Polyhedral and Computational Investigations for Designing Communication Networks with High Survivability Requirements

by
M. Grötschel*
C.L. Monma**
M. Stoer*

August 1992

*Konrad-Zuse-Zentrum für
Informationstechnik Berlin
Heilbronner Str. 10
1000 Berlin 31, Germany

**Bellcore
445 South Street
Morristown, New Jersey 07960

Abstract

We consider the important practical and theoretical problem of designing a low-cost communications network which can survive failures of certain network components. Our initial interest in this area was motivated by the need to design certain "two-connected" survivable topologies for fiber optic communication networks of interest to the regional telephone companies. In this paper, we describe some polyhedral results for network design problems with higher connectivity requirements. We also report on some preliminary computational results for a cutting plane algorithm for various real-world and random problems with high connectivity requirements which shows promise for providing good solutions to these difficult problems.

1 Introduction

This paper focuses on the important practical and theoretical problem of designing survivable communication networks. Our initial interest in this area was motivated by the problem of designing survivable topologies for fiber optic communication networks for the regional telephone companies; see [BC88] and [CMW89] for an overview of this application and a description of a software tool developed at Bellcore. This application requires certain "two-connected" topologies so that special offices can communicate after the failure of any single network link or node.

Our earlier work on two-connected network design problems included structural properties and worst-case analysis of heuristics [MMP90], practical heuristics [MS89], polyhedral results [GM90, GMS92b], and computation with a cutting plane algorthm [GMS92a]. This naturally leads to theoretical and algorithmic questions for network design problems with higher survivability requirements. Structural properties were considered in [BBM90], and practical heuristics were considered in [KM89].

In this paper, we describe polyhedral results, including natural integer programming formulations, classes of valid and facet-defining inequalities, and their associated separation problems for network design problems with higher connectivity requirements. We also report computational results with a cutting plane algorithm on some random problems and on some real-world problems with high survivability demands. The real-world problems we obtained turned out to be quite difficult; indeed, more difficult than we expected based on our success in solving network design problems for the telephone companies. There seem to be several reasons for this: (a) the networks are large (almost 500 nodes), sparse, and have many Steiner nodes; (b) connectivity requirements of three or more seem to be much more difficult than two-connected problems; and (c) the cost structure is very regular which makes proving optimality rather difficult.

In Section 2, we describe the model of network survivability that we have in mind. The polyhedral results are presented in Section 3. The cutting plane implementation and computational results are presented in Section 4.

2 Models of Network Survivability

In this section, we formalize the survivable network design problem that is being considered in this paper. In order to do this, we need to introduce the following notation.

A set V of nodes is given which represent the locations that must be interconnected into a network in order to provide the desired services. A collection E of edges is also specified that represent the possible pairs of nodes between which a direct link can be placed. We let $G = (V, E)$ be the (undirected) graph of possible direct link connections. Each edge $e \in E$ has a nonnegative **fixed cost** c_e of establishing the direct link connection. The graph G may have parallel edges but contains no loops. The cost of establishing a network consisting of a subset $F \subseteq E$ of edges is the sum of the costs of the individual links contained in F. The goal is to build a minimum-cost network so that the required survivability conditions, which we describe below, are satisfied.

If $G = (V, E)$ is a graph, $W \subseteq V$ and $F \subseteq E$ then we denote by $G - W$ and $G - F$ the graph that is obtained from G by **deleting** the node set W and the edge set F, respectively. For notational convenience we write $G - v$ and $G - e$ instead of $G - \{v\}$ and $G - \{e\}$, respectively. The difference of two sets M and N is denoted by $M \backslash N$.

For any pair of distinct nodes $s, t \in V$, an $[s, t]$-**path** P is a sequence of nodes and edges $(v_0, e_1, v_1, e_2, \ldots, v_{l-1}, e_l, v_l)$, where each edge e_i is incident with the nodes v_{i-1} and v_i ($i = 1, \ldots, l$), where $v_0 = s$ and $v_l = t$, and where no node or edge appears more than once in P. A collection $P_1, P_2 \ldots, P_k$ of $[s, t]$-paths is called **edge-disjoint** if no edge appears in more than one path, and is called **node-disjoint** if no node (except for s and t) appears in more than one path. In particular, this implies that parallel edges between a pair of nodes s and t are considered to be node-disjoint $[s,t]$-paths. This is in contrast to standard graph theory; but this modification of the definition was prompted by considerations that arise in practical situations. We say that a graph $G = (V, E)$ is **k-edge (resp. k-node) connected** if there are k-edge (resp. k-node) disjoint paths between every pair of distinct nodes. We note again that this definition of k-node connectivity differs from that of standard graph theory when parallel edges are present.

We measure connectivity not only between two nodes s and t but also between nodes lying in a given node set W. Let $G = (V, E)$ be a graph with at least two nodes and $W \subseteq V$ with $|W| \geq 2$. We set

$\lambda(G, W)$	$:=$	minimum cardinality of a subset F of E, such that two nodes of W are disconnected in $G - F$, and
$\kappa(G, W)$	$:=$	minimum cardinality of a set $S \cup F$, where $S \subseteq V$ and $F \subseteq E$, such that two nodes of W are disconnected in $G - (S \cup F)$.

If $|W| < 2$ or if G has only one node then $\lambda(G, W)$ and $\kappa(G, W)$ are defined to be ∞.

The survivability conditions require that the network satisfy certain edge or node connectivity requirements. To model these survivability conditions, we introduce the concept of node types. For each node $s \in V$ a nonnegative integer r_s, called the **type** of s, is specified.

We say that the network $N = (V, F)$ to be designed satisfies the **node survivability conditions** if, for each pair $s, t \in V$ of distinct nodes, N contains at least $r_{st} := \min\{r_s, r_t\}$ node disjoint $[s,t]$-paths. Similarly, we say that $N = (V, F)$ satisfies the **edge survivabiity**

conditions if, for each pair $s, t \in V$ of distinct nodes, N contains r_{st} edge disjoint $[s,t]$-paths. These conditions ensure that some path between s and t will survive a prespecified level of node or link failures. We introduce further symbols and conventions to denote these node- or edge-survivability models.

Given a graph $G = (V, E)$ and a vector of node types $r = (r_s)_{s\in V}$, we assume — without loss of generality — that there are at least two nodes of the largest type. If we say that we consider the kNCON problem (for G and r) then we mean that we are looking for a minimum-cost network that satisfies the node survivability conditions and where $k = \max\{r_s | s \in V\}$. Similarly, we speak of the kECON problem (for G and r). Furthermore we define

$$\begin{aligned} r(W) &:= \max\{r_u : u \in W\}; \text{ and} \\ \operatorname{con}(W) &:= \min\{r(W), r(V-W)\} \\ &= \max\{r_{st} : s \in W, t \in V - W\} \end{aligned}$$

for any $W \subseteq V$. Let $G = (V, E)$ be a graph. For $Z \subseteq V$, let $\delta_G(Z)$ denote the set of edges with one end node in Z and the other in $V \backslash Z$. It is customary to call $\delta_G(Z)$ a **cut**. If it is clear with respect to which graph a cut $\delta_G(Z)$ is considered, we simply drop the index and write $\delta(Z)$. We also write $\delta(v)$ for $\delta(\{v\})$. If X, Y are subsets of V and $X \cap Y = \emptyset$, we set $[X : Y] := \{ij \in E \mid i \in X, j \in Y\}$; thus $\delta(X) = [X : V \backslash X]$. If $W_1 \ldots, W_p$ are pairwise disjoint subsets of V we denote by $[W_1 : \cdots : W_p]$ the set of edges having their end nodes in different node sets. For any subset of edges $F \subseteq E$, we let $x(F)$ stand for the sum $\sum_{e\in F} x_e$. Consider the following integer linear program for a graph $G = (V, E)$ with edge costs c_e for all $e \in E$ and node types r_s for all $s \in V$:

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

subject to

(i)	$x(\delta(W)) \geq \text{con}(W)$	for all $W \subseteq V, \emptyset \neq W \neq V$;
(ii)	$x(\delta_{G-Z}(W)) \geq \text{con}(W) - \lvert Z \rvert$	for all pairs $s, t \in V, s \neq t$, and for all $Z \subseteq V \backslash \{s,t\}$ with $1 \leq \lvert Z \rvert \leq r_{st} - 1$, and for all $W \subseteq V \backslash Z$ with $s \in W, t \notin W$:
(iii)	$0 \leq x_e \leq 1$	for all $e \in E$;
(iv)	x_e integral	for all $e \in E$.

It follows from Menger's theorem that the feasible solutions of (2.1) are the incidence vectors of edge sets F such that $N = (V, F)$ satisfies all node survivability conditions; i.e., (2.1) is an integer programming formulation of the kNCON problem. Deleting inequalities (ii) from (2.1) we obtain, again from Menger's theorem, an integer programming formulation for the kECON problem. The inequalities of type (i) will be called **cut inequalities** and those of type (ii) will be called **node cut inequalities**.

The polyhedral approach to the solution of the kNCON (and similarly the kECON) problem consists of studying the polyhedron obtained by taking the covex hull of the feasible solutions of (2.1). We set

$$k\text{NCON}(G; r) := \text{conv}\,\{x \in \mathbf{R}^E \mid x \text{ satisfies (2.1) (i), (ii), (iii), (iv)}\}$$
$$k\text{ECON}(G; r) := \text{conv}\,\{x \in \mathbf{R}^E \mid x \text{ satisfies (2.1) (i),(iii),(iv)}\}.$$

3 Polyhedral Results

We briefly mention the idea behind the polyhedral approach and its goal. We consider an integer programming problem like (2.1). We want to turn such an integer program into a linear program and solve it using LP-techniques. To do this, we define a polytope associated with the problem by taking the convex hull of the feasible (integral) solutions of a program like (2.1). Let P be such a convex hull. We know from linear programming theory that, for any objective function c, the linear program $\min c^T x, x \in P$ has an optimum vertex solution (if it has a solution). This vertex solution is a feasible solution of the initial integer program and thus, by construction, an optimum solution of this program.

The difficulty with this approach is that $\min c^T x, x \in P$ is a linear program only "in principle". To provide an instance to an LP-solver, we have to find a different description of P. The polytope P is defined as the convex hull of (usually many) points in $\mathbf{R}^E$, but we need a complete (linear) description of P by means of linear equations or inequalities. The Weyl-Minkowski theorem tells us that both descriptions are in a sense equivalennt, in fact, there are constructive procedures that compute one description of P from the other. However, these procedures are inherently exponential and nobody knows how to make effective use of them, in particular, for NP-hard problem classes.

At present, no effective general techniques are known for finding complete or "good partial" descriptions of such a polytope or large classes of facets. There are a few basic techniques like the derivation of so-called Chvátal cuts (see [C73]). But most of the work is a kind of "art". Valid inequalities are derived from structural insights and the proofs that many of these inequalities define facets use technically-complicated ad-hoc arguments.

If large classes of valid and possibly facet-defining inequalities are found, one tries to use them algorithmically in the framework of a cutting plane algorithm. The principal idea here is the following.

Note that — except for the trivial inequalities — the classes of valid inequalities for the ECON and NCON problem stated in (2.1) contain a number of inequalities that is exponential in the number of nodes of the given graph. Instead of solving an LP with all these inequalities, we solve one with a few "carefully selected" inequalities and we generate new inequalities as we need them. The main difficulty of this approach is in efficiently generating violated inequalities. We state this task formally.

(3.1) Separation Problem (for a class C of inequalities)

Given a vector y decide whether y satisfies all inequalities in C and, if not, output an inequality violated by y. □

We call an algorithm that solves (3.1) an (exact) **separation algorithm** for C, and we say that it runs in polynomial time if its running time is bounded by a polynomial in $|V|$ and the encoding length of y.

A deep result of the theory of linear programming (see Grötschel, Lovász, Schrijver [GLS88]) states (roughly) that a linear program over a class C of inequalities can be solved in polynomial time if and only if the separation problem for C can be solved in polynomial time. Being able to solve the separation problem thus has considerable theoretical consequences.

This result makes use of the ellipsoid method and does not imply the existence of a "practically efficient" algorithm. However, by combining separation algorithms with other LP solvers (like the simplex algorithms) can result in quite successful cutting plane algorithms; see Section 4.

We will now describe several classes of valid inequalities and indicate conditions under which some of these inequalities define facets. It turns out that results of this type are very technical.

3.1 Dimension and Trivial Inequalities

Given a graph G and node types r, we say that an edge e **is essential with respect to** $k\mathrm{ECON}(G;r)$ or $k\mathrm{NCON}(G;r)$ if $k\mathrm{ECON}(G-e;r)$ or $k\mathrm{NCON}(G-e;r)$, resp., is empty. We denote by $k\mathbf{EES}(G;r)$ the set of edges essential for $k\mathrm{ECON}(G;r)$ and by $k\mathbf{NES}(G;r)$ the set of edges essential for $k\mathrm{NCON}(G;r)$.

(3.2) Theorem. Let $G = (V, E)$ be a graph and let $r \in \{0, 1, \ldots, k\}^V$ be given.

(a) Suppose kECON$(G; r)$ is nonempty. Then $k\text{ECON}(G; r) \subseteq \{x \in \mathbf{R}^E \mid x_e = 1 \text{ for all } e \in k\text{EES}(G; r)\}$ and $\dim (kECON(G; r)) = |E| - |kEES(G; r)|$.

(b) Suppose kNCON$(G; r)$ is nonempty. Then $k\text{NCON}(G; r) \subseteq \{x \in \mathbf{R}^E \mid x_e = 1 \text{ for all } e \in k\text{NES}(G; r)\}$ and $\dim (k\text{NCON}(G; r)) = |E| - |k\text{NES}(G; r)|$.

Proof. We only prove (a). (The proof of (b) is analogous). If e is an essential edge then clearly $x_e = 1$ for every incidence vector of a feasible graph, hence for all vectors $x \in k\text{ECON}(G; r)$. So kECON$(G; r)$ can have dimension at most $|E| - |k\text{EES}(G; r)|$. The incidence vectors of $E, E \setminus \{e\}$, for $e \notin k\text{EES}(G; r)$ are linearly independent and feasible. Therefore kECON$(G; r)$ has dimension exactly $|E| - |k\text{EES}(G; r)|$. $\square$

We note that the problem of finding the set of essential edges and thus determining the dimension of kECON$(G; r)$ and kNCON$(G; r)$ can be solved in polynomial time using network flow or graph connectivity algorithms.

The following theorem is a special case of a theorem of Grötschel and Monma [GM90] that characterizes which of the trivial inequalities (2.1)(iii) define facets of the kECON$(G; r)$ and the kNCON$(G; r)$ polytopes.

(3.3) Theorem. Let $G = (V, E)$ be a graph and $r \in \mathbf{Z}_+^V$ such that kECON$(G; r)$ (resp., kNCON$(G; r)$) is full-dimensional. Then

(a) $x_e \leq 1$ defines a facet of kECON$(G; r)$ (resp., kNCON$(G; r)$) for all e;

(b) $x_e \geq 0$ defines a facet of kECON$(G; r)$ (resp., kNCON$(G; r)$) if and only if for every edge $f \neq e$ the polytope kECON$(G - \{e, f\}; r)$ (resp., kNCON$(G - \{e, f\}; r)$) is nonempty.

Proof. We prove this only for kECON$(G;r)$. (For kNCON$(G;r)$, the proof is exactly the same.) Note that the full-dimensionality of kECON$(G;r)$ implies that kEES$(G;r)$ is empty.

(a) The $|E|$ incidence vectors of the sets $E, E\setminus\{f\}$ for all $f \neq e$ are linearly independent, feasible, and all of them satisfy $x_e = 1$, so the dimension of the face induced by $x_e \leq 1$ is at least $|E|-1$. But $x_e = 1$ does not hold for all $x \in k$ECON$(G;r)$, so $x_e \leq 1$ defines a facet.

(b) If the polytope kECON$(G-\{e,f\};r)$ is empty for some f, then $x_e = 0$ implies $x_f = 1$ for all $x \in k$ECON$(G;r)$, but not reversely. Therefore, the face defined by $x_e \geq 0$ is strictly contained in the face defined by $x_f \leq 1$, which is a facet by (a). So $x_e \geq 0$ does not define a facet. If kECON$(G-\{e,f\};r)$ is nonempty for all f, then kEES$(G-e;r)$ is empty, hence kECON$(G-e;r)$ is full-dimensional. Thus there are $\dim(k\mathrm{ECON}(G;r))$ affinely independent vectors in kECON$(G-e;r)$. These vectors, enlarged by a component e of value 0, lie in the face defined by $x_e \geq 0$. Since this face cannot be the whole kECON$(G;r)$ polytope, it is a facet. □

3.2 Cut Inequalities

In this section, we discuss the **cut inequalities** (2.1)(i). $x(\delta(W)) \geq \mathrm{con}(W)$ for all $W \subseteq V$, $\emptyset \neq W \neq V$. The separation problem for cut inequalities is NP-hard if we allow the given vector y to have negative entries. The reason is that the max-cut problem can be reduced to it. But separation can be performed in polynomial time for nonnegative vectors (we are only interested in $0 \leq y \leq 1$) by using network flow or graph connectivity algorithms as follows.

Let us consider the class $\mathcal{C}$ of cut inequalities (2.1)(i) and let y be some vector in $\mathbf{R}^E$ with $0 \leq y_e$ for all $e \in E$. We view the components y_e as capacities of the edges of the given graph $G=(V,E)$ and compute a Gomory-Hu tree (using the Gusfield version that consists of $|V|-1$ calls of a max-flow algorithm and some bookkeeping; see [GH61, G87]). The Gomory-Hu tree has the property that, for any two nodes $u, v \in V$, the minimum capacity of a cut separating u and v is given by the smallest weight of an edge that is contained in the unique path linking u and v in the Gomory-Hu tree. Having the Gomory-Hu tree T with weights w_e for all $e \in T$ we can determine whether y satisfies all cut inequalities as follows. For each edge e, let V_e and V_e' denote the node sets of the two components of $(V, T-e)$. If there is an edge $e \in E$ with

$$w_e < \min\{\max\{r_v \mid v \in V_e\}, \max\{r_v \mid v \in V_e'\}\}$$

then y violates the inequality $x(\delta(V_e)) \geq \mathrm{con}(W)$, otherwise y satisfies all cut inequalities. Gomory-Hu trees can be computed quite efficiently in practice. Thus the class of cut inequalities is very useful for the cutting plane approach from the algorithmic point of view. The class of cut inequalities also contains many facet-defining inequalities, which theoretically justifies the utilization of cut inequalities.

Recently, Hao and Orlin [HO92] devised an algorithm for finding a cut of minimum capacity in a graph, which has the same running time as the max-flow algorithm. But it does not output a Gomory-Hu tree, which is needed in some of our separation routines for the more general class of partition inequalities described in Section 3.4.

The following theorem was proven by Stoer [S92] and gives necessary and sufficient conditions for a cut inequality to define a facet for $k\mathrm{ECON}(G;r)$ when G is $(k+1)$-edge connected and where all nodes have the same type k.

(3.4) Theorem. Let $G=(V,E)$ be a $(k+1)$-edge connected graph, let $r_v = k$ for all nodes $v \in V$, and let $W \neq V$ be a nonempty node set. Define for each $W_i \subseteq W$ with $\emptyset \neq W_i \neq W$ the deficit of W_i as

$$\mathrm{def}_G(W_i) := \max\{0, k - |\delta_{G[W]}(W_i)|\}.$$

Define similarly for $U_i \subseteq V \backslash W$ with $\emptyset \neq U_i \neq V \backslash W$

$$\mathrm{def}_G(U_i) := \max\{0, k - |\delta_{G[V \backslash W]}(U_i)|\}.$$

The cut inequality

$$x(\delta(W)) \geq k$$

defines a facet of the polytope kECON$(G; r)$ of k-edge connected graphs if and only if

(a) $G[W]$ and $G[V \backslash W]$ are connected, and

(b) for all edges $e \in E(W) \cup E(V \backslash W)$,

for all pairwise disjoint nodesets $W_1, \ldots, W_p$ $(p \geq 0)$ of W with $W_i \neq W$ for all i, and for all pairwise disjoint nodesets $U_1, \ldots, U_q$ $(q \geq 0)$ of $V \backslash W$ with $U_i \neq V \backslash W$ for all i, the following inequality holds:

$$\sum_{i=1}^{p} \mathrm{def}_{G-e}(W_i) + \sum_{i=1}^{q} \mathrm{def}_{G-e}(U_i) - \left|[\bigcup_{i=1}^{p} W_i : \bigcup_{i=1}^{q} U_i]\right| \leq k. \quad \square$$

We refer to [S92] for the details of the proof of (2.3) that is rather long and involved. We present a corollary that provides some simple minimal sufficient and some maximal necessary conditions on the connectivity of $G[W]$ and $G[V - W]$; these were originally proven in [GM90].

(3.5) Corollary. Let $G = (V, E)$ be a $(k+1)$-edge connected graph, let $r_v = k$ for all nodes $v \in V$, and let $W \neq V$ be nonempty node set.

(a) If $G[W]$ or $G[V \backslash W]$ is at most $\lceil k/2 \rceil$-edge connected then $x(\delta(W)) \geq k$ does not define a facet of kECON$(G; r)$.

(b) If $G[W]$ and $G[V \backslash W]$ are k-edge connected then $x(\delta(W)) \geq k$ defines a facet of kECON$(G; r)$.

For the case of kECON$(G; r)$ and kNCON$(G; r)$ problems with arbitrary node types, we do not know of any general results characterizing those cut inequalities that are facet-defining. However, in the case of 2ECON$(G; r)$ and 2NCON$(G; r)$ problems, i.e., where $r_v \in \{0, 1, 2\}$ for all nodes v, we do know necessary and coefficient conditions; see [GMS92b].

3.3 Node Cut Inequalities

In this section, we discuss the node cut inequalities (2.1)(ii).

$$
x(\delta_{G-Z}(W)) \geq \mathrm{con}(W) - |Z| \quad \begin{array}{l} \text{for all pairs } s,t \in V, s \neq t \\ \text{for all } Z \subseteq V\backslash\{s,t\} \text{ with } 1 \leq |Z| \leq r_{st} - 1, \text{and} \\ \text{for all } W \subseteq V\backslash Z \text{ with } s \in W, t \notin W. \end{array}
$$

These inequalities are valid for $k\mathrm{NCON}(G;r)$ but are not generally valid for $k\mathrm{ECON}(G;r)$.

As in the case of cut inequalities, the separation problem for node cut and for cut inequalities is NP-hard if the given vector y is allowed to have negative entries. The separation problem is solvable in polynomial time if we restrict the input to nonnegative vectors. The algorithm works as follows.

Consider the class $\mathcal{C}$ of node cut inequalities (2.1)(ii) and of cut inequalities (2.1)(i), let y be some vector in $\mathbf{R}^E$ with $0 \leq y$ for all $e \in E$, and let $G = (V, E)$ be the given graph. We construct a directed graph $D = (N, A)$ from G in the following way. Each node $v \in V$ is replaced by two nodes v', v'' of N that are linked by an arc (v', v'') with capacity one. Each edge $uv \in E$ is replaced by two arcs (u'', v') and (v'', u') each with capacity y_e. Let $S \subseteq V$ be a set of k nodes of maximal type, i.e., $r_s \geq r_t$ for all $s \in S$ and $t \in V\backslash S$. For all $s \in S$ and all $t \neq s$ with $r_{st} \geq 1$ we compute a minimum capacity cut C' of value c' separating s'' and t'. Such a cut is of the form $\delta^+(W') := \{(i,j) : i \in W', j \in N\backslash W'\}$, where $W' \subseteq N$ and $s'' \in W'$. The arcs in C' correspond either to arcs of the form (v', v'') for some $v \in V$ or to arcs of the type (u'', w') for some edge $uw \in E$. Without loss of generality, if $v'' \in W'$ and $v' \notin W'$ then either v'' can be moved to $N\backslash W'$ or v' can be moved to W'. We let

$$
\begin{aligned}
Z &:= \{v : (v', v'') \in C'\}, \\
C &:= \{uw \in E : (u'', w') \in C'\}, \text{and} \\
W &:= \{v \in V : v', v'' \in W'\}.
\end{aligned}
$$

If $c' < r_{st}$ then we have found a node cut inequality violated by y, namely where s, t, Z and W are as defined above and $C = \delta_{G-Z}(W)$. Possibly $Z = \emptyset$, in this case the cut inequality $x(\delta_G(W)) \geq r_{st}$ is violated. If $c' \geq r_{st}$ for all $s \in S$ and $t \neq s$ with $r_{st} \geq 1$ then there are no node cut inequalities violated by y.

We know of no general necessary and sufficient conditions for node cut inequalities to define facets of kNCON$(G;r)$, except in the low-connectivity case, i.e., where $r_v \in \{0,1,2\}$ for all $v \in V$; see [GMS92b]. Some (complicated) necessary conditions and sufficient conditions for the general problem are derived in [S92]. We state these results here omitting the proofs.

(3.6) Theorem. Let $G = (V, E)$ and $r \in \mathbf{Z}_+^V$ be given. Let Z, W, s, t define a node cut inequality (2.1)(i) given by

$$x(\delta_{G-Z}(W)) \geq \text{con}(W) - |Z|.$$

(a) This inequality defines a facet of kNCON$(G;r)$ if the following conditions hold, where $V_j := \{u \in V \mid r_v \geq j\}$ for $j = l, \ldots, k$:

(i) $\kappa(G[W] \cup Z] - e, V_j) \geq j$ for all $e \in E$ and for all $j > \text{con}(W)$;

(ii) $\kappa(G[W \cup Z] - e, V_j \cup Z) \geq j$ for all $e \in E$ and for all $j \leq \text{con}(W)$;

(iii) if $|W|, |\bar{W}| \geq 2$, then the graph $(V, \delta_{G-Z}(W))$ has a matching of size $\text{con}(W) - |Z| + 1$;

(iv) if $|W| = 1$, then $|W|$ has at least $\text{con}(W) - |Z| + 1$ neighbor nodes in $\bar{W}$; and

(v) (i) and (ii) also hold for $\bar{W}$ instead of W.

(b) This inequality defines a facet of kNCON$(G;r)$ only if the following holds. Define for any $U \subseteq W$ (resp. $U \subseteq \bar{W}$).

$$\text{ndef}_G(U) := \quad \max\{0, \text{con}(U) - \text{ number of nodes adjacent to } U \text{ in } G[W \cup Z] \text{ (resp., } G[\bar{W} \cup Z])\}.$$

Then, for all edges $e \in E(W \cup Z) \cup E(\bar{W} \cup Z)$, for all pairwise disjoint nodesets $W_1 \dots, W_p$ of W, with $W_i \neq W$ for $i = 1 \dots, p$ $(p \geq 0)$, and for all pairwise disjoint nodesets $U_1, \dots, U_q$ of $\bar{W}$, with $U_i \neq \bar{W}$ for $i = 1, \dots, q$ $(q \geq 0)$, the following inequality holds:

$$\sum_{i=1}^{p} \mathrm{ndef}_{G-e}(W_i) + \sum_{i=1}^{q} \mathrm{ndef}_{G-e}(U_i) - \left|[\bigcup_{i=1}^{p} W_i : \bigcup_{i=1}^{q} U_i]\right| \leq \mathrm{con}(W) - |Z|. \quad \square$$

3.4 Partition Inequalities

In this section, we discuss the class of partition inequalities for kECON and kNCON problems that generalize the cut inequalities (2.1)(i). For a graph $G = (V, E)$ and $r \in \mathbf{Z}_+^v$, we call a collection $\{W_1, \dots, W_p\} (p \geq 2)$ of subsets of V a **proper partition** of V if

- $W_i \neq \emptyset$, for $i = 1, \dots, p$;
- $W_i \cap W_j = \emptyset$, for $1 \leq i < j \leq p$;
- $\cup_{i=1}^{p} W_i = V$;
- $r(W_i) \geq 1$, for $i = 1, \dots, p$.

Let $[W_1 : \cdots : W_p]$ be the set of all edges having their endpoints in different sets W_i and let $I_1 := \{i \mid \mathrm{con}(W_i) = 1\}$ and $I_2 := \{i \mid \mathrm{con}(W_i) > 1\}$. Then the **partition inequality** induced by $\{W_1, \dots, W_p\}$ is defined as

$$(3.7) \qquad x([W_1 : \cdots : W_p]) \geq \begin{cases} p - 1, & \text{if } I_2 = \emptyset, \\ \lceil \frac{1}{2} \sum_{i \in I_2} \mathrm{con}(W_i) \rceil + |I_1|, & \text{otherwise.} \end{cases}$$

It is not hard to see that the partition inequalities (3.7) are valid for $k\mathrm{ECON}(G; r)$ and $k\mathrm{NCON}(G; r)$. Namely, if $\mathrm{con}(W_i) \geq 2$ for $i = 1, \dots, p$, any feasible survivable subgraph has to use at least $\text{con}(W_i)$ edges of $\delta(W_i)$, so it uses at least $\text{con}(W_i)/2$ edges of $[W_i : \cdots : W_p]$. If there are sets W_i with $\text{con}(W_i) = 1$, then a feasible solution using a minimum number α of edges of $[W_1 : \cdots : W_p]$ would either use at least two edges of each $\delta(W_i)$, $i = 1, \ldots, p$, (in which case α is at least $\lceil \sum_{i \in I_2} \text{con }(W_i)/2 \rceil + |I_1|$), or $|\delta(W_i)| = 1$ for some W_i (in which case induction on $|I_1|$ may be used by deleting W_i from the graph G and the feasible solution). So partition inequalities are valid. The separation problem for partition inequalities is known to be NP-hard; see [GMS92a].

We know of no general necessary and sufficient conditions for partition inequalities to define facets of kECON$(G; r)$ or kNCON$(G; r)$. Some special cases are dealt with in [GM90] and [S92]. We state here one particularly nice result.

(3.8) Theorem. Let $G = (V, E)$ be a complete graph with $k + 1$ parallel edges between each pair of distinct nodes and let $r \in \mathbf{Z}_+^V$. The partition inequality (3.7) given by a proper partition $W_1, W_2, \ldots, W_p$ of V with $p \geq 2$ defines a facet of kECON$(G; r)$ if and only if at least one of the following conditions hold:

(a) $\sum_{i \in I_2} \text{con}(W_i)$ is odd;

(b) $I_1 \neq \emptyset$; or

(c) $p = 2$.

We will prove Theorem (3.8) for the case where each W_i consists of a single node w_i with $r(w_i) \geq 2$ and $\sum_{v \in V} r_v$ is odd. (The general case follows from several lifting results that were not included here; the full details can be found in [S92].) In this case, the partition inequalities (3.7) become

$$x(E) \geq \lceil \sum_{v \in V} r_v/2 \rceil. \tag{3.9}$$

Before presenting the proof, we introduce an algorithm for constructing feasible solutions for kECON$(G; r)$ with a minimum number of edges. This construction procedure is a modification of an algorithm by Chou and Frank [CF70]. We describe this result in full detail and prove some facts which will be useful for our facet proof.

(3.10) Algorithm

Given an instance (G, r) of the kECON problem, where

- $G = (V, E)$ is a complete graph of p nodes with at least $k + 1$ parallel edges between each pair of nodes,
- $r_v \geq 2$ for all $v \in V$, $\sum_{v \in V} r_v$ is even, and there are two nodes of highest type k, find a subgraph N of G, feasible for the kECON problem, using a minimum number of edges of G, namely $\sum_{v \in V} r_v/2$.

Step 1. Choose any order $v_1, \ldots, v_p$ of the nodes in V which is imagined to be cyclic. If the highest node type k is even and there are nodes of odd type, the order should be chosen in such a way that there are two nodes (say u and v) of odd type that separate two nodes (say x and y) of type k in the cycle formed by the edges $v_1v_2, v_2v_3, \ldots, v_pv_1$; i.e., in traversing the cycle, we encounter these four nodes on the order u, x, v, y. (Necessarily there are at least two nodes of highest type k and at least two nodes of odd type, if any.)

Step 2. Construct a subgraph N' from the empty set by adding a cycle through all nodes of type at least i respecting the given order for $i = 2, 4, \ldots, k$ if k is even (or $i = 2, 4, \ldots, k - 1$ if k is odd).

Step 3. Let $\{w_1, w_2, \ldots, w_l\}$ be the set of all nodes of odd type numbered in the way they are met when scanning through $\{v_1, v_2, \ldots, v_p\}$ in the given order. (The starting node of the scan does not matter, nor does the direction. Note also that the number l of nodes of odd type must be even.) Construct the desired subgraph N by adding to N' the matching consisting of edges $w_i w_{l/2+i}$ $(i = 1, \ldots, l/2)$.

Define auxiliary node types

$$r'_v := 2\lfloor r_v/2 \rfloor \text{ for all } v \in V.$$

(3.11) Lemma. Let (G, r) be an instance of the kECON problem as in Algorithm (3.10). Let N' and N be the networks produced by the algorithm in Step 2 and 3, respectively. Then the following results hold:

(a) $|N| = \sum_{v \in V} r_v/2$.

(b) N' is feasible for $k\mathrm{ECON}(G, r')$.

(c) For any $W \subseteq V$, $\emptyset \neq W \neq V$,

$$|\delta_{N'}(W)| \leq \min(r(W), r(V \backslash W))$$

only if W is an interval, which is defined to be a set of the form $\{v_i, v_{i+1}, \ldots, v_j\}$ for $i \leq j$, or $\{v_i, \ldots, v_p, v_1, \ldots, v_j\}$ for $j < i$.

(d) For any $W \subseteq V, \emptyset \neq W \neq V$ with $r(W) \leq r(V \backslash W)$, if $|\delta_{N'}(W)| = |\delta_N(W)| = r'(W) < k$ then W is an interval (as defined in (c)), and W contains only nodes of even type r.

(e) N is feasible for $k\mathrm{ECON}(G, r)$.

Proof (of Lemma). Result (a) clearly holds since each node v has degree exactly r_v. Result (b) holds since Step 2 ensures that each pair of nodes s, t lie on $\min(r'_s, r'_t)/2$ edge-disjoint cycles in N^1.

In order to prove result (c), consider a $W \subseteq V$ with $\emptyset \neq W \neq V$ which is not an interval. Let X be a maximal interval in either W or $V \backslash W$, so that $r(X)$ is minimal with respect to all such maximal intervals in the cyclic order given in Step 1 of the algorithm. Without loss of generality, we assume that $X \subseteq W$. Then the two maximal intervals X', X'' in $V \backslash W$ which precede and follow X in the cyclic order contain nodes of type at least $r(X)$ (since $r(X)$ was chosen to be minimum). By the construction of N', there is at least one edge from X to X' and from X to X'' but no edges from X to $W \backslash X$, so

$$\begin{aligned} |\delta_{N'}(W)| &= |\delta_{N'}(W \backslash X)| + |\delta_{N'}(X)| \geq |\delta_{N'}(W \backslash X)| + 2. \\ \text{By (b), } |\delta_{N'}(W \backslash X)| &\geq \min(r'(W), r'(V \backslash W)) \\ &\geq \min(r(W), r(V \backslash W)) - 1 \end{aligned}$$

which yields the desired result when combined with the previous inequalities.

For (d), consider a set $W \subseteq V$, $\emptyset \neq W \neq V$ with $r(W) \leq r(V \backslash W)$ and $|\delta_{N'}(W)| = |\delta_N(W)| = r'(W) < k$. Since $r'(W) \leq r(W) = \min(r(W), r(V \backslash W))$, (c) implies that W is an interval. Since $|\delta_{N'}(W)| = |\delta_N(W)|$, the cut $\delta_N(W)$ can not contain any matching edge from Step 3 of the algorithms, so all nodes of odd type, are contained either in W or in $V \backslash W$. However, W can not contain all nodes of odd type, because if k is even then W must also contain a node v of highest type $r_v = k$ (because of the requirement on the order in Step 1) and then $r'(W) = r(W) = k$ which yields a contradiction; and if k is odd, the value $r(V \backslash W)$ must be even, contradicting $r(W) \leq r(V \backslash W)$. Therefore, W contains only nodes of even type r.

For (e), suppose that N is not feasible for kECON(G, r). There must be a set $W \subseteq V, \emptyset \neq W \neq V$, with $|\delta_N(W)| < r(W) \leq r(V \backslash W)$. Now by (b), $r'(W) \leq |\delta_{N'}(W)| \leq |\delta_N(W)| \leq r(W) - 1 \leq r'(W)$. So equality holds in all four inequalities. This implies $r'(W) = r(W) - 1 < k$, so by (d), W is an interval and contains only nodes of even type r. But then $r'(W) = r(W)$, a contradiction.

Proof. We now prove Theorem (3.8) for the following situation:

- $G = (V, E)$ is given as in the theorem;
- $\{W_1, \ldots, W_p\}$ $(p \geq 3)$ is a partition of V with $|W_i| = 1$ for all i;
- $r_v \geq 2$ for all nodes in v;
- $\sum_{v \in V} r_v$ is odd.

We prove that the partition inequalities (3.9) induced by such partitions define facets of kECON$(G; r)$. More precisely, we will prove that any inequality $b^T x \geq \beta$ that defines a facet of kECON$(G; r)$ containing the face defined by partition inequality (3.9) is the same up to scalar multiplication; that is, $b = \alpha \cdot 1$ for some scalar α. To this aim, we pick some (unspecified) node w of type at least 3 and show that, for all $e \in \delta(w)$, b_e takes the same value α_w. Nodes of type 2 in G have to be treated differently, as we will see later.

Given w with $r_w \geq 3$, we distinguish two cases:

(i) $V \backslash \{w\}$ contains at least two nodes of highest type k; and

(ii) there are only two nodes of type k, $k \geq 3$, with w being one of those two nodes.

We start with Case (i) as it contains the main idea of proving $b_e = b_f$ for all $e, f \in \delta(w)$. We apply Algorithm (3.10) to G and node types $\bar{r}_v := r_v$ for all nodes $v \in V$ except w, and $\bar{r}_w := r_w - 1$. This setting is a legal input. In Step 1, we choose the order $v_1, \ldots, v_p$ in such a way that v_p is a node of type $\bar{r}(v_p) = k, v_1 := w$, and v_2 is a node of type k or, if this setting of v_2 contradicts the requirements of Step 1 of the algorithm, we choose some node of odd type as v_2. The algorithm then produces a feasible network N with respect to node types $\bar{r}$. This network is not feasible with respect to node types r, because the degree of w is one less than required.

We analyze what sort of cut $\delta_N(W)$ hinders network N to be survivable with respect to r. (It will turn out that $\delta_N(W) = \delta_N(w)$.) So let $\delta_N(W)$ be a cut with $|\delta_N(W)| < \min\{r(W), r(V\backslash W)\}$ and $w \in W$. Since N is feasible with respect to $\bar{r}$, the cardinality of $\delta_N(W)$ is at least $\min\{\bar{r}(W), \bar{r}(V\backslash W)\}$. This implies that $|\delta_N(W)| = \bar{r}(W) < r(W) \leq r(V\backslash W)$, that w is the only node of type $r(W)$ in W, and that, by Lemma (3.11)(c), W is an interval. W cannot contain a node of type k, except possibly $w(= v_1)$. In particular, it does not contain v_p. If v_2 is also of type k, it does not contain v_2 either, so $W = \{w\}$. If v_2 had to be chosen as an odd node not of type k, because of the restrictions in Step 1, we can, for the time being, only conclude that W is some interval starting with v_1. But in which situation are we forced to choose v_2 as an odd node not of type k? Only if $\bar{r}(v_1) = k$ and k is even (and there are nodes of odd type, etc.). Anyway, $\bar{r}(W)$ is even and so is $|\delta_N(W)| = \bar{r}(W)$. But then $\delta_N(W)$ may not contain any of the matching edges added in Step 3 of the algorithm; that is, W either contains all nodes of odd type or none. If it contains all, then also a node $v \neq w$ of type k, which is impossible, because w was supposed to be the only node of type $r(W)$ in W. Consequently, W contains no node of odd type, hence $W = \{w\}$.

So far we have shown that the only cut that may keep N from being feasible with respect to node types r is the cut induced by $\{w\}$. Therefore, it is possible to add any edge $e \in \delta(w)$ to $N = (V, F)$ to make the network feasible. The resulting network (called $N_e = (V, F \cup \{e\})$) uses exactly $(\sum_{v \in V} r_v + 1)/2$ edges, so the incidence vector of $F \cup \{e\}$ satisfies the given partition inequality and also $b^T x \geq \beta$ with equality. Comparing $F \cup \{e\}$ and $F \cup \{f\}$ for any two edges $e, f \in \delta(w)$ proves $b_e = b_f$ for all $e, f \in \delta(w)$, which is the desired result.

In Case (ii), we cannot simply reduce the type of node w, as there would be only one other node u of type k left. So we reduce the types of both u and w to the highest node type in $V\backslash\{u, w\}$, that is, we set $\bar{r}_u := \bar{r}_w := r(V\backslash\{u, w\})$, and $\bar{r}_v := r_v$ for all nodes $v \in V\backslash\{u, w\}$. The node type $\bar{r}_w$ of w is at least 3, because if $\bar{r}_u = \bar{r}_w = 2$, then $k = 2$ and $\bar{r} = 2$ for all nodes in V, contradicting the assumption that $\sum_{v \in V} r_v$ is odd. Let us also pick a set T of $r_w - \bar{r}_w$

parallel uw-edges. With this setting of $\bar{r}$, we have the situation of (i). That means, we can construct a network $N = (V, F)$ such that the addition of any edge $e \in \delta(w)\backslash T$ produces a network $N_e := (V, F \cup \{e\})$, feasible for the ECON problem with nodes types $\bar{r}$. The N_e can be required not to contain T, because the algorithm will use at most $\bar{r}_w$ parallel uw-edges of the $r_w + 1$ existing uw-edges. To each N_e we now add the set T. The resulting subgraph $(V, F \cup \{e\} \cup T)$ is feasible because N_e was already feasible with respect to $\bar{r}$ and the new subgraph contains r_{uw} edge-disjoint $[u, w]$-paths. So, for any two edges $e, f \in \delta(w)\backslash T$, the incidence vectors of $F \cup \{e\} \cup T$ and $F \cup \{f\} \cup T$ are contained in the face defined by the partition inequality (3.9) and must necessarily satisfy $b^T x = \beta$. Therefore $b_e = b_f$ for all $e, f \in \delta(w)\backslash T$, and since T can be chosen freely from the set of parallel uw-edges, $b_e = b_f$ for all $e, f \in \delta(w)\backslash T$.

So far, we have proved that for all nodes $w \in V$ with $r(w) \geq 3$ there exists a number α_w such that, for all edges $e \in \delta(w)$, we have $b_e = \alpha_w$. Any two adjacent nodes s, t of type at least 3 must then have the same value $\alpha_s = \alpha_t =: \alpha$, so $b_e = \alpha$ for all edges that are adjacent to some node of type at least 3. So, if there is at most one node of type 2, we are already done with proving $b = \alpha \cdot 1$.

Now we want to prove that $b_e = \alpha$ also for all edges e between two nodes of type 2. Let W be the set of nodes of type 2, and choose some edge $e = st \in E(W)$. Construct a survivable network N in the graph $G - W$ (with node types r) that uses a minimum number of edges. Let $f = u_f v_f$ and $g = u_g v_g$ be two edges used by N. Now replace f in N by a $[u_f, v_f]$-path P_f using all nodes of $W\backslash\{t\}$, such that s is the second last node in P_f (the last is v_f). Replace also g in N by a $[u_g, v_g]$-path consisting of the nodes u_g, t, and v_g. The network N' thus produced is feasible. Another feasible network N'' is constructed as follows. Replace f in N by a certain $[u_f, v_f]$-path consisting of $P_f - v_f$, edge $e = st$, and some edge tv_f. Both N' and N'' use $\lceil \sum_{v \in V} r_v/2 \rceil$ edges of G. Comparing $b^T x$ for the associated incidence vectors, we see $b(sv_f) + b(u_g t) + b(tv_g) = b_e + b(tv_f) + b_g$. All edges appearing in this expression, except e, are edges in $E\backslash E(W)$, which have b-value α. Therefore also $b_e = \alpha$.

Taking everything together, we have proved $b_e = \alpha$ for all $e \in E$, so the partition inequality (7.9) defines the same face as the inequality $b^T x \geq \beta$, namely a facet of kECON$(G;r)$. $\square$

It can be shown that the separation problem for partition inequalities is NP-hard, even if we restrict attention to vectors y with $0 \leq y \leq 1$, see [GMS92a]. However, there are fast and successful heuristics for the separation of partition inequalities; and our computational experiments have revealed that partition inequalities are very helpful for solving network survivability problems; see Section 4.

3.5 Node Partition Inequalities

In this section, we consider the class of node partition inequalities which generalize the node cut inequalities (2.1)(ii) in a manner similar to how the partition inequalities (3.7) generalize the cut inequalities (2.1)(i). Consider a graph $G = (V, E)$ and $r \in \mathbf{Z}_+^V$. Let $Z_2 \subseteq \ldots \subseteq Z_k \subseteq V$, $k \geq 2$ be node sets with $|Z_j| = j - 1$ for $j = 2, \ldots, k$. Let $\{W_1, \ldots W_p\}$ be a proper partition of $V\backslash Z_k$, such that at least two node sets in the partition contain nodes of type k. Define $I_j := \{i \mid r(W_i) \geq j\}$ for $j = 1, \ldots, k$. The **node partition inequality** induced by $W_1, \ldots, W_p$ and $Z_2, \ldots, Z_k$ is given by

$$(3.12) \qquad x([W_1 : \cdots : W_p : Z_k]) \quad - \quad x(\textstyle\bigcup_{j=2}^{k} \bigcup_{i \in I_j} [Z_j : W_i]) \geq p - 1.$$

(3.13) Theorem. The node partition inequalities (3.12) are valid for kNCON$(G;r)$.

Proof: In the case where $p = 2$, the node partition inequality (3.12) defines a node cut inequality (2.1)(ii) which is known to be valid. The case $p \geq 3$ is treated by induction.

So let $\{W_1, \ldots, W_p\}$ and $Z_2, \ldots, Z_k$ induce a node partition inequality with $p \geq 3$. Let W_p be the node set with smallest value of $r(W_i)$. Define $j := r(W_p)$. Then, for $i = 1, \ldots, p - 1$

the node partition inequality induced by $Z_2, \ldots, Z_k$ and $\{W_1, W_2, \ldots, W_i \cup W_p, \ldots, W_{p-1}\}$ is valid and has right-hand side $p-2$. Note that none of the coefficients of the left-hand side of this new inequality is larger than in the original inequality, and that the coefficients of edges in $[W_i : W_p]$ have dropped to 0. Adding up all these node partition inequalities and the node cut constraint $x(\delta_{G-Z_j}(W_p)) \geq 1$ gives an inequality $a^T x \geq (p-1)(p-2)+1$, where all coefficients of a^T have value at most $p-1$. Dividing this inequality by $p-1$ and rounding up the right-hand side and all coefficients of the left-hand side produces the desired node partition inequality with right-hand side $p-1$. $\square$

It is known to be NP-hard to separate this class of inequalities; see [GM90]. Some necessary and some sufficient conditions for these inequalities to define facets are known only in a few special cases; see [GMS92b] and [S92].

It can be shown that the separation problem for partition inequalities is NP-hard, even if we restrict attention to vectors y with $0 \leq y \leq 1$, see [GMS92a]. However, there are fast and successful heuristics for the separation of partition inequalities; and our computational experiments have revealed that partition inequalities are very helpful for solving network survivability problems; see Section 4.

3.6 r-Cover and Lifted r-Cover Inequalities

A nice combinatorial relaxation of the kECON (and then the kNCON problem) is the r-cover problem that can be defined as follows. Given a graph $G = (V, E)$ and positive integers r_v for all $v \in V$, an **r-cover** is a set $F \subseteq E$ of edges such that $|F \cap \delta(v)| \geq r_v$ for all $r \in V$. Clearly, every solution of an ECON problem defined by a graph G and node types $r \in \mathbf{N}^V$ is an r-cover. Hence, if edge weights are given in addition, a lower bound for the ECON problem can be determined by solving the r-cover problem which means finding a r-cover of minimum weight. This relaxation is of particular interest since the r-cover problem can be solved in polynomial time. The polynomial-time solvability follows from the fact that the r-cover problem can be transformed into a (1-capacitated) b-matching problem.

In his paper [E65] Edmonds not only gave a polynomial time algorithm for solving the b-matching problems, he also determined a complete linear description of the b-matching polytope. Edmonds' blossom inequalities for the 1-capacited b-matching polytope of a graph $G = (V, E)$ can be transformed to the r-cover case. This transformation, see [GMS91], yields the **r-cover inequalities**, valid for ECON$(G; r)$, that have the following form:

$$x(E(H)) + x(\delta(H) \backslash T) \geq \left\lfloor \textstyle\sum_{v \in H}(r_v - |T|)/2 \right\rfloor \quad \begin{array}{l}\text{for all } H \subseteq V, \\ \text{and all } T \subseteq \delta(H).\end{array} \tag{3.14}$$

The separation problem for the class of r-cover inequalities and vectors in the unit hypercube can be solved in polynomial time as follows. Suppose $x^* \in Q^E, x^* \geq 0$ is given. We define a new vector $y^* := 1 - x^*$ and a 1-capacitated b-matching problem by setting $b_v := \deg(v) - r_v$ for all $v \in V$. Then it is easy to see that x^* satisfies all r-cover inequalities (3.14) if and only if y^* satisfies the blossom inequalities for the corresponding b-matching problem which read as follows.

$$y(E(H)) + y(T) \leq \left\lfloor (\sum_{V \in H} b_v + |T|)/2 \right\rfloor \text{ for all } W \subseteq V \text{ and all } T \subseteq \delta(H). \tag{3.15}$$

Padberg and Rao [PR82] have designed a polynomial time algorithm for the separation problem for the class of blossom inequalities (3.13). This algorithm is based on the Gomory-Hu procedure for finding a minimum cut in a graph with nonnegative edge capacities. If the Padberg-Rao algorithm yields an inequality $y(E(H)) + y(\bar{T}) \leq \lfloor (\sum_{v \in H} b_v + |T|)/2 \rfloor$ that is violated by y^* then x^* violates the r-cover inequality $x(E(H)) + x(\delta(H) \backslash T) \geq \lceil (\sum_{v \in H} r_v - |T|)/2 \rceil$.

In case some of the nodes have type 1 the r-cover inequalities can be strengthened as follows

$$x(E(H)) + x(\delta(H) \backslash T) \geq \left\lceil (\sum_{\substack{V \in H_1, \\ r_v \geq 2}} r_v - |T|)/2 \right\rceil + |\{v \in H \mid r_v = 1\}|. \tag{3.16}$$

These inequalities are valid for kECON$(G; r)$ but not for the r-cover polytope. To solve the separation problem for the class of strengthened r-cover inequalities (3.16) heuristically we do the following. We declare all nodes of type one to be of type two and apply the procedure described above for the modified node types. If the Padberg-Rao separation algorithm finds a violated blossom inequality then the corresponding strengthened r-cover inequality (3.16) is violated. However, this trick does not yield an exact separation routine because there may be violated inequalities of type (3.16) that this procedure does not find. This is due to the fact that the transformation of node types may violate one of the requirements for the Padberg-Rao procedure to work correctly; i.e., this method is a heuristic for strengthened r-cover inequalities (3.16) in case nodes of type 1 are present.

We now generalize the r-cover inequalities (3.16) further. Let $G = (V, E)$ be a graph and $r \in \{0, \ldots, k\}^V$. Let $H \neq V$ be a node set, called the **handle**, and let $T \subseteq \delta(H)$. For each $e \in T$ we denote by T_e the set of the two end nodes of e. The sets $T_e, e \in T$, are called **teeth**. For simplicity we also call the edges $e \in T$ teeth in this section. If an edge $e \in T$ is parallel to some edge $f \in T$, we count T_e and T_f as two sets, even if $T_e = T_f$. Let $\{H_1, \ldots, H_p\}, p \geq 3$, be a partition of H into nonempty disjoint node sets such that

- $r(H_i) \geq 1$ for $i = 1, \ldots, p$;
- no more than $\mathrm{con}(H_i) - 1$ teeth intersect any $H_i, i = 1, \ldots, p$;
- at least three H_i are intersected by teeth;
- $\sum_{i \in I_2} \mathrm{con}(H_i) - |T|$ is odd, where $I_2 := \{i \mid \mathrm{con}(H_i) \geq 2\}$.

Let $I_1 := \{i \mid \mathrm{con}(H_i) = 1\}$. We call

$$(3.17) \qquad x([H_1 : \cdots : H_p]) + x(\delta(H) \setminus T) \geq \frac{1}{2}\Big(\sum_{i \in I_2} \mathrm{con}(H_i) - |T| + 1\Big) + |I_1|$$

the **lifted r-cover inequality** induced by $H_1, \ldots, H_p, T$.

It is not hard to show (see [S92]) that the lifted r-cover inequalities (3.17) are valid for kECON$(G; r)$ and hence for kNCON$(G; r)$. Some necessary conditions are known for these inequalities to define facets for kECON$(G; r)$, and one can easily see that these inequalities do not define facets for kNCON$(G; r)$ for "highly" connected graphs G; see [S92]. It is known that the separation problem for lifted r-cover inequalities is NP-hard; see [GMS92a]. We have invented a separation heuristic for class (3.17) that uses some graph manipulation techniques (such as shrinking of edges) and is based on an analysis of the Gomory-Hu tree as provided by the Padberg-Rao algorithm.

3.7 Further Remarks on Valid Inequalities

We note that we know more general versions of some of the classes of valid inequalities introduced before, and we also know further classes of inequalities valid for ECON$(G; r)$ or NCON$(G; r)$. We have omitted the introduction of these classes here since many technical definitions are necessary to describe them, and we do not have general results about the dimensions of the faces that they define, and we could not make any algorithmic use of them so far. Some information about these classes can be found in [S92].

There is a rich literature in polyhedral combinatorics on facet manipulation techniques, i.e., methods by which new types of facets can be derived from known classes. These techniques usually run under the name "facet lifting" or "facet extension". A number of lifing results are shown in [S92]. Included are techniques such as: addition of an edge, addition of a node, and expansion of a node w into a node set W where all edges in $E(W)$ receive a coefficient zero in the lifted inequality. The description of these techniques and the proofs of the associated lifting results are also rather technical and not presented here.

Our results about characterizing those inequalities among the classes of valid inequalities described before that are facet-defining for kECON$(G;r)$ or kNCON$(G;r)$ appear somewhat unsatisfactory. In fact, they are. But having worked on the facial structure of these polytopes for some time we are convinced that general results covering large classes of graphs and node types simultaneously are very hard to obtain. Theorem (3.8) gives a glimpse at the technical subtleties involved in a seemingly simple-looking case. The main difficulty we see is that a slight change of the graph or node type may result in a considerable change of the problem complexity and the polyhedral structure. For instance, if $r_v = 1$ for all $v \in V$, kECON$(G;r)$ is equal to kNCON$(G;r)$, which is nothing but the convex hull of the incidence vectors of all supersets of spanning trees of G; changing a few node types to zero we obtain an NP-hard Steiner tree problem; setting $r_v = 2$ for all $v \in V$, 2NCON$(G;r)$ and 2ECON$(G;r)$ contain the travelling salesman polytope on G.

Based on the observations we think that further investigation should go into the study of more restricted, practically relevant cases and not into investigation of the whole range of kECON or kNCON polytopes.

4 A Cutting Plane Algorithm and Computational Results

In this section, we give an outline of our cutting plane algorithm for the kECON and kNCON problems. We describe it for the kNCON problem. The algorithm for the kECON problem is derived from that for the kNCON problem by skipping all those separation routines that check inequalities that are valid for kNCON$(G;r)$ but not for kECON$(G;r)$.

Our cutting plane procedure starts with solving the LP

(4.1) $\min c^T x$

subject to

$$
\begin{aligned}
x(\delta(v)) &\geq r_v && \text{for all } v \in V \text{ with } r_v \geq 1; \\
0 \leq x_e &\leq 1 && \text{for all } e \in E
\end{aligned}
$$

consisting of at most $|V|$ degree inequalities and the $2|E|$ trivial inequalities. Almost all of these define facets of kNCON$(G;r)$, if kNCON$(G;r)$ is full-dimensional (see Theorem (3.2)). An optimal solution $y \in \mathbf{R}^E$ of this relaxation of the kNCON problem is usually not feasible for the polytope kNCON$(G;r)$. (If it were, we would be finished.)

So in each iteration of the cutting plane algorithm we try to find inequalities (more specifically: cut, node cut, partition, node partition, and lifted r-cover inequalities) that are valid for kNCON$(G;r)$, but are violated by y. Geometrically, such an inequality defines a hyperplane in $\mathbf{R}^E$ separating y from the kECON-polyhedron, a so-called "cutting plane". The heuristics and exact algorithms for finding inequalities violated by a given y are called **separation routines**.

We add all the violated inequalities found by our separation routines to the current LP and solve the revised LP to get a new optimum solution y. (We do not solve the new LP from scratch, but use postoptimization.) We repeat this process until the current optimal LP solution y happens to be feasible for kNCON$(G;r)$, or no further inequalities violated by y are found. In the second case we proceed with a branch&cut method.

In the first case (y feasible), we know that y is optimal, since the present LP is a relaxation of the kNCON problem. Note that feasibility of y is identical with y being a $\{0,1\}$-vector that satisfied all cut constraints (2.1)(i) and node cut constraints (2.1)(ii). This feasibility criterion is easy to check.

Of course, since we are using only a subset of all facet-defining inequalities for kNCON$(G;r)$, we cannot be sure to find an optimal solution with such a cutting plane algorithm for all graphs G, cost functions c, and node types r.

In any case, even if the present fractional solution y is not feasible, its objective function value $c^T y$ provides a lower bound for the kNCON problem, which is increased with every iteration (or at least, it does not drop).

We summarize the cutting plane algorithm:

(4.2) Algorithm (Cutting Plane Algorithm for kNCON)

1. Solve the LP (4.1). Let y be an optimal solution.
2. While y is not feasible for $k\text{NCON}(G; k)$ do:
   (a) find valid inequalities violated by y, add them to the LP, delete some redundant inequalities, and resolve the LP. Let y be a new optimal solution.
   (b) If no violated inequalities can be found, perform branch & cut.

At present, we have a preliminary version of a code for solving survivability problems with higher connectivity requirements. In order to test our code for general kNCON problems, we first used a set of random problems. Later, we also obtained test data for a real-world 3NCON problem, which arose in the design of a communication network on a ship. Both types of test problems have their "drawbacks", however. Most random problems turned out to be too easy, and the ship problem confronted us with many new difficulties.

We first report about our computational results on random problems. We used the same set of random data as Ko and Monma [KM89] used for their high-connectivity heuristics. So we will be able to compare results later. All running times reported are on a SUN 4/50 IPX workstation.

The test set consists of five complete graphs of 40 nodes and five complete graphs of 20 nodes, whose edge costs are independently drawn from a uniform distribution of real numbers between 0 and 20. For each of these 10 graphs, a minimum-cost k-edge connected subgraph for $k = 3, 4, 5$ is to be found. Table 2 reports the number of iterations (minimum and maximum) and the average time (in seconds) taken by our code to solve these problems for $k = 3, 4$, and 5, respectively. Only the time for the cutting plane phase is given.

# Nodes	**# Iterations**			**Average Time (secs)**		
$K =$	3	4	5	3	4	5
20 nodes:	1-2	1-5	1-4	0.43	0.51	0.58
40 nodes:	1-2	1-2	1-4	1.54	1.95	2.36

All problems except one 3ECON instance on 20 nodes were solved in the cutting plane phase. In fact, 20 of the 30 problems were already solved in the first iteration with the initial LP (4.1). For the instances not solved in the first iteration, at most four lifted r-cover inequalities (3.16) had to be added to obtain the optimal solution. Except for one 3ECON instance, no partition inequalities were added. So, the average solution time is mainly the solution time for the first LP.

All optimal solutions for the kECON problems were at the same time feasible and hence optimal for the corresponding kNCON problems, except the one 3ECON problem which could not be solved in the cutting plane phase. There the optimal solution (obtained by branch & cut) is 3-edge connected, but not 3-node connected.

These excellent results were surprising, because we always thought high-connectivity problems to be harder than low-connectivity problems. But this does not seem to be true for random costs.

The high-connectivity heuristics of Ko and Monma performed reasonably well. The relative gap between the heuristic (h) and the optimal solution value (o), namely $100 \times (h - o)/o$, computed for the above set of random problems, ranged between 0.8 and 12.8 with an average of 6.5 % error (taken over all problems).

The second set of test problems were five complete graphs whose 40 nodes were placed randomly in a square, and whose edge weights are the euclidean distances between the end nodes. All nodes are of the same type k, where k ranges between 1 and 5, so five kECON problems are derived from each of the five graphs. The 1ECON problems were solved with a spanning tree algorithm.

Table 1 shows the computational results for these problems. Its entries are from left to right:

K the required connectivity

P the number of partition inequalities used in the cutting plane phase

RC the number of $r$cover inequalities used in the cutting plane phase

IT the number of iterations in the cutting plane phase

COPT the optimal value

GAP the relative error between the optimal value and the lower bound achieved by the cutting plane phase

TT the total running time (with branch and cut, etc.) in minutes.

K	**IT**	**P**	**RC**	**COPT**	**GAP**	**TT**
1				77.20– 95.31		
2	3–10	54–70	0–8	90.69–107.72	0.00–0.23	0–1
3	3–6	55–74	4–24	156.83–184.95	2.12–2.65	33–324
4	1–2	40–46	0–4	221.64–264.39	0.00	0–1
5	1–8	40–44	0–15	307.12–363.19	0.00–0.98	0–186

Table 1: Performance of branch&cut on euclidean problems

All k-edge connectivity problems with even k, except one 2ECON problem, could already be solved in the cutting plane phase. The running time for the cutting plane phase was at most six seconds for all problems. Of the 5ECON problems, two could be solved in the cutting plane phase in two iterations. Two other 5ECON problems were solved by branch and cut in at most three minutes, and only one took 186 minutes. For all 3ECON problems, branch and cut had to be employed to bridge the gap. The relatively large gaps in the 3ECON problems indicate that our separation routines should be enhanced to find the partition and r-cover inequalities still violated by the last LP-solution and that, maybe, other facet-defining classes of inequalities for the kECON polytope for odd k are needed. Because of this poor performance and because of the nonavailability of good upper bounds, the branching procedure degenerated into enumeration for several instances.

Considering the gaps and the running times, we may conclude that k-edge connectivity problems for even k are easier to solve than those for odd $k \geq 3$, and that the k-edge connectivity problems with odd k become easier for "large" k. One reason for the difference between even and odd k may be that, for even k, the only facet-defining partition inequalities are the cut inequalities, which are easy to find.

Concerning the structure of the solutions, all optimal solutions for **even** k were regular graphs, except for one 2ECON instance, where two nodes of degree 3 appeared. For $k \geq 3$, the solutions are not necessarily k-node connected. Some are only 2-node connected.

The optimal values can be said to be roughly linearly increasing in k, from $k \geq 2$. By the way, the lower values in column **COPT** of Table 1 are all due to the same problem instance, and the upper values too.

To give an impression of the solution structures, Figure 1 was included, depicting, for one of the graphs and $k = 1, \ldots, 4$, the optimal kECON solutions in clockwise order.

One real-world application of survivable network design, where connectivities higher than two are needed, is the design of a fiber communication network that connects locations on

Figure 1 is not available as postscriptfile.To get a paper-copy send an e-mail to bibliothek@sc.zib-berlin.de.

Figure 1: Solution of euclidean problems for $k = 1$ to 4

a military ship containing various communication systems. The reason for demanding high survivability of this network is obvious.

The problem of finding a high-connected network topology minimizing the cable installation cost can be formulated as a 3NCON problem. We will describe the characteristics of this problem in the following.

We obtained the graph and edge cost data of a generic ship model. It has the following features. The graph of possible link installations has the form of a three-dimensional grid with 15 layers, 494 nodes, and 1096 edges, which is depicted in Figure 2.

The ship problem comes with three different options: the node types may vary; the costs may be normal or random; and the underlying graph may be reduced or not.

Of the grid's 494 nodes, there are 461 regular nodes (depicted by hollow circles), 30 special nodes (depicted by filled circles) in the main part of the ship and 3 priority nodes (depicted by triangles) in the ship's tower. The special and priority nodes represent various communication, command and control systems to be interconnected by the network. The regular nodes represent fiber junction boxes where the fiber cable can be routed. The notation "shipxyz" will be used to indicate that the regular nodes are of type x, special nodes are of type y and priority nodes are of type z. So "ship013" is the problem, where the three nodes in the tower are of type 3, the 30 special nodes in the body of the ship are of type 1, and all other 461 grid nodes are of type 0.

No branch and cut was used because of the long computation times for some of the problems.

Inequalities that are not almost tight are removed from the LP, but are still kept in a pool of inequalities. But since even this pool grew too much, inequalities are removed from there that have not been violated for more than five iterations in a row.

All problems were solved as ECON problems. No node partition inequalities were used.

The upper bounds were produced by a heuristic for NCON problems (not ECON) that tries to eliminate as many edges from the input graph as possible, starting with the first edge. (This heuristic was written by Bill Cook.) The input graph was the graph of edges formed by the support of the best fractional solution, and the edges were ordered by increasing value of the fractional solution. (Not by increasing costs, as all horizontal edges have the same cost.)

The normal cost structure is highly regular. The costs are roughly proportional to the distances between nodes, with the feature that horizontal distances are much longer than vertical distances. (The grid shown in Figure 1 has been scaled. Also, contrary to the graphical representation, the horizontal layers do not always have the same distance from each other.) With this cost structure, it is much cheaper to route vertically than horizontally. Since there exist many shortest paths between any two nodes, there will also exist many optimum solutions to the survivable network problem. So the problem is highly degenerate. Degeneracy together with the size of the ship problem caused us to run into difficulties. We also considered "random" uniform $[0,1]$ costs which were scaled so that the overall cost of the edges remained the same as in the original problem. These problems were indicated by appending "rand" to the problem name.

We considered "reduced" versions of these problems where we removed some of the "unnecessary" nodes in the lower left and right hand corner of the grid, and also deleted some of the horizontal layers of the grid containing only nodes of type 0. It is not obvious at all that corners of a grid may be cut out and layers may be deleted without affecting the optimum objective function value of the problem. But nevertheless, we used these reductions heuristically to cut down problem sizes in the hope that some optimal solution of the original graph is still contained in the reduced graph. For the "ship023" problem, this hope was confirmed. These reduced problems were denoted by appending "red" to the problem name.

	Original Graph					Reduced Graph				
PROBLEM	**0**	**1**	**2**	**3**	**Edges**	**0**	**1**	**2**	**3**	**Edges**
ship013	461	30	0	3	1096/0	128	28	0	3	325/6
ship023	461	0	30	3	1096/0	249	0	30	3	607/6
ship033	461	0	0	33	1096/0	300	0	0	33	719/9

Table 2: Sizes of ship problems

Figure 2 shows the reduced graph of the "ship013" problem. The result of the reductions can be seen from Table 2, whose columns list, from left to right, the problem names, and, for the original ship graph and the reduced ship graphs, the number of nodes of type 0, 1, 2, and 3, the total number of nodes and the total number of edges/number of forced edges. The forced edges are those edges contained in some cut of size 3 separating two nodes of type 3, which must be contained in any feasible solution. Table 2 shows that the reductions are enormous, yet there are still many more nodes of type 0 than nodes of nonzero type in each problem.

When we applied our code to the reduced graphs, the fractional solutions still looked frequently like paths beginning at some special node and ending in some node of type 0. To cure this problem, we made use of the following type of inequalities

$$x(\delta(v)\setminus\{e\}) \geq x_e \quad \text{for all nodes } v \text{ of type 0 and all } e \in \delta(v).$$

These inequalities (we call them **con0 inequalities**) describe algebraically that nodes of type 0 do not have degree 1 in an edge-minimal solution. This is not true for all survivable networks, but it is true for the optimum solution if all costs are positive. So, although these inequalities are not valid for the kNCON polytope, we used them to force the fractional solutions into the creation of longer paths.

Table 3 gives some computational results of our cutting plane algorithm on the several versions of the ship problem. The entries from left to right are:

PROBLEM	Problem name with "red" for reduced and "rand" for random costs
VAR	Number of edges minus number of forced edges
IT	Number of iteration (i.e., LPs solved)
PART	Number of partition inequalities (3.7) added
RCOV	Number of r-cover inequalities (3.14) added
LB	Lower bound (i.e., optimal LP value)
GAP	(UB-LB)/LB in percent
TIME	in minutes:seconds

PROBLEM	**VAR**	**IT**	**PART**	**RCOV**	**LB**	**UB**	**GAP**	**TIME**
ship013	1088	3252	777261	0	211957.1	217428	2.58	10122:35
ship013red	322	775	200570	0	217428	217428	0	426:47
ship013rand	1088	1273	301190	0	171409.8	171409.9	0	2783:15
ship023	1088	15	4090	0	286274	286274	0	27:20
ship023red	604	12	2372	0	286274	286274	0	1:54
ship023rand	1088	11	3649	0	245905.8	248249.0	0.94	42:31
ship033	1082	42	10718	1	461590.6	483052	4.64	55:26
ship033red	710	40	9817	0	462099.3	483052	4.53	34:52
ship033rand	1082	89	23505	0	422169.6	428535.5	1.50	198:17
ship113	1090	128	17199	0	902691.0	918691	1.77	4724:55
ship113rand	1090	45	4343	0	789280.7	817352.1	3.55	13:33
ship123	1088	61	13210	0	906691.0	930691	2.57	1167:37
ship123rand	1088	1942	54846	9	834951.9	856666.7	2.60	694:20
ship133	1084	176	21564	0	945052.0	1008808	6.74	119:15
ship133rand	1084	582	22788	0	941568.2	969578.7	2.97	195:11
ship223	1085	5	541	0	940925.0	940925	0	0:43
ship223rand	1085	5	625	0	1090417.5	1090430.0	0.001	0:54
ship233	1081	5	532	0	1028193.0	1029176	0.09	0:54
ship233rand	1081	10	599	6	1172230.3	1183636.9	0.97	1:53

Table 3: Performance of the cutting plane algorithm on the ship problems

Table 4 gives a breakdown of how time was spent. The entries from left to right are:

PROBLEM	Problem name with "red" for reduced and "rand" for random costs
PT	Time spent for reduction of the problem (in percent)
LPT	Time spent for LP solving (in percent)
CT	Time spent for separation (in percent)
MT	Time spent on miscellaneous items, input, output, etc. (in percent)
TIME	Total time in minutes:seconds

PROBLEM	**PT**	**LPT**	**CT**	**MT**	**TIME**
ship013	0.0	75.6	23.9	0.5	10122:35
ship013red	0.0	68.5	30.1	1.4	426:47
ship013rand	0.0	64.0	35.1	0.9	2783:15
ship023	0.0	13.1	86.4	0.4	27:20
ship023red	0.1	39.2	58.6	1.9	1:54
ship023rand	0.0	2.7	97.1	0.2	42:31
ship033	0.0	31.2	68.2	0.6	55:26
ship033red	0.0	41.1	58.4	0.5	34:52
ship033rand	0.0	27.6	71.9	0.4	198:17
ship113	0.0	98.4	1.6	0.0	4724:55
ship113rand	0.0	42.0	56.8	1.2	13:33
ship123	0.0	97.9	2.1	0.0	1167:36
ship123rand	0.0	23.9	75.7	0.5	694:19
ship133	0.0	60.4	38.9	0.7	119:15
ship133rand	0.0	32.0	67.3	0.7	195:11
ship223	0.3	18.7	76.7	4.5	0:44
ship223rand	0.2	15.9	79.8	4.1	0:54
ship233	0.2	22.4	74.5	2.9	0:54
ship233rand	0.1	9.5	88.7	1.7	1:53

Table 4: Computation time on the ship problems

Several problems were solved to optimality: ship013 (normal costs, reduced), ship023 (normal costs, reduced and not reduced), and ship223 (normal costs). The ship223 problem with normal costs ended up with a fractional solution of value 940925, but the primal heuristic found an integer solution with the same value.

Reduction paid off by a factor of at least 20 in the computing times of the ship013 and ship023 problems. The reduction of the ship023 problem did not affect the optimal value.

The largest gaps are for the ship033 and ship133 problems for normal costs, reduced or not reduced. More polyhedral investigation is probably required in these cases. By inspection, we found a few more violated partition inequalities. So, also better separation routines for partition inequalities are needed.

The problems ship013 (not reduced, normal and random costs) and ship113 (not reduced, normal costs) took longest. The solution of ship013 (not reduced, normal costs) is distributed over the whole net with many small fractionals. This shows that tree-like problems are not well handled. Problems with many nodes of type 2 are easier than problems without nodes of type 3.

r-cover inequalities (very few, however) were found only for the ship033, ship123, ship233 problems. Therefore better separation routines for r-cover inequalities for this special kind of graph are needed.

As one can see from Table 3 our code is still painfully slow for the problems involving many nodes of type 1. In each iteration, only small progress is made. Although many inequalities are added, they do not produce big structural changes in the fractional solution. A better strategy might be to add structured sets of partition or cut inequalities that somehow anticipate the "escape maneuvers" of the fractional solution.

An optimal solution to the reduced ship023 problems is shown in Figure 4. We do not understand yet why our code solves the ship023 problem rather easily and why there is still a gap after a long running time of our cutting plane algorithms for other problems. Probably, the "small" changes of a few survivability requirements result in more dramatic structural changes of the polyhedra and thus of the inequalities that should be used. It is conceivable that our code has to be tuned according to different survivability requirements settings.

Summarizing our computational results, we can say that for survivability problems with high connectivity requirements, many nodes of type 0 and highly regular cost structure (such as the ship problems) much still remains to be done to speed up our code and enhance the quality of solutions. This is in contrast to our previous work (see [GMS92a]) on applications in the area of telephone network design, where problem instances typically are of moderate size and contain not too many nodes of type 0, and where our approach produces very good lower bounds and even optimum solutions in a few minutes. Yet, we see our work as a promising step towards solving problems with high-connectivity constraints.

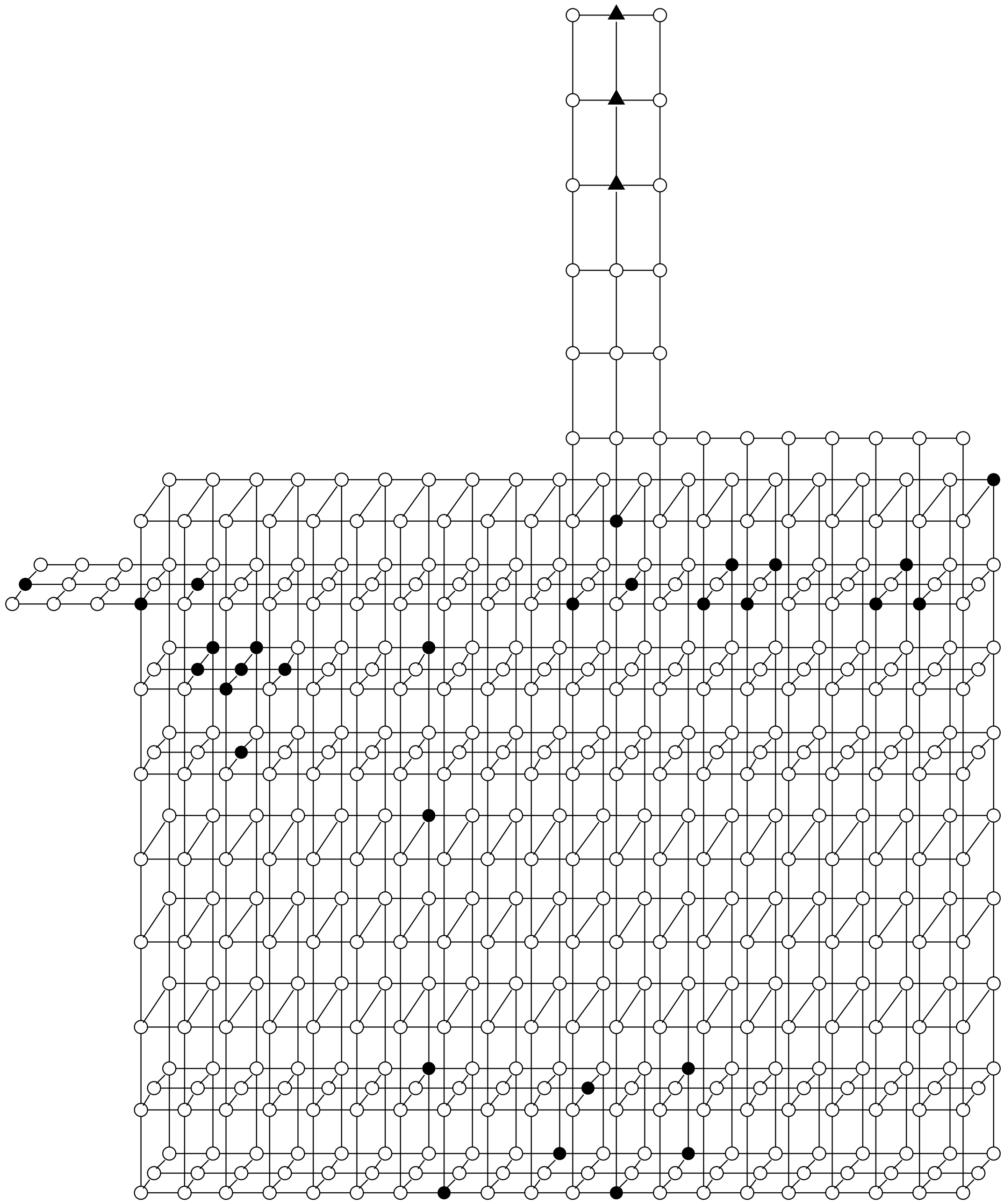

Figure 2: Grid graph of the ship problem

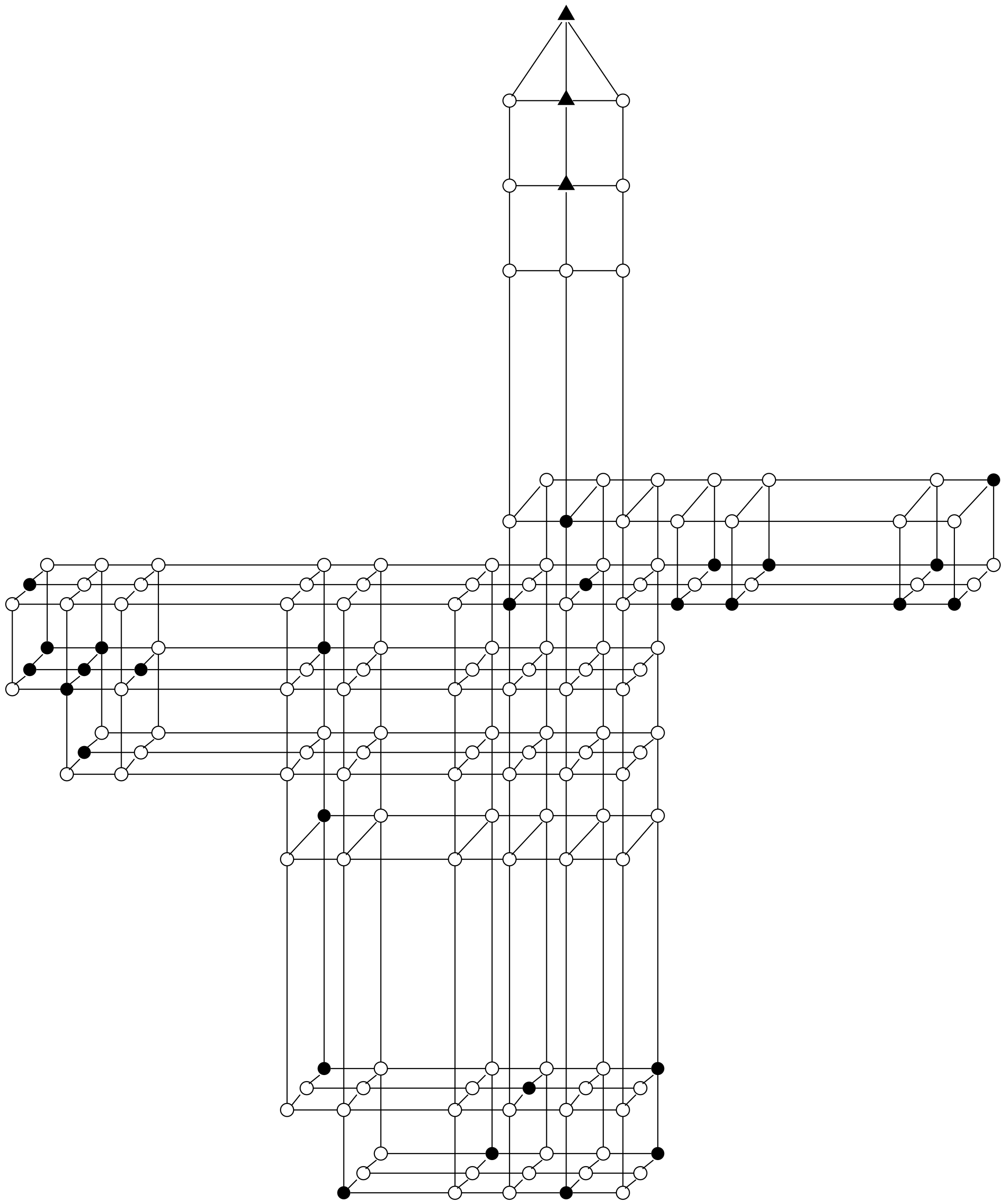

Figure 3: Reduced grid graph of the “ship13” problem

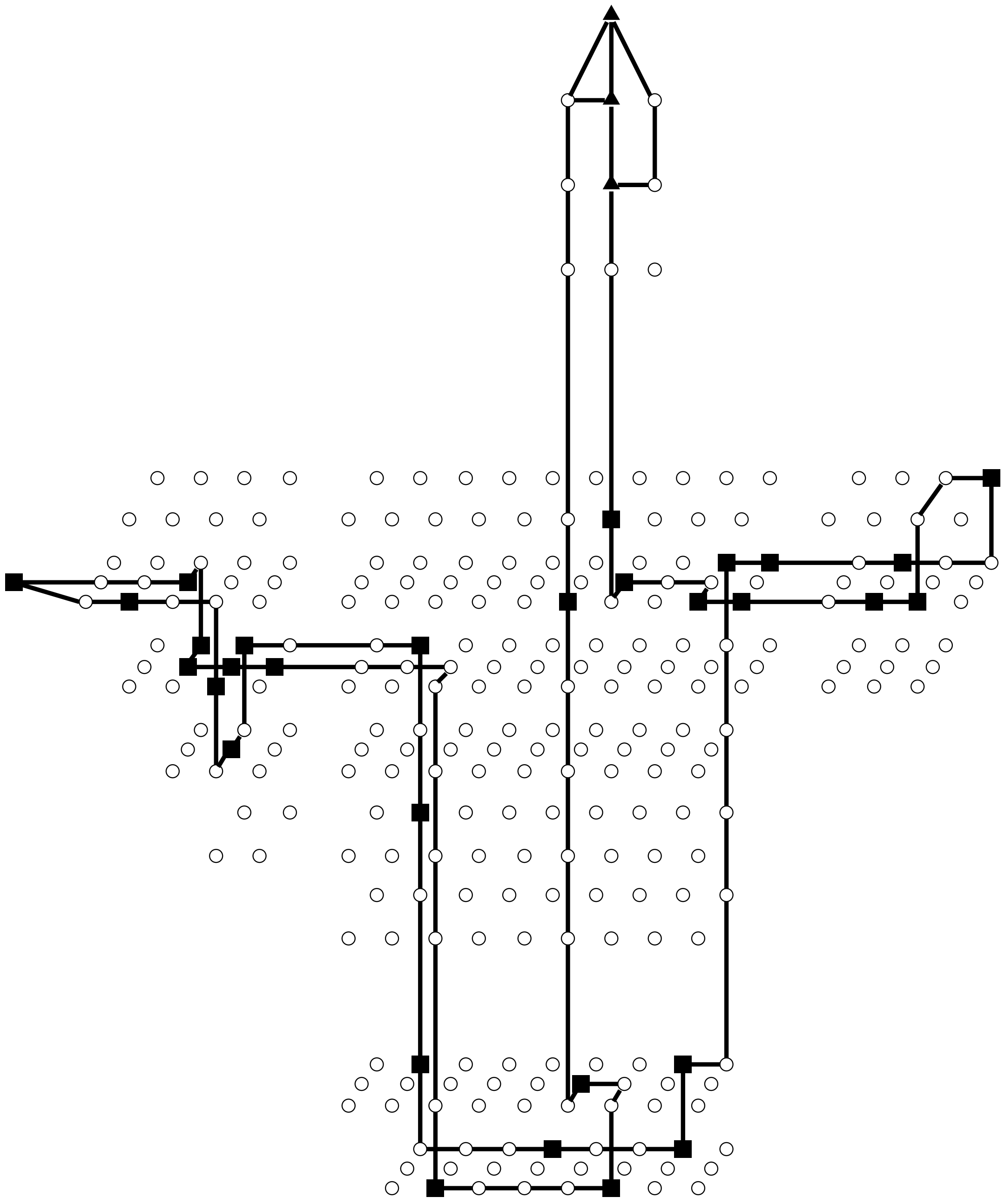

Figure 4: Optimum solution of reduced “ship23” problem

References

[BBM90] D. Bienstock, E. F. Brickell and C. L. Monma. On the structure of minimum-weight k-connected spanning networks. *SIAM Journal on Discrete Mathematics*, 3, 320–329, 1990.

[BC88] Bellcore, FIBER OPTIONS, Software for designing survivable optimal fiber networks. Software Package, December 1988.

[C73] V. Chvátal. Edmonds Polytopes and a hierarchy of combinatorial problems, *Discrete Mathematics* **4**, 305–337, 1973.

[CF70] W. Chou and M. Frank. Survivable communication networks and the terminal capacity matrix. *IEEE Transactions on Circuit Theory*, CT-17: 183–192, 1970.

[CMW89] R.H. Cardwell, C. L. Monma, and T-H. Wu. Computer-aided design procedures for survivable fiber optic networks. *IEEE Journal on Selected Areas in Communications*, **7**, 1188–1197, 1989.

[E65] J. Edmonds. Maximum matching and a polyhedron with $0,1$ vertices. *Journal of Research of the National Bureau of Standards*, **69**, 125–130, 1965.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin, 1988.

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *SIAM Journal on Applied Mathematics*, **9**, 551–570, 1961.

[GM90] M. Grötschel and C. L. Monma. Integer polyhedra associated with certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics*, **3**, 502–523, 1990.

[GMS92a] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research,* **40**, 309–330, 1992.

[GMS92b] 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**, 474–504, 1992.

[GMS91] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral approaches to network survivability. In F. Roberts, F. Hwang, and C. L. Monma, editors, *Reliability of Computer and Communication Networks*, Series in Discrete Mathematics and Computer Science, **5**, 121–141. AMS/ACM, 1991.

[G87] D. Gusfield. Very simple algorithms and programs for all pairs network flow analysis. Computer Science Division, University of California, Davis, 1987.

[HO92] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a graph. *Proceedings of the Third Annual ACM-SIAM-Symposium on Discrete Algorithms*, Orlando, Florida, 165–174, 1992.

[KM89] C. W. Ko and C. L. Monma. Heuristic methods for designing highly survivable communication networks, Technical Report, Bellcore, 1989.

[MMP90] C. L. Monma, B. S. Munson, and W. R. Pulleyblank. Minimum-weight two-connected spanning networks, *Mathematical Programming*, **46**, 153–171, 1990.

[MS89] C. L. Monma and D. F. Shallcross. Methods for designing communication networks with certain two-connected survivability constraints. *Operations Research*, **37**, 541, 1989.

[PR82] M. Padberg, R. Rao. Odd minimum cut sets and b-matchings, *Mathematics of Operations Research*, **7**, 67–80, 1982.

[S92] M. Stoer. Design of survivable networks, Ph.D. thesis, University of Augsburg, 1992, to appear in Lecture Notes in Mathematics, Springer, Heidelberg.