7 MULTICOMMODITY NETWORK FLOW MODELS AND ALGORITHMS IN TELECOMMUNICATIONS

Michel Minoux[1]

[1]Laboratoire Lip6
Université Paris 6
Paris, France
Michel.Minoux@lip6.fr

Abstract: We present an overview of mathematical optimization models and solution algorithms related to optimal network design and dimensioning in telecommunications. All the models discussed are expressed in terms of minimum cost multicommodity network flows with appropriate choice of link (or node) cost functions. Various special cases related to practical applications are examined, including the linear case, the linear with fixed cost case, and the case of general discontinuous nondecreasing cost functions. It is also shown how the generic models presented can accommodate a variety of constraints frequently encountered in applications, such as, among others, constraints on the choice of routes, robustness and survivability constraints. Finally we provide an overview of recent developments in solution algorithms, emphasizing those approaches capable of finding provably exact solutions, which are essentially based on general mixed integer programming techniques, relaxation, cutting-plane generation techniques and branch & cut.
Keywords: Multicommodity flows, network design, mixed integer programming, relaxation, cutting-planes, constraint generation, branch & cut.

7.1 INTRODUCTION

We are concerned here with the development of models and exact solution methods for optimal design and dimensioning of Telecommunication networks.

Expressed in general terms, the basic problem addressed here may be subsumed as follows. Given a list of traffic nodes (sources or sinks), (telephone traffic, data, multimedia traffic) together with known anticipated values for the volume of traffic to be exchanged between nodes, it is required to build a network connecting sources and sinks and capable of handling all the requested traffic flows simultaneously. Achieving this involves the joint resolution of two types of problems:

- Defining a best network topology: determine which pairs of nodes should be connected by transmission links. Thus, the network topology is nothing else but the graph specifying how the traffic sources and sinks are interconnected.

- Defining link capacities (or dimensioning): determine the type and capacity of the transmission equipment to be installed on each link in order to construct the network according to the chosen topology.

A solution to the above network design problem (corresponding to a choice of a topology and link capacities) will be called *feasible* if the network thus constructed is actually capable of (simultaneously) flowing all the traffic requirements between all source-sink pairs. Let us point out here that, in order to check feasibility of a network (topology and dimensioning), we have to be sure that, for each source-sink requirement, we can find one or several paths to flow the corresponding traffic: this is the so-called *traffic routing problem.* Any network design problem therefore assumes proper handling (whether direct or indirect) of routing issues.

Clearly, a network design problem of the type described above will usually have a huge number of feasible solutions, but, among those, some will be less costly than others, where, depending on the context of application considered, cost will refer to various economic criteria to be minimized (investment costs, leasing costs) or maximized (payments received from subscribers). In the present chapter, the optimization criterion considered will be to minimize total network cost defined as the sum of the transmission equipment costs to be installed on the various links of the associated graph.

It will soon become apparent in the sequel that the practical difficulty in solving optimal network design problems depends, to a large extent, on the structure of the objective function, and on how closely it approximates actual costs of transmission equipment. For instance, if an approximation of actual costs by linear cost functions appears to be acceptable, then the problem may be reducible either to continuous linear programming or even to shortest path computations (for which many efficient algorithms are available). By contrast, if a much more accurate representation of reality is required, then step-increasing (discontinuous) cost functions have to be considered, giving rise to large scale integer linear programs, much more difficult to solve in practice. Other possible sources of difficulty in getting optimal solutions (with proof of optimality) are related to the presence of additional constraints derived either from technical restrictions in the use of equipment, or from various network management rules. Examples of this will be found in Sections 7.3.2 and 7.4.4.

7.2 BASIC MODELS: GRAPHS, FLOWS AND MULTICOMMODITY NETWORK FLOWS

The basic mathematical models used to formulate and solve optimal network design problems make use of graph-theoretic and/or linear programming-based models (see e.g. Berge (1970), Gondran and Minoux (1995), and Ahuja et al. (1993)).

The set of all possible topologies for the network to be constructed will typically be described by means of a given (undirected) graph $G = [\mathcal{N}, \mathcal{U}]$ where:

- the node set $\mathcal{N}$ represents the various traffic sources/sinks to be interconnected;
- the edge set $\mathcal{U}$ corresponds to the various pairs of nodes which may be physically connected by installing transmission links (cables, optical carriers, etc., ...).

It should be clear that the graph G defined above represents all the possible network topologies for interconnecting the $N = |\mathcal{N}|$ given nodes, which correspond to the decision of installing, or not installing, transmission equipment on each link $u \in \mathcal{U}$ of G. Thus any a priori possible network topology will correspond to a *partial graph* $G' = [\mathcal{N}, \mathcal{U}']$ of G (defined as a graph on the same node set $\mathcal{N}$, and edge set $\mathcal{U}' \subseteq \mathcal{U}$).

Now, a natural basic model to represent the way traffic between some source node and some sink nodes in G flows through the network, while using transmission resources installed on the links, is the so-called single-commodity network flow model (see Ford and Fulkerson (1962), Gondran and Minoux (1995), and Ahuja et al. (1993)). If $M = |\mathcal{U}|$ denotes the number of edges of G, a single-commodity flow between $s \in \mathcal{N}$ (source) and $t \in \mathcal{N}$ (sink), is a M-vector $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_M)^T$ such that:

- $|\varphi_u|$ represents the amount of transmission resource used on edge $u = (i, j)$;
- having chosen an arbitrary orientation on link $u = (i, j)$ (for instance assuming $i < j$, $\varphi_u > 0$ if the flow runs from i to j, $\varphi_u < 0$ if the flow runs from j to i), Kirchhoff's conservation law holds at each node $k \neq s$, $k \neq t$, in other words:

$$\forall k \in \mathcal{N} \backslash \{s,t\}:$$

$$\sum_{u \in \omega^+(k)} \varphi_u - \sum_{u \in \omega^-(k)} \varphi_u = 0 \tag{7.1}$$

and

$$\sum_{u \in \omega^+(s)} \varphi_u = \sum_{u \in \omega^-(t)} \varphi_u = v(\varphi) \tag{7.2}$$

In the above equations (7.1) and (7.2), $\omega^+(k)$ (resp: $\omega^-(k)$) denotes the subset of edges of the form (i,k) oriented from i to k (resp: the subset of edges of the form (k, j) oriented from k to j). The real value $v(\varphi)$ involved in equation (7.2) is referred to as the *value* of the flow φ. We recall that equations (7.1)-(7.2) may be rewritten in matrix-form as:

$$A \cdot \varphi = v(\varphi) \cdot b \tag{7.3}$$

where A is the so-called node-arc incidence matrix of the directed graph $\widetilde{G}$ deduced from G by choosing an arbitrary orientation on the edges as suggested above. b is a N-vector with one component b_i for each node $i \in \mathcal{N}$, defined as:

$$\begin{cases} b_i = 0, & \forall i \in \mathcal{N} \backslash \{s,t\} \\ b_s = +1 \\ b_t = -1 \end{cases}$$

Now, if the single-commodity flow model recalled above is well suited to representing how traffic flows through the network from one given source node to one given sink node, it is no longer appropriate to represent how to allocate transmission resources to several competing flows between distinct source-sink pairs. For this, we need an extended version of the single-commodity flow model, referred to as the *multicommodity network flow model*, defined as follows.

K single-commodity flows (numbered $k = 1, \ldots, K$) are given, where, for each k, $s(k)$ and $t(k)$ denote the source node and sink node respectively and d_k the requested value of the single commodity flow to be sent between $s(k)$ and $t(k)$ through the network. Then the corresponding multicommodity flow is represented as a M-vector $\psi = (\psi_u)_{u \in \mathcal{U}}$ satisfying the following linear system:

$$(MCF) \qquad \begin{cases} \forall k = 1, \ldots, K: \\ \qquad A\varphi^k = d_k b^k \\ \forall u \in \mathcal{U}: \\ \qquad \psi_u = \sum_{k=1}^{K} |\varphi_u^k| \end{cases}$$

In the above, A denotes the node-arc incidence matrix of the directed graph $\widetilde{G}$ (deduced from G by giving each edge of G an arbitrary orientation); φ^k is the M-vector representing the k^{th} single-commodity flow from $s(k)$ to $t(k)$ and b^k is the N-vector having all components 0 except $b^k_{s(k)} = +1$ and $b^k_{t(k)=-1}$.

7.3 MODELING NETWORK DESIGN PROBLEMS AS MINIMUM COST MULTICOMMODITY FLOWS

We discuss now a very general model, using the multicommodity flow concept, to formulate network design and dimensioning problems as optimization models.

7.3.1 A minimum cost multicommodity flow model

Let $G = [\mathcal{N}, \mathcal{U}]$ denote the graph representing all possible network topologies and suppose we are given a set of multicommodity flow requirements as a list of K source-sink pairs $\{s(k), t(k)\}$ $(k = 1, \ldots, K)$ together with a requested flow value d_k between $s(k)$ and $t(k)$. For each edge $u = (i, j) \in \mathcal{U}$ on which one or several transmission equipment can be installed, we assume that we are given a cost function $\Phi_u : \mathbb{R}^+ \to \mathbb{R}^+$, where $\Phi_u(x_u)$ is the cost of installing the equipment necessary to accommodate a total traffic flow value x_u on link $u = (i, j)$. Note that, in most practical situations, the Φ_u functions will be nonnegative and nondecreasing on $[0, +\infty[$, or a subinterval in $[0, +\infty[$. Indeed, it is realistic to assume that each cost function Φ_u is defined only on an interval $[0, \beta_u]$ (β_u representing the maximum total transmission capacity which can possibly be installed on link $u = (i, j)$).

The problem of optimally designing a network capable of simultaneously handling all the requested flows at minimum cost can thus be formulated as the following min cost multicommodity flow problem:

$$\text{Minimize } \Phi(x) = \sum_{u \in \mathcal{U}} \Phi_u(x_u) \tag{7.4}$$

subject to:

$$(MCMCF) \qquad \forall u \in \mathcal{U}:\; x_u = \sum_{k=1}^{K} |\varphi_u^k| \tag{7.5}$$

$$\forall k = 1, 2, \ldots, K:\; A\varphi^k = d_k b^k \tag{7.6}$$

$$\forall u \in \mathcal{U}:\; 0 \leq x_u \leq \beta_u \tag{7.7}$$

In this model, equations (7.5) and (7.6) define x as the vector representing the multicommodity flow, x_u being the total traffic flow through link u. Equations (7.6) express the single-commodity flow requirement constraints for each source-sink pair composing the multicommodity flow. Constraints (7.7) are bound constraints imposing that, on each link u, the total traffic flow through the link should not exceed the prescribed limit β_u. Finally (7.4) defines the objective function, to be minimized, as the sum of individual link cost functions.

Such a model involves $(K+1)M$ variables: M x_u variables and K unknown vectors $\varphi^1, \varphi^2, \ldots, \varphi^K$, each of dimension M. The total number of constraints, apart from the nonnegativity conditions on x, is $KN + 2M$.

The $(MCMCF)$ model is very generic in the sense that a huge variety of optimal network design problems can be cast into this form, depending mainly on the particular choice for the link cost functions Φ_u. The simplest situation corresponds to the case where the Φ_u functions are linear (cf. §7.4.1) but this may lead to poor approximations of reality. Therefore we will also investigate more accurate models: the linear with fixed cost case (§7.4.2), concave piecewise linear cost functions (see §7.4.3) or step-increasing cost functions (cf. §7.4.4).

7.3.2 Flexibility of the (MCMCF) model

The $(MCMCF)$ model introduced in the previous section can easily handle various constraints arising from practical concerns in network engineering. We illustrate here this flexibility on two typical examples:

(a) *restriction of possible choice of routes*

It frequently occurs that, for a given source-sink pair, routing has to be restricted to a given partial subgraph of G. This corresponds to the situation where, either for technical, managerial, or economical reasons, the flow under consideration is not allowed to use some links or nodes. This restriction can easily be taken into account by just replacing, for each such k $(1 \leq k \leq K)$, equations (7.6) by: $A^k\varphi^k = d_k b^k$ where A^k denotes the node-arc incidence matrix of the partial subgraph allowed for flow k. Clearly the basic structure of the problem is essentially unchanged.

(b) *Handling node costs (switching costs)*

In many cases, the total network cost has to include, not only link costs (representing transmission equipment) but also node costs to represent the cost of switching equipment installed at the nodes. For any given node i, the cost of the necessary switching equipment essentially depends on y_i, the volume of traffic transiting through node i, which can easily be expressed in terms of x_u variables as:

$$y_i = \frac{1}{2}\left(\sum_{u\in\omega(i)} x_u - D_i\right)$$

where

$$D_i = \sum_{k/s(k)=i} d_k + \sum_{k/t(k)=i} d_k$$

is the total volume of traffic originating or terminating at i. Assuming that we are given, for each $i \in \mathcal{N}$, a cost function $\Delta_i(y_i)$ representing the cost of switching equipment as a function of y_i, the problem can be formulated as:

$$(MCMCF1) \qquad \begin{cases} \text{Minimize } \sum_{u\in\mathcal{U}} \Phi_u(x_u) + \sum_{i\in\mathcal{N}} \Delta_i(y_i) \\ \text{s.c:} \\ \forall u \in \mathcal{U}: \quad x_u = \sum_{k=1}^{K} |\varphi_u^k| \\ \forall i \in \mathcal{N}: \quad y_i = \frac{1}{2}\left(\sum_{u\in\omega(i)} x_u - D_i\right) \\ \forall k = 1,\dots,K: \quad A\varphi^k = d_k b^k \\ \forall u \in \mathcal{U}: \quad 0 \le x_u \le \beta_u. \end{cases}$$

Indeed it is possible to show that $(MCMCF1)$ can be reduced to a minimum cost multicommodity flow model with link cost functions only, i.e. to the basic model $(MCMCF)$ (see e.g. Minoux (2001)) at the expense of increasing the number of nodes and edges. In view of this, it is seen that the two models are *essentially equivalent* and this helps in understanding why most published work on network optimization is mainly restricted to the case of link cost functions only.

A third example illustrating the flexibility of the $(MCMCF)$ model will be addressed in §7.5 in connection with the issue of survivability constraints.

7.4 EXACT SOLUTION METHODS FOR (MCMCF)

We provide in this section an overview of known exact solution approaches to the min cost multicommodity flow problem for the most frequently encountered types of cost functions.

7.4.1 Case of linear cost functions: problem (P1)

This is the special case where each link cost function is of the form:

$$\Phi(x_u) = \gamma_u x_u \text{ with } \gamma_u \geq 0$$

(so that Φ_u is non decreasing). Then $(MCMCF)$ can be simply expressed as an ordinary (continuous) linear program as follows.

Each vector φ^k having components unconstrained in sign, it can be rewritten as: $\varphi^{k+} - \varphi^{k-}$, where φ^{k+} and φ^{k-} are two nonnegative vectors with the additional condition that $\forall u$, φ_u^{k+} and φ_u^{k-} cannot be simultaneously nonzero. Then we have $|\varphi_u^k| = \varphi_u^{k+} + \varphi_u^{k-}$, and therefore the resulting problem reads:

$$(P1) \quad \begin{cases} \text{Minimize } \sum\limits_{u \in \mathcal{U}} \gamma_u x_u \\ \text{s.t:} \\ \forall u \in \mathcal{U}: \quad x_u = \sum\limits_{k=1}^{K} \left(\varphi_u^{k+} + \varphi_u^{k-}\right) \\ \forall k = 1, \dots, K: \quad A\varphi^{k+} - A\varphi^{k-} = d_k b^k \\ \qquad\qquad\qquad\qquad \varphi^{k+} \geq 0,\ \varphi^{k-} \geq 0, \\ \forall u \in \mathcal{U}: \quad 0 \leq x_u \leq \beta_u. \end{cases}$$

This gives rise to large scale (continuous) linear programs with special structure. Such problems have been extensively studied since the early 1970s. The first solution approaches have been based on the fact that $(P1)$ features the appropriate structure suited to the Dantzig-Wolfe decomposition technique (or "price decomposition"), see for instance Minoux (1995) and the survey papers Assad (1978) and Kennington (1978). Other approaches, using the decomposable structure of the problem have also been proposed: application of Lagrangean duality in conjunction with the use of subgradient algorithms (Kennington and Shalaby, 1977), resource decomposition for the maximum multicommodity flow problem (Sakarovitch, 1966), partitioning methods (Grigoriadis and White, 1972). More recently, extensive developments related to interior point methods in linear programming have motivated investigation of new approaches to multicommodity flow problems such as $(P1)$ (see e.g. Schultz and Meyer (1991), Choi and Goldfarb (1990), and Chardaire and Lisser (2002)). Also application of parallel computing techniques have been studied (Pinar and Zenios, 1992). It is worth pointing out here that a well-solved special case of $(P1)$ arises when the upper capacity constraints $x_u \leq \beta_u$ are not active (for instance, if, on each link, the β_u value is greater than the total sum of all requirements). In this case it is easily seen that an optimal solution is readily obtained by sending each flow k on a minimum cost chain (w.r.t. the γ_u valuations) between $s(k)$ and $t(k)$ in G. Thus solving the problem reduces to shortest path computations. This nice property is exploited in many approaches to capacitated multicommodity flow problems, in particular those using Dantzig-Wolfe decomposition and Lagrangean relaxation.

7.4.2 The linear with fixed cost case: problem (P2)

In this version of the problem, the cost function on link u is given by:

$$\begin{aligned}\Phi_u(x_u) &= 0 \text{ if } x_u = 0\\ &= \delta_u + \gamma_u x_u \text{ if } x_u > 0.\end{aligned}$$

The corresponding $(MCMCF)$ problem, which we will denote $(P2)$, has been proposed in the literature as an appropriate model for taking into account infrastructure costs (such as digging trenches for cable installation, building towers for antennas, etc., ...). See, for instance, Yaged (1971) and Minoux (1989). Under this model, as soon as some traffic $x_u > 0$ has to be sent on a link $u = (i, j)$, then a fixed cost δ_u has to be paid, in addition to the linear cost $\gamma_u x_u$, proportional to the traffic flow value on link u. As will be seen in §7.4.3, this model is also relevant to approximate some more general cost functions, in particular nonlinear concave cost functions which are well suited to capture economy-of-scale phenomena in Telecommunications. Unfortunately, problem $(P2)$ belongs to the class of NP-hard problems. A possible proof of this uses the reduction of the Steiner tree problem in a graph, (already known to be NP-hard) to a single-commodity flow problem with fixed costs, a special case of $(P2)$ (see, e.g., Minoux (1989) for details).

For the case where the capacity constraints are not binding ($\beta_u = +\infty$, $\forall u \in \mathcal{U}$), an exact Branch & Bound algorithm has been proposed in Minoux (1976a); it makes use of lower bounds, obtained for each subset of edges $V \subset \mathcal{U}$ by carrying out the following two steps:

(a) Compute the minimum weight spanning tree with respect to the δ_u values on the partial graph $[\mathcal{N}, V]$. Let $\delta^*(V)$ denote the minimum value obtained.

(b) For each source-sink pair $k (1 \leq k \leq K)$ determine $\gamma_k^*(V)$, the length of a shortest chain between $s(k)$ and $t(k)$ with respect to the γ_u values, on the partial graph $[\mathcal{N}, V]$.

Then, for each $V \subset \mathcal{U}$, the resulting lower bound for the cost of an optimal network using only edges in V is

$$\mu(V) = \delta^*(V) + \sum_{k=1}^{K} d_k \gamma_k^*(V).$$

This bound can thus be obtained very efficiently, since it reduces to minimum spanning tree and shortest path computations. Exact solutions to problem $(P2)$ can be obtained with such an approach for medium-sized instances, with $|\mathcal{U}|$ typically not exceeding 40 to 50 edges (see Minoux (1976a)). For larger instances, heuristic solution methods providing only approximate solutions will have to be envisaged (see Minoux (1976a) and Minoux (1989)).

We note here that the special case of problem $(P2)$ where we have a set of single-commodity flow requirements (instead of multicommodity requirements) has been investigated by various authors, see Ortega and Wolsey (2003) and Rardin and Wolsey (1993).

7.4.3 The case of piecewise-linear concave cost functions: problem (P3)

If one is willing to consider models still closer to reality, an essential feature of telecommunications problems is the so-called *economy of scale* phenomenon: on each link of the network, the larger the installed capacity, the smaller the average cost (cost per unit capacity installed). Economic studies (see Ellis (1975), for instance) show that if we plot on a capacity/cost diagram the points representing the various transmission systems available to build transmission links (each one of these corresponding to an offered capacity c_u and associated cost γ_u) then these points lie, within a small tolerance, on a curve with equation $c_u = k_u(x_u)^\alpha$; here $k_u > 0$ is a coefficient depending on the link (and, in particular, on the link length) and α is a number close to 0.5, typically in the interval $[0.4, 0.6]$.

Minimum cost multicommodity flow problems with (concave) link cost functions of the form $\Phi_u(x_u) = k_u(x_u)^\alpha$ (where $0 < \alpha < 1$) are extremely difficult problems which, in view of the current state-of-the art, cannot be solved exactly (within acceptable computing times) except for very small sized networks (typically less than $N = 15$ nodes, $K = 20$ commodities and 10 possible paths per commodity). (Note that if approximate solutions are allowed, good heuristics for problems of this type have been described, e.g. those based on the so-called "accelerated greedy algorithm", see Minoux (1976b) and Minoux (1989)).

One possible way to bypass this difficulty is to approximate the cost function $k_u(x_u)^\alpha$ on each link by a piecewise linear concave function obtained, either via tangential approximation or barycentric approximation at a few points to be chosen. In either case, the approximation obtained can be viewed as the pointwise minimum of a finite number of linear with fixed cost functions. This version of the $(MCMCF)$ problem will be denoted $(P3)$. As an example, if we take the function $\Phi_u(x_u) = 100(x_u)^{0.5}$ to be approximated on the interval $[0, 900]$ then the tangential approximation at the points $x_u = 100$, $x_u = 400$, $x_u = 900$, is defined as:

$$\Phi_u^+(x_u) = \mathrm{Min}\{500 + 5x_u\ ;\ 1000 + \frac{5}{2}x_u\ ;\ 1500 + \frac{5}{3}x_u\}.$$

The barycentric approximation at the same points is defined as:

$$\Phi_u^-(x_u) = \mathrm{Min}\{10x_u\ ;\ \frac{2000}{3} + \frac{10}{3}x_u\ ;\ 1200 + 2x_u\}.$$

Note that $\Phi_u^+(x_u)$ and $\Phi_u^-(x_u)$ are upper and lower approximations to Φ_u respectively. Also, we note that in spite of the fairly reduced number of points defining Φ^+ and Φ^-, these constitute fairly good approximations (relative error ≤ 6 % on the range [100,900]). However, on the range [0.100] the relative error may be huge. For instance for: $x_u = 4$ $\Phi_u = 200$ whereas $\Phi_u^+ = 520$ and $\Phi_u^- = 40$ (the relative errors are 60 % and 80 % respectively). Thus, if more accurate approximations are required for small values of the capacity x_u, more points in the range [0.100] will have to be used to build tangential or barycentric approximations.

Now we show that a problem of type $(P3)$ can be readily reduced to a problem of type $(P2)$ on a network with the same node set but an increased number of links. For each link $u \in \mathcal{U}$, let q_u denote the number of linear with fixed cost functions $\varphi_u^t(x_u)$ defining Φ_u by the formula

$$\Phi_u(x_u) = \text{Min}_{t=1,\dots,q_u} \left\{ \varphi_u^t(x_u) \right\} \tag{7.8}$$

Let $\widehat{G}$ be the graph deduced from G by replacing each edge $u = (i,j) \in \mathcal{U}$ by q_u edges $u^1, u^2, \dots, u^{q_u}$ between i and j. Denote $\widehat{\mathcal{U}}$ the edge set of $\widehat{G}$. Now it is easily seen that problem $(P3)$ readily reduces to the solution of $(P2)$ on $\widehat{G}$ where each edge $u^t \in \widehat{\mathcal{U}}$ has an associated linear with fixed cost function $\varphi_u^t(x_{u^t})$.

7.4.4 The case of step-increasing cost functions: Problem (P4)

In the previous section, an appropriate model for taking into account the so-called "economy-of-scale" phenomenon has been discussed. Here, while keeping this capability, we introduce a new model taking into account the essentially discontinuous character of allowed capacity augmentations on the transmission links. The resulting discontinuity of the link cost functions is due to what is commonly referred to as "modularity" of transmission equipment. For instance if we decide to install an additional optical carrier between two nodes i and j, then the capacity of edge (i,j) features an important increase, and the total cost of the network is simultaneously increased by the cost corresponding to this optical carrier. To represent this, we are therefore led to a new minimum cost multicommodity flow problem in which each link cost function is a step-increasing (discontinuous) function of installed capacity. We will denote $(P4)$ the corresponding problem.

A typical step-increasing cost function on link $u \in \mathcal{U}$ can be defined by specifying the finite set $V = \left\{ v_u^0, v_u^1, \dots, v_u^{q(u)} \right\}$ of capacities corresponding to the discontinuity points and the associated costs: $\gamma_u^0 = \Phi_u\left(v_u^0\right), \gamma_u^1 = \Phi_u\left(v_u^1\right), \dots, \gamma_u^{q(u)} = \Phi_u\left(v_u^{q(u)}\right)$, with:

$$0 = v_u^0 < v_u^1 < \cdots < v_u^{q(u)} = \beta_u$$
$$0 = \gamma_u^0 < \gamma_u^1 < \cdots < \gamma_u^{q(u)}.$$

With the above notation, the link cost function Φ_u is defined by:

$$\begin{aligned} \Phi_u(x_u) &= 0 \text{ if } x_u = 0 \\ &= \gamma_u^i \ \ \forall x_u \in \left]v_u^{i-1}, v_u^i\right], \ \ \forall i = 1, \dots, q(u). \end{aligned}$$

Note that Φ_u is defined for x_u in the range $[0, \beta_u]$, the $(P4)$ model therefore includes, by construction, upper bound constraints on variables.

Various special cases of the $(P4)$ model have been studied in the literature, which correspond to simplified forms of the step cost functions Φ_u. In the case where the v_u^i values (resp: the γ_u^i values) are integer multiples of a given capacity value c (resp: of a given cost value γ) we have the so-called "single facility" or "single module" case, studied e.g. in Magnanti and Mirchandani (1993) and Barahona (1996).

Also the "two-facility" case where two types of modules of capacity c_1 and c_2 respectively and corresponding costs γ_1 and γ_2 has been investigated by various authors (see Magnanti et al. (1995), Bienstock and Günlük (1996), and Günlük (1999)). We will provide an overview of previous work on exact solution methods for those special cases in §7.4.4.1 below. However, in practice, there is often significantly more than two types of equipment, and moreover, additional constraints have to be taken into account. A typical example of constraints to be taken into account is the following: if, for a given value x_u of the total traffic flow on link u, one or several modules of capacity c have been used, then for all traffic flow values $x'_u > x_u$, only modules with capacity at least c can be used.

Such a "non-regression" rule is a consequence of the usually rapid increase of traffic requirements over time. As an example, applied to a case with 3 modules of capacity c_1, c_2, c_3 and corresponding costs $\gamma_1, \gamma_2, \gamma_3$ with:

$$\begin{gathered} c_1 < c_2 < c_3 \\ \frac{\gamma_1}{c_1} > \frac{\gamma_2}{c_2} > \frac{\gamma_3}{c_3} \\ \gamma_1 < \gamma_2 < 2\gamma_1 \\ 2\gamma_2 < \gamma_3 < 3\gamma_2 \end{gathered}$$

it leads to the following step-increasing cost function:

$$\begin{aligned} \Phi_u(x_u) &= 0 \ \text{ if } \ x_u = 0 \\ &= \gamma_1 \ \text{ if } \ 0 < x_u \leq c_1 \\ &= \gamma_2 \ \text{ if } \ c_1 < x_u \leq c_2 \\ &= 2\gamma_2 \ \text{ if } \ c_2 < x_u \leq 2c_2 \\ &= \gamma_3 \ \text{ if } \ 2c_2 < x_u \leq c_3 \\ &= 2\gamma_3 \ \text{ if } \ c_3 < x_u \leq 2c_3 \\ &\ \vdots \\ &= (k+1)\gamma_3 \ \text{ if } \ kc_3 < x_u \leq (k+1)c_3 \end{aligned}$$

Since such step-increasing cost functions without well-identified special structure actually arise in practice, investigating solution methods capable of handling arbitrary step-increasing cost functions appears to be necessary. Exact solution methods applicable to this "general case" will be the subject of §7.4.4.2 below.

7.4.4.1 Exact solution methods for special cases. In this subsection, we provide a brief overview of existing exact solution methods applicable to the main two previously mentioned special cases, namely the single-facility case (see Magnanti and Mirchandani (1993), Barahona (1996), and Bienstock et al. (1998)) and the two-facility case (see Magnanti et al. (1995), Bienstock and Günlük (1996), and Günlük (1999)). The idea, common to all these references, is to formulate the problem as a mixed integer linear programming problem, and to apply a Branch & Cut approach (see e.g. Ortega and Wolsey (2003) and Padberg and Rinaldi (1991)). The basic ingredient of Branch & Cut approaches is to generate *valid inequalities* and to exploit them in the framework of Branch & Bound to strengthen the linear relaxation of the problem (the continuous linear program obtained by dropping the integrality constraints on the integer-constrained variables). Strengthening leads to improving the lower bounds, thus to reducing the number of generated nodes in the Branch & Bound tree.

Let us recall here that a *valid inequality* is an inequality necessarily satisfied by any integer solution which, when appended to the linear programming formulation, cuts off the polyhedron defining the set of fractional solutions. Among the simplest and most frequently used valid inequalities for strengthening LP relaxations in the context of 1-facility or 2-facility versions of problem $(P4)$ we mention the so-called "cutset inequalities."

In the single-facility case, where there is only one module of capacity c and cost γ, such inequalities are obtained as follows.

On each edge $u \in \mathcal{U}$ of the network, let us denote $y_u \in \mathbb{N}$ the number of modules of capacity c installed on link u. For each cut separating the node set X into two complementary subsets $S \subset \mathcal{N}$, $\bar{S} = \mathcal{N} \backslash S$, let $\omega(S)$ denote the edge subset of the cut, i.e. the subset of edges having one endpoint in S and one endpoint in $\bar{S}$. Also, we denote $d(S,\bar{S})$ the sum of the d_k requirements taken on all k such that either $s(k) \in S$ and $t(k) \in \bar{S}$ or $s(k) \in \bar{S}$ and $t(k) \in S$. Then, for any such cut $(S,\bar{S})$, the inequality

$$\sum_{u \in \omega(S)} y_u \geq \left\lceil \frac{d(S,\bar{S})}{c} \right\rceil$$

is a valid inequality (where for any $\alpha \in \mathbb{R}$, $\lceil \alpha \rceil$ denotes the smallest integer greater than or equal to α).

Other types of valid inequalities which have been used in this context include the so-called 3-partition inequalities and arc-residual capacity constraints (see Magnanti et al. (1995) and Bienstock et al. (1998)), the rounded metric inequalities (see Bienstock et al. (1998), and in generalized form Minoux (2001)), flow-cut-set facets (Bienstock and Günlük, 1996), and mixed partition inequalities (Günlük, 1999). We refer to the survey in Minoux (2001) for a more detailed presentation of the above.

In the present state, according to the computational results reported by the various authors quoted, exact optimal solutions could only be obtained for problems of moderate size (at most 27 nodes and 51 links) and moreover the instances treated only concern fairly low density matrix requirements (the density of a flow requirement matrix is $2K/N(N-1)$, i.e. the ratio between the number K of source-sink requirements and the maximum possible number of source-sink pairs, i.e. $(N(N-1))/2)$. It is worth pointing out here that the practical difficulty in getting exact optimal solution to $(MCMCF)$ with step-increasing cost functions indeed appears to *increase very rapidly with this density parameter.*

7.4.4.2 Exact solution methods for the general case. The more general version of problem $(P4)$ corresponding to arbitrary step-increasing cost functions has been studied in Dahl and Stoer (1998), Stoer and Dahl (1994), Gabrel et al. (1999), and Gabrel and Minoux (1997).

The approach in Dahl and Stoer (1998) and Stoer and Dahl (1994) basically makes use of the same kind of techniques as those mentioned in §7.4.4.1, namely the identification of valid inequalities leading to strengthening the formulation of the problem expressed as a (mixed) integer linear programming problem. The computational results presented in Dahl and Stoer (1998) involve a few instances with sizes up to 45 nodes and 53 links, but, in all cases, the requirement matrices have low density (less than 7 %). Only one of the instances is solved to exact optimality, in all other cases the integrality gaps (difference between optimal integer solution values and optimal continuous solution values) are quite important (ranging from 22 % to 66 %).

The approach proposed in Gabrel et al. (1999) is somewhat different and may be viewed as a specialization of the so-called Benders method (Benders, 1962). It leads to iteratively solving integer linear programming subproblems constructed along the iterations by appending additional constraints, according to a procedure referred to as *constraint generation.* At each iteration the integer linear programming subproblem is solved (exactly) by applying a standard Mixed Integer Linear programming solver (e.g. CPLEX or XPRESS). More precisely the approach works as follows.

At any current iteration k, the subproblem to be solved, denoted (R_k), is a *relaxation* of $(MCMCF)$ where the constraints (7.5)–(7.6) are replaced by a set of constraints (called *metric inequalities*), each of which expresses a necessary condition for feasibility of the vector $x = (x_u)_{u\in\mathcal{U}}$. Each such *metric inequality* corresponds to a choice of virtual lengths $\lambda_u \geq 0$ $(u \in \mathcal{U})$ assigned to the links of the network, and reads:

$$\sum_{u\in\mathcal{U}} \lambda_u x_u \geq \theta(\lambda) = \sum_{k=1}^{K} d_k \ell_k^*(\lambda) \tag{7.9}$$

where, for each $k = 1,\dots,K$, $\ell_k^*(\lambda)$ denotes the length of the shortest $s(k)-t(k)$ chain in the network with respect to the "lengths" λ_u $(u \in \mathcal{U})$. Moreover, it can be shown that $x = (x_u)_{u\in\mathcal{U}}$ is a feasible solution to the multicommodity flow problem if all inequalities (7.9) deduced for all possible $\lambda \geq 0$ are satisfied.

If we denote J^k the subset of indices of the metric inequalities generated up to iteration k, and λ^j the length vector corresponding to the j^{th} inequality $(j \in J^k)$, then the relaxed subproblem (R_k) reads:

$$(R_k)\qquad \begin{cases} \text{Minimize } \sum\limits_{u\in\mathcal{U}} \Phi_u(x_u) \\ \text{subject to:} \\ \sum\limits_{u\in\mathcal{U}} \lambda_u^j x_u \geq \theta(\lambda^j) \quad \forall j \in J^k \\ x_u \in V_u, \quad \forall u \in \mathcal{U}. \end{cases}$$

As shown in Gabrel et al. (1999) and Minoux (2001), this problem can easily be reformulated as a pure 0-1 integer programming problem (with multidimensional knapsack-type structure). Let $\bar{x}$ be an (exact) optimal solution to (R_k). Then we know that it is an optimal solution to problem $(P4)$ if and only if all the metric inequalities (7.9) are satisfied. The algorithm therefore consists, at each iteration, in identifying metric inequalities violated by the current solution $\bar{x}$. If none can be (provably) found, then $\bar{x}$ is an exact optimal solution to $(P4)$ and the procedure terminates. On the contrary, if such violated inequalities can be identified, then these are appended to (R_k) to build up the new relaxed subproblem (R_{k+1}) at iteration $k+1$. This constraint-generation idea can be implemented in various ways. For instance, one can generate only one inequality at each step, the most violated one according to some chosen criterion ("single constraint generation"); or systematically generate several violated inequalities ("multiple constraint generation"). One can also consider the use of particular subclasses of metric inequalities, e.g. the so-called *bipartition inequalities* (these are particular metric inequalities obtained by setting $\lambda_u = 1$ for all edges of a cut and $\lambda_u = 0$ for the other edges).

In Gabrel et al. (1999), a multiple constraint generation procedure is described, based on the combined use of bipartition inequalities and general metric inequalities, and where violated inequalities are identified by maximizing the ratio

$$\rho = \frac{\theta(\lambda)}{\sum_{u\in\mathcal{U}} \lambda_u \bar{x}_u}$$

($\rho > 1$ corresponds to a metric inequality violated by the current solution $\bar{x}$).

Computational results are reported there on a series of 50 instances with up to 20 nodes, 37 links with cost functions featuring an average number of 6 steps, and *100 % dense requirement matrices.* A sample of these results (taken from Gabrel et al. (1999)) appears in Table 7.1. N is the number of nodes; M is the number of edges; Iter is the total number of main iterations necessary to reach exact optimality; NC is the total number of constraints (= metricinequalities) generated in the process, and GMI is the number of general metric inequalities ($NC - GMI$ is thus the number of bipartition inequalities generated); T is the total computation time in seconds (using CPLEX 4.0 in MIP mode for the solution of the relaxed subproblems) including $T(CG)$, that part of total time devoted to generating constraints.

From these results it is seen that the number of main iterations is always quite limited (it does not exceed 12 on the instances treated) and increases only moderately fast with problem size. Also it is observed that general metric inequalities are very rarely needed to reach optimality, bipartition inequalities almost always suffice. Finally, the computation times needed for generating violated inequalities ($T(CG)$) represent only a very small fraction of total computation time.

Table 7.1 also illustrates the practical difficulty in obtaining exact (guaranteed) optimal solutions to problem $(P4)$ with general step increasing cost functions and fully dense requirement matrices. Of course, network engineering applications often call for the solution of significantly larger instances, and in this case, approximate solution methods (heuristics) have to be envisaged. For problem $(P4)$ many heuristic solution approaches are possible, including the use of standard meta-heuristics (simulated annealing, tabu search, genetic algorithms, see e.g. Reeves (1993)) in connection with the use of specialized local search operators such as link rerouting or flow rerouting. Recently it has been shown in Gabrel et al. (2003a) that the exact constraint-generation procedure of Gabrel et al. (1999) can also be used as the basis of an approximate solution procedure by solving the relaxed subproblems (R_k) *heuristically* rather than exactly. The computational results reported in Gabrel et al. (2003a) suggest that solutions thus obtained are much closer to exact optimality than those produced by several types of link-rerouting and flow-rerouting heuristics.

Table 7.1 Exact optimal solutions to problem $(P4)$ on a series of 10 to 20 node networks with cost functions having average 6-steps per link (taken from (Gabrel et al., 1999)).

N	M	Iter	GMI	NC	T (total)	$T(CG)$
10	16	7	0	25	26	1.1
10	17	9	2	53	171	2.4
10	18	6	0	30	48	1.1
12	21	11	0	68	1 471	4
12	20	7	0	37	150	2.4
12	20	12	0	68	361	4.2
15	26	9	0	79	1 621	8
15	27	12	0	132	10 911	11.2
15	26	8	0	69	984	7.2
20	36	12	0	183	18 795	35
20	35	9	0	103	2 139	25
20	35	12	0	147	12 476	34

7.5 HANDLING SURVIVABILITY CONSTRAINTS: THE (MCMCFS) MODEL

This section is devoted to an important extension to the $(MCMCF)$ model in which *survivability constraints* have to be taken into account. This will be denoted by $(MCMCFS)$. Such constraints frequently appear in practical network engineering applications, in view of the high capacities offered by modern transmission systems (e.g. optical carriers). Survivability constraints express the fact that the network to be set up, not only has to meet the flow requirements under regular operation, but should keep on meeting all the flow requirements (or a fixed minimum percentage α of these) in case of failure of one or several elements (transmission component, switching component). Since simultaneous failure of several transmission or switching components usually occurs with extremely low probability, most practical applications only require to make the network robust against single link failure and/or single node failure.

The $(MCMCFS)$ problem has been actively investigated in the special case of linear cost functions, which leads to solving large scale (continuous) linear programs, to which several types of decomposition techniques can be applied. The reader is referred to Dahl and Stoer (1998), Goemans and Bertsimas (1993), Minoux (1981), Minoux and Serreault (1984), Stoer and Dahl (1994), and to the survey papers Grötschel et al. (1995) and Minoux (1989). Also worth mentioning are published work where survivability constraints are indirectly and approximately taken into account by means of various types of connectivity constraints (e.g. k-connectedness constraints); the papers Grötschel and Monma (1990), Grötschel et al. (1992), and Monma and Shallcross (1989) are relevant references along this line.

We will discuss more thoroughly here recent work on handling survivability constraints in the case where the cost functions on the links are arbitrary step-increasing functions of total flow through the link (cf §7.4.4.2). In order to make the presentation of the model simpler, we will restrict to the case of *link failures* (though fairly straightforward, the extension to node failures would require more intricate notation). Furthermore, for the sake of simplicity, we will assume that when a given link $u = (i, j)$ fails, all the transmission systems corresponding to this link are in failure state. Of course, the $(MCMCFS)$ model presented below would easily extend if a more detailed representation of failure were required, for instance if, on any link (i, j), at most one of the installed transmission systems (modules) can break down at a time (for this, it would be enough to consider multiple copies of edge (i, j), one for each transmission system which can be installed between i and j, each being characterized by its capacity and cost).

To express robustness of the network w.r.t. any link failure, let us introduce, for each $v \in \mathcal{U}$, the *projection operator* π_v defined as follows. For any $x \in \mathbb{R}^M$ (such that $\forall u \in \mathcal{U}, x_u$ represents the total flow through link u): $\pi_v(x) = x'$ where, $\forall u \neq v : x'_u = x_u$ and $x'_v = 0$. With this notation it is observed that $x \in \mathbb{R}^M$ satisfies the survivability constraint w.r.t. the failure of link $v \in \mathcal{U}$ if and only if $\pi_v(x) \in X$ where X denotes the set of solutions to (7.5)-(7.6). The $(MCMCF)$ model with survivability constraints can thus be stated as:

$$(MCMCFS) \quad \begin{cases} \text{Minimize: } \sum_{u \in \mathcal{U}} \Phi_u(x_u) \\ \text{s.t:} \\ \forall v \in \mathcal{U}: \quad \pi_v(x) \in X \\ \forall u \in \mathcal{U}: \quad 0 \leq x_u \leq \beta_u. \end{cases}$$

In the case where the cost functions Φ_u are step-increasing and where, on each link $u \in \mathcal{U}$, V_u denotes the set of discontinuity points of the Φ_u function, the constraint $0 \leq x_u \leq \beta_u$ has to be replaced by $x_u \in V_u$. The resulting optimization problem is a variant of problem $(P4)$ which we denote $(P4S)$.

Exact solution methods for $(P4S)$ have been investigated mainly by Dahl and Stoer (1998), Stoer and Dahl (1994), and Gabrel et al. (2003b), the proposed approaches may be viewed as generalizations of those described in §7.4.4.2.

In Dahl and Stoer (1998), computational results are reported on 23 instances involving networks up to 118 nodes and 134 edges, the link cost functions featuring up to 6 steps. For all the examples treated, the requirement matrices have *very low density* (less than 7 %). Exact optimal solutions are obtained for 8 instances over 23; for the other instances the approximate solutions obtained are sometimes close to exact optimality but the difference may be significant (up to 66 % off optimality), which suggests a huge variability in the practical difficulty of the problem instances.

Gabrel et al. (2003b) describes an extension to problem $(P4S)$ of the constraint-generation approach of Gabrel et al. (1999) (see §7.4.4.2 above). Several alternate strategies for generating constraints (metric inequalities) are described and tested on

30 instances involving 15 and 20 node networks with cost functions having average 6 steps per link and 100% dense requirement matrices. A sample of the results obtained is shown in Table 7.2 (MIP solver used: CPLEX 6.0 on a Sun UltraSparc 30 computer).

Table 7.2 Exact solutions to problem $(P4S)$ on a series of 15 node and 20 node networks with cost functions having average 6 steps on each link. (taken from Gabrel et al. (2003b)).

Instance	N, M	Iter	NC	Time (sec.)
15.1	15, 26	2	313	379
15.2	15, 26	6	461	5 422
15.3	15, 27	3	322	464
15.4	15, 26	3	306	108
15.5	15, 27	2	247	352
15.6	15, 26	4	305	237
20.1	20, 36	6	560	11 162
20.2	20, 35	4	597	3 457
20.3	20, 35	4	598	9 778
20.4	20, 36	8	609	41 752
20.5	20, 37	3	527	9 513
20.6	20, 35	3	550	209

The main conclusion which may be drawn from these experiments is that the exact solution of problem $(P4S)$ leads to computation times comparable to those obtained for problem $(P4)$ (without survivability constraints). The average computing times necessary to reach exact (guaranteed) optimality are significant, but it should be noticed that in all the instances in Table 7.2, *requirement matrices are 100% dense.* The comparison of the results with those presented in Dahl and Stoer (1998) seems to confirm that the practical difficulty of network optimization problems such as $(P4)$ and $(P4S)$ increases very rapidly with the density of requirement matrices. (Clearly, in view of practical application of the optimization models, it would not be realistic to restrict to low density requirement matrices).

For applications requiring solutions to $(P4S)$ on larger networks, approximate solution techniques will have to be envisaged. Among the many possible solution approaches, the one described in Gabrel et al. (2003a), which also readily applies to problem $(P4S)$, appears to be worth considering.

Bibliography

R. Ahuja, T. Magnanti, and J. Orlin. *Networks Flows, Algorithms and Applications*. Prentice Hall, 1993.

A.A. Assad. Multicommodity network flows – a survey. *Networks*, 8:37–91, 1978.

F. Barahona. Network design using cut inequalities. *SIAM Journal on Optimization*, 6(3):823–837, 1996.

J.F. Benders. Partitioning procedures for solving mixed-variable programming problems. *Num. Math.*, 4:238–252, 1962.

C. Berge. *Graphes et Hypergraphes*. Dunod, Paris, 1970.

D. Bienstock, S. Chopra, O. Günlük, and C.Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81:177–199, 1998.

D. Bienstock and O. Günlük. Capacitated network design-polyhedral structure and computation. *INFORMS Journal on Computing*, pages 243–259, 1996.

P. Chardaire and A. Lisser. Simplex and interior-point specialized algorithms for solving non-oriented multicommodity flow problems. *Operations Research*, 50(2):260–276, 2002.

I.C. Choi and D. Goldfarb. Solving multicommodity network flow problems by an interior-point method. *SIAM Proc. in Appl. Math.*, 46:58–69, 1990.

G. Dahl and M. Stoer. A cutting-plane algorithm for multicommodity survivable network design problems. *Informs J. on Computing*, 10(1), 1998.

L.W. Ellis. La loi des volumes économiques appliquée aux télécommunications. *Rev. Telecom 1*, 50:4–20, 1975.

L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, 1962.

V. Gabrel, A. Knippel, and M. Minoux. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters*, 25:15–23, 1999.

V. Gabrel, A. Knippel, and M. Minoux. A comparison of heuristics for the discrete cost multicommodity network optimization problem. *Journal of Heuristics*, 9:429–445, 2003a.

V. Gabrel, A. Knippel, and M. Minoux. Exact solution of survivable network problems with step-increasing cost functions. Unpublished research report, University Paris 6, 2003b.

V. Gabrel and M. Minoux. LP relaxations better than convexification for multicommodity network optimization problems with step-increasing cost functions. In *Acta Mathematica Vietnamica 22*, pages 128–145, 1997.

M. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60(2):145–166, 1993.

M. Gondran and M. Minoux. *Graphs and Algorithms*. John Wiley & Sons, New York, 1995.

M.D. Grigoriadis and W.W. White. A partitioning algorithm for the multicommodity network flow problem. *Mathematical Programming*, 3(2):157–177, 1972.

M. Grötschel and C. Monma. Integer polyhedron associated with certain network design problems with connectivity constraints. *SIAM J. Discrete Math.*, 3:502–523, 1990.

M. Grötschel, C. 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.

M. Grötschel, C. Monma, and M. Stoer. Design of survivable networks. *Handbook in OR and MS*, 27:617–672, 1995.

O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Math. Prog. A*, 86:17–39, 1999.

J. Kennington. Multicommodity flows: a state-of-the art survey of linear models and solution techniques. *Opns. Res.*, 26:209–236, 1978.

J. Kennington and M. Shalaby. An effective subgradient procedure for minimal cost multicommodity flow problems. *Management Science*, 23(9):994–1004, 1977.

T.L. Magnanti and P. Mirchandani. Shortest paths, single origin-destination network design and associated polyhedra. *Networks*, 23:103–121, 1993.

T.L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two-facility network loading problem. *Operations Research*, 43(1):142–157, 1995.

M. Minoux. Multiflots de coûts minimal avec fonctions de coût concaves. *Annales des Télécommunications*, 31:77–92, 1976a.

M. Minoux. Optimisation et planification des réseaux de télécommunications. In G. Goos, J. Hartmanis, and J. Cea, editors, *Optimization Techniques*, pages 419–430. Springer-Verlag, 1976b. Lectures Notes in Computer Science 40.

M. Minoux. Optimum synthesis of a network with nonsimultaneous multicommodity flow requirements. *Studies on Graphs and Discrete Programming*, 11:269–277, 1981. P. Hansen, Editor. Annals of Discrete Mathematics.

M. Minoux. Network synthesis and optimum network design problems: models, solution methods and applications. *Networks*, 19:313–360, 1989.

M. Minoux. *Mathematical Programming. Theory and Algorithms*. John Wiley & Sons, 1995.

M. Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106:19–46, 2001.

M. Minoux and J.Y. Serreault. Subgradient optimization and large scale programming: an application to network synthesis with security constraints. *RAIRO*, 15:185–203, 1984.

C. Monma and D. Shallcross. Methods for designing communication networks with certain two–connected survivability constraints. *Operations Research*, 37:531–541, 1989.

F. Ortega and L.A. Wolsey. A branch-and-cut algorithm for the single-commodity uncapacitated fixed-charge network flow problem. *Networks*, 41(3):143–158, 2003.

M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *S.I.A.M. Review*, 33:60–100, 1991.

M.C. Pinar and S.A. Zenios. Parallel decomposition of multicommodity network flows using a linear quadratic penalty algorithm. *ORSA Journal on Computing*, 4(3):235–249, 1992.

R.L. Rardin and L.A. Wolsey. Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *European J. Operational Res.*, 71:95–109, 1993.

C.R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications, 1993.

M. Sakarovitch. The multicommodity maximum flow problem. ORC Report 65-25, University of California, Berkeley, 1966.

G.L. Schultz and R.R. Meyer. An interior-point method for block-angular optimization. *SIAM J. Optimization*, 1(4):583–602, 1991.

M. Stoer and G. Dahl. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68:149–167, 1994.

B. Yaged. Minimum cost routing for static network models. *Networks*, 1(2):139–172, 1971.