Topology-Constrained Network Design

Bernard Fortz

Topology-Constrained Network Design

Bernard Fortz

1 Introduction

Many network design problems considered in this book aim at optimizing simultaneously the decisions on opening links (with an associated fixed cost) and the capacity to allocate to these links in order to satisfy a set of demands, with a variable routing cost associated to these demands. However, in certain situations, the demand is not known in advance, or involves a lot of uncertainty, leading to an approach in two phases, where the topological design of the network (considering only fixed cost of opening links) is considered first, and the decisions on routing and capacity allocation taken in a second (later) stage. This approach is relevant when the fixed costs are high compared to routing and capacity costs, and/or when topological decisions do not affect too much capacity decisions. For example, in telecommunications, fiber optic cables have a virtually unlimited capacity, and the limitation of capacity arises from equipment placed in the nodes of the network (routing cards). While decisions to dig a trench to lay a cable are very costly and must be taken over a long term horizon, increasing capacity by adding or upgrading equipment into nodes is relatively simple and cheap.

In this chapter, we study models and techniques for long-term planning of the first phase, i.e., we only deal with topological aspects. Two main issues appear in the planning process of networks: economy and survivability. Economy refers to the construction cost, which is expressed as the sum of the edge costs, while survivability refers to the restoration of services in the event of node or link failure. The goal is then to determine a set of links connecting all nodes under some survivability criteria.

For example, in telecommunications, the network is seen as a given set of nodes and a set of possible fiber links that have to be placed between these nodes to achieve connectivity and survivability at minimum cost. Until about 25 years ago, the limited

Bernard Fortz
Computer Science Department, Université libre de Bruxelles, Belgium and INOCS, INRIA Lille Nord-Europe, France, e-mail: bernard.fortz@ulb.ac.be

capacity of copper cables resulted in highly diverse routing. The developments in fiber-optic technology have led to components that are cheap and reliable, having an almost unlimited capacity. The introduction of such a technology has made hierarchical routing and bundling of traffic very attractive. This approach has resulted in sparse, even treelike network topologies with larger amounts of traffic carried by each link. Trees satisfy the primary goal of minimizing the total cost while connecting all nodes. However, only one node or edge breakdown causes a tree network to fail in its main objective of enabling communication between all pairs of nodes.

This means that some survivability constraints have to be considered while building the network. Losing end-to-end customer service could lead to dramatic loss of revenue for commercial service providers. Constructing network topologies that provide protection against failures has become one of the most important problems in the field of network design.

The most studied models deal with k-connectivity requirements, i.e., the ability to restore network service in the event of a failure of at most $k-1$ components of the network. Among them, the minimum-cost two-connected spanning network problem consists in finding a network with minimal total cost for which two node-disjoint paths are available between every pair of nodes. This means that two-connected networks are able to deal with a single link or node failure. Two-connected networks have been found to provide a sufficient level of survivability in most cases, and a considerable amount of research has focused on so-called *low-connectivity constrained* network design problems, i.e., problems for which each node j is characterized by a requirement $r_j \in \{0,1,2\}$ and $\min\{r_i, r_j\}$ node-disjoint paths between every pair of nodes i, j are required.

Two-connectivity seems a sufficient level of survivability for most networks, since the probability of dealing with two simultaneous failures is usually very low for fiber optics technologies used in telecommunications networks. However, it turns out that the optimal solution of this problem is often very sparse (in many cases such as a Hamiltonian cycle). In such a topology, primary routing paths and re-routing paths in case of failure might become very long. This introduces another difficulty as it causes large delays in the network.

To avoid this, two kinds of solutions have been proposed in the literature. The first one imposes a constraint on the length of the paths (in terms of number of links crossed), the so-called *hop-constrained* models. The second approach consists of imposing that each edge belongs to at least one cycle whose length is bounded by a given constant, which ensures the existence of an alternate short path in case of failure.

The chapter is organized as follows. After introducing in Section 2 the specific notation and some fundamental definitions used in the chapter, we begin by the simplest but fundamental problem: the design of connected networks (and in particular the minimum spanning tree problem) is covered in Section 3. Next we turn our attention to networks requiring a higher level of survivability in Section 4. Sections 5 and 6 consider problems in which the length of re-routing paths in case of failure is limited, by introducing hop constraints and rings of bounded lengths, respectively. Section 7 provides links to the relevant literature that was used as basis to this chapter, as well as interesting references to dig further. We conclude in Section 8 with some perspectives on future trends in topological network design.

2 Notation and Definitions

Most models considered in this chapter are based on *undirected graphs* as capacity is not involved and links are usually bi-directional. Therefore, the given sets of nodes and possible connections are represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is the set of *nodes* and $\mathcal{E}$ is the set of *edges* that represent the possible pairs of nodes between which a direct connection can be established. The graph $\mathcal{G}$ may have parallel edges but should not contain loops. Throughout this chapter, $n = |\mathcal{N}|$ and $m = |\mathcal{E}|$ will denote the number of nodes and edges of $\mathcal{G}$.

Given a subset of nodes $\mathcal{S} \subset \mathcal{N}$, the edge set

$$\delta(\mathcal{S}) = \{\{i,j\} \in \mathcal{E} \mid i \in \mathcal{S},\ j \in \mathcal{N} \backslash \mathcal{S}\}$$

is called the *cut* induced by $\mathcal{S}$. We write $\delta_{\mathcal{G}}(\mathcal{S})$ to make clear — in case of possible ambiguities — with respect to which graph the cut induced by $\mathcal{S}$ is considered. For a single node $i \in \mathcal{N}$, we denote $\delta(i) = \delta(\{i\})$. The set

$$\mathcal{E}(\mathcal{S}) = \{\{i,j\} \in \mathcal{E} \mid i \in \mathcal{S},\ j \in \mathcal{S}\}$$

is the set of edges having both end nodes in $\mathcal{S}$. We denote by $\mathcal{G}(\mathcal{S}) = (\mathcal{S}, \mathcal{E}(\mathcal{S}))$ the subgraph induced by edges having both end nodes in $\mathcal{S}$. If $\mathcal{E}(\mathcal{S})$ is empty, $\mathcal{S}$ is an *independent set*. $\mathcal{G}/\mathcal{S}$ is the graph obtained from $\mathcal{G}$ by contracting the nodes in $\mathcal{S}$ to a new node w (retaining parallel edges). Given two subsets of nodes $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, the subset of edges having one endpoint in each subset is denoted by

$$[\mathcal{S}_1 : \mathcal{S}_2] = \{\{i,j\} \in \mathcal{E} \mid i \in \mathcal{S}_1,\ j \in \mathcal{S}_2\}.$$

We denote by $\mathcal{N} - z = \mathcal{N} \backslash \{z\}$ and $\mathcal{E} - e = \mathcal{E} \backslash \{e\}$ the subsets obtained by removing one node or one edge from the set of nodes or edges. $\mathcal{G} - z$ denotes the graph $(\mathcal{N} - z, \mathcal{E} \backslash \delta(z))$, i.e., the graph obtained by removing a node z and its incident edges from $\mathcal{G}$. This is extended to a subset $\mathcal{Z} \subset \mathcal{N}$ of nodes by the notation $\mathcal{G} - \mathcal{Z} = (\mathcal{N} \backslash \mathcal{Z}, \mathcal{E} \backslash (\delta(\mathcal{Z}) \cup \mathcal{E}(\mathcal{Z})))$.

Each edge $e = \{i,j\} \in \mathcal{E}$, has a *fixed cost* $c_e = c_{ij}$ representing the cost of establishing the direct link connection, and a *length* $d_e = d_{ij} = d(i,j)$. It is assumed throughout this work that these edge lengths satisfy the *triangle inequality*, i.e.,

$$d(i,j) + d(j,k) \geq d(i,k) \quad \text{for all } i,j,k \in \mathcal{N}.$$

The cost of a network $(\mathcal{N}, \mathcal{F})$ where $\mathcal{F} \subseteq \mathcal{E}$ is a subset of possible edges is denoted by $c(\mathcal{F}) = \sum_{e \in \mathcal{F}} c_e$. The *distance* between two nodes i and j in this network is denoted by $d_{\mathcal{F}}(i,j)$ and is given by the length of a shortest path linking these two nodes in $\mathcal{F}$.

Without loss of generality, all costs are assumed to be nonnegative, because an edge e with a negative cost c_e will be contained in any optimum solution.

For any pair of distinct nodes $s, t \in \mathcal{N}$, an $[s,t]$-*path* $\mathcal{P}$ 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 to 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 $\mathcal{P}$. A collection $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ 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 (other than s and t) appears in more than one path. A *cycle* (containing s and t) is a set of two node-disjoint $[s,t]$-paths. A *Hamiltonian cycle* is a cycle using each node of the network exactly once. A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is k-*edge-connected* (resp., k-*node-connected*) if, for each pair s, t of distinct nodes, $\mathcal{G}$ contains at least k edge-disjoint (resp., node-disjoint) $[s,t]$-paths.

When the type of connectivity is not mentioned, we assume node-connectivity. The *edge connectivity* (resp., *node-connectivity*) of a graph is the maximal k for which it is k-edge-connected (resp., k-node-connected). A 1-edge-connected network is also 1-node-connected, and we call it simply *connected*. A cycle-free graph is a *forest* and a connected forest is a *tree*. A *connected component* of a graph is a maximal connected subgraph. If $\mathcal{G} - e$ has more connected components than $\mathcal{G}$ for some edge e, we call e a *bridge*. Similarly, if $\mathcal{Z}$ is a node set and $\mathcal{G} - \mathcal{Z}$ has more connected components than $\mathcal{G}$, we call $\mathcal{Z}$ an *articulation set* of $\mathcal{G}$. If a single node forms an articulation set, the node is called *articulation point*.

Node and edge-disjoint $[s,t]$-paths are related to cuts and articulation sets by Menger's theorem:

Theorem 1 (Menger (1927)).

1. *In a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, there is no cut of size $k - 1$ or less disconnecting two given nodes s and t, if and only if there exist at least k edge-disjoint $[s,t]$-paths in $\mathcal{G}$.*
2. *Let s and t be two nonadjacent nodes in $\mathcal{G}$. Then there is no articulation set $\mathcal{Z}$ of size $k - 1$ or less disconnecting s and t, if and only if there exist at least k node-disjoint $[s,t]$-paths in $\mathcal{G}$.*

In order to formulate network design problems as integer linear programs, we associate with every subset $\mathcal{F} \subseteq \mathcal{E}$ an *incidence vector* $y^{\mathcal{F}} = (y_e^{\mathcal{F}})_{e \in \mathcal{E}} \in \{0,1\}^{|\mathcal{E}|}$ by setting

$$y_e^{\mathcal{F}} = \begin{cases} 1 & \text{if } e \in \mathcal{F}, \\ 0 & \text{otherwise}. \end{cases}$$

Conversely, each vector $y \in \{0,1\}^{|\mathcal{E}|}$ induces a subset

$$\mathcal{F}^y = \{e \in \mathcal{E} \mid y_e = 1\}$$

of the edge set $\mathcal{E}$. For any subset of edges $\mathcal{F} \subseteq \mathcal{E}$ we define

$$y(\mathcal{F}) = \sum_{e \in \mathcal{F}} y_e.$$

3 Connected Networks

A fundamental constraint in topological network design is to ensure all nodes can communicate. This translates into the constraint that a path must exist between any pair of nodes in the graph, or in other terms, the constructed graph must be connected.

The problem of finding a minimum cost connected network is polynomially solvable: if costs are non-negative, there exists an optimal solution with the minimum number $n-1$ of edges in a connected graph, hence the problem reduces to the well-known *Minimum Spanning Tree* problem. A simple algorithm to solve it is the greedy algorithm: start with an empty solution; order the edges of $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ by increasing costs; iteratively consider each edge in this sorted list and add it to the solution if it does not form a cycle with the edges already selected.

The property that a spanning tree is a graph with $n-1$ edges and without cycles is the basis of the greedy algorithm described above. This property also leads to an integer programming formulation of the problem. Let y_e be a binary variable indicating whether edge $e \in \mathcal{E}$ is part of the spanning tree. Then the minimum spanning tree problem can be formulated as

$$\text{Minimize} \quad \sum_{e \in \mathcal{E}} c_e y_e \tag{1}$$

$$\text{Subject to} \quad y(\mathcal{E}) = n-1, \tag{2}$$

$$y(\mathcal{E}(\mathcal{S})) \leq |\mathcal{S}| - 1 \quad \forall \emptyset \neq \mathcal{S} \subset \mathcal{N}, \tag{3}$$

$$y_e \in \{0,1\}, \qquad \forall e \in \mathcal{E}, \tag{4}$$

where constraint (2) imposes the cardinality constraint and constraints (3) are subtour elimination constraints that eliminate all possible cycles from the solution. Although there is an exponential number of subtour elimination constraints, they can be separated in polynomial time by a simple minimum cut computation.

Primal-dual arguments can be used to show that the linear programming relaxation of formulation (1)-(4) is integer, and its extreme points coincide with the incidence vectors of spanning trees. Moreover, the same arguments can be used to prove the correctness of the greedy algorithm.

Another formulation for the minimum spanning tree problem is obtained by considering a spanning tree as a connected subgraph with $n-1$ edges. Connectivity can be imposed by forcing each cut in the graph to contain at least one edge, leading to the cut-set formulation

$$\text{Minimize} \quad \sum_{e \in \mathcal{E}} c_e y_e \tag{5}$$

$$\text{Subject to} \quad y(\mathcal{E}) = n-1, \tag{6}$$

$$y(\delta(\mathcal{S})) \geq 1 \quad \forall \emptyset \neq \mathcal{S} \subset \mathcal{N}, \tag{7}$$

$$y_e \in \{0,1\}, \quad \forall e \in \mathcal{E}, \tag{8}$$

where subtour elimination constraints have been replaced by cut inequalities (7).

In general, the linear relaxation of (5)-(8) has fractional extreme points and therefore its polytope strictly contains the polytope induced by (1)-(4). Cut inequalities can be generalized: Consider a partition $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p$ of $\mathcal{N}$ into p nonempty subsets. Any spanning tree contains at least $p-1$ edges joining the sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p$, leading to the valid inequality

$$\frac{1}{2}\sum_{i=1}^{p} y(\delta(\mathcal{S}_i)) \geq p-1 \tag{9}$$

that is usually called *partition inequality* or *multi-cut inequality*. Cut inequalities (7) are a special case of (9) with $p = 2$. Again, it is possible to show that the polyhedron of the linear relaxation of the formulation given by (6), (9), (8) has integer extreme points. Hence the linear relaxations of the multi-cut formulation and of the subtour formulation are equivalent and both define the convex hull of incidence vectors of spanning trees.

The subtour and multi-cut formulations are ideal, but are not usable as such in practice, as they suffer from an exponential number of constraints. However, a cutting-plane approach can be used, as both classes of inequalities can be separated in polynomial time.

Another approach is to model the problem with extended formulations, by introducing new sets of variables but keeping the number of constraints polynomial. One of these extended formulations makes use of directed flows between an arbitrarily chosen *root node* $r \in \mathcal{N}$ and all the other nodes to impose connectivity. Indeed, the constructed graph is connected if and only if there exists a path from r to every other node $k \in \mathcal{N} \setminus \{r\}$. To this aim, let us consider each node $k \neq r$ as a commodity, where one unit of flow originates at node r and must be delivered to node k. In order to represent these flows, we define $\mathcal{A} = \{(i,j), (j,i) | \{i,j\} \in \mathcal{E}\}$ as the directed set of arcs obtained by replacing each edge by two arcs in opposite direction. As in previous chapters, let x_{ij}^k be the flow of commodity k in arc (i,j). We can then formulate the minimum spanning tree problem as:

$$\text{Minimize } \sum_{e \in \mathcal{E}} c_e y_e \tag{10}$$

$$\begin{aligned}
\text{Subject to } \quad & \textstyle\sum_{j \in \mathcal{N}_i^+} x_{ij}^k - \sum_{j \in \mathcal{N}_i^-} x_{ji}^k = w_i^k, && \forall i \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{r\} && (11)\\
& x_{ij}^k + x_{ji}^{k'} \leq y_e, && \forall e = \{i,j\} \in \mathcal{E}, \forall k, k' \in \mathcal{N} \setminus \{r\}, && (12)\\
& y(\mathcal{E}) = n-1, && && (13)\\
& x_{ij}^k \geq 0, && \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{N} \setminus \{r\}, && (14)\\
& y_e \in \{0,1\}, && \forall e \in \mathcal{E}, && (15)
\end{aligned}$$

where, for each $i \in \mathcal{N}$, we define

$$\mathcal{N}_i^+ = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}, \quad \mathcal{N}_i^- = \{j \in \mathcal{N} : (j,i) \in \mathcal{A}\}. \tag{16}$$

Flow balance constraints (25) define a path between r and k with

$$w_i^k = \begin{cases} 1, \text{ if } i = r, \\ -1, \text{ if } i = k, \\ 0, \text{ otherwise.} \end{cases} \tag{17}$$

Each edge is given a natural direction as flows are directed away from the root r of the tree. Constraints (12) ensure that flow is sent only on edges present in the spanning tree, and always in the same direction. Finaly (13) are the usual cardinality constraints.

This formulation can also be seen as an application of the max flow-min cut theorem, and again leads to a complete description of the convex hull of incidence vectors of spanning trees (or to be more precise, the projection of the formulation on the space of y-variables defines this polyhedron).

Now consider the general case where costs are not restricted to be positive. Then a minimum cost connected network is not necessarily a tree, as all negative cost edges should belong to the optimal solution, possibly creating cycles. Hence the subtour formulation is not valid anymore. However, the multi-cut and the flow formulations become valid if cardinality constraints (6) and (13) are removed. Moreover, the obtained formulations are ideal again in the sense that their linear relaxation (or its projection) describe the convex hull of incidence vectors of connected networks.

4 Survivable Networks

The major problem with the models presented above is that the topology tends to be sparse, as costs are minimized, the most extreme case being the minimum spanning tree. However, networks are subject to failures. For example, in the context of telecommunications, a network is seen as a set of gateway nodes (routers or telephone offices) and (fiber) links that are placed between nodes. If connectivity is the only constraint imposed on the network, a single link or node failure will disconnect it, which is clearly not acceptable.

In this context, survivability refers to the restoration of services in the event of node or link failure, or, in other words, a network is survivable if there exists a prespecified number of node-disjoint or edge-disjoint paths between any two nodes. Again, the only costs considered are construction costs, like the cost of digging trenches and placing a fiber cable into service.

A considerable amount of research has focused on low-connectivity constrained network design problems. These models can be described informally as follows: a set of nodes that have to be connected by a network is given. These nodes are classified according to their importance, namely the

- *special nodes*, for which a “high” degree of survivability has to be ensured in the network to be constructed;
- *ordinary nodes*, which have to be simply connected to the network;

- *optional nodes*, which may not be part of the network at all.

The pairs of nodes between which a direct transmission link can be placed are also given, together with the cost of placing the fiber cable and putting it into service. The problem now consists in determining where to place fiber cables so that the construction cost, i.e., the sum of the fiber cable costs, is minimized and certain survivability constraints are ensured. For instance, we may require that

- the destruction of any single link may not disconnect any two special nodes, or
- the destruction of any single node may not disconnect any two special nodes.

These requirements are equivalent to ask that there exist

- at least two edge-disjoint paths, or
- at least two node-disjoint paths

between any two special nodes.

Higher survivability levels may be imposed by requiring the existence of three or more paths between certain pairs of nodes according to their importance class. However, up to now, low-connectivity requirements have been found to provide a sufficient level of survivability for telecommunications operators.

In graph-theoretic language, the set of nodes and possible link connections can be represented again by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The survivability requirement or importance of a node is modeled by node types. In particular, each node $s \in \mathcal{N}$ has associated nonnegative integer r_s, the *type* of s. Sometimes, we also write $r(s)$ instead of r_s. A network $(\mathcal{N}, \mathcal{F})$, where $\mathcal{F} \subseteq \mathcal{E}$ is a subset of the possible links, is said to satisfy the *node-connectivity requirements*, if, for each pair $s, t \in \mathcal{N}$ of distinct nodes, $(\mathcal{N}, \mathcal{F})$ contains at least

$$r(s,t) = \min\{r_s, r_t\}$$

node-disjoint $[s,t]$-paths.

Similarly, we say that $(\mathcal{N}, \mathcal{F})$ satisfies the *edge-connectivity requirements*, if, for each pair $s, t \in \mathcal{N}$ of distinct nodes, the network contains at least $r(s,t)$ edge-disjoint $[s,t]$-paths. If all node types have the same value k, it is equivalent to request that $(\mathcal{N}, \mathcal{F})$ is k-node-connected or k-edge-connected.

We consider here the low-connectivity requirements, i.e., node types $r_s \in \{0, 1, 2\}$. Using our previous classification of nodes,

- special nodes have type 2,
- ordinary nodes have type 1, and
- optional nodes have type 0.

To shorten notation, we extend the type function r to sets by setting

$$\begin{aligned} r(\mathcal{S}) &= \max\{r_s \mid s \in \mathcal{S}\} \text{ for all } \mathcal{S} \subseteq \mathcal{N}, \text{ and} \\ \operatorname{con}(\mathcal{S}) &= \max\{r(s,t) \mid s \in \mathcal{S}, t \in \mathcal{N} \backslash \mathcal{S}\} \\ &= \min\{r(\mathcal{S}), r(\mathcal{N} \backslash \mathcal{S})\} \qquad\qquad \forall \mathcal{S} \subset \mathcal{N}, \emptyset \neq \mathcal{S} \neq \mathcal{N}. \end{aligned}$$

We write $\operatorname{con}_G(\mathcal{S})$ to make clear with respect to which graph $\operatorname{con}(\mathcal{S})$ is considered.

We can now formulate the connectivity constrained network design problem as the following integer linear program:

$$\text{Minimize} \sum_{e \in \mathcal{E}} c_e y_e \tag{18}$$

$$\begin{aligned}
\text{Subject to} \qquad y(\delta(\mathcal{S})) &\geq \operatorname{con}(\mathcal{S}) && \forall \mathcal{S} \subset \mathcal{N},\ \emptyset \neq \mathcal{S} \neq \mathcal{N}, && (19)\\
y(\delta_{G-z}(\mathcal{S})) &\geq \operatorname{con}_{G-z}(\mathcal{S}) && \forall z \in \mathcal{N},\ \mathcal{S} \subset \mathcal{N} \backslash \{z\}, &&\\
& && \qquad \emptyset \neq \mathcal{S} \neq \mathcal{N} \backslash \{z\}, && (20)\\
y_e &\in \{0,1\} && \forall e \in \mathcal{E}. && (21)
\end{aligned}$$

It follows from Menger's Theorem that, for any feasible solution y of this program, the subgraph $(\mathcal{N}, \mathcal{F}^y)$ of $\mathcal{G}$ defines a network satisfying the node-connectivity requirements. Removing (20), we obtain an integer linear program for edge-connectivity requirements. Inequalities (19) are called *cut inequalities*, while inequalities (20) are called *node cut inequalities*.

The connectivity constrained network design problem is NP-hard in general. In particular :

- If $r_s \in \{0,1\}$, $\forall s \in \mathcal{N}$, it reduces to the well-known NP-hard Steiner tree problem in networks.
- If $r_s = 2$, $\forall s \in \mathcal{N}$, it consists in determining a minimum cost two-connected network. This last problem is NP-hard even if the graph is complete and costs satisfy the triangle inequality, since with an algorithm for this problem, one could decide whether a graph has a Hamiltonian cycle by associating a cost equal to 1 to all graph edges and cost equal to 2 to all non-graph edges.

However, for some particular connectivity requirements or costs, or when the underlying graph $\mathcal{G}$ is restricted, the problem may become polynomially solvable:

- If $r_s = 1$, $\forall s \in \mathcal{N}$, the problem reduces to the minimum cost connected network problem studied in the previous section.
- If $r_s = 1$ for exactly two nodes of $\mathcal{N}$ and $r_s = 0$ for all the other nodes, the problem becomes a shortest path problem.
- If $r_s = k$, $k \geq 2$, for exactly two nodes of $\mathcal{N}$ and $r_s = 0$ for all the other nodes, the problem becomes a k-shortest paths problem.
- If $r_s \in \{0,1\}$, $\forall s \in \mathcal{N}$, the problem reduces to the Steiner tree problem in networks. This problem is NP-hard in general, but solvable in polynomial time in the case where either the number of nodes of type 0 or the number of nodes of type 1 is restricted.

The general formulation described above and many of its special cases received a lot of attention. Polyhedral results have been obtained for different special cases of the model (see Section 7 for references to surveys on the subject).

An important class of valid inequalities for k-node-connected networks are obtained by a generalization of partition inequalities (9). First, we observe that the deletion of $k-1$ nodes from a k-node-connected network leaves a connected graph. Thus, if $\mathcal{Z} \subseteq \mathcal{N}$ is a node set with exactly $k-1$ nodes and $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p$ ($p \geq 2$) is a partition of $\mathcal{N} \backslash \mathcal{Z}$ into p nonempty subsets, the inequality

$$\frac{1}{2} \sum_{i=1}^{p} y(\delta_{\mathcal{G}-\mathcal{Z}}(\mathcal{S}_i)) \geq p-1 \tag{22}$$

is valid for the polytope of k-node-connected networks. These inequalities are called *node-partition inequalities*, and can be seen as a generalization of node cut inequalities (20).

When $r(s) = 2$ for all $s \in \mathcal{N}$, partition inequalities can be generalized further: consider again a patition $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p$ ($p \geq 2$) of $\mathcal{N}$ and let $\mathcal{F} \subseteq \delta(\mathcal{S}_1)$ with $|\mathcal{F}|$ odd. The *$\mathcal{F}$-partition inequality* is defined as

$$\frac{1}{2} \sum_{i=1}^{p} y(\delta(\mathcal{S}_i)) - y(\mathcal{F}) \geq p - \left\lceil \frac{|\mathcal{F}|}{2} \right\rceil. \tag{23}$$

To show it is valid, simply add valid inequalities

$$\begin{aligned} y(\delta(\mathcal{S}_i)) \geq 2 \quad & \forall\, i = 2, \ldots, p, \\ -y_e \geq -1 \quad & \forall e \in \mathcal{F}, \\ y_e \geq 0 \quad & \forall e \in \delta(\mathcal{S}_1) \setminus \mathcal{F}, \end{aligned}$$

divide by 2 and round up to obtain (23).

The formulation above is in the space of design variables only, and involves an exponential number of constraints. As in Section 3, it is also possible to obtain valid polynomial-size models by the introduction of flow variables. We do it here only for edge-connectivity requirements. Assuming that we want to construct a network in which there are $r(s,t)$ edge-disjoint paths between nodes s and t, we can define a set $\mathcal{K}$ of commodities where we have one commodity for each node pair s,t such that $r(s,t) > 0$, with a flow requirement of $r(s,t)$ where the source and sink of commodity k are arbitrarily chosen between s and t.

$$\text{Minimize} \sum_{e \in \mathcal{E}} c_e y_e \tag{24}$$

$$\text{Subject to} \quad \textstyle\sum_{j \in \mathcal{N}_i^+} x_{ij}^k - \sum_{j \in \mathcal{N}_i^-} x_{ji}^k = w_i^k, \quad \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}, \tag{25}$$

$$x_{ij}^k + x_{ji}^k \leq y_e, \qquad \forall\, e = \{i,j\} \in \mathcal{E}, \forall\, k \in \mathcal{K}, \tag{26}$$

$$x_{ij}^k \geq 0, \qquad \forall\, (i,j) \in \mathcal{A}, \forall\, k \in \mathcal{K}, \tag{27}$$

$$y_e \in \{0,1\}, \qquad \forall\, e \in \mathcal{E}, \tag{28}$$

Flow balance constraints (25) define $r(s,t)$ paths between s and t for each commodity $k \in \mathcal{K}$ corresponding to the node pair s,t, with

$$w_i^k = \begin{cases} r(s,t), \text{ if } i = s, \\ -r(s,t), \text{ if } i = t, \\ 0, \text{ otherwise}, \end{cases} \tag{29}$$

and (26) impose that these paths are edge-disjoint.

Applying simple max-flow / min-cut arguments, it is quite easy to see that the linear relaxation of the flow formulation is equivalent to the formulation involving only design variables with an exponential number of constraints.

5 Hop Constraints

The models from the previous section usually lead to designs that are very sparse. In general, the survivability constraints alone may not be sufficient to guarantee a cost effective routing with a good quality of service. The reason for this is that the routing paths may be too long, leading to unacceptable delays. Since in most of the routing technologies, delay is caused at the nodes, it is usual to measure the delay in a path in terms of its number of intermediate nodes, or equivalently, its number of arcs (or hops). Thus, to guarantee the required quality of service, one can impose a limit on the number of arcs in the routing paths.

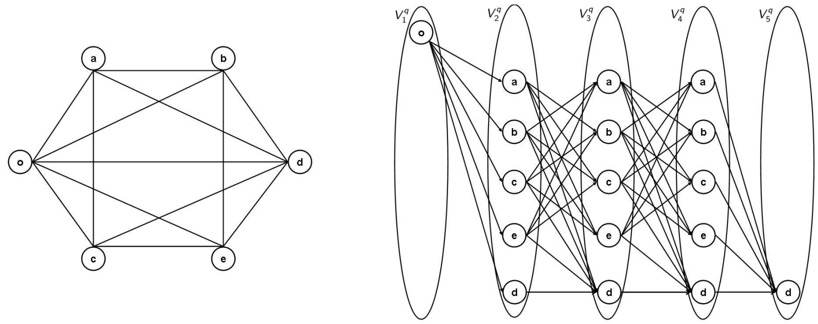
Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with nonnegative edge costs c_e, $e \in \mathcal{E}$, and a set of node pairs $\mathcal{K}$ (sometimes called commodities), we study the problem of constructing a minimum cost set of edges so that the induced subgraph contains at least K edge-disjoint paths with at most L edges between each pair in $\mathcal{K}$.

In theory, we could formulate the limit on the number of hops in a path in the space of design variables only. This approach was used by several authors (see Section 7) but has a major drawback: it leads to formulations with an exponential number of constraints, some of which are really hard to interpret and also difficult to handle numerically (the associated separation problem being NP-hard).

In this chapter, we prefer to use extended formulations, once again based on multi-commodity flow variables, that allow to model the limit on the path length more naturally. Moreover, these formulations imply all the valid inequalities known for formulations in the space of design variables. More precisely, the projection of the extended formulations presented here on the space of design variables strictly contains the polyhedron defined by formulations in the space of design variables presented in the literature so far.

The basic idea is to use a layered representation of graph $\mathcal{G}$ to implicitly force each path to use at most L edges. We model the subproblem associated with each commodity with a directed graph composed of $L+1$ layers as illustrated in Figure 1.

Namely, from the original non-directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, we create a directed layered graph $\mathcal{G}^q = (\mathcal{N}^q, \mathcal{A}^q)$ for each commodity $q \in \mathcal{K}$, where $\mathcal{N}^q = \mathcal{N}_1^q \cup \ldots \cup$



(a) (b)

Fig. 1 Original network (a) and its alternative (or associated) layered representation (b) when $L = 4$

$\mathcal{N}^q_{L+1}$ with $\mathcal{N}^q_1 = \{o(q)\}$, $\mathcal{N}^q_{L+1} = \{d(q)\}$ and $\mathcal{N}^q_l = \mathcal{N} \backslash \{o(q)\}$, $l = 2, \ldots, L$. Let v^q_l be the copy of $v \in \mathcal{N}$ in the l-th layer of graph $\mathcal{G}^q$. Then, the arc sets are defined by $\mathcal{A}^q = \{(i^q_l, j^q_{l+1}) \mid \{i, j\} \in \mathcal{E}, i^q_l \in \mathcal{N}^q_l, j^q_{l+1} \in \mathcal{N}^q_{l+1}, l \in \{1, \ldots, L\}\} \cup \{d(q)^l, d(q)^{l+1}, l \in \{2, \ldots, L\}\}$, see Figure 1. In the sequel, an (undirected) edge in $\mathcal{E}$ with endpoints i and j is denoted $\{i, j\}$ while a (directed) arc between $i^q_l \in \mathcal{N}^q_l$ and $j^q_{l+1} \in \mathcal{N}^q_{l+1}$ is denoted by (i, j, l) (the commodity q is omitted in the notation as it is often clear from the context).

Note that each path between $o(q)$ and $d(q)$ in the layered graph G^q is composed of exactly L arcs (hops), which correspond to a maximum of L edges (hops) in the original one. In fact this is the main idea of this transformation, since in the layered graph any path is feasible with respect to the hop-constraints. The usual multi-commodity flow equations defined in this layered graph yield the following model:

$$\text{Minimize} \sum_{e \in \mathcal{E}} c_e y_e \tag{30}$$

Subject to

$$\sum_{j \in \mathcal{N}^+_i} x^{lq}_{ij} - \sum_{j \in \mathcal{N}^-_i} x^{l-1\,q}_{ji} = w^q_i, \quad \forall i \in \mathcal{N}^q, \ \forall l \in \{2, \ldots, L+1\}, \ \forall q \in \mathcal{K} \tag{31}$$

$$\sum_{l=1}^{L} \left(x^{lq}_{ij} + x^{lq}_{ji} \right) \le y_e, \qquad \forall e = \{i, j\} \in \mathcal{E}, \forall q \in \mathcal{K}, \tag{32}$$

$$x^{lq}_{ij} \ge 0, \ integer, \qquad \forall (i, j, l) \in \mathcal{A}^q, \forall q \in \mathcal{K}, \tag{33}$$

$$y_e \in \{0, 1\}, \qquad \forall e \in \mathcal{E}, \tag{34}$$

$$\tag{35}$$

Flow balance constraints (31) define K paths between $o(q)$ and $d(q)$ in the layered graph $\mathcal{G}^q$ with

$$w_i^k = \begin{cases} K, \text{ if } i = o(q), \\ -K, \text{ if } i = d(q), \\ 0, \text{ otherwise.} \end{cases} \tag{36}$$

Constraints (32) guarantee that paths are edge-disjoint and only use installed edges.

This model can become very large when the number of commodities and the size of the network increase, but it can be solved efficiently using Benders decomposition. It is also interesting to mention that if there is only one commodity, and with non-negative costs, the linear relaxation of this model is always integral when $L \leq 3$.

6 Rings

As stated before, models from Section 4 lead to very sparse designs. In fact, it turns out that the optimal solution of the two-connected network problem is often a Hamiltonian cycle. Hence, any edge failure implies that the flow that was routed on that edge must be rerouted, using all the edges of the network, an obviously undesirable feature.

It is therefore necessary to add extra constraints to limit the region of influence of the traffic that it is necessary to reroute if a connection is broken. Hop constraints presented above are a possible way to achieve this. Another approach presented in this section is based on the technology of *self-healing rings*, quite popular in telecommunications networks. Self-healing rings are cycles in the network equipped in such a way that any link failure in the ring is automatically detected and the traffic rerouted by the alternative path in the cycle. It is natural to impose a limited length of these rings. This is equivalent to set a bound on the length of the shortest cycle including each edge.

This leads to the problem of designing a minimum cost network with the following constraints:

1. The constructed network contains at least two node-disjoint paths between every pair of nodes (*2-connectivity constraints*),
   and
2. each edge of the network belongs to at least one cycle whose length is bounded by a given constant K (*ring constraints*).

This problem is called the *Two-Connected Network with Bounded rings (2CNBR) problem*. The length of an edge can be unitary (i.e., similar to hop constraints), or can be weighted, to represent for example the physical delay for flowing through a given edge.

A useful tool to analyze feasible solutions of 2CNBR is the *restriction of a graph to bounded rings*. Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a constant $K > 0$, we define for each subset of edges $\mathcal{F} \subseteq \mathcal{E}$ its restriction to bounded rings $\mathcal{F}_K$ as

$$\mathcal{F}_K = \left\{ e \in \mathcal{F} : \begin{array}{l} e \text{ belongs to at least one cycle} \\ \text{of length less than or equal to } K \text{ in } \mathcal{F} \end{array} \right\}.$$

The subgraph $\mathcal{G}_K = (\mathcal{N}, \mathcal{E}_K)$ is the *restriction of* $\mathcal{G}$ *to bounded rings*. Note that an edge $e \in \mathcal{E} \backslash \mathcal{E}_K$ will never belong to a feasible solution of 2CNBR.

Further we denote by $\mathcal{Y}$ the set of incidence vectors of subsets $\mathcal{F} \subseteq \mathcal{E}$ such that

1. $\mathcal{F}$ is two-connected,
2. $\mathcal{F} = \mathcal{F}_K$.

Then, the 2CNBR problem consists in

$$\text{Minimize } \sum_{e \in \mathcal{E}} c_e y_e$$

$$\text{Subject to } \; y \in \mathcal{Y}.$$

Checking that $\mathcal{G}_K$ is two-connected, i.e., that $\mathcal{Y}$ is nonempty, can be done in polynomial time. We therefore assume in the remainder of this chapter that there always exists a feasible solution to the problem.

Since all costs c_e, $e \in \mathcal{E}$ are assumed to be nonnegative, there always exists an optimal solution of 2CNBR whose induced graph is minimal with respect to inclusion. More precisely, if $\mathcal{F}_K$ is two-connected, as $\mathcal{F} \supseteq \mathcal{F}_K$, $\mathcal{F}$ is also two-connected and the cost of $\mathcal{F}$ is greater than or equal to the cost of $\mathcal{F}_K$. We can thus relax the constraints and just require that $\mathcal{F}_K$ is two-connected for a set of edges $\mathcal{F}$ to be feasible. Hence, 2CNBR can be equivalently formulated as

$$\text{Minimize } \sum_{e \in \mathcal{E}} c_e y_e$$

$$\begin{aligned} \text{Subject to } \; & \mathcal{F}_K^y \text{ is two-connected}, \\ & y_e \in \{0,1\}, \qquad \forall e \in \mathcal{E}. \end{aligned}$$

In order to formulate the problem using only design variables y, observe that if a subset of edges $\mathcal{F} \subseteq \mathcal{E}$ is such that $(\mathcal{G} - \mathcal{F})_K$ is not two-connected, then $\mathcal{G} - \mathcal{F}$ does not contain a feasible solution. Therefore each feasible solution contains at least one edge from $\mathcal{F}$.

As we are only interested in minimal feasible solutions, this is sufficient to formulate the 2CNBR problem as the following integer linear program:

$$\text{Minimize } \sum_{e \in \mathcal{E}} c_e y_e \tag{37}$$

$$\text{Subject to } \; y(\mathcal{F}) \geq 1, \; \forall \mathcal{F} \subseteq \mathcal{E}, \; (\mathcal{G} - \mathcal{F})_K \text{ is not two-connected}, \tag{38}$$

$$y_e \in \{0,1\}, \; \forall e \in \mathcal{E}. \tag{39}$$

Constraints (38) are called *subset constraints*.

As feasible solutions of 2CNBR are two-connected graphs, valid inequalities for the design of 2-connected networks are also valid for 2CNBR. In particular, cut constraints (19) and node-partition inequalities (22) are very important in branch-and-cut strategies to solve the problem. However, for general 2-connected networks, cut constraints are facet-defining under very mild conditions, while in many cases they do not define facets for 2CNBR. By studying conditions for these inequalities to define facets, it is possible to strengthen them.

Given a subset of nodes $\mathcal{S} \subseteq \mathcal{N}$, $\emptyset \neq \mathcal{S} \neq \mathcal{N}$, the cut constraint imposes that there are at least two edges leaving $\mathcal{S}$, i.e.,

$$y(\delta(\mathcal{S})) \geq 2.$$

To characterize which cut constraints define facets, it is useful to know, for any pair of edges $e, f \in \delta(\mathcal{S})$, if there exists a solution whose incidence vector lies in the face $y(\delta(\mathcal{S})) = 2$, i.e., if there exists a feasible solution containing e and f but no other edge of $\delta(\mathcal{S})$). This is the case if and only if

$$C_{e,f} = \mathcal{E}(\mathcal{S}) \cup \mathcal{E}(\mathcal{N} \backslash \mathcal{S}) \cup \{e, f\}$$

is feasible, i.e., if $(C_{e,f})_K$ is two-connected.

A useful tool to represent and analyze the vectors belonging to the face defined by a cut constraint is the *ring-cut graph*: Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a constant $K > 0$, and a subset of nodes $\mathcal{S} \subset \mathcal{N}$, $\emptyset \neq \mathcal{S} \neq \mathcal{N}$, the ring-cut graph $RCG_{\mathcal{S},K} = (\delta(\mathcal{S}), RCE_{\mathcal{S},K})$ induced by $\mathcal{S}$ is the graph defined by associating one node to each edge in $\delta(\mathcal{S})$ and by the set of edges

$$RCE_{\mathcal{S},K} = \left\{ \{e, f\} \subseteq \delta(\mathcal{S}) : (C_{e,f})_K \text{ is two-connected} \right\}.$$

With the help of the ring-cut graph, it is possible to characterize which cut constraints are facet-defining. Moreover, the ring-cut graph can be used to derive new valid inequalities for 2CNBR. If $\mathcal{F} \subseteq \delta(\mathcal{F})$ is an independent subset in the ring-cut graph $RCG_{\mathcal{S},K}$, then

$$y(\mathcal{F}) + 2y(\delta(\mathcal{S}) \backslash \mathcal{F}) \geq 3 \tag{40}$$

is a valid inequality for the 2CNBR problem. Inequalities (40) are called *ring-cut inequalities*, and are also very useful to strengthen formulations of 2CNBR.

7 Bibliographical Notes

7.1 Connected Networks

The greedy algorithm for the Minimum Spanning Tree problem is due to Kruskal (1956). The tree polytope is a special case of the matroid polytope first described by Edmonds (1971). For a review on polyhedral results for tree related problems, see Magnanti and Wolsey (1995). Another compact formulation for the minimum spanning tree problem (not described here) was proposed by Martin (1991).

Grötschel, Monma and Stoer studied in detail network design problems with connectivity constraints. A survey of their work can be found in Grötschel et al. (1995a) and Stoer (1992). A later survey by Kerivin and Mahjoub (2005) concentrates on polyhedral aspects for some special cases and presents a general branch-and-cut algorithm. They also consider problems with bounded rings and hop-constrained problems.

7.2 Survivable Networks

In their earliest work on the subject, Grötschel and Monma (1990) introduced a general model mixing edge and node survivability requirements. They examined the dimension of the associated polytope and proved facet results for cut and node-cut inequalities.

They also described completely the polytope of the (1-)connected network problem, based on the work of Cornuéjols et al. (1985), by the introduction of partition inequalities. The first separation algorithm for these inequalities was proposed by Cunningham (1985) and requires $|\mathcal{E}|$ min-cut computations. Barahona (1992) reduced this computing time to $|\mathcal{N}|$ min-cut computations. $\mathcal{F}$-partition inequalities were first proposed by Mahjoub (1994).

Low-connectivity constrained network design problems have been introduced by Monma and Shallcross (1989) and Stoer (1992), who introduced most of the terminology and models presented in Section 4. For high-connectivity requirements, the reader is referred to Grötschel et al. (1995b) and Stoer (1992).

Directed multi-commodity flow formulations were studied by Magnanti and Raghavan (2005), and they introduced improved directed flow models that are much stronger than the cut formulations for some variants of the problem (in particular when there are many nodes with unitary requirements).

The complexity of the minimum cost two-connected network problem was established by Eswaran and Tarjan (1976). An in-depth survey of Steiner tree problems was made by Winter (1987). The algorithm for solving the Steiner tree problem in polynomial time when either the number of nodes of type 0 or the number of nodes of type 1 is restricted is due to Lawler (1976). At the time of writing, the most recent and efficient exact approaches for some variants of the Steiner tree problem are due to Fischetti et al. (2017).

7.3 Hop Constraints

Hop-constraints were considered by Balakrishnan and Altinkemer (1992) as a means of generating alternative base solutions for a network design problem. Later on, Gouveia (1998) presented a layered network flow reformulation whose linear programming bound proved to be quite tight. This reformulation has, then, been used in several network design problems with hop-constraints (Pirkul and Soni, 2003; Gouveia and Magnanti, 2003; Gouveia et al., 2003) and even some hop-constrained problems involving survivability considerations (more on this below). It is also interesting to point out that the apparently simple general network design problem with $L = 2$ already contains a complex structure (Dahl and Johannessen, 2004) who also conduct a computational study of this variation of the problem.

The K-edge-disjoint L-hop-constrained network design problem was first studied by Huygens et al. (2007) who only consider $L \leq 4$ and $K = 2$. The node-disjoint variant was studied by Gouveia et al. (2006) and later by Gouveia et al. (2008) who consider a more complicated version. A summary of these results is also presented in the survey by Kerivin and Mahjoub (2005). Itaí et al. (1982) and later Bley (2003) study the complexity of this problem for the node-disjoint and the edge-disjoint cases. More recently, Bley and Neto (2010) also studied the approximability of the problem for $L = 3$ and $L = 4$.

Some authors focused on the formulation of some variants of the problem in the space of design variables. For $K = 1$, Dahl (1999) has provided such a formulation and shown that it describes the corresponding convex hull for $L \leq 3$. Later on, Dahl et al. (2004) have shown that finding such a description for $L \geq 4$ would be much more complicated. For $K \geq 2$ the results are even worse. Huygens et al. (2004) have extended Dahl's result for $K = 2$ and $L \leq 3$. For $L \geq 4$, the only interesting result for the moment is the one given by Huygens and Mahjoub (2007) for $L = 4$ and $K = 2$ where a valid formulation has been given. However, in terms of valid inequalities and with the exception of the well known L-path cut inequalities, nothing is known for larger values of L. This may also explain why the only cutting plane method for the more general problem with several sources and several destinations by Huygens et al. (2007) only considers $L \leq 3$.

The extended formulation for the general case presented in Section 5 was introduced by Botton et al. (2013) who also proposed an efficient Benders decomposition method to solve it. They also applied this algorithm to a generalisation of the model with so-called reliable edges (Botton et al., 2015). For the single commodity case, Botton et al. (2018) proposed valid inequalities that completely describe the polyhedron of incidence vectors of feasible solutions for $K = 2$ and $L = 3$ (hence for arbitrary costs), extending results by Bendali et al. (2010) for non-negative costs.

A version of the problem where the hop constraint limit is different in the nominal graph and in case of failures was introduced by Gouveia and Leitner (2017) and an efficient branch-and-cut approach for this problem was proposed by Gouveia et al. (2018).

Another application of extended formulations for hop constrained problems in the context of distribution networks was proposed by De Boeck and Fortz (2018), where some preprocessing techniques to reduce the size of the resulting formulations are also discussed.

Recently, Gouveia et al. (2019) published a survey of layered graph approaches for hop constrained problem.

7.4 Rings

The Two-Connected Network with Bounded Rings problem was first studied by Fortz et al. (2000). More polyhedral results can be found in Fortz (2000); Fortz and Labbé (2002). Fortz and Labbé (2004) studied the particular case of unit edge lengths. The edge-connectivity version of the problem was studied by Fortz et al. (2006).

Other network design problems involve the creation of rings of bounded lengths. Goldschmidt et al. (2003) study the problem of connecting subsets of customers to a concentrator through a self-healing ring connected to a backbone ring of concentrators. This problem, called the *SONET Ring Assignment Problem* (SRAP) has the drawback that many instances are infeasible. Carroll et al. (2013) studied a generalization of the SRAP, called the *Ring Spur Assignment Problem* (RSAP). In this problem, the objective is to design a set of bounded disjoint local rings that are interconnected by a backbone ring, like in the SRAP. Since no SRAP solution exist in some real world instances, locations that have no possible physical route due to limitations of geography can be connected to local rings by spurs off the local rings.

8 Conclusions and Perspectives

Topological network design is a very important topic, and is often used as first step of network planning, considering a long-term horizon where demand is not known in advance. Decoupling the decisions on capacity and routing from the physical design is possible because technology in telecommunications make these two sets of decisions independent, as fiber optics cables have almost infinite capacity, but capacity restrictions arise from equipments that are subsequently added to the nodes.

However, when considering topological design, the simplest model that consists of building a connected network (and that can be solved in polynomial time), is not sufficient as survivability and quality of service aspects must be ensured: one must guarantee that redundancy exists in the network to cover different cases of failures, and that paths used for connecting demand nodes are not too long. Unfortunately, as soon as such constraints are added, all problems become NP-hard and more challenging. Therefore, there is also quite a lot of literature (not covered in this chapter) dealing with the development of heuristics for the problems presented here.

Concerning exact methods, problems with survivable requirements only (as those presented in Section 4) have been extensively studied from the polyhedral point of view. The problems are usually modeled in the "natural" space of design variables and several classes of facet-inducing valid inequalities have been proposed in the literature, leading to powerful branch-and-cut algorithms able to solve realistic size instances very efficiently.

As soon as constraints on the length of the paths are added, formulating the problems in the space of design variables only becomes much more challenging, and most approaches rely on extended formulations. But, despite the progress of modern solvers, these formulations quickly become intractable because of their large size, and decomposition methods like Lagrangian relaxation, Benders decomposition or branch-and-price are often used.

Interesting open problems involve a better understanding of Benders cuts arising from Benders decomposition methods for these problems in order to develop more efficient separation algorithms, as currently most approaches rely on solving linear programs for the separation. Given that large size instances remain intractable, there is also a large unexplored research direction in matheuristics, i.e., methods using structural knowledge from exact methods to derive effective heuristics and provide a certificate of quality for the solutions obtained. To the best of our knowledge, the state-of-the-art heuristics are meta-heuristics that do not use any knowledge gained from exact method development.

On a more managerial side, it would also be interesting to measure more precisely, on realistic instances, the loss in solution quality due to the separation of topological design and capacity allocation / routing decisions compared to an integrated strategy.

References

Balakrishnan A, Altinkemer K (1992) Using a hop-constrained model to generate alternative communication network design. ORSA Journal on Computing 4:192–205

Barahona F (1992) Separating from the dominant of the spanning tree polytope. Operations Research Letters 12:201–203

Bendali F, Diarrassouba I, Mahjoub A, Mailfert J (2010) The edge-disjoint 3-hop-constrained paths polytope. Discrete Optimization 7(4):222–233

Bley A (2003) On the complexity of vertex-disjoint length-restricted path problems. Computational Complexity 12(3-4):131–149

Bley A, Neto J (2010) Approximability of 3- and 4-hop bounded disjoint paths problems. In: Proceedings of IPCO 2010, Lausanne, Switzerland, Springer, no. 6080 in LNCS, pp 205–218

Botton Q, Fortz B, Gouveia L, Poss M (2013) Benders decomposition for the hop-constrained survivable network design problem. INFORMS Journal on Computing 25(1):13–26, http://joc.journal.informs.org/content/25/1/13.full.pdf+html

Botton Q, Fortz B, Gouveia L (2015) On the hop-constrained survivable network design problem with reliable edges. Computers & Operations Research 64(0):159 – 167

Botton Q, Fortz B, Gouveia L (2018) The 2 edge-disjoint 3-paths polyhedron. Annals of Telecommunications 73(1):29–36, DOI 10.1007/s12243-017-0615-2, URL https://doi.org/10.1007/s12243-017-0615-2

Carroll P, Fortz B, Labbé M, McGarraghy S (2013) A branch-and-cut algorithm for the ring spur assignment problem. Networks 61(2):89–103

Cornuéjols G, Fonlupt F, Naddef D (1985) The traveling salesman problem on a graph and some related integer polyhedra. Mathematical Programming 33:1–27

Cunningham W (1985) Optimal attack and reinforcement of a network. Journal of ACM 32:549–561

Dahl G (1999) Notes on polyhedra associated with hop-constrained walk polytopes. Operations Research Letters 25:97–100

Dahl G, Johannessen B (2004) The 2-path network problem. Networks 43:190–199

Dahl G, Foldnes N, Gouveia L (2004) A note on hop-constrained walk polytopes. Operations Research Letters 32(4):345–349

De Boeck J, Fortz B (2018) Extended formulation for hop constrained distribution network configuration problems. European Journal of Operational Research 265(2):488–502

Edmonds J (1971) Matroids and the greedy algorithm. Mathematical Programming 1(1):127–136

Eswaran K, Tarjan R (1976) Augmentation problems. SIAM Journal on Computing 5:653–665

Fischetti M, Leitner M, Ljubić I, Luipersbeck M, Monaci M, Resch M, Salvagnin D, Sinnl M (2017) Thinning out steiner trees: a node-based model for uniform edge costs. Mathematical Programming Computation 9(2):203–229

Fortz B (2000) Design of Survivable Networks with Bounded Rings, Network Theory and Applications, vol 2. Kluwer Academic Publishers

Fortz B, Labbé M (2002) Polyhedral results for two-connected networks with bounded rings. Mathematical Programming 93(1):27–54

Fortz B, Labbé M (2004) Two-connected networks with rings of bounded cardinality. Computational Optimization and Applications 27(2):123–148

Fortz B, Labbé M, Maffioli F (2000) Solving the two-connected network with bounded meshes problem. Operations Research 48(6):866–877

Fortz B, Mahjoub A, Mc Cormick S, Pesneau P (2006) Two-edge connected subgraphs with bounded rings: Polyhedral results and branch-and-cut. Mathematical Programming 105:85–111

Goldschmidt O, Laugier A, Olinick EV (2003) SONET/SDH ring assignment with capacity constraints. Discrete Applied Mathematics 129:99–128

Gouveia L (1998) Using variable redefinition for computing lower bounds for minimum spanning and steiner trees with hop constraints. INFORMS Journal on Computing 10:180–188

Gouveia L, Leitner M (2017) Design of survivable networks with vulnerability constraints. European Journal of Operational Research 258(1):89 – 103, DOI https://doi.org/10.1016/j.ejor.2016.09.003, URL http://www.sciencedirect.com/science/article/pii/S0377221716307172

Gouveia L, Magnanti TL (2003) Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. Networks 41(3):159–173

Gouveia L, Patrício P, Sousa A (2006) Telecommunications Planning: Innovations in Pricing, Network Design and Management, Springer, New York, chap Compact models for hop-constrained node survivable network design, pp 167–180

Gouveia L, Patrício P, Sousa A (2008) Hop-contrained node survivable network design: An application to MPLS over WDM. Networks and Spatial Economics 8(1):3–21

Gouveia L, Joyce-Moniz M, Leitner M (2018) Branch-and-cut methods for the network design problem with vulnerability constraints. Computers & Operations Research 91:190–208

Gouveia L, Leitner M, Ruthmair M (2019) Layered graph approaches for combinatorial optimization problems. Computers & Operations Research 102:22–38

Gouveia LE, Patrício P, de Sousa A, Valadas R (2003) MPLS over WDM Network Design with Packet Level QoS Constraints based on ILP Models. In: Proceedings of IEEE Infocom, pp 576–586

Grötschel M, Monma C (1990) Integer polyhedra arising from certain design problems with connectivity constraints. SIAM Journal on Discrete Mathematics 3:502–523

Grötschel M, Monma C, Stoer M (1995a) Design of Survivable Networks, Handbooks in OR/MS, vol 7 on Network models, North-Holland, chap 10, pp 617–672

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

Huygens D, Mahjoub AR (2007) Integer programming formulations for the two 4-hop-constrained paths problem. Networks 49(2):135–144

Huygens D, Mahjoub A, Pesneau P (2004) Two edge-disjoint hop-constrained paths and polyhedra. SIAM Journal on Discrete Mathematics 18(2):287–312

Huygens D, Labbé M, Mahjoub AR, Pesneau P (2007) The two-edge connected hop-constrained network design problem: Valid inequalities and branch-and-cut. Networks 49(1):116–133

Itaí A, Perl Y, Shiloach Y (1982) The complexity of finding maximum disjoint paths with length constraints. Networks 2:277–286

Kerivin H, Mahjoub AR (2005) Design of survivable networks: A survey. Networks 46(1):1–21

Kruskal J (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society 7:48–50

Lawler E (1976) Combinatorial Optimization : Networks and Matroids. Holt, Rinehart and Wilson, New-York

Magnanti TL, Raghavan S (2005) Strong formulations for network design problems with connectivity requirements. Networks 45(2):61–79

Magnanti TL, Wolsey LA (1995) Optimal trees. Handbooks in Operations Research and Management Science 7:503–615

Mahjoub A (1994) Two-edge connected spanning subgraphs and polyhedra. Mathematical Programming 64:199–208

Martin RK (1991) Using separation algorithms to generate mixed integer model reformulations. Operations Research Letters 10(3):119–128

Menger K (1927) Zur allgemeinen kurventheorie. Fundamenta Mathematicae 10:96–115

Monma C, Shallcross D (1989) Methods for designing communications networks with certain two-connected survivability constraints. Operations Research 37(4):531–541

Pirkul H, Soni S (2003) New formulations and solution procedures for the hop constrained network design problem. European Journal of Operational Research 148:126–140

Stoer M (1992) Design of Survivable Networks, Lecture Notes in Mathematics, vol 1531. Springer-Verlag

Winter P (1987) Steiner problems in networks : a survey. Networks 17:129–167