in V_N$, N contains $\varrho_{st} := \min\{\varrho_s, \varrho_t\}$ node-disjoint paths connecting them. We can relax the problem by replacing the node-disjointness with edge-disjointness, and obtain the *k-Edge-Connected Steiner Network Problem* (kECON). For simplicity, we define $\mathcal{R}_i := \{v \in V \mid \varrho_v = i\}$ for all $0 \leq i \leq k$, and call the set $\mathcal{R} := \bigcup_{i>0} \mathcal{R}_i$ the *customer nodes*. We can assume that $|\mathcal{R}_2| \geq 2$, since otherwise we obtain the traditional Steiner tree problem.

A lot of research has been conducted on this problem, both in the fields of effective heuristics and approximation algorithms, see [11] for an overview. However, these are beyond the scope of this paper, as we will concentrate on the exact ILP formulations. This is based to the fact that recent advances in computational power and ILP solvers, when used in conjunction with strong models, allow to solve real-world instances for other network problems to provable optimality within reasonable time bounds; see, e.g., [1, 3, 16]. Furthermore, ILP formulations also often form the basis of approximation schemata.

For kECON and kNCON, Grötschel, Monma and Stoer [7] described cut-based integer linear programs (ILP). Apart from such formulations, the problem can also be formulated in terms of multi-commodity flow, as done by Raghavan [17].

Regarding 2ECON, it has been shown that for both concepts formulations based on *oriented* graphs are stronger than undirected formulations: an *orientation* of an undirected graph G is a directed graph G', which is obtained by transforming each edge of G into a directed arc. Robbins [18] showed that a graph G is 2-edge-connected if and only if there exists an orientation G' with directed paths $(v \to w)$ and $(w \to v)$ for every pair of nodes v, w. This fact has been exploited by Chopra [5] for solving 2ECON via directed graphs, who proved his formulation to be polytope-wise superior to the undirected formulation mentioned above. Goemans [6] and Stoer [19, pp. 31–32] extended this formulation to kECON for the case that all connectivity requirements are 0, 1 or even; later Magnanti and Raghavan [15] extended it for general k.

It has been an open problem [17, p. 183],[19, pp. 32,134] whether a similar orientation technique can be used for kNCON-type problems. Recently, Chimani, Kandyba, and Mutzel [4] showed that this is indeed possible for the *2-Root-Connected Steiner Network Problem* (2RSN) and its prize-collecting variant (2RPCSN), where the node-wise 2-connectedness is required only with a special root node $r \in V$; i.e., each node $v \in \mathcal{R}$ has to have ϱ_v node-disjoint paths with r. A certain orientability property for the feasible networks of this problem was shown, and was used to obtain a novel ILP formulation based on directed cuts for this problem.

In this paper, we present how the more prominent 2NCON problem can be transformed in a variant of 2RSN (Section 2) and derive two new ILP formulations based on different concepts: one based on multi-commodity flow (Section 3) and one based on directed cuts, only requiring easily separable constraints (Section 4). We show the

theoretical advantages of both new formulations compared to the previously known ILPs. Furthermore, we prove that our approaches are equivalent from the polyhedral point of view (Section 5). Nonetheless, the cut formulation has certain advantages in practice, as we show in an experimental study (Section 6).

2 Directed 2NCON

The main problem with node-disjointness is that Robbins' theorem can only be exploited for 2ECON. Furthermore, it is in general not possible to orient an undirected 2-connected graph such that for every pair of nodes $v, w \in \mathcal{R}_2$, there are two node-disjoint directed paths, one from v to w and one from w to v. Up until now, this was the main hindrance why there were no orientation-based formulations for 2NCON.

We require the following theorem, shown in [4], as a foundation to show the validity all the formulations below. We rephrase it such that it is more useful in the following:

Theorem 1 ([4]). *Let $G = (V, E)$ be a graph with some root node $r \in V$ and the property that each non-trivial (i.e., larger than a single edge) 2-connected component contains r. Then there exists an orientation G' such that:*
- *For each node $v \in V \setminus \{r\}$ which does not share a common non-trivial 2-connected component (block) with r, G' contains a directed path $(r \to v)$.*
- *For each node $v \in V \setminus \{r\}$ which shares a common non-trivial block with r, G' contains a directed path $(r \to v)$ and a directed path $(v \to r)$, which are node-disjoint except for r and v.*

The proof of this theorem allows an insight which will be of particular use for our problem:

Corollary 1. *Let the graph G be defined as in Theorem 1, and let there be only a single non-trivial block. Then there exist two orientations G' and G'' with the properties of Theorem 1 and:*
- *Only one single arc in G' is directed towards r.*
- *Only one single arc in G'' is directed outwards from r.*

Proof. The idea of the proof for Theorem 1 works as follows: we first direct all the edges of the trivial blocks from the node nearer to r to the node farther away. We then consider the non-trivial blocks separately and orient them as follows: We start with a directed cycle through r and label the contained nodes in an increasing fashion. Thereby, the root r obtains the smallest label. Hence there is only a single edge $\hat{e}$ which is directed from a larger label number to the smaller number, i.e., the root. Then the proof proceeds by incrementally choosing a non-oriented path between two labeled nodes: we orient these paths and label their nodes uniquely in such a way that the property of increasing labels remains true. Hence we require only a single edge $\hat{e}$ per block to be directed towards r. If G only contains a single non-trivial block, we

get the corollary's claim for G'. By inverting all orientations in this block, we obtain G''. □

Based on these results we can therefore introduce a directed variant of 2NCON and show that it is equivalent to the original 2NCON problem.

Definition 1 (D2NCON). Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}^+$, a vector of connectivity requirements $\varrho \in \{0, 1, 2\}^{|V|}$, and a node $r \in V$ with $\varrho_r = 2$. The *Directed 2-Node-Connected Steiner Network Problem* (D2NCON) is to find a subgraph $N = (V_N, E_N)$ of G and an orientation $N' = (V_N, A_N)$ of N which minimize $\sum_{e \in E_N} c_e$ under the restriction that N' may only contain a single edge oriented towards r and the following connectivity properties are satisfied for each node $v \in V \setminus \{r\}$:

- If $\varrho_v = 1$, N' contains a directed path $(r \to v)$.
- If $\varrho_v = 2$, N' contains a directed path $(r \to v)$ and a directed path $(v \to r)$, which are node-disjoint except for r and v.

Theorem 2. *Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}^+$, and a vector of connectivity requirements $\varrho \in \{0, 1, 2\}^{|V|}$. Choose $r \in V$ with $\varrho_r = 2$ arbitrarily. For this input, any solution of D2NCON can be transformed into an equivalent solution of 2NCON with the same objective value, and vice versa.*

Proof. D2NCON $\Longrightarrow$ 2NCON. Let N' be an optimal orientation for D2NCON. Assume that there are at least two non-trivial blocks B_1 and B_2 containing the nodes $v_1, v_2 \in \mathcal{R}_2$, respectively. Then r has to be a cut vertex contained in both blocks. But since a valid orientation requires directed paths $(v_1 \to r)$ and $(v_2 \to r)$ there have to be at least two edges being directed towards r, which is a contradiction to the feasibility of the solution. Hence we know that N' will only contain a single non-trivial block, and therefore the corresponding subgraph N clearly is a feasible solution for 2NCON.

2NCON $\Longrightarrow$ D2NCON. Let N be a feasible solution for 2NCON. We show that for every choice of r there is an orientation N' which is feasible for D2NCON: any solution of 2NCON has the property that all $\mathcal{R}_2$ nodes lie in a common block. Hence we can choose any of these nodes as our root r and apply Theorem 1 and Corollary 1, which guarantees the existence of an orientation required by D2NCON. □

Hence, we can find formulations for the D2NCON problem, and by solving D2NCON we automatically solve 2NCON as well. In the following sections, the problem input will always be defined as given above; the root node $r \in \mathcal{R}_2$ is chosen arbitrarily. Let $\mathcal{R}'_i := \mathcal{R}_i \setminus \{r\}$, for $0 \le i \le 2$, and $\mathcal{R}' := \mathcal{R} \setminus \{r\}$. Furthermore, let $\bar{G} = (V, A)$ be the bidirected graph obtained from G by replacing every undirected edge $\{u, v\} \in E$ by two directed arcs $(u, v), (v, u) \in A$ with costs $c_{uv} = c_{vu} = c_{\{u,v\}}$.

Remark. One may try to model node-connectivity by only computing edge-connectivity in a modified underlying graph, by replacing each node by a directed arc. This is not valid in our case as the orientability theorems require bidirectedness of the underlying graphs.

3 Multi-Commodity Flow for D2NCON

We start with presenting a novel ILP formulation (DFlow) based on multi-commodity flow for D2NCON, and therefore for 2NCON. As there has been much research on flow-based formulations for the latter problem, we compare our formulation to the currently strongest one, by Raghavan [17, pp. 180–181][4], and show that our formulation is theoretically beneficial.

The idea is to consider $\bar{G}$ and send exactly one unit of flow from the root to each $\mathcal{R}'$ node. Furthermore, we send one unit of flow from each $\mathcal{R}'_2$ customer back to the root. Thereby it has to be ensured that the pairs of forward- and backward-flows do not use common nodes and edges except for v and r. We define the set of commodities $\mathcal{C} = \{(r,v) \mid v \in \mathcal{R}'\} \cup \{(v,r) \mid v \in \mathcal{R}'_2\}$; a flow of commodity $\chi \in \mathcal{C}$ on the arc $(i,j) \in A$ is modeled by the variable f_{ij}^{χ}. Finally, we introduce the variables x_{ij} which are 1, if the solution network contains the arc $(i,j) \in A$.

$$\text{DFlow}: \qquad \min \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} \tag{1}$$

$$\sum_{(i,v)\in A} f_{iv}^{\chi} - \sum_{(v,i)\in A} f_{vi}^{\chi} = \begin{cases} -1, \text{ if } v = s \\ 1, \text{ if } v = t \\ 0, \text{ else} \end{cases} \qquad \forall \chi = (s,t) \in \mathcal{C}, \forall v \in V \tag{2}$$

$$\sum_{(i,w)\in A} \left(f_{iw}^{(v,r)} + f_{iw}^{(r,v)} \right) \leq 1 \qquad \forall v \in \mathcal{R}'_2, \forall w \in V \setminus \{r,v\} \tag{3}$$

$$0 \leq f_{ij}^{\chi} \leq x_{ij} \qquad \forall (i,j) \in A, \forall \chi \in \mathcal{C} \tag{4}$$

$$x_{vw} + x_{wv} \leq 1 \qquad \forall \{v,w\} \in E \tag{5}$$

$$\sum_{(i,r)\in A} x_{ir} = 1 \tag{6}$$

$$x_{vw} \in \{0,1\} \qquad \forall (v,w) \in A \tag{7}$$

The *flow-conservation constraints* (2) define the sent flow and ensure the flow balances, while the *coupling constraints* (3) ensure the node-disjointness of the pairs of forward and backward flow, by guaranteeing that at most one unit of flow per commodity pair is sent into any node. The inequalities (4) ensure the flow capacities and bind the flow

[4] Various aspects of the considered problems were studied in papers by Stoer and Raghavan (with coauthors) [7–9, 15]. For simplicity and common notations we reference their theses [17, 19], including page numbers when suitable.

variables f to the network-defining variables x. For the latter, (5) ensures that we have a unique orientation for the selected edges, and (6) requires that only one arc is selected which is oriented towards the root r.

Theorem 3. *An optimal solution for* DFlow *gives an optimal solution for the corresponding 2NCON problem.*

Proof. Clearly, the flow-constraints guarantee the existence of the directed paths from r to all customer nodes, and for each $\mathcal{R}'_2$ customer we also have a directed path backwards to r. Due to Robbins' theorem [18] this guarantees that every $\mathcal{R}'_2$ customer belongs to the same edge-biconnected component as r. Corollary 1, Theorem 2, (3) and (6) guarantee that this component is 2-node-connected. □

Note that if we consider this ILP without the coupling constraints, all resulting feasible solutions for x would induce feasible solutions for the 2ECON problem and vice versa. In fact, the accordingly reduced ILP is equivalent to Raghavan's ILP for 2ECON [17, pp. 159–161,188]. Raghavan also presented a formulation for the 2NCON problem by computing two multi-commodity flows g and h simultaneously [17, p. 180–181]: g represents directed flow for the induced 2ECON problem, h represents an non-oriented flow with node-disjointness constraints.[5] The two flows are bound to each other only by their common use of the z_e variables, for $e \in E$, which define whether the given undirected edge e is contained in the solution network or not. For notational simplicity we can write both z_{vw} and z_{wv} for $e = \{v, w\}$. We denote Raghavan's formulation as MFlow (mixed flow):

$$\text{MFlow}: \qquad \min \sum_{e \in E} c_e \cdot z_e \tag{8}$$

$$\sum_{(i,v)\in A} g^{\chi}_{iv} - \sum_{(v,i)\in A} g^{\chi}_{vi} = \begin{cases} -1, \text{ if } v = s \\ 1, \text{ if } v = t \\ 0, \text{ else} \end{cases} \quad \forall \chi = (s,t) \in \mathcal{C}, \forall v \in V \tag{9}$$

$$g^{\chi}_{vw} + g^{\chi'}_{wv} \leq z_{vw} \qquad \forall \{v,w\} \in E, \forall \chi, \chi' \in \mathcal{C} \tag{10}$$

$$g^{\chi}_{vw} \geq 0 \qquad \forall (v,w) \in A, \forall \chi \in \mathcal{C} \tag{11}$$

$$\sum_{(i,v)\in A} h^{\chi}_{iv} - \sum_{(v,i)\in A} h^{\chi}_{vi} = \begin{cases} -2, \text{ if } v = s \\ 2, \text{ if } v = t \\ 0, \text{ else} \end{cases} \quad \forall \chi = (s,t) \in \mathcal{D}, \forall v \in V \tag{12}$$

$$0 \leq h^{\chi}_{vw} \leq z_{vw} \qquad \forall (v,w) \in A, \forall \chi \in \mathcal{D} \tag{13}$$

$$\textstyle\sum_{(v,i)\in A} h^{\chi}_{vi} \leq 1 \qquad \forall \chi = (s,t) \in \mathcal{D}, v \in V \setminus \{s,t\} \tag{14}$$

$$z_e \in \{0,1\} \qquad \forall e \in E \tag{15}$$

[5] Note that this formulation has been developed for general k, where it is called *improved undirected flow formulation with node-disjointness constraints.*

Thereby, we consider an arbitrary ordering $\langle v_1, v_2, v_3, \ldots \rangle$ of the nodes of $\mathcal{R}_2$ and define $v_0 := v_{|\mathcal{R}_2|}$. We then obtain the "cyclic" commodity set $\mathcal{D} := \{(v_i, v_{i+1}) \mid 0 \leq i < |\mathcal{R}_2|\}$ [17, pp. 89–92].

Let $\mathcal{P}_{DF}$ and $\mathcal{P}_{MF}$ be the polyhedra of the feasible solutions of the LP relaxations for DFlow and MFlow, respectively. We then consider their projections into the space of z variables, i.e., $\text{proj}_z(\mathcal{P}_{DF}) = \{z \in [0,1]^{|E|} \mid (x,f) \in \mathcal{P}_{DF},\ z_{ij} = x_{ij} + x_{ji}\ \forall \{i,j\} \in E\}$ and $\text{proj}_z(\mathcal{P}_{MF}) = \{z \in [0,1]^{|E|} \mid (z,g,h) \in \mathcal{P}_{MF}\}$. We also consider extended projections including the flow variables $f \in [0,1]^{|A|\cdot|\mathcal{C}|}$, i.e., variables not in the objective function. Let $\text{proj}_{z,f}(\mathcal{P}_{DF}) = \{(z,f) \mid (x,f) \in \mathcal{P}_{DF}, z_e = x_{ij} + x_{ji}\ \forall e = \{i,j\} \in E\}$ be the projection of $\mathcal{P}_{DC}$ into the variable space of z and retaining the flow f. Let $\text{proj}_{z,f}(\mathcal{P}_{MF}) = \{(z,f) \mid (z,g,h) \in \mathcal{P}_{MF}, f = g\}$ be the projection of $\mathcal{P}_{MF}$ ignoring the h flow. In other words, we identify the flows f and g.

We show that the lower bounds obtained by the LP relaxations of our new formulation are at least as tight as those of the mixed flow formulation. Therefore note that the flow f is a kind of natural fusion of the flow g and the node-disjointness properties of h.

Theorem 4. *The* DFlow *formulation is at least as strong as the* MFlow *formulation, i.e.,* $\text{proj}_z(\mathcal{P}_{DF}) \subseteq \text{proj}_z(\mathcal{P}_{MF})$. *Furthermore we even have* $\text{proj}_{z,f}(\mathcal{P}_{DF}) \subset \text{proj}_{z,f}(\mathcal{P}_{MF})$

Proof. We show that for any feasible solution (x,f) of DFlow we can obtain a feasible solution (z,g,h) of MFlow, using $\text{proj}_{z,f}$ as described above. Based on these, it is easy to see that (9), (10), (11), and (15) are satisfied. It remains to show that we can always find a feasible flow solution h within the network G with the projected edge capacities z. Let $\chi := (s,t) \in \mathcal{D}$. If $\chi \in \mathcal{C}$, we can choose $h_e^\chi := f_e^\chi + f_e^{(t,s)}$, which satisfies (12), (13), (14) due to (4) and proj_z.

Assume $\chi \notin \mathcal{C}$. We look at the maximum (s,t)-flow in G with capacities z and consider any corresponding minimum (s,t)-cut; let S be the cut set containing s, and $V \setminus S$ contains t. W.l.o.g. assume that $r \in S$. Since (2) is satisfied for the commodities (r,t) and (t,r), the maximum undirected (r,t)-flow, and therefore also the maximum undirected (s,t)-flow h^χ, is at least 2. Assume we cannot send h^χ without violating (14). Then there exists a single node w such that the total capacity ϱ of the cut edges which do not send their flow through w is less than one. If $w \neq r$, (3) guarantees that we can send two (undirected) flow units between r and t whereby at most one unit is sent through w. This is a contradition to ϱ. If $w = r$, (6) guarantees an in-flow into r of exactly 1 for both the (s,r) and the (t,r) flow. Hence, using the two 1-flows $f^{(s,r)}$ and $f^{(t,r)}$ we can send an undirected flow of at least 1 between s and t without using r, which is again a contradition to ϱ.

To establish the second claim it is enough to construct a feasible flow g in MFlow which sends more than one unit into the root r. This is infeasible for DFlow as (6) is violated. □

Observation 1. DFlow is more compact than MFlow.

I.e., our novel formulation requires less variables and less constraints than MFlow. Furthermore, our formulation answers the question by Raghavan [17, p. 183], whether his flow variables g and h can be bounded together more tightly. Note that Corollary 1 is crucial for the validity of our approach, which explains why this compact formulation could not be used legitimately before.

4 Directed-Cut for D2NCON

We now present an ILP formulation (DCut) based on directed cuts. Its number of variables is independent of the size of $\mathcal{R}$ as it only requires variables x_e for all $e \in A$. On the other hand, it requires an exponential number of constraints. In the following, we will see that these constraints are of the traditional cut type and therefore easily and polynomially separable within a Branch-and-Cut approach. As seen in, e.g., [4, 12], cut formulations may clearly outperform flow formulations in practice, which has been the main motivation to develop DCut. The ILP itself is a variant of the ILP for the 2RSN problem [4], guaranteeing that only one edge is directed into the arbitrarily chosen r.

Let $S \subset V$, then $\delta^+_G(S) := \{(s,t) \in A \mid s \in S, t \in V \setminus S\}$ denotes the arcs leaving S, and $\delta^-_G(S) := \{(s,t) \in A \mid s \in V \setminus S, t \in S\}$ the arcs entering S. If the graph G is clear from the context, we will omit the subscript. Furthermore, we use the shorthands $G_w := G \setminus \{w\}$, for some $w \in V$, and $x(B) := \sum_{e \in B} x_e$ for some $B \subseteq A$.

$$\text{DCut}: \qquad \min \sum_{(i,j) \in A} c_{ij} \cdot x_{ij} \tag{16}$$

$$x_{vw} + x_{wv} \leq 1 \qquad \forall \{v,w\} \in E \tag{17}$$

$$x(\delta^-(S)) \geq 1 \qquad \forall S \subseteq V \setminus \{r\}, S \cap \mathcal{R}' \neq \emptyset \tag{18}$$

$$x(\delta^+(S)) \geq 1 \qquad \forall S \subseteq V \setminus \{r\}, S \cap \mathcal{R}'_2 \neq \emptyset \tag{19}$$

$$x(\delta^-_{\bar{G}_w}(S_1)) + x(\delta^+_{\bar{G}_w}(S_2)) \geq 1 \qquad \begin{array}{l} \forall w \in V \setminus \{r\}, \forall S_1, S_2 \subseteq V \setminus \{r,w\}, \\ S_1 \cap S_2 \cap \mathcal{R}'_2 \neq \emptyset \end{array} \tag{20}$$

$$\textstyle\sum_{(i,r) \in A} x_{ir} = 1 \tag{21}$$

$$x_{vw} \in \{0,1\} \qquad \forall (v,w) \in A \tag{22}$$

As before, (17) guarantees the unique orientation of chosen edges. The constraints (18) and (19) ensure the existence of the required paths, and (20) assures the node-disjointness of these paths. Finally, (21) requires exactly one edge being directed towards the root. Based on this description and Corollary 1 we obtain:

Theorem 5. *An optimal solution for* DCut *gives an optimal solution for the corresponding 2NCON problem.*

We can compare this formulation with the common and currently best known cut-formulation presented in [19, p. 14], which we denote by UCut since it is based on undirected cuts. Therefore we define $\delta_G(S) := \{\{s,t\} \in E \mid s \in S, t \in V \setminus S\}$ as the edges with one incident node in S and the other one in its complement. We use the variables z_e for all $e \in E$ which are set to 1 if the corresponding edge is selected into N, and 0 otherwise.

$$\text{UCut}: \qquad \min \sum_{e \in E} c_e \cdot z_e \tag{23}$$

$$z(\delta(S)) \geq 1 \qquad \forall S \subseteq V, \emptyset \neq S \cap \mathcal{R}_1 \neq \mathcal{R}_1 \tag{24}$$

$$z(\delta(S)) \geq 2 \qquad \forall S \subseteq V, \emptyset \neq S \cap \mathcal{R}_2 \neq \mathcal{R}_2 \tag{25}$$

$$z(\delta_{G_w}(S)) \geq 1 \qquad \forall w \in V, \forall S \subseteq V, \emptyset \neq S \cap (\mathcal{R}_2 \setminus \{w\}) \neq \mathcal{R}_2 \setminus \{w\} \tag{26}$$

$$z_e \in \{0,1\} \qquad \forall e \in E \tag{27}$$

We can show that our (rooted, directed) DCut formulation is stronger than the (unrooted, undirected) UCut formulation. Let $\mathcal{P}_{DC}$ and $\mathcal{P}_{UC}$ be the polyhedra of the feasible solutions of the LP relaxations for DCut and UCut, respectively. We can use the natural projection $x_{vw} + x_{wv} = z_{e'}$ for all $e' = \{v,w\} \in E$ in order to obtain $\text{proj}_z(\mathcal{P}_{DC})$.

Theorem 6. *We have* $\text{proj}_z(\mathcal{P}_{DC}) \subset \mathcal{P}_{UC}$*, i.e., the* DCut *formulation is strictly stronger than the* UCut *formulation.*

Proof. We can show that $\text{proj}_z(\mathcal{P}_{DC}) \neq \mathcal{P}_{UC}$ using the triangle graph with $\varrho = 1$ for each node. The solution $z_e = 0.5$ for each edge e is feasible for UCut, but there is no corresponding flow in DCut which would be feasible. We can obtain an example with $|\mathcal{R}_2| \geq 2$ by attaching a feasible network of $\mathcal{R}_2$ nodes to one of the triangle's nodes.

Hence it remains to show that $\text{proj}_z(\mathcal{P}_{DC}) \subseteq \mathcal{P}_{UC}$, i.e., that we can generate the undirected constraints from their directed counterparts. Recall that for any set $S \subset V$ we have $z(\delta(S)) = z(\delta(V \setminus S)) = x(\delta^-(V \setminus S)) + x(\delta^-(V))$. Consider any constraint (24) with its corresponding set S. If $r \in S$, we can use (18) and have $x(\delta^-(V \setminus S)) \geq 1$. Analogously, if $r \notin S$, we have $x(\delta^-(S)) = x(\delta^+(V \setminus S)) \geq 1$. Therefore, in both cases we have $z(\delta(S)) \geq 1$ and the undirected constraint is hence satisfied.

Consider any constraint (25) with its corresponding set S; we show: $z(\delta(S)) = x(\delta^-(S)) + x(\delta^+(S)) \geq 2$. If $r \in V \setminus S$, the inequalities (18) and (19) directly give the above formula. If $r \in S$, we can consider the cut set $V \setminus S$ instead of S, as $z(\delta(S)) = z(\delta(V \setminus S))$. Using the same argument for the graph G_w we can generate the inequalities (26) from the inequalities (20) with $S_1 = S_2$. □

5 Strength of D2NCON formulations

5.1 Polyhedral Comparison

Let $\text{proj}_x(\mathcal{P}_{DF})$ be the natural projection obtained from $\mathcal{P}_{DF}$ by only considering its x variables.

Theorem 7. *We have* $\text{proj}_x(\mathcal{P}_{DF}) = \mathcal{P}_{DC}$*, i.e., the* DFlow *formulation and the* DCut *formulation are equally strong.*

Proof. $\text{proj}_x(\mathcal{P}_{DF}) \subseteq \mathcal{P}_{DC}$: We show that if (x, f) is feasible for DFlow, then x is feasible for DCut, i.e., x satisfies the constraints (18), (19), and (20).

Assume there is a set $S \subseteq V \setminus \{r\}$ with $v \in S \cap \mathcal{R}'$ and $x(\delta^-(S)) < 1$. This would imply that the minimum (r, v)-cut is less than 1. By the max-flow min-cut theorem this implies that the maximum (r, v)-flow is less than 1, which is a contradiction to the corresponding flow constraint (2) with commodity (r, v). Analogously, we can show that the constraints (19) are also satisfied.

Let $v \in \mathcal{R}'_2$. Since f satisfies DFlow, we know that there is exactly one unit of flow of commodity (r, v) going from r to v, and one unit commodity (v, r) going from v to r. Hence the total amount of flow between r and v is 2. The constraints (3) ensure that deleting any node $w \neq r$ in G can decrease this amount by at most one flow unit. Hence there is an (undirected) max-flow of at least 1 between r and v in G_w, and therefore the minimum undirected cut between any $v \in \mathcal{R}'_2$ and r in any G_w is at least 1. Since for any sets S_1 and S_2 in (20), there exists an $\mathcal{R}'_2$ customer node in $S_1 \cap S_2$, the sum of the respective cut sizes has to be at least 1.

$\mathcal{P}_{DC} \subseteq \text{proj}_x(\mathcal{P}_{DF})$: We show that if x is feasible for DCut, then there exists a flow f such that (x, f) is feasible for DFlow. Clearly, all DFlow constraints only dealing with x-variables are satisfied as they are identical to the DCut formulation. It remains to show that we can fit flow into the network using the x-values as capacities. Note that the flows of each commodity are mostly independent of each other, as only the coupling constraints (3) define a dependency between the forward and the backward flow for each $v \in \mathcal{R}'_2$. It is clear that we can find a flow from r to any $v \in \mathcal{R}'_1$, since DCut's constraint (18) guarantees a minimum cut between r and v of at least 1. Analogously, because of (18) and (19), we can also find a forward and a backward flow $\left(f^{(r,v)}, f^{(v,r)}\right)$ for each $v \in \mathcal{R}'_2$, and it remains to show that there always exists such a pair of flows which satisfies (3) for all nodes $w \in V \setminus \{r, v\}$. Let us assume there exists no such pair of flows. Then let $\left(\hat{f}^{(r,v)}, \hat{f}^{(v,r)}\right)$ be the flows satisfying (2) and (4) where the maximal violation of (3) is minimal. Let $\hat{w}$ be a node where such a maximal violation occurs, cf. Fig. 1(a). The paths used by the flow can then be divided into multiple paths which go through $\hat{w}$ (P), and multiple paths which do not go through $\hat{w}$ (Q). Let $\alpha > 1$ be the amount of flow over P, then we have $\beta = 2 - \alpha < 1$ as the flow over Q.

Due to constraint (20) we know that there exists a set Q^+ of additional paths not going through $\hat{w}$ over which we can send $\beta^+ > 0$ amount of flow, such that $\beta + \beta^+ = 1$. Hence we can look at the flow pair $\left(\tilde{f}^{(r,v)}, \tilde{f}^{(v,r)}\right)$, where the flow over $\hat{w}$ is only 1, and the second flow unit is sent over that paths of Q and Q^+.[6] Since the original flow was minimal in terms of constraint violation, this new pair of flows has to have a different node $\tilde{w}$ over which at least α flow units are sent, say $\gamma \geq \alpha$. Clearly, $\tilde{w}$ has to be part of Q^+. Even when Q^+ contributes all of its flow units to γ, we have $\gamma = \gamma' + \beta^+$ and have:

$$\alpha \leq \beta^+ + \gamma' = 1 - \beta + \gamma' = 1 - 2 + \alpha + \gamma' \quad \Longrightarrow \quad 1 \leq \gamma'$$

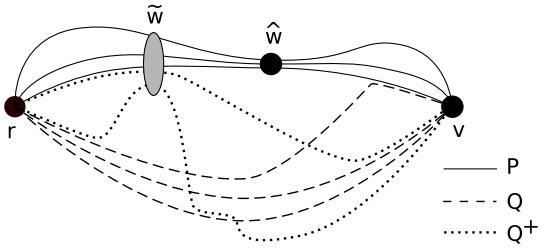
The paths of Q will not contribute to γ, since then we could modify the original flow $\left(\hat{f}^{(r,v)}, \hat{f}^{(v,r)}\right)$ such that (3) is less violated for $\hat{w}$ (without introducing an additional violation of at least α). Thus we have that γ' has to be contributed by paths of P. Since the new flow sends exactly 1 unit over P we have the fact that all paths in P go through $\hat{w}$ and $\tilde{w}$: otherwise we could choose a path going through $\hat{w}$ and not through $\tilde{w}$ and further reduce the flow through the latter.

Hence we know that in $\left(\hat{f}^{(r,v)}, \hat{f}^{(v,r)}\right)$ both $\hat{w}$ and $\tilde{w}$ have a through-flow of α, and the paths in P can be subdivided into subpaths between r and $\tilde{w}$, $\tilde{w}$ and $\hat{w}$, and $\hat{w}$ and v, assuming, w.l.o.g. that $\tilde{w}$ is closer to r than $\hat{w}$. But then, we can iterate the above argument, send only 1 unit of flow through $\hat{w}$ and $\tilde{w}$, and find an additional node $\check{w}$ which has too much through-flow. This argument can be iterated ad infinitum, thus requiring an infinitely large graph, which states a contradition. □

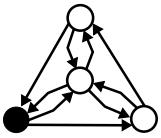
5.2 Additional Cut-Constraints

Recall that UCut without (26) is the traditional undirected cut formulation for 2ECON. It has been shown in [19] that the latter formulation can be strengthened by adding certain classes of valid inequalities which are NP-hard to separate. Chopra [5] showed that his directed 2ECON cut formulation inherently includes one of these classes, namely the *partition inequalities*; in [19, pp. 130–134] it was shown that the latter formulation also includes the class of the (polynomially separable) *Prodon inequalities*. Moreover, Raghavan showed that his improved undirected multi-commodity flow formulation for kECON, which for $k = 2$ is equivalent to both directed flow and directed cut formulations for 2ECON, also includes the *odd-hole inequalities* and the *combinatorial-design inequalities* [17, pp. 165–180].

[6] We can choose this new pair of flows such that (2) and (4) still holds, since, even if the newly routed flow over Q^+ sends the total amount β^+ in a single direction (say from r to v), we know that $\hat{f}$ satisfies (2) and therefore sends at least $\alpha - 1 = \beta^+$ units into each direction. In the modified flow we can hence remove enough directed flow per direction from P to allow valid flow using Q^+ instead.



(a) Sketch for proof part of $\mathcal{P}_{DC} \subseteq \text{proj}_x(\mathcal{P}_{DF})$



(b) The node-partition inequalities are not satisfied.

Fig. 1. In (b), all nodes have connectivity requirement 2, the root node is denoted by the black circle, and all arcs have an x-value of 0.5.

By dropping the constraints (20) and (21) from DCut, we obtain the directed cut formulation for 2ECON. Hence we can conclude:

Proposition 1. DCut *and* DFlow *inherently ensure the validity of the partition, Prodon, odd-hole, and combinatorial-design inequalities.*

On the other hand we can show:

Proposition 2. *None of the above formulations induces the undirected* node-partition inequalities *[19, pp. 91–94], i.e., undirected partition inequalities where one node is removed from the graph. This result constitutes a negative answer for the open question [17, p. 183] whether* MFlow *would induce this constraint class.*

Proof. See Figure 1(b) for an example, which denotes the x variables of the arcs. Once more, we use the natural projection $x_{vw} + x_{wv} = z_{vw}$ for all $\{v, w\} \in E$ to obtain an undirected network: when removing the central node, the partition inequality with three partition sets is violated. Yet the solution is feasible for the LP relaxation of DCut. □

5.3 Algorithmical Remarks

From the point of formulation strength, using DFlow instead of DCut might seem like a reasonable choice in general, as both the number of variables and constraints are bound by a polynomial. But in practice, the latter has certain advantages: it requires much less variables, especially when $\mathcal{R}$ is large. Furthermore, its drawback of an exponential number of constraints can turn out to be beneficial, as the actual computation

of an optimal solution will in general not require all of these constraints. Therefore, traditional Branch-and-Cut techniques can be expected to be highly efficient, since all constraints in DCut can be easily separated using simple polynomial max-flow algorithms, see, e.g., [22].

6 Experiments

The main purpose of our experimental study was to obtain an unskewed comparison between our two formulations DCut and DFlow. Therefore, we implemented both formulations using CPLEX 10.0's Branch-and-Bound framework, without any preprocessing or primal heuristics. The additionally necessary separation routines for DCut were implemented in C++ using LEDA 5.1.1 and the efficient max-flow algorithm of [2]. For specific separation strategies see [4, 12]. All the tests were performed on an a single core of an Intel Core 2 Duo E4300 with 1.80GHz, 2GB of RAM per process, and a time limit of 2 hours per problem instance.

We use three different benchmark sets, which have also been used and thoroughly described in [4] for the evaluation of 2RSN formulations. See Table 1 for a summary of our experimental results and Figure 2 for diagrams visualizing them.

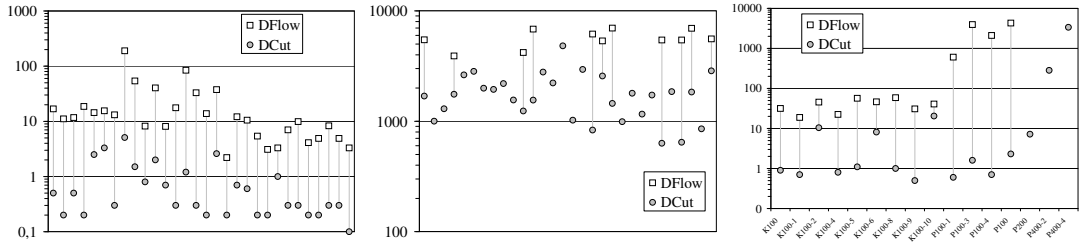
ClgS instances. The first benchmark set *ClgS* contains 25 instances based on real-world map data of the city district Cologne-Ossendorf. The underlying graph has 190 nodes and 377 edges. The instances differ in the customer nodes, and have 3–6 $\mathcal{R}_1$, and 2–3 $\mathcal{R}_2$ customers.

Each instance of this group was solved to optimality within a few seconds, whereby DFlow was on average 3 times faster than DCut. This is due to the fact that the underlying LPs of DFlow are rather small for this small number of customer, and the overhead of DCut's cut separation routines is comparably expensive. This picture already shifts to the favor of DCut, when we consider the related test set *ClgS*$^+$ [4], which only differs in that is has more customer nodes. Note that neither of both approaches requires any branching.

Grid instances. We consider the artificial instances of [20] based on grid graphs with 100, 400, 900, and 1600 nodes. For each graph size there are 2×15 instances, using two different edge cost functions, respectively. Any instance has 5–13 $\mathcal{R}_1$ customers, and 3–8 $\mathcal{R}_2$ customers.

	ClgS	**Grid 100**	**Grid 400**	**Grid 900**	**Grid 1600**	**K**	**P 100**
DFlow	0.5/0.5	11.4/22.2	209.6/281.01	1530.7/1737.2	5463.4/5569.3 (36.7%)	40.7/39.02	3013.8/2131.5
DCut	1.1/1.65	0.3/0.9	21.45/27.08	196.6/495.35	1739.7/1822.68	1/4.9	1.2/1.3

Table 1. Median/average CPU time in seconds. The percentage of successful instances is given in brackets if not 100%. See Figure 2(c) for the *P* instances with 200 and 400 nodes.



(a) Grid instances with 100 nodes (b) Grid instances with 1600 nodes (c) PCSTLib$^+$ instances

Fig. 2. Comparison of DFlow and DCut. The vertical axis gives the CPU time in seconds, while the different instances per instance set are plotted on the horizontal axis in alphabetical order. For the PCSTLib$^+$ instances, the labeled names give the group and specify whether an instance has 100, 200, or 400 nodes. If an instance was not solved to provable optimality within the time limit, the diagram does not contain a data point for the corresponding formulation.

Eventhough the number of customer nodes is only slightly higher than for the *ClgS* instances, this is enough for DCut to clearly outperform DFlow, cf. Table 1. There are only 2 instances (both with 900 nodes) where DFlow is slightly faster. The effect of the increased number of customers is further amplified by the larger underlying graphs, as this results in an even larger increase of variables for DFlow. While the cut approach is able to solve all problems, the flow approach solves only 36.7% of the largest instances within 2 hours.

PCSTLib$^+$ instances. The PCSTLib benchmark [10], was used in several studies, e.g., [13, 14], and contains originally random graphs divided into two groups K and P, where 15%–27% and 34%–50% of the nodes are customers, respectively. The former were generated to be similar to street map layouts. We consider the augmented *PCSTLib$^+$* [4, 20] benchmark, where roughly 1/3 of the customer nodes are selected to be in $\mathcal{R}_2$. For the experiments, we restricted ourselves to the graphs with up to 100 nodes for group K, and up to 400 nodes for group P. Note that some of these instances are infeasible for 2NCON as the underlying graph does not allow any two node-disjoint paths between certain customers. Hence we report only on the feasible instances.

Again, DCut is clearly faster than DFlow: roughly 8 times for the K instances; and over 1600 times faster for the P instances with 100 nodes, on average. The larger graphs cannot be solved by DFlow, but DCut finds provably optimal solution for all of them; cf. Fig. 2(c).

References

1. D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study.* Princeton University Press, 2006.

2. B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1997.
3. M. Chimani, M. Kandyba, I. Ljubić, and P. Mutzel. Obtaining optimal k-cardinality trees fast. 2008. To appear in ALENEX'08.
4. M. Chimani, M. Kandyba, and P. Mutzel. A new ILP formulation for 2-root-connected prize-collecting Steiner networks. In L. Arge, M. Hoffmann, and E. Welzl, editors, *ESA*, volume 4698 of *Lecture Notes in Computer Science*, pages 681–692. Springer, 2007.
5. S. Chopra. Polyhedra of the equivalent subgraph problem and some edge connectivity problems. *SIAM J. Discrete Math.*, 5(3):321–337, 1992.
6. M. X. Goemans. *Analysis of linear programming relaxations for a class of connectivity problems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.
7. M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral Approaches to Network Survivability. In *Reliability of Computer and Communication Networks, Proc. Workshop 1989*, volume 5 of *Discrete Mathematics and Theoretical Computer Science*, pages 121–141. American Mathematical Society, 1991.
8. M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operatios Research*, 40(2):309–330, 1992.
9. M. Grötschel, C. L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization*, 2(3):474–504, 1992.
10. D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting steiner tree problem: Theory and practice. In *Proceedings of 11th ACM-SIAM Symposium on Distcrete Algorithms*, pages 760–769, San Fransisco, CA, 2000.
11. H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.
12. I. Ljubić. *Exact and Memetic Algorithms for Two Network Design Problems*. PhD thesis, Technische Universität Wien, 2004.
13. I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming, Series B*, 105(2–3):427–449, 2006.
14. A. Lucena and M. G. C. Resende. Strong lower bounds for the prize-collecting steiner problem in graphs. *Discrete Applied Mathematics*, 141(1-3):277–294, 2003.
15. T. L. Magnanti and S. Raghavan. Strong formulations for network design problems with connectivity requirements. *Networks*, 45(2):61–79, 2005.
16. T. Polzin and S. V. Daneshmand. Improved algorithms for the Steiner problem in networks. *Discrete Applied Mathematics*, 112(1-3):263–300, 2001.
17. S. Raghavan. *Formulations and Algorithms for the Network Design Problems with Connectivity Requirements*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.
18. H.E. Robbins. A theorem on graphs with an application to a problem of traffic control. *American Mathematical Monthly*, 46:281–283, 1939.
19. M. Stoer. *Design of Survivable Networks*. Springer-Verlag, 1992. volume 1531 of Lecture Notes in Mathematics.
20. D. Wagner, G. R. Raidl, U. Pferschy, P. Mutzel, and P. Bachhiesl. A multi-commodity flow approach for the design of the last mile in real-world fiber optic networks. In *Operations Research Proceedings 2006*, pages 197–202. Springer-Verlag, 2006.
21. P. Winter. Steiner problem in networks: A survey. *Networks*, 17(2):129–167, 1987.
22. L. A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.