Annals of Operations Research 106, 19–46, 2001

Discrete Cost Multicommodity Network Optimization Problems and Exact Solution Methods

MICHEL MINOUX Michel.Minoux@lip6.fr
University Paris 6, 4, place Jussieu, 75005 Paris, France

Abstract. We first introduce a generic model for discrete cost multicommodity network optimization, together with several variants relevant to telecommunication networks such as: the case where discrete node cost functions (accounting for switching equipment) have to be included in the objective; the case where survivability constraints with respect to single-link and/or single-node failure have to be taken into account. An overview of existing exact solution methods is presented, both for special cases (such as the so-called single-facility and two-facility network loading problems) and for the general case where arbitrary step-increasing link cost-functions are considered. The basic discrete cost multicommodity flow problem (DCMCF) as well as its variant with survivability constraints (DCSMCF) are addressed. Several possible directions for improvement or future investigations are mentioned in the concluding section.

1. Introduction

Network optimization problems which, among other issues, include topological optimization and optimum network loading (or dimensioning), arise in many important areas of applications, such as telecommunications, transportation systems, distribution and logistics, see for instance [13,20,21,29,36,38,43]. Among existing surveys covering a wide variety of network optimization models, we refer to [3,19,36,41,43]. In these models, the cost functions to be minimized are in most cases linear, linear with fixed costs, or nonlinear but continuous. In the latter case, apart from a few very specially structured problems (see, e.g., [18]) exact solution algorithms can only be expected when convexity is present.

We are concerned here with a very general model for network optimization expressed in terms of minimum cost multicommodity flows, with discrete (nonconvex, discontinuous, step increasing) cost functions on the links (also, as will be seen in the paper, discrete costs of switching equipment may easily be taken into account within the same model). Investigations of (simplified versions of) this class of network optimization problems have started only a few years ago with the work by [32] and [4] on the so-called "single-facility network loading problem"; and by [34] and [6] on the so-called "two-facility network loading problem". Contrary to the above-mentioned contributions, the general model discussed in the present paper does not assume any special structure of the link cost functions, besides the fact that they are discontinuous and step-increasing, therefore appropriate to capture two major aspects of actual cost structure in telecommunication network engineering:

- the discreteness of capacity and cost increase on the links (transmission equipment) as well as at the nodes (switching equipment),
- the so-called "economy of scale" phenomenon, which is basically that the cost per unit capacity installed is decreasing with the maximum capacity provided by an equipment.

The purpose of the present paper is twofold. First we illustrate the flexibility of the generic model proposed by showing various possible applications (including the single-facility and two-facility versions of the problem). This will be the subject of section 2. The second objective is to provide an overview of the currently available exact solution algorithms which have been proposed so far, with a discussion of the computational results obtained. This will be the subject of section 3. Finally, in section 4, several possible directions for improvement or future investigations are mentioned.

2. Modeling discrete cost network optimization problems

2.1. Single-commodity and multicommodity flow models

We assume here familiarity with the basic concepts and tools from graph theory, network flow theory and linear programming, for which numerous textbooks are available.
A network optimization problem will be specified as follows.
First, we are given a graph $G = [\mathcal{N}, \mathcal{U}]$ where:

- $\mathcal{N}$ is the set of nodes representing the various sources or destinations of traffic (corresponding, for instance, to customers, switching centers, etc.) which have to be interconnected through communication links; we denote $N = |\mathcal{N}|$;
- $\mathcal{U}$ corresponds to the set of possible physical links on which transmission equipment (cables, optical carriers, etc.) may be installed in order to allow traffic to be flowed through the network. Each possible transmission equipment which is eligible for capacity augmentation on a given link is characterized by its capacity (expressed in a given unit, e.g., Mb/sec) and its cost. (At this stage we do not consider switching equipment at the nodes; we will show later how they may be handled.) We denote $M = |\mathcal{U}|$.

Note that in this paper we will refer to G as an *undirected graph* ($\mathcal{U}$ will therefore be a set of *edges*) which corresponds to considering, on each link (i, j) the *global flow* on the link without distinguishing that part of traffic flowing from i to j from that part of traffic flowing from j to i. Obviously, all the models discussed here would readily extend in case of an application requiring a distinct representation for the traffic flow from i to j and for the traffic flow from j to i. In such a case, G would have to be assumed directed ($\mathcal{U}$ being a set of *arcs*).

In addition to the graph G representing all possibilities for capacity installation or capacity augmentation, a set of traffic requirements between some (or all) pairs of nodes has to be specified. Thus a list of requirements between pairs of nodes will be assumed to be given in the following form: K denoting the number of pairs, for each $k = 1, \dots, K$, d_k will denote the required amount of flow to be sent through the network between the nodes $s(k)$ ("source") and $t(k)$ ("destination").

Remark. Again, we consider here an undirected model, where, for any given pair of nodes in the list, say k, d_k represents the total amount of communication needs from $s(k)$ to $t(k)$ and from $t(k)$ to $s(k)$. The model discussed below would readily extend to a context of application where it would be necessary to distinguish the traffic flows from $s(k)$ to $t(k)$ from the traffic flows from $t(k)$ to $s(k)$.

A natural basic model of communication needs between nodes of a telecommunication network is in terms of network flow theory [1,12]. In particular, the so-called single-commodity flow (SCF) model allows one to represent the use of resources (transportation or communication facilities) when there is only one flow requirement between two nodes, a source s and a sink t. The flow through the network is then represented as a vector $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_M)$ where:

- M is the number of links,
- $|\varphi_u|$ $(1 \leqslant u \leqslant M)$ is the amount of resource used on link (or edge) u,
- having chosen an (arbitrary) orientation on link $u = (i, j)$ (e.g., $\varphi_u > 0$ if the flow runs from i to j and $\varphi_u < 0$ if it runs from j to i), then Kirchhoff's conservation law holds at each node i $(i \neq s, i \neq t)$, i.e.,

$$\forall i \in \mathcal{N},\ i \neq s,\ i \neq t\colon \quad \sum_{u \in \omega^+(i)} \varphi_u - \sum_{u \in \omega^-(i)} \varphi_u = 0, \tag{1}$$

$$v(\varphi) = \sum_{u \in \omega^+(s)} \varphi_u = \sum_{u \in \omega^-(t)} \varphi_u, \tag{2}$$

$\omega^+(i)$ and $\omega^-(i)$ denote, respectively, the subsets of arcs originating at i and terminating at i. $v(\varphi)$ is nothing but the total amount of flow leaving node s (source) or entering note t (sink) and is called *the value* of flow φ. (In our model below, $v(\varphi)$ will be constrained to be equal to the prescribed flow requirement between s and t in the network.)

Let $\widetilde{G}$ denote the directed graph deduced from G by choosing, on each edge $u = (i, j)$ the orientation from i to j if $i < j$ and from j to i if $i > j$ (here an arbitrary ordering of the nodes is used). The basic relations (1) and (2) defining a flow with source s and sink t can be rewritten in matrix notation as:

$$\text{(SCF)} \qquad A \cdot \varphi = v(\varphi) \cdot b \tag{3}$$

where A is the $N \times M$ node-arc incidence matrix of graph $\widetilde{G}$ and b is an N-vector with all components 0, except $b_s = +1$ and $b_t = -1$.

Now, in the network problems we want to solve, there are several distinct traffic requirements to be flowed simultaneously on the network, and these have to share the common capacity resources on the links. To represent such a situation, we need the concept of *a multicommodity flow*.

In algebraic form, a multicommodity flow can be viewed as a M-vector $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_M)$ (each component corresponding to an edge or an arc of the network) defined by the following linear system:

$$\text{(MCF)} \qquad \begin{cases} \forall u \in \mathcal{U}: & \Psi_u = \sum_{k=1}^{K} |\varphi_u^k|, \\ \forall k = 1, 2, \dots, K: & A \cdot \varphi^k = d_k \cdot b^k, \end{cases}$$

where φ^k denotes the M-vector representing the kth single-commodity flow (between $s(k)$ and $t(k)$) and, $\forall k$, b^k is the N-vector with all components 0, except $b^k_{s(k)} = +1$ and $b^k_{t(k)} = -1$ and d_k the amount of flow to be sent from $s(k)$ to $t(k)$.

On each edge or arc of G, Ψ_u thus denotes the sum of the amounts of resources (transmission or communication facilities) used by the various constitutive single-commodity flows.

Later on, in the paper, we will denote by $X \subset \mathbb{R}_+^M$ the polyhedron representing the set of all feasible multicommodity flows:

$$X = \{\mathbf{x} \in \mathbb{R}^M \mid \mathbf{x} \geqslant \Psi \text{ for some } \Psi \text{ satisfying (MCF)}\}.$$

Thus $\mathbf{x} = (\mathbf{x}_u)_{u \in \mathcal{U}}$ belongs to X if and only if a feasible multicommodity flow exists when, on each edge $u \in \mathcal{U}$, the total capacity installed is $\mathbf{x}_u$ (of course $\mathbf{x}_u \geqslant 0$ on each edge u).

Several linear representations of X (as a system of linear equality and inequality constraints involving the $\mathbf{x}$ variables and possibly other variables) are known, including the so-called node-arc formulation and arc-chain formulation (for an overview, see, e.g., Assad [2], Kennington [30], Minoux [43]).

Later in the paper we will use the following representation of X involving the $\mathbf{x}$ variables only. For any $\lambda = (\lambda_1, \dots, \lambda_M) \in \mathbb{R}_+^M$, let $\theta(\lambda)$ denote the quantity

$$\theta(\lambda) = \sum_{k=1}^{K} d_k \times l_k^*(\lambda),$$

where $l_k^*(\lambda)$ is the length of the shortest chain joining $s(k)$ and $t(k)$ in G, when each edge $u \in \mathcal{U}$ is given length $\lambda_u \geqslant 0$ (note that $\theta(\lambda)$ may be interpreted as the value of the minimum cost multicommodity flow solution when, on each edge u, the cost function $\Phi_u(\mathbf{x}_u)$ is linear of the form $\lambda_u \mathbf{x}_u$).

Then $\mathbf{x} = (\mathbf{x}_u)_{u \in \mathcal{U}}$ belongs to X if and only if, for all $\lambda \in \mathbb{R}_+^M$, we have

$$\sum_{u \in \mathcal{U}} \lambda_u \mathbf{x}_u \geqslant \theta(\lambda) \tag{4}$$

(see, e.g., [48] or [24, chapter 6]). The above inequalities (4) are often referred to as "metric inequalities".

2.2. A general min-cost multicommodity flow model

Suppose we are given a set of nodes $\mathcal{N} = \{1, 2, \dots, N\}$ and a set of edges $\mathcal{U}$ representing all the existing and *possible links* between the nodes in $\mathcal{N}$ (one of the expected results of the optimization model is to specify on which links capacity should be installed in order to minimize the total cost criterion). Moreover, we are given multicommodity flow requirements, i.e., a list of source-sink pairs $s(k)t(k)$ $(k = 1, \dots, K)$ with corresponding prescribed flow values d_k $(k = 1, \dots, K)$, each d_k representing, for instance, the amount of commodity which should be sent between $s(k)$ and $t(k)$ in the network to be built. Associated with each possible link $u \in \mathcal{U}$, we shall assume that we are given a *cost function* Φ_u giving, for any value $\mathbf{x}_u$ of the total (multicommodity) flow on u, the cost $\Phi_u(\mathbf{x}_u)$. We assume here that the cost function Φ_u on each link is a discrete discontinuous step-increasing function of the total flow on link u defined on a given interval $[0, \beta_u]$ as follows:

Let $V_u = \{v_u^0, v_u^1, \dots, v_u^{q(u)}\}$ be a finite set of values representing the discontinuity points of the Φ_u function and denote

$$\begin{aligned} \gamma_u^0 &= \Phi_u(v_u^0), \\ \gamma_u^1 &= \Phi_u(v_u^1), \\ \gamma_u^2 &= \Phi_u(v_u^2), \\ &\vdots \\ \gamma_u^{q(u)} &= \Phi_u(v_u^{q(u)}), \end{aligned}$$

with $0 = v_u^0 < v_u^1 < v_u^2 < \cdots < v_u^{q(u)} = \beta_u$ and $0 = \gamma_u^0 < \gamma_u^1 < \gamma_u^2 < \cdots < \gamma_u^{q(u)}$.

With this notation we have:

$$\Phi_u(\mathbf{x}_u) = 0 \quad \text{if } \mathbf{x}_u = 0 \quad \text{and} \quad \forall i = 1, \dots, q(u)\colon \quad \Phi_u(\mathbf{x}_u) = \gamma_u^i \quad \text{for all } \mathbf{x}_u \in \left]v_u^{i-1}, v_u^i\right].$$

Figure 1 shows a typical cost function of this kind. Note here that the cost function $\Phi_u(\mathbf{x}_u)$ is not defined for values of $\mathbf{x}_u$ greater than $\beta_u = v_u^{q(u)}$, therefore our model will include bound constraints of the form: $0 \leqslant \mathbf{x}_u \leqslant \beta_u$.

Using the algebraic model introduced in section 1.2, finding a minimum cost multicommodity flow with the above cost functions is equivalent to the following mathematical programming problem:

$$(\text{DCMCF}) \qquad \begin{cases} \text{Minimize } \Phi(\mathbf{x}) = \displaystyle\sum_{u \in \mathcal{U}} \Phi_u(\mathbf{x}_u) \\ \text{subject to} \\ \mathbf{x}_u = \displaystyle\sum_{k=1}^{K} \left|\varphi_u^k\right| \quad (\forall u \in \mathcal{U}), \\ A \cdot \varphi^k = d_k b^k \quad (k = 1, \dots, K), \\ \forall u\colon \quad 0 \leqslant \mathbf{x}_u \leqslant \beta_u \end{cases}$$

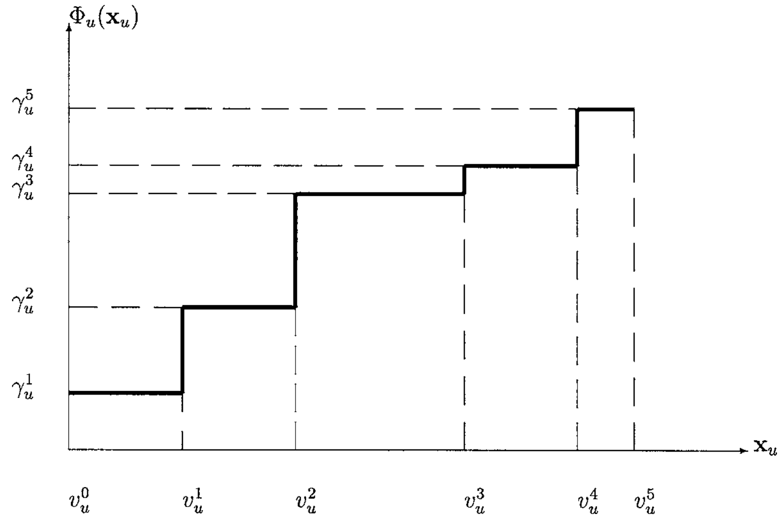

Figure 1. A typical cost function on one link $u \in \mathcal{U}$.

("minimum discrete cost multicommodity flow" problem).

An equivalent formulation using the feasible multicommodity flow polyhedron X is:

$$(\text{DCMCF}') \quad \begin{cases} \text{Minimize } \Phi(\mathbf{x}) = \sum_{u \in \mathcal{U}} \Phi_u(\mathbf{x}_u) \\ \text{subject to} \\ \forall u\colon \quad 0 \leqslant \mathbf{x}_u \leqslant \beta_u, \\ \mathbf{x} \in X. \end{cases}$$

We note that assuming, as above, upper bound constraints on the total capacity which may be installed on the links is actually not restrictive. Indeed, setting these upper bounds to the value $\sum_k d_k$ is equivalent to imposing no actual restriction on the way flows are routed on the network.

2.3. Versatility of the model: known special cases and new applications

The general problem (DCMCF) introduced above includes as special cases a number of network optimization problems studied in the literature.

(a) The so-called single-facility capacitated network loading problem, where capacity expansion on any given link u can be done by installing an integer number of units of a given basic facility characterized by its capacity C and its cost γ_u (see, e.g., [4,7]). Figure 2 shows a typical link cost function corresponding to the single-facility case.

(b) The so-called two-facility capacitated network loading problem, which generalizes the previous model in that, on each link u, capacity expansion can be achieved by means of two types of facilities, one "small" facility with capacity C^1 and cost γ_u^1 and a "large" facility with capacity C^2 and cost γ_u^2 with $\gamma_u^1/C^1 > \gamma_u^2/C^2$ (to comply

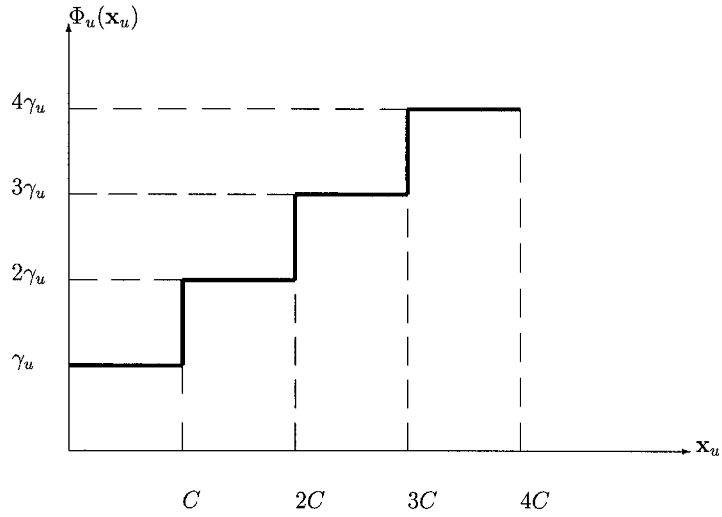

Figure 2. Link cost function for the single-facility network loading problem.

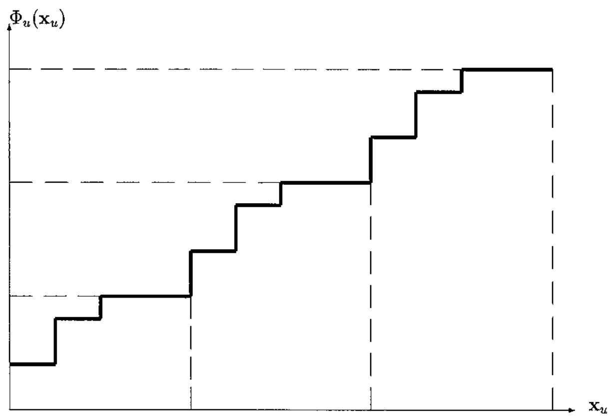

Figure 3. Link cost function for the two-facility network loading problem.

with the economy of scale phenomenon) (see [6,34]). Figure 3 shows a typical link cost function corresponding to the two-facility case.

Note, however, that the cost function shown in figure 3 may still not correspond well to the needs of practical applications for various reasons.

One of the reasons is that, in practice, due to the rapid increase of communication needs, once a big capacity system has been installed on a link, the smaller capacity system should not be considered anymore. This constraint is not easy to handle within a two-facility model, since it significantly alters the mathematical structure of the problem. However, in the framework of our general model, such a constraint is readily taken into account by changing the cost function of figure 3 by the cost function shown in figure 4.

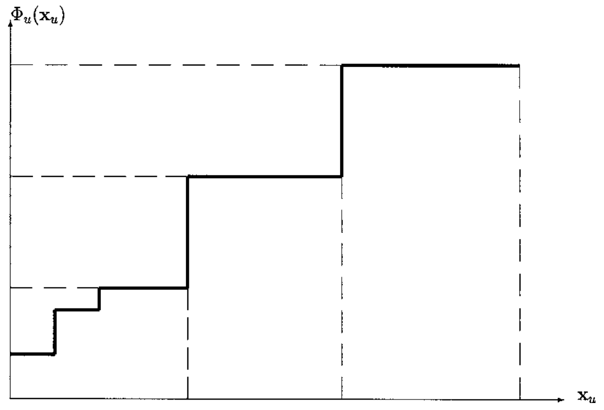

Figure 4. Link cost function for the two-facility case with the additional constraint that the smaller facility is no longer used, once the larger facility has been used.

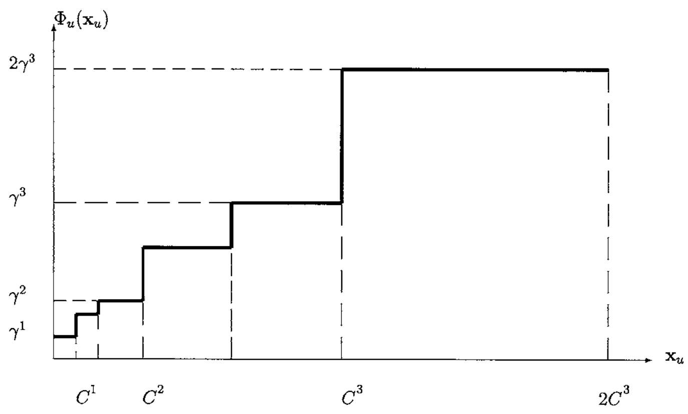

Figure 5. A typical link cost function corresponding to the case of three facilities (C^1, γ^1), (C^2, γ^2), (C^3, γ^3) with the constraint that, once a facility has been installed, smaller facilities are not installed any more.

Another reason which restricts the applicability of the single-facility or two-facility models is that, in practice, the number of available types of equipment may be significantly larger than 2. Considering our general model with arbitrary step increasing cost functions induces no limitation of this kind and enables proper handling of a huge variety of situations, such as the one shown in figure 5 (which corresponds to the case of 3 facilities with the additional constraint that, once a facility has been installed, smaller facilities are not installed any more).

2.4. Including node costs (switching equipment) in the model

In this section, we show that our model (DCMCF) remains essentially unchanged if discrete (step-increasing) node costs (representing installation of switching equipment) have to be taken into account.

The cost of switching equipment at each node $i \in \mathcal{N}$ essentially depends on the total volume of traffic, say $\mathbf{y}_i$, transiting through node i. As for link capacity increases, we will consider that node capacity augmentations may be performed by discrete amounts, corresponding to various available equipment types, each of which is specified by a (capacity, cost) pair. Thus we will assume that, for each $i \in \mathcal{N}$, we are given a step-increasing function $\Delta_i(\mathbf{y}_i)$ representing total switching cost at i as a function of $\mathbf{y}_i$.

We note that the $\mathbf{y}_i$ variables may easily be expressed in terms of the variables φ^k involved in the formulation (DCMCF) given above. Indeed, for the kth commodity, at any node $i \neq s(k)$, $i \neq t(k)$ the total flow of that commodity through node i is equal to $\frac{1}{2}\sum_{u\in\omega(i)} |\varphi_u^k|$ where $\omega(i)$ denotes, as usual, the set of arcs in $\widetilde{G}$ which have i as an endpoint (whether initial or terminal endpoint).

We therefore obtain:

$$\mathbf{y}_i = \frac{1}{2} \sum_{\substack{k/s(k)\neq i\\ t(k)\neq i}} \sum_{u\in\omega(i)} |\varphi_u^k|$$

which leads to the following statement of the problem

$$(\text{DCMCF}_1) \quad \begin{cases} \text{Minimize } \sum_{u\in\mathcal{U}} \Phi_u(\mathbf{x}_u) + \sum_{i\in\mathcal{N}} \Delta_i(\mathbf{y}_i) \\ \text{subject to} \\ \forall u \in \mathcal{U}\colon \quad \mathbf{x}_u = \sum_{k=1}^{K} |\varphi_u^k|, & (5) \\ \forall i \in \mathcal{N}\colon \quad \mathbf{y}_i = \frac{1}{2} \sum_{\substack{k/s(k)\neq i\\ t(k)\neq i}} \sum_{u\in\omega(i)} |\varphi_u^k|, & (6) \\ \forall k = 1,\ldots,K\colon \quad A\varphi^k = d_k b^k, & (7) \\ \forall u \in \mathcal{U}\colon \quad 0 \leqslant \mathbf{x}_u \leqslant \beta_u. & (8) \end{cases}$$

It is seen above that equations (6) are quite similar to equations (5) and, indeed, it is easy to check that (DCMCF_1) may be reformulated as a problem of type (DCMCF) on a new graph deduced from $\widetilde{G}$ by essentially splitting each node i into two distinct nodes i' and i'' and adding news arcs (i', i'') and (i'', i') along which the flow values are exactly the values of the flow transiting through node i in $\widetilde{G}$.

Thus (DCMCF_1) and (DCMCF) appear to be essentially equivalent: any algorithm solving (DCMCF) can be used to solve (DCMCF_1).

3. Overview of exact solution methods and computational results

We first provide in section 3.1 an overview of recent results obtained on some important special cases, mainly the single-facility and two-facility versions of the problem.

We next discuss available solution methods and computational results for (DCMCF) in the general case of arbitrary step-increasing cost functions (with no extra constraint).

The version of the problem with arbitrary step-increasing cost functions and survivability constraints (later on denoted (DCSMCF)) is discussed in section 3.3.

In this section, please keep in mind that the notation x_u, y_u indeed have a meaning different from $\mathbf{x}_u$, $\mathbf{y}_i$ used in the previous section.

3.1. Solution algorithms for special versions: single-facility and k-facility cases ($k \leqslant 3$)

The main stream of research directed towards obtaining exact solutions relies on the so-called "branch and cut" approach. More precisely, starting from an integer (or mixed-integer) LP formulation, valid inequalities and/or facet-defining inequalities are identified for strengthening the initial formulation, and possibly those corresponding to sub-problems generated in the course of a branch and bound process.

One justification for such an approach is that, as noted by many authors, the bounds provided by LP relaxations are usually weak, and strengthening the formulations prior to applying branch and bound is compulsory.

To our knowledge, one of the first published work dealing with k-facility network optimization problems ($k \leqslant 3$) is the paper by Magnanti and Mirchandani [32] which is restricted to the case of single-origin-destination. (Note that the above draws however from previous work [33,45].)

It is shown in [32] that, in the single-facility case, the problem reduces to a shortest path problem between s and t (origin and destination of the flow) and that this property is intimately related to an integrality property (with respect to the design variables) of the polyhedron obtained by adding to the flow equations and upper bounding constraints the set of so-called $s - t$ cut inequalities which are obtained as follows.

Let C denote the capacity of one module of the facility to be installed and, on each arc $u = (i, j)$ of the network, let $y_u \in \mathbb{N}$ denote the number of modules installed on link u. For any $s - t$ cut forming a partition of $\mathcal{N}$ into S and $\overline{S} = \mathcal{N} \backslash S$ with $s \in S$, $t \in \overline{S}$, denote $\omega(S)$ the set of edges having an endpoint in S and the other in $\overline{S}$ and $y(S, \overline{S}) = \sum_{u \in \omega(S)} y_u$. Then the corresponding cutset inequality reads:

$$y(S, \overline{S}) \geqslant \left\lceil \frac{d}{C} \right\rceil, \tag{9}$$

where d is the flow requirement between s and t and $\lceil d/C \rceil$ denotes the "smallest integer greater than or equal to d/C".

It is shown in [32] that the cut inequalities (9) above may be extended to the two- and three-facility versions of the problems. In the two-facility case, for instance, with two types of modules having capacity 1 and C, respectively, the extended cut inequalities read

$$x(S, \overline{S}) + ry(S, \overline{S}) \geqslant r \left\lceil \frac{d(S, \overline{S})}{C} \right\rceil, \tag{10}$$

where $d(S, \overline{S})$ is the total flow requirement between nodes in S and nodes in $\overline{S}$; $r = d(S, \overline{S}) \bmod C$; and x_u (respectively y_u) is the number of low capacity (respectively medium capacity) modules installed.

But, unfortunately, the nice result of the single-facility case does not extend to these more general versions of the problem (which turn out to be strongly NP-hard) and it can only be shown that (10) are typically facets of the associated integer polyhedron.

Further facet-defining inequalities are also derived in [32] for the three-facility case.

As suggested in [32] the various types of inequalities above (in particular, inequalities (10)) may be extended to the multicommodity case (where several single-commodity flow requirements between a given set of origin-destination pairs have to be simultaneously met).

A more detailed study of the multicommodity case for the two-facility version of the problem is provided in [34], where a polyhedral approach is described based on the cutset inequalities (10) (which are shown to be facet-defining if $d(S, \overline{S}) > 0$ and the subgraphs induced by S and $\overline{S}$ are both connected), and two other classes of inequalities: the so-called "arc-residual capacity" inequalities and "3-partition" inequalities.

The "arc-residual capacity" constraints are deduced from the polyhedral structure of the constraints of the subproblems arising when applying the Lagrangian relaxation principle to the individual flow conservation equations. (Indeed this Lagrangian subproblem decomposes into many subproblems, one for each arc in the network.)

Again, x_u (respectively y_u) denoting the number of low capacity (respectively medium capacity) modules installed, such an inequality for arc $u \in \mathcal{U}$ typically reads:

$$\sum_{k \in L} |\varphi_u^k| - x_u - r_L y_u \leqslant D_L - \mu_L r_L, \tag{11}$$

where $L \subset \{1, 2, \dots, K\}$ is any subset of the commodities,

$$D_L = \sum_{k \in L} d_k, \quad \mu_L = \left\lceil \frac{D_L}{C} \right\rceil \quad \text{and} \quad r_L = D_L \bmod C.$$

The "3-partition" inequalities are obtained by partitioning the node set into three (nonempty) subsets S_1, S_2, S_3.

For any i, j, $1 \leqslant i \leqslant 3$, $i \leqslant j \leqslant 3$, $i \neq j$, we denote $d(S_i, S_j)$ the sum of all requirements between S_i and S_j. Similarly we denote $x(S_i, S_j)$ (respectively $y(S_i, S_j)$) the sum of the values x_u (respectively y_u) for all edges having one endpoint in S_i and the other endpoint in S_j. For all distinct $i, j, k \in \{1, 2, 3\}$ denote:

$$q_i = \left\lceil \frac{d(S_i, S_j) + d(S_i, S_k)}{C} \right\rceil,$$
$$r_i = \big(d(S_i, S_j) + d(S_i, S_k)\big) \bmod C$$

and $\underline{r} = \min\{r_1; r_2; r_3\}$. Then the corresponding 3-partition inequality reads:

$$x(S_1, S_2) + x(S_1, S_3) + x(S_2, S_3) + \underline{r}\big(y(S_1, S_2) + y(S_1, S_3) + y(S_2, S_3)\big) \geqslant \left\lceil \frac{\underline{r}(q_1 + q_2 + q_3)}{2} \right\rceil. \tag{12}$$

Computational results on a series of 126 test problems are reported in [34]. They involve graphs with up to 15 nodes 34 arcs and 21 commodities (thus the requirement matrices have densities less than 20%). Exact solutions are obtained for instances with up to 10 nodes. For all instances, including the larger ones, the various cuts introduced appear to significantly reduce the integrality gaps.

The work of Barahona [4], which is limited to the single-facility version of the problem, is essentially based on the use of the same cutset inequalities (9) and some special version of *multicut inequalities* aimed at ensuring connectivity of the constructed network. The separation problem for the cutset inequalities (9) is shown to reduce to the well known (NP-hard) max-cut problem.

Computational results on three small instances (13 nodes, 7 nodes and 10 nodes, respectively) taken as subgraphs ("backbone networks") of larger network design problems are mentioned. Only approximate solutions (to within small gaps) are obtained. Note that in these experiments (due to the small sizes of the problems addressed), the separation subproblem for the cutset inequalities (9) is solved exactly.

Polyhedral results and computational experiments on the single-facility version of the problem have been presented in [7]. Two formulations of the problem are compared. Formulation F1 is based on the node-arc formulation of the multicommodity flow problem (this is referred to in the paper as "the multicommodity formulation"). It involves both design variables and flow variables. Formulation F2 is based on metric inequalities of the form (4) (this is referred to in the paper as "the capacity formulation").

The valid inequalities proposed in connection with F1 are basically the cut inequalities (9), the 3-partition inequalities (similar to those in [34]) and the "flow-cut-set" inequalities previously described in [6].

The valid inequalities proposed in connection with F2 are essentially the "rounded metric inequalities" and a special subclass of these referred to in the paper as "partition inequalities". We first explain the idea of "rounded metric inequalities". We know (see section 2.1) that a metric inequality is obtained by assigning to each link $u \in \mathcal{U}$, a "length" $\lambda_u \geqslant 0$ and reads

$$\sum_{u \in \mathcal{U}} \lambda_u y_u \geqslant \theta(\lambda) = \sum_{k=1}^{K} d_k \ell_k^*(\lambda),$$

where y_u denotes the number of modules of capacity 1 installed on link u, and where $\forall k = 1, \ldots, K$, $\ell_k^*(\lambda)$ is the length of the shortest chain joining $s(k)$ and $t(k)$ in G with respect to the "lengths" λ_u on the edges. Also we know that we may assume all λ_u's *rational*, therefore we may restrict to the case of integral λ_u values. Since, in the model considered, the requirement values d_k may be fractional numbers, $\theta(\lambda)$ may be fractional. In such a case, in view of the integrality of the y variables, the following inequality is valid:

$$\sum_{u \in \mathcal{U}} \lambda_u y_u \geqslant \lceil \theta(\lambda) \rceil \tag{13}$$

and is called a *rounded metric inequality*.

Partition inequalities (as defined in [7]) form a special class of the above which corresponds to choosing in any possible way, a partition of the node set $\mathcal{N}$ into p subsets $V_1, V_2, \ldots, V_p$, imposing $\lambda_u \geqslant 0$ for any edge having its endpoints in two distinct subsets of the partition and choosing $\lambda_u = 0$ for any other edge. Conditions are given under which partition inequalities are facet-defining.

Computational results on both formulations F1 and F2 are provided in [7] on two series of data sets, one corresponding to a basic network ("New York") with $N = 15$ and $M = 22$ (44 arcs when each edge has been replaced by two opposite arcs); the second one corresponding to a network with $N = 27$ and $M = 51$ (102 arcs).

Note that in the first data set, the requirement matrix is fully dense, but in the second one, it is quite sparse (20 pairs of nodes only have traffic to exchange, thus the density is less than 6%).

Exact optimal (integer) solutions to basic instances of the problem and some of their variants (obtained by drawing random requirement matrices while restricting to single source traffic patterns) are obtained within reasonable computing time (they typically range between $1'$ and $45'$ CPU) with both formulations using branch and cut.

However, the variants obtained by drawing random requirements while allowing *multiple sources* could not be solved exactly with either formulation within reasonable time limits (5 hours CPU for F2 and 1 hour CPU for F1) but the gaps obtained are always small (less than 3.5%) with an average of about 2.5% for the capacity formulation and less 1.0% for the multicommodity formulation (thus formulation F1 happened to perform significantly better on these instances).

Recent polyhedral investigations of the two-facility version of the problem have been presented by Bienstock and Günlük [6] and Günlük [28]. The results obtained in [6] include the use of cutset inequalities and 3-partition inequalities similar to those (independently) obtained in [34], but also a new type of facets referred to as "flow-cut-set facets" involving both the design variables x and y and the flow variables. The branch-and-cut algorithm presented in [28] uses various known types of inequalities (metric inequalities, spanning-tree inequalities, see [47]) together with a new class of inequalities (called "mixed partition inequalities") obtained by combining bipartition and three-partition inequalities.

Computational results in [6] are provided for a few instances involving two distinct network structures (one with 15 nodes and 22 links, the other with 16 nodes and 49 links). The traffic requirement matrices are fully dense, and, for the instances corresponding to the first network structure, there are existing capacities on each link (existing capacities can be used without extra cost) and a linear flow cost. It is shown that the strengthened formulation using all three classes of valid inequalities leads to reduced integrality gaps (0.4% within about 10 minutes CPU for data set 1, and about 20% for data set 2). In the case of data set 1, the times needed to obtain exact optimal integer solutions by using branch and bound on the strengthened formulation is between 10 to 15 seconds CPU, while applying branch and bound on the initial formulation takes much more time (hours for some of the instances).

The computational results reported in [28] involve three data sets, the first two corresponding to those in [6]. Most instances of these two data sets are solved to optimality, however no comment on how these results compare with those in [6] is provided. The third data set is borrowed from [7] and corresponds to a 27 node 51 link network. Apart from the two instances with very sparse traffic matrices, the other instances could not be solved exactly within 3 hours CPU (the final integrality gaps being quite small).

3.2. *Solution algorithms for the general case of arbitrary step-increasing cost functions*

We now turn to describe the available exact solution methods for the general model (DCMCF) introduced in section 2.2. The relevant references on this subject are Stoer and Dahl [51], Dahl and Stoer [11], and Gabrel, Knippel and Minoux [16] (see also [14,15] for previous related work).

The first two references also include discussion of a variant of the problem where *survivability constraints* are considered, in addition to the basic constraints of (DCMCF), but of course most of the polyhedral results stated there apply to both variants, with and without survivability constraints. The variant of the problem with survivability constraints will be discussed in more detail in section 3.3.

The first class of valid inequalities proposed in [51], called "band inequalities", are facet-defining inequalities for a relaxation of the problem composed of a single metric inequality of the form

$$\sum_{u\in\mathcal{U}} \lambda_u \mathbf{x}_u \geqslant \theta(\lambda)$$

which is rewritten as:

$$\sum_{u\in\mathcal{U}} \sum_{t=1}^{q(u)} g_u^t \underline{y}_u^t \geqslant \theta(\lambda), \tag{14}$$

where the $\underline{y}_u^t$ variables are 0–1 variables satisfying the following "ordering constraints":

$$\forall u \in \mathcal{U}:\quad 1 \geqslant \underline{y}_u^1 \geqslant \underline{y}_u^2 \geqslant \cdots \geqslant \underline{y}_u^{q(u)} \tag{15}$$

and where, for all u, the total capacity $\mathbf{x}_u$ installed on link u is expressed, in terms of the $\underline{y}_u^t$ variables as:

$$\mathbf{x}_u = \sum_{t=1}^{q(u)} \underline{y}_u^t \left(v_u^t - v_u^{t-1}\right)$$

(hence, the g_u^t values are the coefficients of the $\underline{y}_u^t$ variables after substituting the above expression of $\mathbf{x}_u$ in the metric inequality).

The set of integral solutions to (14), (15) may be viewed as a knapsack polytope with additional ordering constraints.

Denote by I the set of all pairs of indices (u, t) for all $u \in \mathcal{U}$ and $t = 1, 2, \ldots, q(u)$ and, for any subset of edges $S \subset \mathcal{U}$, $I(S)$ the set of all pairs (u, t) for $u \in S$, $t = 1, 2, \ldots, q(u)$.

Let $F \subset \mathcal{U}$ denote the support of the metric inequality considered, i.e., $F = \{u \mid u \in \mathcal{U},\ \lambda_u > 0\}$. A *band* B of F is a subset of $I(F)$ containing one and exactly one element of the form $(u, \bar{t}_u)$ with $\bar{t}_u \in \{1, 2, \ldots, q(u)\}$, for each $u \in F$.

Given a band $B \subset I(F)$, for all $u \in F$, the corresponding index $\bar{t}_u$ will be denoted t_u^B. We say that a band B is *valid* if $g(B^<) < \theta(\lambda)$ where:

$$g(B^<) = \sum_{u \in F} \sum_{t < t_u^B} g_u^t.$$

A band B' of F is said to be *above* a band B of F if $t_u^B \leqslant t_u^{B'}$ for all $u \in F$, and strict inequality holds for at least one $u \in F$.

Given a band B on F, the associated band inequality is defined as:

$$\underline{y}(B) = \sum_{u \in F} \underline{y}_u^{t_u^B} \geqslant 1. \tag{16}$$

It is shown in [51] that, whenever B is a valid band, the associated band inequality (16) is valid for the problem. Moreover if $|F| \geqslant 2$, (16) defines a *facet* of the convex hull of the integer solutions to (14), (15) if and only if there is no valid band above B. Note that, in the computational experiments, the band inequalities used are most often derived from cut inequalities of the form $\mathbf{x}(S, \overline{S}) \geqslant d(S, \overline{S})$, a special case of metric inequality.

Other types of valid inequalities and facets are also described in [51], including partition inequalities (which are used to express connectedness conditions), and other inequalities specific to the version of the problem where survivability constraints are considered (the so-called strengthened band inequalities, cut inequalities and the lifted two-cover inequalities, see section 3.3 below for more details).

The computational results described in [11] include instances of both versions of the problem with and without survivability constraints.

The results shown have been obtained using cut inequalities and band inequalities derived from cut inequalities. A heuristic procedure (based on an LP duality approach) is proposed for the separation of band inequalities. Testing feasibility of the solutions obtained is carried out by using a continuous LP solver (CPLEX) based on an arc-chain formulation of the feasible multicommodity flow problems. CPLEX is also used to solve the successive continuous LPs in the process of generating valid inequalities.

Results are shown in [11] for 4 instances: two instances corresponding to test set C (37 nodes, 44 edges and cost functions with up to 4 steps) and two instances corresponding to test set D (45 nodes, 53 edges and cost functions similar to those in test set C). Computation times are short (typically a few seconds) and an optimal solution is obtained for one of the instances of test set D. However for the other three instances large integrality gaps are obtained (54,5%, 66,8% and 22,7%, respectively). As indicated by the authors, other types of inequalities would be needed to reduce the gap for such instances of the problem.

It should be noted here that the above-mentioned test sets feature very sparse requirement matrices (less than 4% density for test set C and less than 7% density for D).

Gabrel, Knippel and Minoux [16] describe a constraint-generation (BENDERS-type [5,35,40]) approach for exact solution of the same general min-cost multicommodity flow problem with arbitrary step-increasing cost functions. This exploits the description of the feasible multicommodity flow polyhedron X as a large set of metric inequalities of type (4) as recalled in section 2.1. When only a few metric inequalities are used, a *relaxation* of the problem is obtained, which is eventually tightened by adding new metric inequalities violated by the optimal solution to the relaxed subproblem. The process stops when the (exact) optimal solution $\bar{\mathbf{x}}$ to the current relaxed subproblem satisfies all possible metric inequalities (even those not explicitly stated in the subproblem), i.e., when $\bar{\mathbf{x}} \in X$.

At the current iteration k of the constraint-generation approach, let J^k be the index set of metric inequalities generated so far corresponding to λ^j, $j \in J^k$. The current relaxed subproblem to be solved reads

$$(\mathrm{R}_k) \qquad \begin{cases} \text{Minimize } \displaystyle\sum_{u \in \mathcal{U}} \Phi_u(\mathbf{x}_u) \\ \text{subject to} \\ \displaystyle\sum_{u \in \mathcal{U}} \lambda_u^j \mathbf{x}_u \geqslant \theta(\lambda^j) \quad \forall j \in J^k, \\ \mathbf{x}_u \in V_u \quad \forall u \in \mathcal{U} \end{cases}$$

(we recall that $V_u = \{v_u^0, v_u^1, \ldots, v_u^{q(u)}\}$ denotes the set of discontinuity points of the cost function on edge u, see section 2.2).

(R_k) is reformulated as a pure 0–1 integer linear program by introducing, for each link, u, the $q(u)$ 0–1 variables $\underline{y}_u^1, \underline{y}_u^2, \ldots, \underline{y}_u^{q(u)}$ satisfying the ordering constraints (15) and expressing the $\mathbf{x}_u$ variables as:

$$\forall u \in \mathcal{U}: \quad \mathbf{x}_u = \sum_{t=1}^{q(u)} \underline{y}_u^t \left(v_u^t - v_u^{t-1}\right) \tag{17}$$

and the objective function as:

$$z = \sum_{u \in \mathcal{U}} \sum_{t=1}^{q(u)} \underline{y}_u^t \left(\gamma_u^t - \gamma_u^{t-1}\right). \tag{18}$$

Thus (R_k) reduces to the following 0–1 integer linear programming problem (ILP_k):

$$(\mathrm{ILP}_k)\quad \begin{cases} \text{Minimize } z = \sum_{u\in\mathcal{U}} \sum_{t=1}^{q(u)} \underline{y}_u^t \left(\gamma_u^t - \gamma_u^{t-1}\right) \\ \text{subject to} \\ \sum_{u\in\mathcal{U}} \lambda_u^j \left(\sum_{t=1}^{q(u)} \underline{y}_u^t \left(v_u^t - v_u^{t-1}\right)\right) \geqslant \theta\left(\lambda^j\right) \quad \forall j \in J^k, \\ \forall u \in \mathcal{U},\ \forall t = 2, \dots, q(u):\quad \underline{y}_u^t \leqslant \underline{y}_u^{t-1}, \\ \forall t = 1, \dots, q(u):\quad \underline{y}_u^t \in \{0, 1\}. \end{cases}$$

(In the computational experiments reported in [16], (ILP_k) is solved by using a standard commercial LP-software, namely CPLEX 4.0 in MIP mode.)

In order to solve the separation subproblem, the criterion proposed in [16] to select the metric inequalities violated by the current optimal integer solution $\bar{\mathbf{x}}$ to (R_k) is taken to be the ratio:

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

($\rho > 1$ thus corresponding to a violated inequality). The problem of finding one "most violated" inequality (maximizing ρ) in the general class of metric inequalities, may then be stated as the "auxiliary problem"

$$(\mathrm{AP})\quad \begin{cases} \text{Find } \lambda \text{ maximizing } \theta(\lambda) \\ \text{under the constraints} \\ \sum_{u\in\mathcal{U}} \lambda_u \bar{\mathbf{x}}_u = 1, \\ \lambda \geqslant 0 \end{cases}$$

which is solved (approximately) by using a subgradient-type algorithm.

However, in order to improve the computational efficiency of the procedure, (AP) is not systematically solved at each step, but only when attempts at generating violated *bipartition inequalities* have been unsuccessful (bipartition inequalities correspond to setting $\lambda_u = 1$ for all edges u separating two subsets of nodes in the network and $\lambda_u = 0$ for all other edges). The problem of finding a bipartition inequality maximizing the ratio $\theta(\lambda)/\sum_{u\in\mathcal{U}} \lambda_u \bar{\mathbf{x}}_u$ may be stated as finding $S \subset \mathcal{N}$ and $\overline{S} = \mathcal{N} \backslash S$ maximizing the ratio: $\rho = d(S, \overline{S})/\bar{\mathbf{x}}(S, \overline{S})$, where $d(S, \overline{S})$ denotes the sum of all requirements crossing the cut $(S, \overline{S})$ and $\bar{\mathbf{x}}(S, \overline{S}) = \sum_{u\in\omega(S)} \bar{\mathbf{x}}_u$ is the sum of capacities on the edges of the cut in the solution $\bar{\mathbf{x}}$. This problem is NP-hard (since max-cut is a special case) but is solved approximately via a fast local search algorithm inspired from Kernighan and Lin [31]. The procedure implementing this algorithm to find a near-optimal cut $\omega(S, \overline{S})$ containing a specific edge $(i_0, j_0) \in \mathcal{U}$ and maximizing ρ is denoted MAX-RATIO-CUT[i_0, j_0]. Using this device, several violated bipartition inequalities are systematically looked for at each step (multiple constraint generation) by running MAX-RATIO-CUT[i_0, j_0] for all possible $(i_0, j_0) \in \mathcal{U}$, to obtain a set of near-optimal cuts covering all the edges (in practice, it is observed that $\mathcal{O}(N)$ cuts are thus computed).

Of course, among the resulting bipartition inequalities, only those achieving $\rho > 1$ are actually added to the current relaxed subproblem. The comparison carried out between single constraint generation and the multiple constraint generation scheme above shows clear superiority of the latter (with multiple constraint generation, the total number of main iterations does not exceed 14 for all the instances solved and appears to increase quite slowly with problem size). Moreover, the necessity of solving the auxiliary problem (AP) occurs only very rarely (in only two instances over 50).

Systematic computational testing of the procedure on a series of 50 instances for networks up to 20 nodes and 37 edges is reported in [16]. In these instances, the link cost functions have an average number of 6 steps. Also an important feature of the instances considered, as opposed with the test problems in [11,51], is that the requirement matrices are *fully dense* (the practical difficulty of (DCMCF) seems to be much increased for larger values of the density parameter).

The average total number of generated constraints is about 60 for 12-node networks, 90 for 15-node networks, and 150 for 20-node networks.

The computation times necessary to reach exact optimality (using CPLEX 4.0 in MIP mode to solve the relaxed subproblems (ILP_k)) increase quite rapidly with problem size: about 500 sec on average for 12-node instances; 4400 sec on average for 15-node instances; 22000 sec on average for 20-node instances. Also worth pointing out in these results is the variability in the computation times, which lie in the range [22 sec–1471 sec] for 12-node networks, [565 sec–13473 sec] for 15-node networks, and [2139 sec–51644 sec] for 20-node networks.

In all cases, the time taken by the (multiple) constraint generation process appears to be quite negligible (less than 1% of total time for the larger problems).

3.3. *Solution algorithms for the case of arbitrary step-increasing cost functions and survivability constraints*

We address here an important extension of our basic general model (DCMCF) to handle situations where *survivability constraints* have to be imposed. Such constraints tend to arise more and more frequently in applications, due to the very high capacities provided by modern transmission equipment, such as optical carriers.

Survivability constraints express the fact that, in addition to meeting the given multicommodity requirements, the network to be designed should remain feasible with respect to the given requirements in case of any failure situation out of a given list of possible failure situations. In practice, a typical failure situation is when one link or one node in the network fails. Simultaneous failure of several links or nodes having very small probability, most practical applications only require survivability with respect to single-link and/or single-node failure. Until recently, previous work on the optimum survivable network design problem has concentrated on the case of linear cost functions.

This gives rise to large scale linear programs with special structure which can be tackled via various kinds of LP decomposition techniques, see [23,39,44] and the surveys in [27,43].

Also various simplified ways of handling survivability constraints, e.g., by considering various types of connectivity constraints have been extensively studied, see [9,25,26,46].

Handling survivability constraints in the single-facility network loading problem has been addressed in [8]. Various ways of strengthenning a given cutset inequality (of type (9)) are suggested there, through a polyhedral investigation of some basic polytopes arising when stating the inequalities expressing the survivability conditions for the given cut. However no computational result with this approach is reported in [8].

Here we concentrate on the extended version of (DCMCF) to include survivability constraints while keeping completely general step-increasing link cost functions. For simplicity, we restrict to the case of link failures only, since including node failures in the model would only result in making the notation slightly more intricate.

Again, let the network structure be represented by the (undirected) graph $G = [\mathcal{N}, \mathcal{U}]$ and suppose that the multicommodity flow requirements are given by the associated list of K source-sink pairs $s(k)$ and $t(k)$ (for $k = 1, \ldots, K$) with corresponding requirement values d_k. In order to express survivability of the network with respect to any single link failure, we introduce, for each link $v \in \mathcal{U}$, the operator π_v defined as follows. For any vector $\mathbf{x} \in \mathbb{R}^M$ (where $\forall u \in \mathcal{U}$, $\mathbf{x}_u$ represents total flow through link u in the network) $\pi_v(\mathbf{x}) = \mathbf{x}'$ where:

$$\begin{aligned} &\forall u \neq v, \quad \mathbf{x}'_u = \mathbf{x}_u, \\ &\mathbf{x}'_v = 0 \end{aligned}$$

(π_v is thus the projection operator of $\mathbb{R}^M$ onto $\mathbb{R}^{M-1}$ spanned by coordinates $1, 2, \ldots, v-1, v+1, \ldots, M$).

With this notation, the survivability constraint corresponding to failure of link $v \in \mathcal{U}$ may be easily expressed as: $\pi_v(\mathbf{x}) \in X$. In view of this, the discrete minimum cost survivable multicommodity flow problem to be solved may be formulated as:

$$\text{(DCSMCF)} \qquad \begin{cases} \text{Minimize } \displaystyle\sum_{u \in \mathcal{U}} \Phi_u(\mathbf{x}_u) \\ \text{subject to} \\ \forall v \in \mathcal{U}\text{:} \quad \pi_v(\mathbf{x}) \in X, \\ \forall u \in \mathcal{U}\text{:} \quad \mathbf{x}_u \in V_u. \end{cases} \tag{19}$$

("Discrete cost survivable multicommodity flow problem".)

Note that, in the above, the condition $\mathbf{x} \in X$ has not been stated, since it is implied by each of the conditions (19) (this is so because $\mathbf{x}' = \pi_v(\mathbf{x}) \leqslant \mathbf{x}$ and $\mathbf{x}' \in X, \mathbf{x}' \leqslant \mathbf{x} \Rightarrow \mathbf{x} \in X$).

The only previous references we are aware of dealing with this problem (when arbitrary step-increasing cost functions are considered) are the already mentioned works by [11,51] and also [17]. In order to deal with (DCSMCF), several valid inequalities specific to the survivability case have been proposed in [51] and [11]. Only the so-called "strengthened band inequalities" derived from cut inequalities have been used in the experiments. They are obtained as follows. Consider a metric inequality of the form (14) which is valid for the nominal state of the network (i.e., when no edge failure occurs), F its support, and B a band of F.

If, $\forall v \in F$:

$$\sum_{u \in F \setminus \{v\}} \sum_{t < t_u^B} g_u^t < \theta(\lambda)$$

then it can be shown that the inequality

$$\underline{y}(B) \geqslant 2 \quad \text{("strengthened band inequality")} \tag{20}$$

(see equation (16) in section 3.2) is valid and defines a facet under rather weak conditions.

As illustrated by the numerical results reported in [11], these inequalities appear very efficient at reducing the integrality gap. Experiments on 23 instances drawn from 4 test sets involving survivability constraints have been tried (test set A concerns a 27 node 51 edge network with six step link cost functions; test set B a 118 node 134 edge network with 5 step cost functions; test sets C and D have already been mentioned in section 3.2). For most of these instances (18 over 23) the integrality gaps are reduced (less than 3%) and exact optimal integer solutions are obtained on 8 of the instances, the computing times being always less than a few minutes.

As already pointed out for test sets C and D, in all these test problems the requirement matrices are very sparse (6% for test set A, and less than 2% density for test set B).

The constraint-generation approach described in [16] has also recently been extended to the version of (DCMCF) with survivability constraints in [17]. Several constraint-generation strategies described below have been proposed and compared computationally.

Under strategy A, at each step and for each of the failure cases, the multiple constraint generation procedure of [16] is applied (see section 3.2). Thus all the violated bipartition inequalities which are determined by applying the procedure MAX-RATIO-CUT[i_0, j_0] for all possible $(i_0, j_0) \in \mathcal{U}$ are added to the current problem. Since, for each failure case, the number of resulting inequalities is approximately $\mathcal{O}(N)$, the total number of constraints generated using strategy A is $\mathcal{O}(MN)$, and this will tend to reduce the total number of main iterations. However, having a large number of constraints in the subproblems may result in increased computational effort in solving the subproblems, therefore the other two strategies B and C described below aim at reducing the number of generated bipartition inequalities.

Strategy B consists in considering only one bipartition inequality for each failure case, namely the one achieving largest value of the ratio ρ. Among the M possible candidates, only those achieving $\rho > 1$ are actually added to the subproblem. Thus clearly at most M inequalities may be generated, at each step, under strategy B.

Table 1
Average computation times in seconds (over the 20-node instances) as reported in [16] for solving (DCMCF) and in [17] for solving (DCSMCF).

(DCMCF) results from [16] using CPLEX 4.0		22159 sec
(DCSMCF) results from [17] using CPLEX 6.0	with strategy A	7949 sec
	with strategy B	8702 sec
	with strategy C	6687 sec

Under *Strategy C*, the same bipartition inequalities as those computed by strategy A are first determined. However only a few of them are actually selected to be added to the current subproblem, according to the following selection procedure. First, all the violated bipartition inequalities found are sorted according to decreasing values of the ratio ρ. Next, the subset S of selected inequalities is determined as follows. The first inequality in the ordered list (the one achieving largest ratio) is put in S. Then the inequalities having rank i (for $i = 2, 3, \ldots$) are examined in turn. The ith inequality in the list is selected (added in S) if at least one edge in the associated cut does not belong to any of the cuts corresponding to the previously selected inequalities. In this way, at most M inequalities may be generated at each step under strategy C.

From the computational results provided in [17], none of the above strategies come up with a clear superiority as compared with the other two. For 15-node networks, from a series of 10 test problems, the best strategy in terms of computing times required to get an exact optimal solution is A for 5 of the instances and C for 5 of the instances. For 20-node networks, A is best for 7 instances, B for 1 instance and C for 4 instances.

In terms of total number of main iterations (i.e., total number of subproblems solved) strategy A clearly outperforms B and C. For 20-node networks, for instance, the average number of main iterations is 4.5 for strategy A, 12.4 for B and 10.1 for C.

The main conclusion, however, to be drawn from the results in [17] is that the computing times necessary to find exact optimal solutions to the version of the problem with survivability constraints (DCSMCF) appear to be comparable to those reported in [16] for the problem without survivability constraints (DCMCF). For 20 node instances, we indicate in table 1 above the average computation times reported in [16] for solving (DCMCF) (using CPLEX 4.0 in MIP mode to solve the subproblems) and those reported in [17] for solving (DCSMCF) (using CPLEX 6.0 for the subproblems) with strategies A, B, C, respectively.

4. Possible improvements and directions for future investigations

From the various contributions mentioned in section 3, it appears that both the discrete cost multicommodity flow problem (DCMCF) and its variant with survivability constraints (DCSMCF) are extremely difficult combinatorial optimization problems for which only fairly small instances can be solved exactly with currently available techniques. For the general case (arbitrary step-increasing cost functions) and 100% dense requirement matrices, which corresponds to the most difficult cases, instances with about 20 nodes, 40 links and cost functions featuring on average 6 steps per link, are typical of the current upper limits if guaranteed exact optimal solutions are expected.

Being able to solve exactly significantly larger instances (50 nodes, 100 links, say) represents a real challenge which is not likely to be met without major improvements in the currently available techniques. In this section, possible improvements and directions for future research are discussed. In section 4.1 the possibility of strengthening the formulation of the subproblems (R_k) considered in [16,17] is discussed and a method for building a strengthened formulation is proposed. In section 4.2 other directions for future investigations are discussed in connection with the approaches based on polyhedral results.

4.1. Strengthened metric inequalities for (DCMCF) and (DCSMCF)

We consider here the possibility of strengthening the formulation of each subproblem (R_k) used in the approach proposed in [16] and recalled in section 3.2. The constraints in (R_k) are all the metric inequalities (most of them being bipartition inequalities) generated so far. Each metric inequality reads:

$$\sum_{u\in\mathcal{U}} \lambda_u^j \mathbf{x}_u \geqslant \theta(\lambda^j) \tag{21}$$

for some $\lambda^j \in \mathbb{R}_+^M$ and

$$\theta(\lambda^j) = \sum_{k=1}^{K} d_k \times \ell_k^*(\lambda^j).$$

Since, in (R_k), each variable $\mathbf{x}_u$ is constrained to belong to the finite discrete set $V_u = \{v_u^0, v_u^1, \ldots, v_u^{q(u)}\}$, the left-hand side of (21) cannot be smaller than the minimum value $\theta^*(\lambda^j)$ of the knapsack-like subproblem:

$$\mathrm{SP}[\lambda^j] \qquad \begin{cases} \theta^*(\lambda^j) = \min \sum_{u\in\mathcal{U}} \lambda_u^j \mathbf{x}_u \\ \text{subject to} \\ \sum_{u\in\mathcal{U}} \lambda_u^j \mathbf{x}_u \geqslant \theta(\lambda^j), \\ \forall u \in \mathcal{U}\colon \quad \mathbf{x}_u \in V_u \end{cases}$$

which may be equivalently rewritten as:

$$\mathrm{SP}[\lambda^j] \qquad \begin{cases} \theta^*(\lambda^j) = \min \sum_{u\in\mathcal{U}} \alpha_u \\ \text{subject to} \\ \sum_{u\in\mathcal{U}} \alpha_u \geqslant \theta(\lambda^j), \\ \forall u \in \mathcal{U}\colon \quad \alpha_u \in \{\lambda_u^j v_u^0, \lambda_u^j v_u^1, \ldots, \lambda_u^j v_u^{q(u)}\}. \end{cases}$$

Once the optimal value $\theta^*(\lambda^j)$ has been computed, it is seen that the inequality:

$$\sum_{u\in\mathcal{U}} \lambda_u^j \mathbf{x}_u \geqslant \theta^*(\lambda^j) \tag{22}$$

is a *valid inequality* for (R_k). Of course, from the definition, one always has $\theta^*(\lambda^j) \geqslant \theta(\lambda^j)$, but, apart from exceptional cases, strict equality holds, in which case (22) is stronger than (21). We call it a *strengthened metric* inequality.

We note that $\mathrm{SP}[\lambda^j]$ can be solved by applying, e.g., a *dynamic programming* algorithm as classically done for solving knapsack problems in pseudopolynomial time (see, e.g., [50, chapter 16] or [42, chapter 7]). Of course, such a procedure may not be very efficient in the case of general metric inequalities where most λ_u^j are arbitrary positive real numbers. But this is not the case of *bipartition inequalities* for which most λ_u^j are 0 and only a few λ_u^j are 1, those corresponding to the edges of a cut. Even for a 50-node 100-edge network with average node degree 4, the average cardinality of a cut will usually be well below 50 (this is because telecommunication networks usually correspond to almost planar graphs and, if we apply Euler's formula to a planar graph G with $N = 50$, $M = 100$, the number of nodes in the topological dual graph G^* is $2 - N + M = 52$, an upper bound for the cardinality of any cut in G).

In view of the above, it is seen that it will be possible to efficiently compute the $\theta^*(\lambda^j)$ values for bipartition inequalities therefore leading to a *strengthened subproblem*:

$$(\mathrm{R}_k^*) \qquad \begin{cases} \text{Minimize } \sum_{u\in\mathcal{U}} \Phi_u(\mathbf{x}_u) \\ \text{subject to} \\ \sum_{u\in\mathcal{U}} \lambda_u^j \mathbf{x}_u \geqslant \theta^*(\lambda^j) \quad \forall j \in J^k, \\ \mathbf{x}_u \in V_u, \quad \forall u \in \mathcal{U}. \end{cases}$$

Clearly, since in most cases $\theta^*(\lambda^j) > \theta(\lambda^j)$, the lower bounds derived from the linear relaxation of (R_k^*) will be tighter than those derived from the linear relaxation of (R_k), opening the way to improved computational efficiency in the solution of the subproblems.

We note that the strengthened metric inequalities suggested above also apply to the approach described in [17] for (DCSMCF).

4.2. *Suggested directions for future investigations*

From the overview of existing techniques presented in section 3 above, it appears that one of the most promising approaches for exact solution of (DCMCF) or (DCSMCF) is based on a thorough exploitation of existing or new polyhedral results for strengthening the various possible LP formulations of the problem itself, or of the subproblems encountered in its solution. First we observe that all available polyhedral results have not been fully exploited yet computationally. This is the case, for instance, of the "lifted two-cover inequalities" and of the "partition inequalities" derived in [51].

Also we observe that considering *alternative formulations* may suggest the use of available polyhedral results relevant to those formulations. As an example of this, we would like to suggest a possible reformulation of the subproblem (R_k) stated in section 3.2.

Let us associate with each edge $u \in \mathcal{U}$, $q(u) + 1$ 0–1 variables $\mu_u^0, \mu_u^1, \dots, \mu_u^{q(u)}$ satisfying

$$\sum_{t=0}^{q(u)} \mu_u^t = 1,$$

the $\mathbf{x}_u$ variables being expressed as:

$$\forall u \in \mathcal{U}: \quad \mathbf{x}_u = \sum_{t=0}^{q(u)} v_u^t \mu_u^t$$

and the objective function as:

$$z = \sum_{u \in \mathcal{U}} \sum_{t=0}^{q(u)} \gamma_u^t \mu_u^t.$$

With the above notation, (R_k) may be reformulated as:

$$(\mathrm{ILP}_k)' \qquad \begin{cases} \text{Minimize } z = \displaystyle\sum_{u \in \mathcal{U}} \sum_{t=0}^{q(u)} \gamma_u^t \mu_u^t \\ \text{subject to} \\ \displaystyle\sum_{u \in \mathcal{U}} \lambda_u^j \left(\sum_{t=0}^{q(u)} v_u^t \mu_u^t \right) \geqslant \theta(\lambda^j) \quad \forall j \in J^k, \\ \forall u \in \mathcal{U}: \quad \displaystyle\sum_{t=0}^{q(u)} \mu_u^t = 1, \\ \forall t = 0, 1, \dots, q(u): \quad \mu_u^t \in \{0, 1\}. \end{cases}$$

We note that the solution set of $(\mathrm{ILP}_k)'$ is the intersection of solution sets W of knapsack problems with multiple choice constraints which, for the sake of notational simplicity, may be reformulated as:

$$\begin{cases} -\displaystyle\sum_{j=1}^{n} a_j x_j \leqslant -b, & (23) \\ \displaystyle\sum_{j \in S_i} x_j = 1 \quad \forall i \in I, & (24) \\ x \in \{0, 1\}^n, \end{cases}$$

where $a_j > 0$ for $j = 1, \ldots, n$, $b > 0$ and all S_i are disjoint,

$$\bigcup_{i \in I} S_i = \{1, 2, \ldots, n\}.$$

Valid inequalities for closely related problems, namely knapsack problems with GUB (Generalized Upper Bound) constraints have been described in [52]. We extend below the proof from [52] to solution sets W defined by (23), (24), to show that essentially the same inequalities as those described there are valid inequalities for the case of multiple choice (MC) constraints.

First let us introduce some useful notation.

We say that a subset $C \subseteq \{1, 2, \ldots, n\}$ is a MC-cover ("Multiple Choice Cover") for W iff:

(i) $|C \cap S_i| \leqslant 1, \forall i \in I$,

(ii) $-\sum_{j \in C} a_j > -b$.

Given a MC-cover C, we denote $I^+ = \{i \in I \mid C \cap S_i \neq \emptyset\}$ and, for any $i \in I^+$:

$$S_i^+ = \{j \in S_i \mid a_j \leqslant a_\ell \text{ for } \ell \in C \cap S_i\}.$$

We can then state

Proposition 1. Let $W = \{x \mid x \in \{0, 1\}^n,\ x \text{ satisfies (23) and (24)}\}$ and C be any MC-cover for W. Then the inequality:

$$\sum_{i \in I^+} \sum_{j \in S_i \setminus S_i^+} x_j + \sum_{i \in I \setminus I^+} \sum_{j \in S_i} x_j \geqslant 1 \tag{25}$$

is *valid* for W.

Proof. Consider $\bar{x} \in W$ and suppose that (25) does not hold, in other words that

$$\sum_{i \in I^+} \sum_{j \in S_i \setminus S_i^+} \bar{x}_j + \sum_{i \in I \setminus I^+} \sum_{j \in S_i} \bar{x}_j = 0. \tag{26}$$

(26) implies that each of the two terms on the left-hand side is 0, which means that the only nonzero components of $\bar{x}$ are for $j \in S_i^+$, $i \in I^+$. Moreover, since $\bar{x} \in W$, $\forall i \in I^+$ there is at most one $\bar{x}_j = 1$, $j \in S_i^+$.

In view of this, and using the definition of S_i^+, we can write:

$$\sum_{j=1}^{n} a_j \bar{x}_j \leqslant \sum_{j \in C} a_j$$

hence

$$-\sum_{j=1}^{n} a_j \bar{x}_j \geqslant -\sum_{j \in C} a_j > -b$$

which contradicts the fact that $\bar{x} \in W$, and completes the proof. □

Clearly, the inequalities (25) appear to be potentially useful in a branch and cut approach for solving the subproblems $(\text{ILP}_k)'$. It is possible to show (this was suggested by one of the referees) that the band inequalities (16) for (ILP_k), when expressed in terms of the μ_u^t variables (using the linear transformations $\underline{y}_u^t = \sum_{r=t}^{q(u)} \mu_u^r$) just give rise to valid inequalities of the form (25) for $(\text{ILP}_k)'$ and conversely. Therefore, the two formulations (ILP_k) reinforced by (16) and $(\text{ILP}_k)'$ reinforced by (25) appear to be essentially equivalent, at least from a theoretical standpoint. However they *may not be computationally equivalent* since, depending on implementation details, multiple-choice constraints (24) may be handled more efficiently by some MIP solvers, while ordering constraints (15) would be handled more efficiently by others. Carrying out a full computational comparison between the two approaches in an interesting subject for future investigations.

Acknowledgments

Three anonymous referees are gratefully acknowledged for their constructive remarks and suggestions on the first version of the paper.

References

[1] R. Ahuja, T.K. Magnanti and J.B. Orlin, *Networks Flows, Algorithms and Applications* (Prentice-Hall, 1993).
[2] A.A. Assad, Multicommodity network flows – a survey, Networks 8 (1978) 37–91.
[3] A. Balakrishnan, T.L. Magnanti and P. Mirchandani, Network design, in: *Automated Bibliographies in Combinatorial Optimization*, eds. M. Dell'Amico, F. Maffioli and S. Martello (Wiley, 1998).
[4] F. Barahona, Network design using cut inequalities, SIAM Journal on Optimization 6(3) (1996) 823–837.
[5] J.F. Benders, Partitioning procedures for solving mixed-variable programming problems, Numer. Math. 4 (1962) 238–252.
[6] D. Bienstock and O. Günlük, Capacitated network design-polyhedral structure and computation, INFORMS Journal on Computing (1996) 243–259.
[7] D. Bienstock, S. Chopra, O. Günlük and C.Y. Tsai, Minimum cost capacity installation for multicommodity network flows, Mathematical Programming 81 (1998) 177–199.
[8] D. Bienstock and G. Muratore, Strong inequalities for capacitated survivable network design problems, Math. Programming Ser. A 89 (2000) 127–147.
[9] S.C. Boyd and T. Hao, An integer polytope related to the design of survivable communication networks, SIAM Journal of Discrete Math. 6(4) (1993) 612–630.
[10] S. Chopra, I. Gilboa and S. Trilochan-Sastry, Algorithms and extended formulations for one and two facility network design, in: *Proceedings 5th IPCO Conf.*, Lecture Notes in Computer Science (1996) pp. 44–57.
[11] G. Dahl and M. Stoer, A cutting-plane algorithm for multicommodity survivable network design problems, INFORMS Journal on Computing 10(1) (1998).
[12] L.R. Ford and D.R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, 1962).

[13] L. Fratta, M. Gerla and L. Kleinrock, The flow-deviation method: an approach to store-and-forward communication network design, Networks 3 (1973) 97–133.
[14] V. Gabrel and M. Minoux, Large scale LP-relaxations for minimum cost multicommodity network flow problems with step-increasing cost functions and computational results, Internal Research Report, MASI 96/17, University Paris 6, France (1996).
[15] V. Gabrel and M. Minoux, LP relaxations better than convexification for multicommodity network optimization problems with step-increasing cost functions, Acta Mathematica Vietnamica 22 (1997) 128–145.
[16] V. Gabrel, A. Knippel and M. Minoux, Exact solution of multicommodity network optimization problems with general step cost functions, Operations Research Letters 25 (1999) 15–23.
[17] V. Gabrel, A. Knippel and M. Minoux, Exact solution of survivable network problems with step-increasing cost functions, in: *ALGOTEL 2* (May 10–12, 2000).
[18] C. Gallo, C. Sandi and C. Sodini, An algorithm for the min concave cost flow problem, European J. Operational Research 4 (1980) 248–255.
[19] B. Gendron, T.G. Crainic and A. Frangioni, Multicommodity capacitated network design, in: *Telecommunications Network Planning*, eds. B. Sanso and P. Soriano (Kluwer Academic, 1999) pp. 1–19.
[20] G. Geoffrion and G. Graves, Multicommodity distribution system design by Benders decomposition, Management Science 5 (1974) 822–844.
[21] M. Gerla and L. Kleinrock, On the topological design of distributed computer networks, IEEE Trans. Comput. COM-25 (1977) 48–60.
[22] M. Goemans and D. Bertsimas, Survivable networks, linear programming relaxations and the parsimonious property, Mathematical Programming 60(2) (1993) 145–166.
[23] R.E. Gomory and T.C. Hu, An application of generalized linear programming to network flows, SIAM J. Appl. Math. 10 (1962) 260–283.
[24] M. Gondran and M. Minoux, *Graphs and Algorithms* (Wiley, New York, 1984).
[25] M. Grötschel and C. Monma, Integer polyhedron associated with certain network design problems with connectivity constraints, SIAM J. Discrete Math. 3 (1990) 502–523.
[26] 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 (1992) 309–330.
[27] M. Grötschel, C. Monma and M. Stoer, Design of survivable networks, in: *Handbook in OR and MS*, Vol. 7 (1995) pp. 617–672.
[28] O. Günlük, A branch-and-cut algorithm for capacitated network design problems, Math. Programming Ser. A 86 (1999) 17–39.
[29] H.H. Hoang, Topological optimization of networks: A nonlinear mixed integer model employing generalized Benders decomposition, IEEE Trans. Automatic Control AC-27 (1982) 164–169.
[30] J. Kennington, Multicommodity flows: A state-of-the art survey of linear models and solution techniques, Operations Research 26 (1978) 209–236.
[31] B.W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Techn. J. 49(2) (1970) 291–307.
[32] T.L. Magnanti and P. Mirchandani, Shortest paths, single origin-destination network design and associated polyhedra, Networks 33 (1993) 103–121.
[33] T.L. Magnanti, P. Mirchandani and R. Vachani, Modeling and solving the capacitated network loading problem, Working Paper #709, KATZ graduate School of Business, University of Pittsburgh, PA (1991).
[34] T.L. Magnanti, P. Mirchandani and R. Vachani, Modeling and solving the two-facility capacitated network loading problem, Operations Research 43(1) (1995) 142–157.
[35] T.L. Magnanti, P. Mireault and R.T. Wong, Tailoring Benders decomposition for uncapacitated network design, Mathematical Programming Study 26 (1986) 112–154.

[36] T.L. Magnanti and R.T. Wong, Network design and transportation planning-models and algorithms, Transportation Science 18 (1984) 1–55.

[37] T.L. Magnanti and R.T. Wong, A dual ascent approach to fixed charge network design problems (1999) to appear.

[38] M. Minoux, Optimisation et planification des réseaux de télécommunications, in: *Optimization Techniques*, eds. G. Goos, J. Hartmanis and J. Cea, Lecture Notes in Computer Science 40 (Springer, 1976) pp. 419–430.

[39] M. Minoux, Optimum synthesis of a network with non-simultaneous multicommodity flow requirements, in: *Studies on Graphs and Discrete Programming*, ed. P. Hansen, Annals of Discrete Mathematics 11 (1981) pp. 269–277.

[40] M. Minoux, Subgradient optimization and Benders decomposition for large scale programming, in: *Mathematical Programming*, eds. R.W. Cottle, M.L. Kelmanson and B. Korte (1984) pp. 271–288.

[41] M. Minoux, Network synthesis and dynamic network optimization, in: *Surveys in Combinational Optimization*, eds. S. Martello, G. Laporte, M. Minoux and C. Ribeiro, Annals of Discrete Mathematics 31 (1986) pp. 283–324.

[42] M. Minoux, *Mathematical Programming. Theory and Algorithms* (Wiley, 1986).

[43] M. Minoux, Network synthesis and optimum network design problems: Models, solution methods and applications, Networks 19 (1989) 313–360.

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

[45] P.M. Mirchandani, *Polyhedral Structure of a Capacitated Network Design Problem with an Application to the Telecommunication Industry*, Ph.D. dissertation (MIT Press, Cambridge, MA, 1989).

[46] C. Monma and D. Shallcross, Methods for designing communication networks with certain two-connected survivability constraints, Operations Research 37 (1989) 531–541.

[47] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).

[48] A. Onaga and O. Kakusho, On feasibility conditions of multicommodity flows in networks, IEEE Trans. on Circuit Theory CT 18 (1971) 425–429.

[49] A. Ouorou, Décomposition proximale des problèmes de multiflots à critère convexe. Application aux problèmes de routage dans les réseaux de communication, Thèse de Doctorat, Université de Clermont-Ferrand (1995).

[50] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization. Algorithms and Complexity* (Prentice-Hall, 1982).

[51] M. Stoer and G. Dahl, A polyhedral approach to multicommodity survivable network design, Numerische Mathematik 68 (1994) 149–167.

[52] L.A. Wolsey, Valid inequalities for 0–1 knapsacks and MIPS with generalized upper bound constraints, Discrete Appl. Math. 29 (1990) 251–261.