Available online at www.sciencedirect.com

ScienceDirect

Discrete Optimization 5 (2008) 415–433

DISCRETE OPTIMIZATION

www.elsevier.com/locate/disopt

Partition inequalities for capacitated survivable network design based on directed p-cycles[☆]

Alper Atamtürk[a,*], Deepak Rajan[b]

[a] Industrial Engineering & Operations Research, University of California, Berkeley, CA 94720-1777, United States
[b] IBM T. J. Watson Research Center, Hawthorne, NY 10532, United States

Received 24 December 2005; accepted 12 August 2007
Available online 24 October 2007

Abstract

We study the design of capacitated survivable networks using directed p-cycles. A p-cycle is a cycle with at least three arcs, used for rerouting disrupted flow during edge failures. Survivability of the network is accomplished by reserving sufficient slack on directed p-cycles so that if an edge fails, its flow can be rerouted along the p-cycles.

We describe a model for designing capacitated survivable networks based on directed p-cycles. We motivate this model by comparing it with other means of ensuring survivability, and present a mixed-integer programming formulation for it. We derive valid inequalities for the model based on the minimum capacity requirement between partitions of the nodes and give facet conditions for them. We discuss the separation for these inequalities and present results of computational experiments for testing their effectiveness as cutting planes when incorporated in a branch-and-cut algorithm. Our experiments show that the proposed inequalities reduce the computational effort significantly.

Keywords: Survivable network design; Mixed-integer rounding; p-cycles; Polyhedra; Partition inequalities

1. Introduction

Given a directed graph, a commodity set (origin, destination, demand triples), and costs for flow and capacity, the capacitated network design problem (NDP) is to install batches of a capacity unit on the edges of the graph and route the flow of commodities so that the flow on each arc is no more than the capacity installed and all demands are met at minimum total cost. NDP is $\mathcal{NP}$-hard even in the case of a single commodity [13]. We refer the reader to Balakrishnan et al. [6] for a review of the network design problem.

The network design problem becomes significantly more difficult when the network has to be designed so as to survive failures of its edges. As a simultaneous failure of multiple edges occurs infrequently (at least for the telecommunication networks we aim to address), here we focus on single-edge failures. Then, a network is said to be *survivable* if sufficient capacity exists on the edges of the network so that disrupted flow can be rerouted in the

☆ Research is supported, in part, by NSF Grant 0070127.
* Corresponding author.
E-mail addresses: atamturk@berkeley.edu (A. Atamtürk), drajan@us.ibm.com (D. Rajan).

1572-5286/$ - see front matter
doi:10.1016/j.disopt.2007.08.002

event of any edge failure. To do so requires the installation of spare capacity on the network. Various approaches that attempt to minimize the capacity requirement of survivable networks have been developed. Soriano et al. [36] present an overview of survivable network design problems and a synthesis of the related literature.

The most capacity-efficient survivable networks can be designed by formulating the problem as a capacitated network design problem for every failure scenario, linked by common capacity variables across the scenarios [1,15,24]. Scenario formulations for optimization problems under uncertainty have origins in Dantzig [16]. However, a serious disadvantage of such scenario models is that the size of their formulation equals the size of the network design model (NDP) times the number of failure scenarios (number of edges in this case), which renders the approach impractical even for relatively small networks. Furthermore, an optimal solution to a scenario formulation may call for rerouting the flow of commodities that are not disrupted by the failure. As it is not practical to manipulate undisrupted flow while restoring disrupted flows, this approach, which is also referred to as *global rerouting* (GNP), is not popular in practice. Xiong and Mason [41] cite fast flow recovery requirement as a reason. Nevertheless, GNP serves an important purpose as it provides a lower bound on the capacity requirement of a survivable network. Methodologies implemented in practice usually involve some form of local rerouting, either by enforcing a ring-like topology (dedicated protection) on the network [2,14,17,35], or by shared local protection schemes [21,23,39], and have higher capacity requirement compared to GNP.

We study a hybrid approach for designing survivable networks as proposed by Grover and Martens [19], in which cycles of the network are used for shared protection of disrupted flow, but no specific topology (e.g. ring structure) is imposed on the network. In this approach, one imposes no restriction on no-failure routing, but utilizes specific failure-flow patterns for rerouting the disrupted flow. Using undirected cycles with at least three edges (*p-cycles*) as failure-flow patterns has been shown to be capacity efficient and to achieve fast rerouting times [34,37]. Grover and Stamatelakis [18] solve the survivable network design problem in a hierarchical fashion, in which they first solve the network design problem without the survivability requirement and then assign spare capacity on a subset of undirected p-cycles covering each edge that may fail. The current paper has certain key distinctions from these earlier works. First, it takes a non-hierarchical approach to ensure survivability. Second, rather than assigning integral spare capacity to (undirected) p-cycles, we reserve sufficient slack, which may not necessarily be a multiple of the capacity unit, along directed arcs of the p-cycles. Reserving slack for the flow on a link that may fail rather than covering the capacity of the link leads to a more capacity-efficient survivable network, because flow that must be rerouted is always less than or equal to the capacity of the failed link and slacks on working links may also be utilized for rerouting the disrupted flow. Third, due to the exponentially large number of potential p-cycles available, we employ a column generation technique to pick the p-cycles to consider survivability.

Rajan and Atamtürk [32] introduce the survivability model studied in the current paper (SNP) and compare SNP with NDP and GNP computationally. They conclude that while the capacity efficiency of SNP is very close to GNP, the computational effort required to solve SNP is similar to NDP. This positive conclusion is the main motivation for the polyhedral study of SNP in the current paper.

Rajan and Atamtürk [33] consider a simpler survivability model (SDC), which is more conservative than SNP in terms of capacity usage. In Section 2, we explain the distinctions between SNP and SDC, and compare the capacity requirement for the two in order to emphasize the capacity efficiency of the model studied here. Bienstock and Muratore [12] and Balakrishnan et al. [8] study capacitated survivability models with global rerouting and give strong inequalities for them.

Balakrishnan et al. [7] and Magnanti and Raghavan [28] study un-capacitated survivable network design problems with connectivity requirements and describe strong formulations. Un-capacitated problems mainly differ from capacitated problems in two ways: each commodity is described in terms of number of edge-disjoint paths required; each edge, if chosen, can support all the flow on that edge (one can think of the capacity as infinity). Thus, un-capacitated network design problems are more combinatorial in nature, and are generalized by their capacitated analogues.

The focus of this paper is a polyhedral study of the SNP model for designing capacitated survivable networks. In Section 3 we introduce valid inequalities for SNP based on the minimum capacity requirement between partitions of the nodes and give facet conditions for them. We discuss the separation for these inequalities and incorporate them in a branch-and-cut algorithm to solve survivable network design problems using directed p-cycles. In Section 4 we present computational results, which demonstrate that the proposed inequalities reduce computational effort significantly when used as cuts. We conclude in Section 5 with a few final remarks.

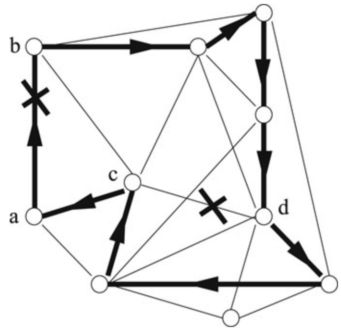

Fig. 1. A directed p-cycle covers reverse directional arcs and chords.

Notations and assumptions

Let $G = (N, E)$ be an undirected graph with node set N and edge set E. Let F be the set of all ordered pairs (arcs) from E, that is, $F = \{(ij), (ji) \ : \ [ij] \in E\}$, where (ij) denotes the arc from node i to node j, and $[ij]$ denotes the (undirected) edge between nodes i and j. Let $G' = (N, F)$ denote the corresponding directed graph. When the end nodes are not relevant, we use $a \in F$ to index the arcs. For $I \subseteq F$ let $[I] := \{[a] \in E : a \in I\}$. We assume, if necessary by introducing edges with high capacity cost, that the graph G is complete. Let K be a commodity set, in which each commodity $k \in K$ is specified by the triple (s^k, t^k, d^k) denoting demand d^k (a non-negative rational number) at the destination node t^k from the source node s^k.

We assume that there is no pre-installed capacity on the network and that capacity is installed in batches of a single type of facility, e.g., fiber-optic cable, having a fixed capacity. Demands as well as edge capacities are expressed in units of this basic facility capacity; so demands are often fractional. We consider a version of the problem where capacity installed on an (undirected) edge can be used to route (directed) flow up to this capacity in both directions, which is typical in telecommunication networks [11,38]. For a vector $v \in \mathbb{R}^X$ indexed on a set X we define $v\langle H\rangle := \sum_{i \in H} v_i$ for $H \subseteq X$ with $v\langle\emptyset\rangle = 0$. For simplicity of notation, we denote a singleton set $\{a\}$ by a. We use ϵ to denote an infinitesimal constant.

2. Survivable network design with p-cycles

In this section we present the survivability model studied in the paper and highlight its capacity efficiency.

A *directed p-cycle* is a simple directed cycle of G' with at least three arcs. Directed p-cycles are used for rerouting disrupted flow during edge failures. This is accomplished by reserving sufficient slack on the directed p-cycles of the network. We will refer to a directed p-cycle simply as p-cycle.

In the survivability model studied in this paper (SNP) a p-cycle is used for rerouting disrupted flow on the reverse directional arcs for the p-cycle as well as on the chords of the p-cycle. Consider, for instance, the p-cycle with clockwise direction, shown with bold arcs in Fig. 1. Since arc (ab) is on the p-cycle, if edge $[ab]$ fails, the flow on arc (ba) may be rerouted from node b to node a using the p-cycle illustrated in the figure or it may be split and rerouted on a number of p-cycles containing arc (a, b). If the chord edge $[cd]$ fails, the same p-cycle may be used to reroute the flow from node c to node d using the upper section of the p-cycle, and the flow from node d to c using the lower section of the p-cycle. As the example illustrates, p-cycles cover flow on multiple arcs. To ensure survivability, sufficient slack is reserved for the p-cycles so that for each arc, the sum of reserved slack for all p-cycles that cover the arc is at least the flow on the arc.

On the other hand, in the SDC model [33] p-cycles do not cover the chords, which makes SDC significantly more conservative than SNP in terms of capacity usage. After introducing a mathematical formulation for SNP, we will compare the capacity requirement of the two models.

2.1. Mathematical formulation

Now we present a mathematical formulation for SNP. Let $x_e \in \mathbb{Z}_+$ be the amount of capacity installed on edge $e \in E$ and $y_a^k \in \mathbb{R}_+$ be the amount of flow of commodity k on arc $a \in F$. We use g_a^k to denote the cost of unit flow of commodity $k \in K$ on arc a and h_e to denote the cost of unit capacity on edge e. For each node–commodity pair let $b_i^k = d^k$ for $i = s^k$, $b_i^k = -d^k$ for $i = t^k$, and $b_i^k = 0$ for $i \in N \setminus \{s^k, t^k\}$.

Let $\mathcal{C}$ be the set of simple directed cycles of G' with at least three arcs (p-cycles). For $c \in \mathcal{C}$ we let variable z_c denote the slack reserved for p-cycle c. Of crucial note, as slack is reserved to cover flows, it is not required to be a multiple of the capacity unit; hence, z_c is a continuous variable. Let α_a^c be 1 if p-cycle c includes arc a, and 0 otherwise. Similarly, let ρ_e^c be 1 if edge e is a chord of p-cycle c, and 0 otherwise. Using these definitions SNP is formulated as the following mixed-integer program:

$$\min \sum_{a \in F} \sum_{k \in K} g_a^k y_a^k + \sum_{e \in E} h_e x_e$$

$$\text{s.t.:} \quad \sum_{(ij) \in F} y_{ij}^k - \sum_{(ji) \in F} y_{ji}^k = b_i^k, \quad \forall i \in N,\ \forall k \in K, \tag{1}$$

$$\text{(SNP)} \quad \sum_{k \in K} y_{ij}^k - \sum_{c \in \mathcal{C}} \rho_{[ij]}^c z_c - \sum_{c \in \mathcal{C}} \alpha_{ji}^c z_c \le 0, \quad \forall (ij) \in F, \tag{2}$$

$$\sum_{k \in K} y_{ij}^k + \sum_{c \in \mathcal{C}} \alpha_{ij}^c z_c \le x_{[ij]}, \quad \forall (ij) \in F, \tag{3}$$

$$x_e \in \mathbb{Z}_+,\ \forall e \in E;\ z_c \in \mathbb{R}_+,\ \forall c \in \mathcal{C};\ y_a^k \in \mathbb{R}_+,\ \forall a \in F,\ \forall k \in K.$$

Constraints (1) are for flow balance. Constraints (2) ensure that flow on each arc (ij) is no more than the total slack reserved for p-cycles which cover (ij), i.e., either include arc (ji) or have chord $[ij]$. Constraints (3) ensure that capacity installed on edge $[ij]$ is large enough to accommodate the flow on arc (ij) as well as the total slack reserved for p-cycles that include (ij).

The formulation for the network design problem without the survivability requirement (NDP) can be obtained from the formulation above by simply dropping constraints (2) and the p-cycle variables z. Thus, SNP has only one additional constraint for each arc compared to NDP. On the other hand, the number of p-cycle variables is exponential in the number of the arcs. Rajan and Atamtürk [32] give a column generation algorithm for solving the linear programming (LP) relaxation of SNP. They show that the pricing problem for p-cycle variables is $\mathcal{NP}$-hard and propose an effective polynomial heuristic to generate p-cycle variables. In Section 4.1, we present a mixed-integer programming formulation to solve the pricing problem exactly.

2.2. Comparison of the survivability models

Since in the SDC model p-cycles do not cover flow on chords, the formulation of SDC does not contain the second term in constraints (2). Thus, SDC is a restriction of SNP and, therefore, requires more excess capacity than SNP. On the other hand GNP, which allows global rerouting of all flows during failures, is a relaxation of SNP and, consequently, requires less capacity than SDC and SNP. However, it is very difficult to solve as it has $\mathcal{O}(|E|)$ times as many constraints as NDP.

In order to provide an empirical evidence on the relative capacity efficiency of the survivability models SDC, SNP, and GNP, in Fig. 2 we summarize computational results on capacity usage from Rajan and Atamtürk [32] and Rajan and Atamtürk [33]. These results are for randomly generated graphs with 75% edge density and 50% demand density, which are available on-line at http://ieor.berkeley.edu/~atamturk/data. Our experiments showed that the variation across random instances was minimal; here, we present the results for one randomly chosen instance for each size. The vertical axis of the chart in the figure shows the ratio of the installed capacity for the survivable model and the capacity of the NDP without the survivability requirement. For example, for a graph with 5 nodes, SDC requires more than double the capacity to achieve survivability. On the average, SDC requires 80% excess capacity, whereas SNP requires only about 45% spare capacity. GNP, which provisions the lowest possible capacity for survivable networks, requires about 30% spare capacity. Thus SNP requires only an additional 12% over the GNP bound, whereas SDC provisions 38% excess capacity over this lower bound. Finally, we remark that for graphs with more than 10 nodes it was not possible to solve even the LP relaxation of GNP in an hour with the default CPLEX solver [32]. Nevertheless, the chart in Fig. 2 clearly demonstrates the capacity-efficiency of SNP, which is the main motivation for the polyhedral study in this paper.

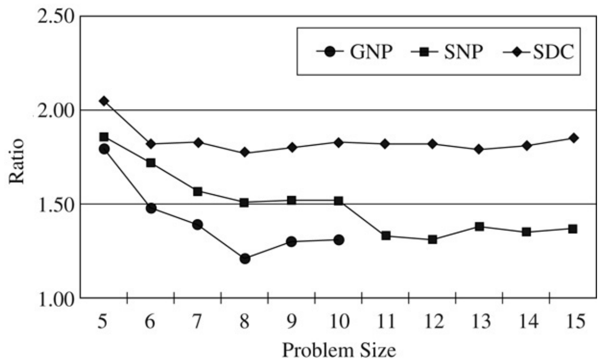

Fig. 2. Comparing the capacity efficiency of the survivability models.

Table 1
Comparison of minimum capacity requirement ($m = |[AB]|$)

Model	NDP	GNP	SNP	SDC
$c(d_A)$	$\lceil d_A \rceil$	$\lceil \lceil d_A \rceil m/(m-1) \rceil$	$\lceil \lceil d_A \rceil m/(m-1) \rceil$	$2\lceil d_A \rceil$
Reference	[26]	[8,12]	Remark 3 in Section 3.2	[33]

Now we provide further insight into the capacity efficiency of SNP observed empirically in the experiments. The relative efficiency of the models can be stated in terms of feasible capacity vectors for the models.

Proposition 1. *Let $\mathcal{X}(\cdot)$ denote the set of feasible integral capacity vectors with respect to a model. Then $\mathcal{X}(SDC) \subseteq \mathcal{X}(SNP) \subseteq \mathcal{X}(GNP) \subseteq \mathcal{X}(NDP)$.*

For a partitioning (A, B) of N with $A \neq \emptyset$ and $B \neq \emptyset$, let $[AB]$ denote the set of edges with one end in A, the other in B, and d_A the total demand of nodes in B from the nodes in A, i.e., $d_A = \sum_{k \in K_A} d^k$, where $K_A = \{k \in K : s^k \in A, t^k \notin A\}$, and let $c(d_A)$ be the minimum integral capacity required on the edges $[AB]$, which may be different for each model. Then, for every feasible capacity vector x we have

$$x\langle [AB] \rangle \geq c(d_A) \tag{4}$$

and inequality (4) is tight for the solution minimizing $x\langle [AB] \rangle$. Although we know from Proposition 1 that $c(d_A)$ for SNP is in between the values for SDC and GNP, Table 1 presents a clearer picture. This table shows the known values for $c(d_A)$ for different models. The minimum integral capacity requirement for SNP over the edges of a given partition indeed equals the lower bound from GNP. Moreover, as the number of edges m increases, the gap between the minimum requirement for SNP and SDC becomes larger, whereas the minimum capacity requirement for SNP, just like GNP, gets closer to NDP.

In conclusion, SNP is a highly capacity-efficient model for designing survivable capacitated networks, even though it allows rerouting of only disrupted flow.

3. Partition inequalities for SNP

In this section we introduce partition inequalities for SNP. Partition inequalities are known to improve the linear programming (LP) relaxations of NDP significantly [3,4,11,20,25]; however, their separation problem is $\mathcal{NP}$-hard [10]. Magnanti et al. [26] introduce partition inequalities for NDP in terms of the integral capacity variables and Magnanti et al. [27] extend them to the case with two capacity types. Barahona [9] presents a cutting-plane algorithm based on the partition inequalities. Bienstock and Günlük [11], Chopra et al. [13] generalize these inequalities further to include non-zero coefficients for the continuous flow variables.

Balakrishnan et al. [8] and Bienstock and Muratore [12] derive partition inequalities for survivability models with global rerouting (GNP), and Rajan and Atamtürk [33] for SDC. Many partition inequalities for NDP and SDC can be derived as strengthened metric inequalities [22,30] that incorporate the respective survivability restrictions [31]. For a review of partition inequalities for NDP, and recent work on tight metric inequalities for NDP, we refer the reader to Avella et al. [5].

Let (A, B) be a partition of the nodes such that $A \neq \emptyset$ and $B \neq \emptyset$, and let $G'_A = (A, F_A), G'_B = (B, F_B)$ be the induced subgraphs defined by A and B. Let AB be the arcs directed from A to B, and BA be the arcs directed from B to A. We let $m := |AB|$ and use $\overline{\mathcal{C}}$ to denote the set of p-cycles that cross the partition. Let $K_A := \{k \in K : s^k \in A,\ t^k \notin A\}$, $K_B := \{k \in K : s^k \in B,\ t^k \notin B\}$, and $K' := K \setminus K_A \setminus K_B$. Also define $d_A = \sum_{k \in K_A} d^k$, $d_B := \sum_{k \in K_B} d^k$, $y_a^A := \sum_{k \in K_A} y_a^k$, and $y_a^B := \sum_{k \in K_B} y_a^k$. Without loss of generality, we assume that $d_A \geq d_B$.

For any $I \subseteq AB$, we define $\overline{I}$ as the set of arcs oriented in the reverse direction of the arcs in I and $\widehat{I} = I \cup \overline{I}$. So, in particular, $\overline{AB} = BA$ and $\widehat{AB} = AB \cup BA$.

Definition 1. For arc $a \in AB$ let $\mathcal{C}^a$ be the set of all p-cycles that cover arc a. In other words,

$$\mathcal{C}^a := \{c \in \mathcal{C} : \alpha_a^c + \rho_{[a]}^c = 1\}.$$

Observe that any p-cycle in C^a necessarily crosses the partition (A, B) and does not include arc a.

Definition 2. For arc $a \in AB$ and $I \subseteq AB$ and let $\mathcal{C}_I^a$ be the set of p-cycles that cover arc a and cross the partition from A to B using only arcs in I. In other words,

$$\mathcal{C}_I^a := \{c \in \mathcal{C}^a : \alpha_b^c = 0 \text{ for } b \in AB \setminus I\}.$$

The intuition behind considering partitions for SNP can be explained as follows: For an arc $a \in AB$, consider the failure of edge $[a]$. Survivability by SNP requires the sum of undisrupted flow from A to B and total slack reserved for p-cycles covering arc a to be at least the sum of demand of commodities K_A. In other words,

$$y^A \langle AB \setminus a \rangle + z \langle \mathcal{C}^a \rangle \geq d_A. \tag{5}$$

Inequality (5) follows directly from constraints (1) and (2) and it is the main observation that leads to the polyhedral results in the paper. Next we review the mixed-integer rounding (MIR) argument used in the validity proofs.

Lemma 1 (*[29,40]*). *For $x \in \mathbb{Z}$, $y \in \mathbb{R}$ constraints $y + x \geq b$ and $y \geq 0$ imply the* mixed-integer rounding inequality

$$y + (b - \lfloor b \rfloor)x \geq (b - \lfloor b \rfloor)\lceil b \rceil.$$

In the following two subsections we derive two classes of partition inequalities for SNP. These inequalities contain non-zero coefficients for the flow as well as p-cycle variables in addition to the capacity variables. Important special cases and separation issues are discussed afterwards.

3.1. P-cycle flow partition inequalities

The first class of inequalities for SNP is a variation of (5) using capacity variables. Throughout let $\eta := \lceil d_A \rceil$ and $r := d_A - \lfloor d_A \rfloor$.

Proposition 2. *For partition (A, B), arc $a \in AB$, and arc set $I \subseteq AB \setminus a$, the* p-cycle flow partition inequality

$$y^A \langle I \rangle + z \langle \mathcal{C}_I^a \rangle + r x \langle [AB \setminus I \setminus a] \rangle \geq r\eta \tag{6}$$

is valid for SNP.

Proof. As $a \notin I$, by separating terms we rewrite (5) as

$$y^A \langle I \rangle + z \langle \mathcal{C}_I^a \rangle + y^A \langle AB \setminus I \setminus a \rangle + z \langle \mathcal{C}^a \setminus \mathcal{C}_I^a \rangle \geq d_A. \tag{7}$$

Since every p-cycle in $\mathcal{C}^a \setminus \mathcal{C}_I^a$ includes an arc in $AB \setminus I \setminus a$, we have

$$z \langle \mathcal{C}^a \setminus \mathcal{C}_I^a \rangle \leq \sum_{c \in \mathcal{C}^a \setminus \mathcal{C}_I^a} \sum_{b \in AB \setminus I \setminus a} \alpha_b^c z_c \leq \sum_{b \in AB \setminus I \setminus a} \left(\sum_{c \in \mathcal{C}} \alpha_b^c z_c \right). \tag{8}$$

Due to constraints (3) the total capacity installed on edges $[AB \setminus I \setminus a]$ (namely $x\langle[AB \setminus I \setminus a]\rangle$) is at least the flow $y^A\langle AB \setminus I \setminus a\rangle$ and the total slack reserved for p-cycles using arcs $AB \setminus I \setminus a$, the last term in (8). Then, we can relax inequality (7) as

$$y^A\langle I\rangle + z\langle \mathcal{C}_I^a\rangle + x\langle[AB \setminus I \setminus a]\rangle \geq d_A. \tag{9}$$

Applying mixed-integer rounding in Lemma 1 to inequalities (9) and $y^A\langle I\rangle + z\langle \mathcal{C}_I^a\rangle \geq 0$, we obtain the p-cycle flow partition inequality (6). ■

Remark 1. Observe that if $I = AB \setminus a$, then the p-cycle flow partition inequality (6) is dominated by (5) as $\mathcal{C}^a = \mathcal{C}^a_{AB\setminus a}$ and $d_A \geq r\eta$.

Let us denote the convex hull of feasible solutions to SNP as conv(SNP). Next we show that p-cycle flow partition inequalities induce facets for conv(SNP).

Theorem 1. *For any non-empty partition (A, B) of G, arc $(ij) \in AB$, and arc set $I \subsetneq AB \setminus (ij)$, the p-cycle flow partition inequality (6) is facet-defining for* conv(*SNP*) *if $|M^1 \cup M^2| \leq \eta$ for all disjoint matchings M^1, M^2 in $[AB \setminus I \setminus (ij)]$, $r > 0$, and $|A| \neq 2$, $|B| \neq 2$.*

Proof. Consider the face $\mathcal{F}$ of conv(SNP) induced by (6), i.e.,

$$\mathcal{F} = \left\{(x, y, z) \in \text{conv(SNP)} : \ y^A\langle I\rangle + z\langle \mathcal{C}_I^{ij}\rangle + rx\langle[AB \setminus I \setminus (ij)]\rangle = r\eta\right\}.$$

In order to prove that $\mathcal{F}$ is a maximal proper face, i.e., a facet, we will employ a common technique in polyhedral combinatorics that shows that any hyperplane containing $\mathcal{F}$ is equivalent to

$$y^A\langle I\rangle + z\langle \mathcal{C}_I^{ij}\rangle + rx\langle[AB \setminus I \setminus (ij)]\rangle = r\eta \tag{10}$$

up to multiplication by a scalar and addition of multiples of the equality constraints (1), implying that there are enough affinely-independent points in $\mathcal{F}$ to define the coefficients of (10).

Then consider a hyperplane defined on (x, y, z) as

$$\sum_{a\in F}\sum_{k\in K} \pi_a^k y_a^k + \sum_{e\in E} \beta_e x_e + \sum_{c\in\mathcal{C}} \kappa_c z_c = \gamma. \tag{11}$$

Let $Y := AB \setminus (ij)$. For each commodity k, consider a spanning arborescence with arc set T_k rooted at the source node s^k of commodity k such that $T_k \cap \widehat{Y} = \emptyset$. Observe that this implies T_k must include either arc (ij) or arc (ji). By adding appropriate multiples of (1) to (11) (for commodity k) for all nodes in depth-first order of T_k, we can eliminate the coefficients of the flow variables corresponding to arcs in T_k in (11). Hence, we may assume, without loss of generality, that $\pi_a^k = 0,\ \forall a \in T_k,\ \forall k \in K$ in (11).

The following definitions are needed for the proof: Let $\tilde{\mathcal{C}}$ be the set of p-cycles that cross the partition exactly once among the arcs in $\widehat{Y}$, and are Hamiltonian cycles on the graph G'. Since any cycle crosses a partition an even number of times, this requires that any p-cycle in $\tilde{\mathcal{C}}$ must use either arc (ij) or (ji). For a set of arcs $Q \subseteq Y$ we define $\tilde{\mathcal{C}}^Q$ as the set of p-cycles in $\tilde{\mathcal{C}}$ that use some arc $a \in \widehat{Q}$.

Now we will define a pivotal point Δ in $\mathcal{F}$ that will be used to construct other convenient points in $\mathcal{F}$ for simple interchange arguments to establish the coefficients of (11): Because the subgraphs G'_A and G'_B are complete and $|A| \neq 2$, $|B| \neq 2$, there exist p-cycles in G'_A and G'_B covering all arcs in these subgraphs. In solution Δ, all commodities in $K' = K \setminus K_A \setminus K_B$ are routed using arcs in $F \setminus \widehat{AB}$ and covered using p-cycles in $\mathcal{C} \setminus \overline{\mathcal{C}}$. The rest of the solution Δ is defined as follows (see Fig. 3): Pick $(ab) \in Y \setminus I$ arbitrarily and let $y_{ab}^A = d_A$ and $y_{ji}^B = d_B$; for a p-cycle $c1 \in \tilde{\mathcal{C}}$ that contains both (ba) and (ij), let $z_{c1} = d_A + \epsilon$ so that flow on both (ab) and (ji) is covered by $c1$ (recall that $d_A \geq d_B$), and the capacity variables $x_{[ab]} = \eta$ and $x_{[ij]} = \eta$. Finally let $y_a = 0$ for all $a \in \widehat{AB} \setminus (ab) \setminus (ji)$ and $z_c = 0$ for all $c \in \overline{\mathcal{C}} \setminus c1$, $x_{[a]} = 0$ for all $a \in Y \setminus I$, and x_e for all $e \in [I]$ is large. Because $r > 0$ and $c1 \notin \mathcal{C}^{ij}$, Δ is feasible and satisfies (10). We define Y_a and Z_a as the total flow on arc a and slack reserved for p-cycles using a, respectively. Observe that for the solution Δ, we have $Y_{ef} < Z_{fe}$ and $Y_{ef} + Z_{ef} < x_{[ef]}$ for $(ef) \in \{(ab), (ji)\}$. This is an important property of Δ that helps us to construct other convenient points.

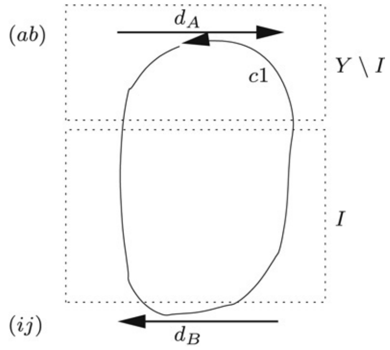

Fig. 3. Feasible solution $\varDelta$.

To simplify Eq. (11) our first strategy is to find the variables that do not appear in (11), i.e., those that have zero coefficients. We start with showing that $\beta_e = 0,\ \forall e \in E \setminus [Y]$; these coefficients correspond to capacity variables which are not part of the partition. Consider $\varDelta$. For any edge $e \in E \setminus [Y]$, we increase the capacity by one unit to obtain a new feasible solution $\varDelta'$, which still satisfies (10). Substituting $\varDelta$ and $\varDelta'$ into (11), we see that $\beta_e = 0,\ e \in E \setminus [Y]$. Similarly, $\beta_e = 0,\ e \in [I]$. In the rest of the proof, for $\varDelta$ we assume w.l.o.g. that x_e is large ($>2d\langle K\rangle$ is sufficient) for all $e \in [I] \cup E \setminus [Y]$ (We refer to this as *Assumption* $\mathcal{A}1$).

We now show that $\kappa_c = 0,\ \forall c \in \mathcal{C} \setminus \overline{\mathcal{C}}$; these coefficients correspond to p-cycles that do not cross the partition. For any $c \in \mathcal{C} \setminus \overline{\mathcal{C}}$, we increase z_c by ϵ to obtain a new solution $\varDelta'$ that satisfies (10). $\varDelta'$ is feasible due to the assumption $\mathcal{A}1$. Substituting $\varDelta$ and $\varDelta'$ in (11), we see that $\kappa_c = 0,\ c \in \mathcal{C} \setminus \overline{\mathcal{C}}$. In the rest of the proof, for $\varDelta$ we assume w.l.o.g. that z_c is large for all $c \in \mathcal{C} \setminus \overline{\mathcal{C}}$, but such that there still exists some slack on every edge $e \in E \setminus [Y]$ (We refer to this as *Assumption* $\mathcal{A}2$).

Next we prove that $\pi_a^k = 0,\ \forall a \in F \setminus Y,\ k \in K$; these coefficients correspond to the flow variables for all arcs that do not cross the partition. For commodity k, consider an arc $(ab) \in F \setminus (Y \cup T_k)$. Let s be the nearest common ancestor of a and b in T_k (s equals a or b if there is a directed path between them in T_k). By increasing y_{ab}^k and the flow on arcs in the path from s to a in T_k by ϵ and decreasing the flow on arcs in the path from b to s in T_k by ϵ, we obtain a new feasible solution $\varDelta'$ satisfying (10) by Assumptions $\mathcal{A}1$ and $\mathcal{A}2$. Substituting $\varDelta$ and $\varDelta'$ in (11), we see that $\pi_{(ab)}^k = 0$.

We now show that $\kappa_c = 0,\ \forall c \in \tilde{\mathcal{C}}^{Y\setminus I}$, and that $\pi_a^k = 0,\ \forall a \in \widehat{Y \setminus I},\ \forall k \in K$. We obtain a new feasible solution $\varDelta'$ from $\varDelta$ by increasing z_{c1} by ϵ. $\varDelta'$ satisfies (10) (recall that $c1 \notin \mathcal{C}^{ij}$). Substituting $\varDelta$ and $\varDelta'$ in (11), we have $\kappa_{c1} = 0$. Alternatively, we can obtain $\varDelta'$ that satisfies (10) from $\varDelta$ by introducing ϵ units on a new p-cycle $c2$ that uses the same edges as $c1$ but flows in the reverse direction (note $c2 \notin \mathcal{C}_I^{ij}$). Substituting $\varDelta$ and $\varDelta'$ in (11), we have $\kappa_{c2} = 0$. From $\varDelta$, we obtain a new solution $\varDelta'$ that satisfies (10) by decreasing y_{ba}^k and increasing y_{ij}^k by ϵ for any commodity k. Substituting $\varDelta$ and $\varDelta'$ in (11), we have $\pi_{ab}^k = 0$ for all $k \in K$ since $\pi_{ij}^k = 0$ (recall $(ij) \in F \setminus Y$). For any commodity k, by increasing flow on arcs (ab) and (ba) by ϵ, we obtain another new solution $\varDelta'$ from $\varDelta$ that satisfies (10). Substituting $\varDelta$ and $\varDelta'$ in (11), we get $\pi_{ba}^k = 0$ for all $k \in K$. Since (ab) was chosen arbitrarily from $Y \setminus I$, such pairs of solutions can be constructed for all arcs $a \in Y \setminus I$. Substituting such pairs of $\varDelta$ and $\varDelta'$ into (11), we get $\kappa_c = 0,\ \forall c \in \tilde{\mathcal{C}}^{Y\setminus I}$ and $\pi_a^k = 0$ for all $a \in \widehat{Y \setminus I},\ \forall k \in K$.

For $Q \subseteq Y$ let $(\tilde{\mathcal{C}}_f^Q, \tilde{\mathcal{C}}_b^Q)$ be the partitioning of $\tilde{\mathcal{C}}^Q$, depending on whether $c \in \tilde{\mathcal{C}}^Q$ uses an arc in Y or $\overline{Y}$. For any p-cycle $c \in \tilde{\mathcal{C}}_b^I$, since $\tilde{\mathcal{C}}_b^I \cap C_I^{ij} = \emptyset$ we can increase the allocation for the cycle by ϵ units to obtain a new feasible solution $\varDelta'$ that satisfies (10). Substituting $\varDelta$ and $\varDelta'$ into (11), we obtain $\kappa_c = 0,\ c \in \tilde{\mathcal{C}}_b^I$. Thus we have shown that $\kappa_c = 0,\ \forall c \in \tilde{\mathcal{C}} \setminus \tilde{\mathcal{C}}_f^I$.

For the rest of the proof, we need an intermediate technical result that holds due to the assumption of the theorem on disjoint matchings: Toward this goal, we also need solutions $\varDelta_1$ and $\varDelta_2$ that satisfy equation (10) and Assumptions $\mathcal{A}1$ and $\mathcal{A}2$; see Fig. 4. Comparing them with $\varDelta$ and other derived feasible points satisfying (10), we evaluate the rest of the coefficients. $\varDelta_1$ is defined by modifying $\varDelta$ as follows: For an arbitrarily chosen arc $(ef) \in I$, we decrease flow variable y_{ab}^A to $d_A - r$ and route r using arc (ef) instead; thus, $y_{ef}^A = r$. We reduce the allocation on p-cycle $c1$ so that $z_{c_1} = d_A - r$ and introduce a new p-cycle variable $c2 \in \tilde{\mathcal{C}}_b^I$ to cover the flow on arc (ef); thus, we set $z_{c2} = r$. Finally, we reduce the capacity installed on edge $[ab]$ by one unit and set $x_{[ab]} = \eta - 1$. $\varDelta_2$ is obtained by modifying

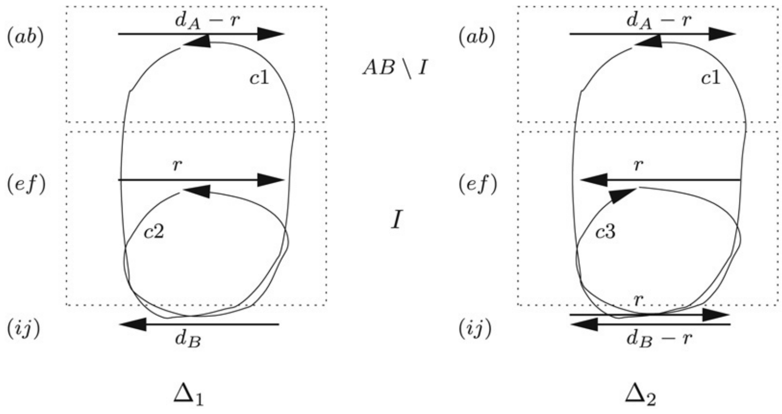

Fig. 4. Feasible solutions Δ_1 and Δ_2.

Δ_1 as follows: We set $y^A_{ef} = 0$ and route this flow on (ij) instead; i.e., $y^A_{ij} = r$. We reduce y^B_{ji} to $d_B - r$ and route the remaining flow on (fe) instead; i.e., $y^A_{fe} = r$. We replace p-cycle $c2$ with a new p-cycle $c3 \in \tilde{\mathcal{C}}^I_f$ that goes through the same edges as $c2$, but in the reverse direction. Hence, both points Δ_1 and Δ_2 satisfy (10).

Claim 1. *For any p-cycle $c \in \overline{\mathcal{C}} \setminus \tilde{\mathcal{C}}$, either $\kappa_c = 0$ or there exists a p-cycle $c0 \in \tilde{\mathcal{C}}^I_f$ such that $\kappa_c = \kappa_{c0}$.*

Proof. For p-cycle $c \in \overline{\mathcal{C}} \setminus \tilde{\mathcal{C}}$, let $T \subseteq Y \setminus I$ and $W \subseteq \overline{Y \setminus I}$ be the arcs p-cycle c uses to cross the partition. Note that $[W]$ and $[T]$ are two disjoint matchings of $[Y \setminus I]$. Letting $t = |T|$ and $w = |W|$, from the assumption of the theorem on all disjoint matchings in $[Y \setminus I]$, we have $t + w \le \eta$. Consider the solution Δ' with $z_c > 0$ obtained as follows:

Case 1. $T \ne \emptyset$. We construct Δ' from Δ. Without loss of generality, suppose that Δ is chosen so that $(ab) \in T$. Let $T' = T \setminus (ab)$ and $t' = |T'|$. Observe that $t' + w \le \eta - 1$. Let $x_e = 1$ for all $[T' \cup W]$ and $x_{[ab]} = \eta - t' - w$. For demands in K_A, we send $1 - \epsilon$ flow on each arc in T' and $\overline{W}$ and $d_A - (t' + w)(1 - \epsilon)$ on (ab). We reduce allocation of z_{c1} to $\eta - (t' + w) - \epsilon$, and introduce a new p-cycle in $\tilde{\mathcal{C}}^{Y \setminus I}_b$ for each arc in $T' \cup \overline{W}$, setting its allocation to $1 - \epsilon$. Δ' satisfies (10) as well because the total flow on I, slack reserved for p-cycles in C^{ij}_I and the total capacity on $[Y \setminus I]$ are unchanged. Now, by introducing p-cycle c with allocation ϵ, we get another solution Δ'' that satisfies (10). Substituting Δ' and Δ'' into (11), we get $\kappa_c = 0$.

Case 2a. $T = W = \emptyset$. In this case, p-cycle c does not use any arcs in $\widehat{Y \setminus I}$.

2a.1. If p-cycle c covers arc (ij) and uses some arc $(ef) \in I$, consider solution Δ_2; see Fig. 4. Assume without loss of generality that Δ_2 is chosen so that $\alpha^c_{ef} = 1$. We obtain Δ' from Δ_2 by setting $z_c = \epsilon$ and reducing the slack reserved for p-cycle c_3 to $r - \epsilon$. Substituting Δ_2 and Δ' into (11), we obtain $\kappa_c = \kappa_{c3}$, where $c3 \in \tilde{\mathcal{C}}^I_f$.

2a.2. If p-cycle c does not cover arc (ij), we obtain Δ' from Δ by setting $z_c = \epsilon$. Substituting Δ and Δ' into (11), we see that $\kappa_c = 0$.

Case 2b. $T = \emptyset$ and $W \ne \emptyset$. Without loss of generality, suppose Δ_2 is chosen so that $ba \in W$. Let $W' = W \setminus (ba)$ and $w' = |W'|$. Observe that $w' \le \eta - 1$. We construct Δ' from Δ_2 as follows: Let $x_{[a]} = 1$ for all $a \in W'$ and $x_{[ab]} = \eta - 1 - w'$. For demands in K_A, we send 1 unit of flow on each arc in W', $d_A - w' - r$ on (ab), and r units on (ij). We reduce allocation of z_{c1} to $\eta - w' - \epsilon$, and introduce a new p-cycle in $\tilde{\mathcal{C}}^{Y \setminus I}_b$ for each arc in W' with allocation $1 - \epsilon$.

2b.1. If p-cycle c covers arc (ij), we obtain another solution Δ'' that satisfies (10) by increasing allocation to p-cycle $c3$ to $r - \epsilon$, and adding p-cycle c with allocation ϵ. Substituting Δ' and Δ'' into (11), we obtain $\kappa_c = \kappa_{c3}$, where $c3 \in \tilde{\mathcal{C}}^I_f$.

2b.2. If p-cycle c does not cover arc (ij), we obtain Δ'' from Δ' by setting $z_c = \epsilon$. Substituting Δ' and Δ'' into (11), we see that $\kappa_c = 0$. ■

Next we show that $\kappa_c = 0,\ \forall c \in \overline{\mathcal{C}} \setminus \mathcal{C}^{ij}_I$. Any p-cycle $c \in \overline{\mathcal{C}} \setminus C^{ij}_I \setminus \tilde{\mathcal{C}}$ contains either (ij) or some arc $(ab) \in Y \setminus I$, since it does not cover flow on arc (ij) or use any arc in I. If p-cycle c uses any arc in $AB \setminus I$, we get $\kappa_c = 0$ from Claim 1 (Case 1). If it does not, then c contains (ij), and does not cover (ij). Now, depending on whether c uses any arcs on $\overline{AB \setminus I}$, we get $\kappa_c = 0$ from either Case 2.a.2 or 2.b.2 of Claim 1.

Now we show that $\beta_e = \beta$ for all $e \in [Y \setminus I]$. Since in the definition of Δ the choice of $(ab) \in Y \setminus I$ is arbitrary, let Δ' be the solution with arc $(ef) \in Y \setminus I$, different from (ab). Substituting Δ and Δ' in (11), we get $\beta_{[ab]} = \beta_{[ef]}$. Thus, $\beta_e = \beta$ for all $e \in [Y \setminus I]$.

In the rest of the proof we will determine the remaining coefficients in (11): $\beta = \beta_{[a]},\ a \in Y \setminus I, \pi^k_a,\ a \in I, k \in K$, and $\kappa_c,\ c \in \mathcal{C}^{ij}_I$.

First, we show that $\pi^k_{fe} = 0$ for all $k \in K$. From Δ_1, we can obtain an new solution Δ' that satisfies (10) by increasing y^k_{fe} and y^k_{ij} by ϵ for $k \in K$. Substituting Δ and Δ' in (11), we have $\pi^k_{fe} = 0$ for all $k \in K$.

Second, we show that $\pi^k_{ef} = 0$ for any $k \notin K_A$. For $k \notin K_A$, by increasing flow on arcs (ef) and (fe) by ϵ, we can obtain another new solution Δ' from Δ that satisfies (10). Substituting Δ and Δ' in (11), we get $\pi^k_{fe} = 0$ for all $k \notin K_A$.

Third, we show that for any commodity $k \in K_A$, $\pi^k_{ef} = \beta/r$. Substituting Δ and Δ_1 in (11), we get $-r\pi^k_{ab} + r\pi^k_{ef} - (r + \epsilon)\kappa_{c1} + r\kappa_{c2} - \beta_{[ab]} = 0$. Since $\pi^k_{ab} = \kappa_{c1} = \kappa_{c2} = 0$, we have $\pi^k_{ef} = \beta/r$ for all $k \in K_A$.

Fourth, we show that for p-cycle $c3$, $\kappa_{c3} = \beta/r$. Substituting Δ_1 and Δ_2 in (11), we get $-r\pi^k_{ef} + r\pi^k_{ij} - r\kappa_{c2} + r\kappa_{c3} = 0$. Since $\pi^k_{ij} = \kappa_{c2} = 0$ and $\pi^k_{ef} = \beta/r$, we have $\kappa_{c3} = \beta/r$.

Since (ef) is chosen from I arbitrarily, we have $\kappa_c = \beta/r$, $\forall c \in \tilde{\mathcal{C}}^I_f$. Similarly we obtain $\pi^k_a = 0$, if $a \in \overline{I}$, $\forall k \in K$, $\pi^k_a = 0$ if $a \in I$, $\forall k \notin K_A$, and $\pi^k_a = \beta/r$ if $a \in I$, $\forall k \in K_A$.

Furthermore, for all $c \in \mathcal{C}^{ij}_I$, there exists a p-cycle $c1 \in \tilde{\mathcal{C}}^I_f$ such that $\kappa_c = \kappa_{c1}$. To see why this is true, consider any p-cycle $c \in \mathcal{C}^{ij}_I$. This p-cycle c contains some arc $(ab) \in I$ and covers the flow on arc (ij). By Claim 1, $\kappa_c = \kappa_{c1}$, where $c1 \in \tilde{C}^I_f$ and uses (ab) (either Case 2.a.1 or Case 2.b.1 depending on whether c contains any arcs from $\overline{AB \setminus I}$). Therefore, we have $\kappa_c = \beta/r$ for all $c \in \mathcal{C}^{ij}_I$.

Finally, plugging Δ in (11), we obtain $\gamma = \eta\beta$. This gives $\beta_{[a]} = \gamma/\eta$, for all $a \in AB \setminus I$, $\pi^k_a = \gamma/(r\eta)$ if $a \in I$, $\forall k \in K_A$, and $\kappa_c = \gamma/(r\eta)$ for all $c \in \mathcal{C}^{ij}_I$. Thus, we have shown that (11) is a multiple of (10). Multiplying (11) by $r\eta/\gamma$, we obtain (10). ■

Remark 2 (*Necessity*). Theorem 1 establishes sufficient facet conditions for (6). The condition $|A| \neq 2, |B| \neq 2$ is assumed for convenience. However, the remaining facet conditions are necessary: If $r = 0$, it follows from the MIR procedure that p-cycle flow partition inequality is dominated by (9) and $y^A\langle I\rangle + z\langle \mathcal{C}^a_I\rangle \geq 0$.

For any two disjoint matchings M^1, M^2 in $[AB \setminus I \setminus a]$, there is a p-cycle c crossing the partition using the edges in the matchings. Consequently, if $|M^1 \cup M^2| > \eta$, then $z_c > 0$ implies $x\langle AB \setminus I \setminus a\rangle > \eta$, in which case (6) is not tight. In other words, $z_c = 0$ for all points on the face of conv(SNP) defined by (6).

3.2. P-cycle flow subset-Q inequalities

The next class of inequalities are obtained by considering subsets of p-cycle flow partition inequalities (6) simultaneously. For $Q \subseteq AB$ with $q := |Q| \geq 2$, let $\eta_q = \lceil q\eta/(q-1)\rceil$ and $r_q = q\eta - (q-1)\lfloor q\eta/(q-1)\rfloor$.

Proposition 3. *For partition (A, B) and arc sets $Q \subseteq AB$ with $q \geq 2$, $I \subseteq AB \setminus Q$, the* p-cycle flow subset-Q inequality

$$\frac{q}{r}\, y^A\langle I\rangle + \frac{1}{r}\sum_{a\in Q} z\langle \mathcal{C}^a_I\rangle + r_q x\langle [Q]\rangle + (r_q + 1)x\langle [AB \setminus Q \setminus I]\rangle \geq r_q\eta_q \tag{12}$$

is valid for SNP.

Proof. Consider the p-cycle flow partition inequality (6) defined by $a \in AB$ and $I \subseteq AB \setminus a$, which is

$$y^A\langle I\rangle + z\langle \mathcal{C}^a_I\rangle + rx\langle [AB \setminus I \setminus a]\rangle \geq r\eta. \tag{13}$$

For $Q \subseteq AB \setminus I$, adding (13) for each arc $a \in Q$ and dividing the resulting inequality by $r(q-1)$, we have

$$\frac{q}{rq - r} y^A\langle I\rangle + \frac{1}{rq - r}\sum_{a\in Q} z\langle \mathcal{C}^a_I\rangle + x\langle [Q]\rangle + \frac{q}{q-1}x\langle [AB \setminus Q \setminus I]\rangle \geq \frac{q\eta}{q-1}. \tag{14}$$

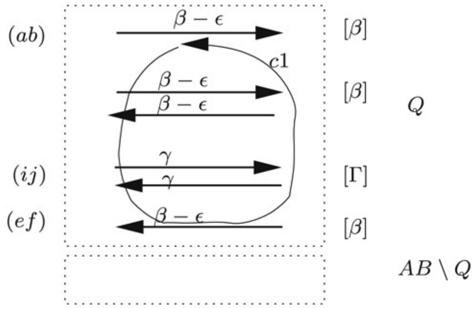

Fig. 5. Feasible solution Δ_Q.

We obtain (12) by applying mixed-integer rounding (Lemma 1) to (14) after splitting the fourth term into $x\langle[AB \setminus Q \setminus I]\rangle$ and $x\langle[AB \setminus Q \setminus I]\rangle/(q-1)$ and by treating $x\langle[AB \setminus Q \setminus I]\rangle$ as an integer variable and the remaining $x\langle[AB \setminus Q \setminus I]\rangle/(q-1)$ as a non-negative continuous variable, and then multiplying the resulting MIR inequality by $q-1$. ■

The next theorem states conditions under which p-cycle flow subset-Q inequalities (12) are strong for SNP. We need to define the following notation: For a matching M in $[AB]$ and $P \subseteq [AB]$ let $M_P = M \cap P$.

Theorem 2. *For any non-empty partition (A, B) of G and arc set $Q \subseteq AB$ with $q > 2$, the p-cycle flow subset-Q inequality (12) with $I = \emptyset$ is facet-defining for* conv(*SNP*) *if $|M^1_Q \cup M^2_Q| + \lceil \frac{1}{r_q}|M^1_{AB\setminus Q} \cup M^2_{AB\setminus Q}|\rceil(r_q+1) \le \eta_q$ for all disjoint matchings M^1, M^2 in $[AB]$, $r_q > 0$, $|A| \ne 2$, $|B| \ne 2$, and either $q = m$ or $\eta m/(m-1) < \lfloor \eta q/(q-1)\rfloor$.*

Remark 3. Note that partition inequality (4) is a special case of (12) with $I = \emptyset$ and $Q = AB$. In order to show that the right-hand side of (4) is tight with $c(d_A) = \lceil \lceil d_A \rceil m/(m-1)\rceil$, below we illustrate a feasible point of SNP, Δ_Q, satisfying

$$r_q x\langle[Q]\rangle + (r_q+1)x\langle[AB \setminus Q]\rangle = r_q \eta_q \tag{15}$$

for any $Q \subseteq AB$. Indeed, Δ_Q is the pivotal point used for manipulation to describe other points satisfying (15) for proving Theorem 2 with similar exchange arguments as in the proof of Theorem 1.

Just like the point Δ in the proof of Theorem 1, for the point Δ_Q all commodities in $K' = K \setminus K_A \setminus K_B$ are routed using arcs in $F \setminus \widehat{AB}$ and covered using p-cycles in $\mathcal{C} \setminus \overline{\mathcal{C}}$. The rest of the point Δ_Q is defined as follows (see Fig. 5): Pick $(ab), (ef) \in Q$ arbitrarily ($q > 2$). Let $\beta = \eta_q - \eta$ and $\Gamma = \eta_q - \beta\lfloor \eta_q/\beta\rfloor$ (note that $\eta_q > \eta$). We set capacities of arbitrary $\lfloor \eta_q/\beta\rfloor$ edges in $[Q]$ to β, one of them to Γ, and the rest to zero. This can be done as $q > \eta_q/\beta$. Let $[H]$ be the set of edges with capacity β. Note that, because $\eta_q \ge 2\beta$, we have $h := |H| \ge 2$. W.l.o.g let $[ab]$ and $[ef]$ be two edges in $[H]$; i.e., $x_{[ab]} = x_{[ef]} = \beta$ and let $x_{[ij]} = \Gamma$.

We send the commodities in K_A using arcs $Q \setminus (ef)$ and the commodities in K_B using arcs $\overline{Q} \setminus (ba)$ as follows: For commodities in K_A, we route $\beta - \epsilon$ units on arcs $H \setminus (ef)$; and the remainder $\gamma = d_A - h(\beta - \epsilon)$ is sent on arc (ij). Similarly, for commodities in K_B, we route up to $\beta - \epsilon$ units on arcs $\overline{H} \setminus (fe)$ until all the flow is sent. If $d_B > h(\beta - \epsilon)$, the rest is sent on arc (ji). In Fig. 5, we present the case, where $d_A = d_B$ and $h = 3$.

Now, to cover the flow on these arcs, consider p-cycle $c1 \in \tilde{\mathcal{C}}$ (defined in the proof of Theorem 1) that contains both (ba) and (ef) and let $z_{c1} = \beta - \epsilon/2$ so that flow on all arcs in $\widehat{Q}$ (including both (ab) and (ji)) is covered by $c1$. Since we did not route any flow on arcs (ba) and (ef), the capacity constraints on edges $[ab]$ and $[ef]$ are not violated. Finally let $y_a = 0$ for all $a \in \widehat{AB \setminus Q}$ and $z_c = 0$ for all $c \in \overline{\mathcal{C}} \setminus c1$, and $x_e = 0$ for all $e \in [AB \setminus Q]$. Hence, Δ_Q is feasible and satisfies (15).

Remark 4 (*Necessity*). The first two conditions of Theorem 2 are necessary: For any two disjoint matchings M^1, M^2 in $[AB]$, there is a p-cycle c crossing the partition using the edges in the matchings. If $|M^1_Q \cup M^2_Q| + \lceil \frac{1}{r_q}|M^1_{AB\setminus Q} \cup M^2_{AB\setminus Q}|\rceil(r_q+1) > \eta_q$, then $z_c > 0$ implies that $r_q x\langle Q\rangle + (r_q+1)x\langle AB \setminus Q\rangle > r_q\eta_q$, in which case (12) is not tight. In other words, $z_c = 0$ for all points on the face of conv(SNP) defined by (12).

It follows from the MIR procedure that p-cycle flow subset-Q inequality (12) is dominated by p-cycle flow partition inequalities (6) and non-negativity on the continuous variables if $\eta q/(q-1)$ is an integer, or equivalently $r_q = 0$ (in particular, if $q = 2$).

The assumption $|A| \neq 2$, $|B| \neq 2$ is made for convenience. When $Q \neq AB$, the last condition is sufficient to ensure that there are solutions with positive capacity variables for edges in $[AB \setminus Q]$ satisfying (12) at equality.

3.3. Special cases

We obtain special cases of the p-cycle flow partition inequality (6) and p-cycle flow subset-Q inequality (12) that contain only integer capacity variables by letting $I = \emptyset$. For partition (A, B) and arc $a \in AB$, we refer to

$$x\langle[AB \setminus a]\rangle \geq \eta \tag{16}$$

as the *survivable partition inequality*. On the other hand, for partition (A, B) and $Q \subseteq AB$ such that $q \geq 2$,

$$r_q x\langle[Q]\rangle + (r_q + 1)x\langle[AB \setminus Q]\rangle \geq r_q \eta_q \tag{17}$$

is referred to as the *survivable subset-Q inequality*.

Inequalities (16) and (17) are introduced in Bienstock and Muratore [12] and Balakrishnan et al. [8], respectively, as valid inequalities for GNP and they deserve special attention because separation for them is easier than for the general inequalities (6) and (12).

Corollary 1. *For any non-empty partition (A, B) of G and arc $(ij) \in AB$, the survivable partition inequality (16) is facet-defining for* $\operatorname{conv}(\mathcal{X}(SNP))$ *if $r > 0$, $|A| \neq 2$, and $|B| \neq 2$.*

Corollary 2. *For any non-empty partition (A, B) of G and arc set $Q \subseteq AB$ with $q \geq 2$, the survivable subset-Q inequality (17) is facet-defining for* $\operatorname{conv}(\mathcal{X}(SNP))$ *if $r_q > 0$, $|A| \neq 2$, $|B| \neq 2$, and either $q = m$ or $\eta m/(m-1) < \lfloor \eta q/(q-1) \rfloor$.*

3.4. Separation

In this section we discuss separation for the partition inequalities of Sections 3.1 and 3.2 for a given partition (A, B) of the graph.

3.4.1. P-cycle flow partition inequalities

Given a fractional solution $(\bar{x}, \bar{y}, \bar{z})$ to linear programming (LP) relaxation of SNP, and a partition (A, B), we are interested in either finding $a \in AB$ and $I \subseteq AB \setminus a$ such that the corresponding p-cycle flow partition inequality (6) is violated, or proving that no such inequality exists. Unfortunately, we do not know an efficient way for finding a set I that would give a violation, if there is any. Given I, it is easy to find an appropriate a in $\mathcal{O}(m - |I|)$ time; however, there are exponentially many choices for I.

Therefore, we present a weaker inequality for which separation is easier. If there is a set I and an arc a for which the weaker inequality is violated, then so is the corresponding p-cycle flow partition inequality (6). Clearly, the converse is not true. Thus this approach can be used as a heuristic method for finding p-cycle flow partition cuts. The weakening given below is based on counting the number of times a p-cycle uses arcs in the set I, which is $\alpha^c\langle I\rangle$.

Proposition 4. *For partition (A, B), arc $a \in AB$, and arc set $I \subseteq AB \setminus a$, inequality*

$$y^A\langle I\rangle + \sum_{c \in \mathcal{C}^a} \alpha^c\langle I\rangle z_c + r x\langle[AB \setminus I \setminus a]\rangle \geq r\eta \tag{18}$$

is valid for SNP. Furthermore, for a given partition (A, B), separation for inequalities (18) can be done in $\mathcal{O}(m^2)$ time.

Proof. To see that (18) is a weakening of (6) first observe that $z\langle \mathcal{C}^a \setminus \mathcal{C}^a_{AB\setminus I}\rangle \geq z\langle \mathcal{C}^a_I\rangle$. But then, $\alpha^c\langle I\rangle \geq 1$ for all $c \in \mathcal{C}^a \setminus \mathcal{C}^a_{AB\setminus I}$, and $\alpha^c\langle I\rangle = 0$ for all $c \in \mathcal{C}^a_{AB\setminus I}$.

For efficient separation we introduce to the LP formulation auxiliary variables $z_a^b = \sum_{c \in \mathcal{C}^a} \alpha_b^c z_c$ for each pair $a, b \in F$. Given arc a, inequality (18) with the smallest left-hand side can be calculated in $\mathcal{O}(m)$ time as follows: Since $\bar{z}_a^b$ is the total slack reserved for all p-cycle variables that use arc b and cover arc a, the contribution of arc $b \in AB \setminus a$ to the left-hand side of (18) is $\bar{y}_b^A + \bar{z}_a^b$ if it is included in the set I, and is $r\bar{x}_{[b]}$ otherwise. Thus, we find inequality (18) with the smallest left-hand side for arc a by choosing I as

$$I = \left\{ b \in AB \setminus a : \bar{y}_b^A + \bar{z}_a^b < r\bar{x}_{[b]} \right\},$$

which can be done in linear time. Repeating this for each arc $a \in AB$, inequality (18) with the smallest left-hand side for partition (A, B) can be obtained in $\mathcal{O}(m^2)$ time. ∎

We can further reduce the separation effort, by considering a weaker inequality that includes all p-cycle variables using arcs in I.

Proposition 5. *For partition (A, B), arc $a \in AB$, and arc set $I \subseteq AB \setminus a$, inequality*

$$y^A\langle I\rangle + \sum_{c \in \overline{\mathcal{C}}} \alpha^c \langle I\rangle z_c + rx\langle [AB \setminus I \setminus a]\rangle \geq r\eta \tag{19}$$

is valid for SNP. Furthermore, for a given partition (A, B), separation for inequalities (19) can be done in $\mathcal{O}(m)$ time.

Proof. Since $\mathcal{C}^a \subseteq \overline{\mathcal{C}}$ and $\alpha^c\langle I\rangle z_c \geq 0$ for $c \in \overline{\mathcal{C}}$, (19) is a weakening of (18).

For efficient separation we introduce to the LP formulation auxiliary variables $z^b = \sum_{c \in \overline{\mathcal{C}}} \alpha_b^c z_c$ for each $b \in F$. Thus $\bar{z}^b$ is the total slack reserved for p-cycles that cross the partition and contain arc b. Now, define $f_b = \min\{y_b^A + \bar{z}^b, r\bar{x}_{[b]}\}$. Then, inequality (19) with the smallest left-hand side for partition (A, B) is obtained by setting $a = \arg\min_{b \in AB}\{f_b\}$ and $I = \{b \in AB \setminus a : y_b^A + \bar{z}^b < r\bar{x}_{[b]}\}$, which is done in $\mathcal{O}(m)$ time. ∎

3.4.2. Survivable partition inequalities

We now discuss the separation for survivable partition inequalities (16). Observe that for a given partition (A, B) there are only n survivable partition inequalities, each of which can be checked for violation by $\bar{x}$ in $\mathcal{O}(m)$, giving us a trivial $\mathcal{O}(m^2)$ algorithm. However, this can be accomplished more efficiently as shown below.

Proposition 6. *For a given partition (A, B), separation for survivable partition inequalities (16) can be done in $\mathcal{O}(m)$ time.*

Proof. Compute $\overline{X}_m = \bar{x}\langle [AB]\rangle$ and $b = \arg\max\{a \in AB : \bar{x}_{[a]}\}$ in $\mathcal{O}(m)$ time. If $\overline{X} - \bar{x}_{[b]} \geq \eta$, then no violated inequality (16) exists. However, if $\overline{X} - \bar{x}_{[b]} < \eta$, then the most violated survivable partition inequality for partition (A, B) is given by $x\langle [AB \setminus b]\rangle \geq \eta$. ∎

3.4.3. P-cycle flow subset-Q inequalities

In this section we discuss separation for the p-cycle flow subset-Q inequalities. Given a fractional solution $(\bar{x}, \bar{y}, \bar{z})$ to the LP relaxation of SNP and a partition (A, B), we are interested in either finding $Q \subseteq AB$ with $q \geq 2$ and $I \subseteq AB \setminus Q$ such that corresponding p-cycle flow subset-Q inequality (12) is violated, or proving that no such inequality exists.

Since p-cycle flow subset-Q inequalities are derived from multiple p-cycle flow partition inequalities (6), their separation is more involved. As with the p-cycle flow inequalities, we present weaker inequalities than (12) for which separation is easier. The separation algorithm for these weaker inequalities can then be used as heuristics for finding p-cycle flow subset-Q cuts.

Proposition 7. *For partition (A, B) and arc sets $Q \subseteq AB$ with $q \geq 2$, $I \subseteq AB \setminus Q$, inequality*

$$\frac{q}{r} y^A\langle I\rangle + \frac{q}{r} \sum_{c \in \overline{\mathcal{C}}} \alpha^c\langle I\rangle z_c + r_q x\langle [Q]\rangle + (r_q + 1)x\langle [AB \setminus Q \setminus I]\rangle \geq r_q \eta_q \tag{20}$$

is valid for SNP. Furthermore, for a given partition (A, B), separation for inequalities (20) can be done in $\mathcal{O}(m^2 \log m)$ time.

Proof. In the proofs of Propositions 4 and 5 we showed that $z\langle \mathcal{C}_I^a \rangle \le \sum_{c \in \mathcal{C}^a} \alpha^c \langle I \rangle z_c \le \sum_{c \in \overline{\mathcal{C}}} \alpha^c \langle I \rangle z_c$ for $a \in AB$. Then

$$\sum_{a \in Q} z\langle \mathcal{C}_I^a \rangle \le \sum_{a \in Q} \sum_{c \in \mathcal{C}^a} \alpha^c \langle I \rangle z_c \le q \sum_{c \in \overline{\mathcal{C}}} \alpha^c \langle I \rangle z_c,$$

which implies that (20) is a weakening of (12).

For fixed q, the inequality (20) with the smallest left-hand side can be found in $\mathcal{O}(m \log m)$ time as follows: Observe that r_q and η_q are fixed when q is fixed. For efficient separation we introduce to the LP formulation auxiliary variables $z^a = \sum_{c \in \overline{\mathcal{C}}} \alpha_a^c z_c$ for each $a \in F$. Thus $\overline{z}^a$ is the total slack reserved for p-cycles that cross the partition and contain arc a. Let $f_a = \min\{q/r(y_a^A + \overline{z}^a), (r_q + 1)\overline{x}_{[a]}\}$. The contribution of arc $a \in AB$ to the left-hand side of (20) is $r_q \overline{x}_{[a]}$ if it is included in the set Q, and f_a otherwise. So we sort the arcs in AB in non-increasing order of $f_a - r_q \overline{x}_{[a]}$ and assign the first q elements to Q, which is done in $\mathcal{O}(m \log m)$ time. Among the remaining arcs in AB, we assign arc a to I if $q/r(y_a^A + \overline{z}^a) < (r_q + 1)\overline{x}_{[a]}$. Then, since $q \le m$, by repeating this procedure for each q, separation for (20) is completed in $\mathcal{O}(m^2 \log m)$ time. ■

3.4.4. Survivable subset-Q inequalities

We now discuss the separation problem for the survivable subset-Q inequalities (17). Given a fractional solution $(\overline{x}, \overline{y}, \overline{z})$ to the LP relaxation of SNP and a partition (A, B), we are interested in either finding $Q \subseteq AB$ with $q \ge 2$ such that the corresponding survivable subset-Q inequality (17) is violated, or proving that no such inequality exists.

Proposition 8. *For a given partition (A, B), the separation for survivable subset-Q inequalities (17) can be done in $\mathcal{O}(m \log m)$ time.*

Proof. First observe that the survivable subset-Q inequality can be rewritten as

$$x\langle [AB] \rangle + 1/r_q x\langle [AB \setminus Q] \rangle \ge \eta_q,$$

where the first term is constant for a given $\overline{x}$ and is computed in linear time. Then for a fixed q the left-hand side of the inequality is minimized by picking $[Q]$ as the set of edges with the q largest $\overline{x}_e$. Then, after sorting $\overline{x}_e$, $e \in [AB]$ in $\mathcal{O}(m \log m)$ time, partial sums $x\langle [AB \setminus Q] \rangle$ for all $2 \le q \le m$ can be computed in linear time incrementally. ■

4. Computational experiments

In this section we present computational experiments conducted by using the partition inequalities as cutting planes for solving SDP. All experiments are performed using CPLEX Version 10.1 MIP solver on a 3MHz Intel Pentium4 Linux workstation with 1GB main memory. Each instance is run up to either five hours or 1,000,000 branch-and-bound nodes, whichever is reached first.

4.1. Solving the LP relaxation

We solve the LP relaxation of SNP with exponentially many p-cycle variables using column generation. If u and v denote the dual variables for constraints (2) and (3) of the LP relaxation, respectively, then the reduced cost of a p-cycle variable z_c, $c \in \mathcal{C}$ can be stated as

$$\sum_{ij \in F} ((u_{ji} - v_{ij})\alpha_{ij}^c + u_{ij}\rho_{[ij]}^c). \tag{21}$$

Rajan and Atamtürk [32] show that the pricing problem for p-cycle variables is $\mathcal{NP}$-hard and describe an effective polynomial heuristic for identifying p-cycle variables with negative reduced cost.

Here we follow an alternative approach, in which we formulate the pricing problem for p-cycle variables as a mixed-integer program and solve it with CPLEX. Solving the pricing problem exactly allows us to solve the LP relaxation of SNP to optimality and, thus, ensures that we have a lower bound on the optimal value for SNP. A p-cycle variable z_c with negative reduced cost can be identified by solving a minimum weight p-cycle problem on G', where the weight of a p-cycle is defined as in (21). Letting $\chi_a,\ a \in F$ and $\sigma_i,\ i \in N$ be binary variables indicating the arcs and nodes on the p-cycle and $\tau_e,\ e \in E$ the chord edges of the p-cycle, we formulate the problem as

$$\min \sum_{ij \in A} (u_{ji} - v_{ij})\chi_{ij} + \sum_{[ij] \in E} (u_{ij} + u_{ji})\tau_{[ij]}$$

$$\text{s.t.:} \quad \chi\langle\delta(i)\rangle = \sigma_i, \quad i \in N, \tag{22}$$

$$\text{(PPC)} \qquad \chi\langle\delta(S)\rangle \geq \sigma_i + \sigma_j - 1, \quad i \in S,\ j \in N \setminus S,\ S \subset N, \tag{23}$$

$$\sigma_i - \chi_{ij} - \chi_{ji} \geq \tau_{[ij]}, \quad [ij] \in E, \tag{24}$$

$$\chi \in \{0,1\}^F,\ \sigma \in [0,1]^N,\ \tau \in [0,1]^E.$$

Here $\delta(S)$ denotes the set of arcs leaving node set S. Constraints (22) and (23) define a simple directed cycle in G'. Observing that $u \leq 0$, for a node i on the p-cycle, constraint (24) allows $[ij]$ to be picked as a chord edge in an optimal solution if only if neither (ij) nor (ji) is on the p-cycle. Constraint (24) also eliminates cycles with only two arcs as $\chi_{ij} = \chi_{ji} = 1$ is infeasible. Thus, feasible directed cycles are limited to p-cycles. Note that binary restriction on variables σ and τ is not necessary; hence, they are modeled as continuous variables between 0 and 1.

Subtour elimination constraints (23) are added to PPC as they are violated. Their separation problem is a simple i–j min-cut problem. Our experience with pricing p-cycle variables by solving PPC with CPLEX has shown this approach to be quite practical. In our experiment CPLEX solves each pricing problem very fast. Moreover, it produces many p-cycles with negative reduced cost early in the branch-and-bound algorithm before finding an optimal solution. We add all found p-cycles with negative reduced cost to the LP formulation at each pricing phase.

4.2. Adding cutting planes

After the LP relaxation of SNP is solved to optimality with column generation, using only the p-cycle variables generated so far, a branch-and-cut algorithm is started. Because the pricing problem PPC is no longer valid after cutting planes are added to the formulation, we do not generate further p-cycles. However, we keep in the formulation all p-cycle variables ever found even if they are non-basic. Nevertheless, we may not find a true optimal solution to SNP because we do not consider other p-cycles once its LP relaxation is solved and, thus, the solution approach is a heuristic one. The computational results presented in the next subsection show, however, that the objective gap between the optimal LP and the MIP solutions found is small especially for larger instances.

We generate cutting planes from all unbalanced partitions with up to three nodes on one side of the partition. Recall that a necessary facet condition for p-cycle flow partition inequalities is that the size of the matchings $M_1 \cup M_2$ crossing the partition should be sufficiently small. Therefore, unbalanced partitions with small number of nodes on one side of the partition are more likely to produce strong inequalities. So the number of partitions considered (pre-selected for separation of cutting planes) is $\mathcal{O}(|N|^3)$.

The inequalities are added to the formulation in a hierarchical manner starting with the ones with fastest separation algorithms. Thus, given a fractional LP solution, we first look for violated survivable partition inequalities (16). If no more violated cuts of this class are found, we look for violated survivable subset-Q inequalities (17), and then p-cycle flow partition inequalities (6). We use the exact separation methods described in Propositions 6 and 8 to find violated survivable partition and subset-Q cuts and the heuristic method in Proposition 4 to find p-cycle flow partition cuts. We add the most violated subset-Q inequality for each q (see Proposition 8), not just the most violated one for each partition and do not generate p-cycle flow subset-Q inequalities in these experiments.

When deriving the inequalities in Section 3 we used the demand for commodities K_A because they must cross the partition. For sparse graphs, we may use the demand for $\overline{K} \supseteq K_A$ in writing the inequalities. For example, if there is a commodity $k \in K'$ such that all paths between s^k and t^k cross the partition (A, B), then $k \in \overline{K}$.

Preliminary experiments have shown that only a few violated cuts are found at the nodes of the search tree other than the root node. Therefore, in the experiments presented here, the cut separation routines are applied only at the root node of the tree.

4.3. Results

The experiments are performed on three randomly generated data sets: the first two sets consist of graphs with average node degrees four and eight; the third set consists of graphs with 75% density. The largest graph has 17 nodes,

Table 2
Objective values

$\|N\|$	Degree 4			Degree 8			75% density		
	LP	MIP	% gap	LP	MIP	% gap	LP	MIP	% gap
5	74.5	92.5	19.5	74.5	92.5	19.5	74.5	92.5	19.5
6	160.6	194.6	17.5	153.8	171.6	10.4	160.6	194.6	17.4
7	135.5	150.3	9.8	129.4	144.6	10.5	135.5	150.3	9.8
8	181.5	205.0	11.5	166.6	184.6	9.8	166.6	184.6	9.8
9	223.4	232.4	3.9	216.0	227.8	5.2	218.9	230.2	4.9
10	289.6	304.4	4.9	254.0	274.5	7.5	266.9	283.4	5.8
11	350.5	367.7	4.7	340.8	360.7	5.5	342.6	361.7	5.3
12	400.1	413.3	3.2	363.8	384.2	5.3	363.1	383.3	5.3
13	466.5	483.8	3.6	424.2	441.4	3.9	410.3	431.1	4.8
14	528.4	540.5	2.2	498.0	508.8	2.1	487.0	498.9	2.4
15	716.4	730.9	2.0	659.9	676.4	2.5	622.6	640.1	2.7
16	1005.8	1017.2	1.1	945.9	965.4	2.0	922.7	946.0	2.5
17	1076.1	1093.7	1.6	1020.1	1039.3	1.9	985.4	1008.7	2.3

and 34 and 68 edges for the first two sets, and 102 edges for the third set. The demand density of each instance is set to 50%, i.e., between every pair of nodes, there exists a commodity k with probability 0.5. Demand d^k for each commodity is drawn from Uniform[0, 2]. The objective coefficients are assigned by giving each node a uniformly generated random coordinate on the unit plane. The flow cost for each arc–commodity pair, g_a^k, equals the Euclidean length of the arc, and the capacity cost of the corresponding edge, $h_{[a]}$, equals $20 \times g_a^k$. The data set is available on-line at http://ieor.berkeley.edu/~atamturk/data.

Before we present the detailed computational results on the effectiveness of the cutting planes described in the paper, we list the objective values for the LP relaxation and the integer solutions found for the instances in the data set in Table 2. The columns under the heading `LP` give the optimal values for LP relaxation of SNP. Since we price the p-cycle variables exactly (see Section 4.1), this value represents a true lower bound on the instances. The columns under the headings `MIP` and `% gap` give the objective value of the best integer solution found and the percentage gap between LP and MIP values. Because we do not generate p-cycles after cuts are added to the formulation, MIP values are only upper bounds on the optimal values for SNP. Nevertheless, in Table 2 we see that the gap between the MIP and LP bounds is quite small, especially for larger instances.

Detailed results for graphs with degree 4 are presented in Table 3. The columns under the heading `Root LP` give the number of p-cycle variables generated by the column generation algorithm and the time to solve the LP relaxation (in seconds) at the root node of the branch-and-bound tree. Compared to the overall solution time, the time spent for solving the root LP relaxation, hence solving PPCs for pricing p-cycle variables, is quite small.

The columns under the heading `Default` show the performance of default CPLEX without adding any of the cutting planes described in the paper; though CPLEX adds its own cuts. The gap improvement shown here is the percentage of the LP gap closed by the CPLEX cuts. Finally, under the heading `With Cuts`, we report the number of cuts added for each class (16), (17), (6), the percentage of the LP gap closed, the number of branch-and-bound nodes, and the total time spent (in seconds) when the cutting planes are used in the computations. With the addition of partition cuts, the improvement in the LP gap increases from an average of 21%–58%, which reduces the number of branch-and-bound nodes explored and the total solution time significantly. We should emphasize that the reported LP gap reduction with the cuts is with respect to the restricted formulation, which contains only the p-cycle variables generated before the branch-and-cut algorithm starts.

We observe that many more p-cycle flow partition cuts are added compared to subset-Q cuts. Additional computations (not reported in the table) without p-cycle flow cuts indicate that p-cycle flow partition cuts are effective in reducing the computational effort especially for the larger problems even though the incremental gap improvement may not be large. The importance of being able to improve the LP relaxation becomes clear, especially, in the case of the instance with 12 nodes, for which the cuts were not as effective as for other instances; hence the corresponding abnormally large solution time.

In Table 4 we report the results of the experiments for graphs with degree 8. Three of the largest instances could not be solved with default CPLEX within the limits of the experiments. Two of these three could not be solved with

Table 3
Experiments with degree 4 graphs

$\|N\|$	Root LP		Default			With Cuts					
	pcyc vars	cpu sec	% gap impr	b&b nodes	cpu sec	surv part	surv subQ	pcyc flow	% gap impr	b&b nodes	cpu sec
5	28	0	28	100	0	6	2	0	67	28	0
6	40	0	35	214	0	20	2	10	88	17	0
7	46	0	20	619	0	7	0	7	73	149	0
8	48	0	39	758	1	9	4	3	70	327	0
9	74	0	22	3 096	6	6	6	7	60	1 057	2
10	62	0	20	8 401	16	15	0	3	53	2 481	5
11	98	1	24	4 131	19	7	0	4	53	1 742	8
12	114	2	15	628 363	2893	5	2	16	29	480 423	2543
13	148	3	12	71 411	434	7	2	9	47	4 470	37
14	168	4	8	38 450	210	10	0	9	67	2 325	15
15	140	5	14	170 240	1032	3	1	3	50	81 722	544
16	152	5	17	144 697	1073	9	1	5	42	20 692	194
17	172	10	24	80 908	895	6	0	3	53	41 843	488

Table 4
Experiments with degree 8 graphs

$\|N\|$	Root LP		Default			With Cuts					
	pcyc vars	cpu sec	% gap impr	b&b nodes	cpu sec (egap)	surv part	surv subQ	pcyc flow	% gap impr	b&b nodes	cpu sec (egap)
5	28	0	28	100	0	6	2	0	67	28	0
6	38	0	17	148	0	21	0	7	100	2	0
7	58	0	27	1 217	2	8	4	6	80	22	0
8	58	0	28	1 314	3	10	14	6	61	540	2
9	86	0	27	13 989	41	14	1	4	45	2 278	9
10	94	1	20	348 546	1371	14	8	6	65	15 542	76
11	116	1	35	200 984	1176	6	4	4	50	12 007	87
12	130	2	29	929 762	7264	5	0	3	38	75 748	615
13	144	4	29	998 047	10 622	10	6	10	65	35 626	420
14	160	5	27	34 665	303	10	7	2	36	12 424	138
15	156	8	18	1000 000	(0.31)	7	0	3	41	1000 000	(0.24)
16	206	12	20	1000 000	(0.68)	9	0	11	50	388 676	6632
17	204	13	26	765 401	(0.4)	4	3	5	47	821 301	(0.34)

the addition of the cuts either. We report the end gap (egap), the gap between best known upper bound and lower bound, for these problems (with the subset of p-cycles included in the formulation) instead of the solution time. For the smaller problems that were solved also by default CPLEX, a comparison of branch-and-bound nodes and solutions times shows that the partition cuts lead to a substantial reduction in the computational effort.

In Table 5 we report the results of the experiments for graphs with 75% edge density. The positive effect of the partition cuts is also apparent for this case. The cutting planes improve the LP gap and reduce the computation time significantly. Four of the larger instances could not be solved with either default CPLEX or with the addition of the cuts within the limits of the experiments. However, the end gap is generally smaller with the cuts.

5. Concluding remarks

We presented a polyhedral study of a model for designing capacitated networks that can survive edge failures by explicitly reserving slack on p-cycles of the underlying directed graph. Even though the disrupted flow only is rerouted, the capacity requirement for the model is close to the one achieved by global rerouting models. The minimum capacity requirement over partitions of the networks achieved by the proposed model equals the one for global rerouting.

We derived strong valid inequalities based on survivability conditions for flows and p-cycles across partitions of the network. The validity of the inequalities are proved via mixed-integer rounding arguments. Alternative proofs based

Table 5
Experiments with 75% density graphs

$\|N\|$	Root LP		Default			With Cuts					
	pcyc vars	cpu sec	% gap impr	b&b nodes	cpu sec (egap)	surv part	surv subQ	pcyc flow	% gap impr	b&b nodes	cpu sec (egap)
5	28	0	28	100	0	6	2	0	67	28	0
6	40	0	35	214	0	20	2	10	88	17	0
7	46	0	20	619	0	7	0	7	73	149	0
8	56	0	28	1 527	3	11	9	5	61	381	1
9	70	0	42	4 560	10	9	5	9	54	5 649	14
10	72	1	24	52 506	140	21	21	8	76	2 756	12
11	106	1	32	65 897	348	7	2	5	53	10 940	72
12	134	3	20	1000 000	(0.6)	6	0	5	35	1000 000	(0.1)
13	134	3	38	778 914	8384	8	8	2	48	444 083	6144
14	142	3	33	313 796	3704	13	8	7	50	32 067	431
15	182	10	28	843 201	(0.4)	10	1	4	50	770 546	(0.5)
16	208	12	30	579 801	(0.8)	11	7	7	57	593 801	(0.2)
17	210	12	26	474 101	(0.8)	12	8	3	43	479 521	(0.7)

on strengthening of metric inequalities with survivability restrictions are given in Rajan [31]. The computational experiments show clearly the effectiveness of the partition cuts in reducing the computational effort of a branch-and-cut algorithm.

In this study we assumed that the network had no existing capacity and that a single type of facility was available. If there are multiple types of facilities, inequalities in this paper can be generalized using similar arguments as in Atamtürk [3], which gives inequalities for network design problems with no survivability requirement for an arbitrary number of facilities with varying capacities. Existing capacities can be handled indirectly by introducing a new facility variable and fixing it to one or directly using mixed-integer rounding as done in Bienstock and Günlük [11]. The other assumption we made in the paper was that installed capacity on an edge could serve flow in both directions up to this capacity. If this is not the case, i.e., if capacities must be installed separately in each direction, we may do so by duplicating each edge and fixing to zero one of the flow variables in reverse direction for each copy.

Acknowledgments

We are thankful to two anonymous referees whose thoughtful suggestions proved to be valuable in improving the presentation of this paper.

References

[1] D. Alevras, M. Grötschel, R. Wessäly, Cost-efficient network synthesis from leased lines, Annals of Operations Research 76 (1998) 1–20.
[2] K. Altinkemer, Topological design of ring networks, Computers and Operations Research 21 (1994) 421–431.
[3] A. Atamtürk, On capacitated network design cut-set polyhedra, Mathematical Programming 92 (2002) 425–437.
[4] A. Atamtürk, D. Rajan, On splittable and unsplittable flow capacitated network design arc-set polyhedra, Mathematical Programming 92 (2002) 315–333.
[5] P. Avella, S. Mattia, A. Sassano, Metric inequalities and the network loading problem, in: D. Bienstock, G. Nemhauser (Eds.), in: Lecture Notes in Computer Science, vol. 3064, Springer-Verlag, 2004, pp. 16–32.
[6] A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Network design, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, John Wiley & Sons, 1997.
[7] A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Connectivity-splitting models for survivable network design, Networks 43 (2004) 10–27.
[8] A. Balakrishnan, T.L. Magnanti, J.S. Sokol, Y. Wang, Modeling and solving the single facility line restoration problem, Operations Research 50 (2002) 617–635.
[9] F. Barahona, Network design using cut inequalities, SIAM Journal on Optimization 6 (1996) 823–837.
[10] D. Bienstock, 2001. Personal communication.
[11] D. Bienstock, O. Günlük, Capacitated network design — Polyhedral structure and computation, INFORMS Journal on Computing 8 (1996) 243–259.
[12] D. Bienstock, G. Muratore, Strong inequalities for capacitated survivable network design problems, Mathematical Programming 89 (2000) 127–147.
[13] S. Chopra, I. Gilboa, S.T. Sastry, Source sink flows with capacity installation in batches, Discrete Applied Mathematics 85 (1998) 165–192.

[14] S. Cosares, D.N. Deutsch, I. Saniee, O.J. Wasem, SONET toolkit: A decision support system for designing robust and cost-effective fiber-optic network, Interfaces 25 (1995) 20–40.
[15] G. Dahl, M. Stoer, A cutting plane algorithm for multicommodity survivable network design problems, INFORMS Journal on Computing 10 (1998) 1–11.
[16] G.B. Dantzig, Linear programming under uncertainty, Management Science 1 (1955) 197–206.
[17] O. Goldschmidt, A. Laugier, E.V. Olinick, SONET/SDH ring assignment with capacity constraints, Discrete Applied Mathematics 129 (2003) 99–128.
[18] W. Grover, D. Stamatelakis, Cycle-oriented distributed pre-configuration: Ring-like speed with mesh-like capacity for self-planning network restoration, in: Proceedings of IEEE International Conference on Communications, 1998, pp. 537–543.
[19] W.D. Grover, R.G. Martens, Optimized design of ring-mesh hybrid networks, in: Proceedings of IEEE/VDE Design of Reliable Communication Networks 2000, 2000, pp. 291–297.
[20] O. Günlük, A branch-and-cut algorithm for capacitated network design problems, Mathematical Programming 86 (1999) 17–39.
[21] M. Herzberg, S.J. Bye, A. Utano, The hop-limit approach for spare capacity assignment in survivable networks, IEEE/ACM Transactions on Networking 3 (1995) 775–784.
[22] M. Iri, On an extension of the max-flow min-cut theorem to multicommodity flows, Journal of the Operations Research Society of Japan 13 (1971) 129–135.
[23] J.L. Kennington, M.W. Lewis, The path restoration version of the spare capacity allocation problem with modularity restrictions: Models, algorithms, and an empirical analysis, INFORMS Journal on Computing 13 (2001) 181–190.
[24] A. Lisser, R. Sarkissian, J.P. Vial, Survivability in telecommunication networks, Technical Report 1995.3, University of Geneva, 1995.
[25] T.L. Magnanti, P. Mirchandani, Shortest paths, single origin-destination network design, and associated polyhedra, Networks 23 (1993) 103–121.
[26] T.L. Magnanti, P. Mirchandani, R. Vachani, The convex hull of two core capacitated network design problems, Mathematical Programming 60 (1993) 233–250.
[27] T.L. Magnanti, P. Mirchandani, R. Vachani, Modeling and solving the two-facility capacitated network loading problem, Operations Research 43 (1995) 142–157.
[28] T.L. Magnanti, S. Raghavan, Strong formulations for network design problems with connectivity requirements, Networks 45 (2005) 61–79.
[29] G.L. Nemhauser, L.A. Wolsey, A recursive procedure for generating all cuts for 0-1 mixed integer programs, Mathematical Programming 46 (1990) 379–390.
[30] K. Onaga, O. Kakusho, On feasibility conditions of multi-commodity flows in networks, Transactions on Circuit Theory 18 (1971) 425–429.
[31] D. Rajan, Designing capacitated survivable networks: Polyhedral analysis and algorithms, Ph.D. Thesis, University of California-Berkeley, Berkeley, USA, 2004.
[32] D. Rajan, A. Atamtürk, Survivable network design: Routing of flows and slacks, in: G. Anandalingam, S. Raghavan (Eds.), Telecommunications Network Design and Management, Kluwer Academic Publishers, 2002, pp. 65–81.
[33] D. Rajan, A. Atamtürk, A directed cycle based column-and-cut generation method for capacitated survivable network design, Networks 43 (2004) 201–211.
[34] D.A. Schupke, G.C. Gruber, A. Autenrieth, Optimal configuration of p-cycles in WDM networks, in: IEEE International Conference on Communications, 2002.
[35] J.B. Slevinsky, W.D. Grover, M.H. MacGregor, An algorithm for survivable network design employing multiple self-healing rings, in: Proceedings of IEEE GLOBECOM 1993, 1993, pp. 1568–1572.
[36] P. Soriano, C. Wynants, R. Séguin, M. Labbé, M. Gendreau, B. Fortz, Design and dimensioning of survivable SDH/SONET networks, in: B. Sansò, P. Soriano (Eds.), Telecommunications Network Planning, Kluwer Academic Publishers, Netherlands, 1998, pp. 147–168.
[37] D. Stamatelakis, W.D. Grover, Theoretical underpinnings for the efficiency of restorable networks using pre-configured cycles ("p-cycles"), IEEE Transactions on Communications 48 (2000) 1262–1265.
[38] S.P.M. van Hoesel, A.M.C.A. Koster, R.L.M.J. van de Leensel, M.W.P. Savelsbergh, Bidirected and unidirected capacity installation in telecommunication networks, Discrete Applied Mathematics 133 (2004) 103–121.
[39] B. Venables, W.D. Grover, M.H. MacGregor, Two strategies for spare capacity placement in mesh restorable networks, in: Proceedings of IEEE International Conference on Communications 1993, 1993, pp. 267–271.
[40] L.A. Wolsey, Integer Programming, John Wiley and Sons, New York, 1998.
[41] Y. Xiong, L.G. Mason, Restoration strategies and spare capacity requirements in self-healing ATM networks, IEEE/ACM Transactions on Networking 7 (1999) 98–110.