MULTICOMMODITY FLOW PROBLEMS IN TELECOMMUNICATIONS NETWORKS
Abdel Lisser\textsuperscript{1} and Philippe Mahey\textsuperscript{2}

\textsuperscript{1}Université de Paris Sud, LRI - UMR 8623,
Bât 490, 91405 Orsay Cedex, France
lisser@lri.fr

\textsuperscript{2}Université Blaise Pascal, LIMOS - UMR 6158,
B.P. 10125, 63173, Aubière Cedex, France
Philippe.Mahey@isima.fr

Abstract: The purpose of this chapter is to investigate multicommodity flow problems that appear in the network design and operation of modern broadband packet-switched networks. We present arc-node and arc-path models and analyze specialized formulations corresponding to hard to solve instances like the minimax congestion problem and the capacity assignment of data networks in the presence of failures. Decomposition methods are studied to cope with the coupling constraints which define interactions between commodities on critical arcs or the combinatorial choice between normal and spare capacities. We focus here mainly on continuous flow models with linear or convex costs.

Keywords: Multicommodity flow problems, network design, packet-switched networks, arc-node models, arc-path models, minimax congestion, capacity assignment, decomposition methods, continuous flow models.

1.1 INTRODUCTION

In the early eighties, a new broadband technology was developed at Bellcore, called SONET (Synchronous Optical Network) (Advisory, 1990) and normalized for European countries under the name of SDH (Synchronous Digital Hierarchy). SDH and new optical fibers gave rise to the high speed transmission networks era. Due to SDH systems and equipment, difficult multiplexing and demultiplexing functionalities which characterize the previous technology become elementary operations and
can be handled easily. The transmission capacities for telecommunications systems, grow exponentially and rates of Gigabit per second, Gb/s, are quickly achieved.

Transmission facilities gave rise to new multimedia services (voice, data, and video) with flexible bandwidth, and highly demanding in terms of bandwidth and quality of service. Virtual private networks, as well as Internet, came into sight with dazzling developments. Moreover, the competition brought about by deregulated telecommunication markets and customer requirements decreased in large scale the transmission costs and highlighted network survivability in order to provide a high quality of service.

SDH facilities use a pair of fibers per system and their high capacity simplifies network architectures. Therefore, the graphs underlying the networks are sparse and the traffic per cable is very high. This makes the network more sensitive to system or equipment failures and increases survivability issues.

Recent advances in wavelength technology, namely Wavelength Digital Technology WDM (Brackett, 1990), allow to handle several wavelengths, each carrying 2.5 Gb/s per fiber, raising the speed up to Terabit per second, Tb/s, through a long distance. The savings in the optical fibers as well as in the optical amplifiers (10 times SDH amplifying capabilities) are significant.

The fundamental task in telecommunication network design consists in transporting different services in order to connect different customers. This operation is called routing. In order to achieve this task, graph theory and optimization fields provide the multicommodity flow model.

It is well known that telecommunications networks can be modeled as graphs where the nodes correspond to telecommunication centers (local, central offices, ...) while the arcs represent the cables. Each arc, i.e. cable, has a given capacity according to the equipment it connects. This capacity is shared by individual demands. To route a given demand or commodity through the network, a cost should be paid according to different used arcs. This leads to a minimum cost multicommodity netflow problem, or MCNF for short. This problem has been widely studied in the literature both for theoretical (Schrijver, 2003) and practical purposes (Ahuja et al., 1993).

Recent applications in telecommunications in the last decade are: Traffic management of Asynchronous Transfer Mode (ATM) networks (Medova, 1998); Virtual path routing for survivable ATM networks (Murakami and Kim, 1996); Design of centralized telecommunication networks (Gupta and Pirkul, 1996); Design of telecommunication loss networks (Girard and Sanso, 1998; Medhi and Gupitan, 1975); Design of fiber transport networks (Yoon et al., 1998); Routing and wavelength assignment in optical networks (Banerjee and Mukherjee, 1996); Terminal layout with hop constraints (Gouveia, 1996); Design of token ring LANs (LeBlanc et al., 1996); Routing in virtual circuit data networks with unreliable links (Sung and Lee, 1995); Telephone network traffic management (Chang et al., 1992; Chang, 1994); Ring network design (Arbib et al., 1992); Backbone network design in communications network (Bogdanowicz, 1993); Routing in packet-switched networks (Ribeiro and Elbaz, 1992); Virtual circuit routing in computer networks (Yee and Lin, 1992; Zhang and Hartmann, 1992); Fair integration of routing and flow-control in communication networks (Chang, 1992); Design of lightwave networks (Labourdette and Acampora, 1991); Design of surviv-
able telecommunication networks (Lisser et al., 1998; 2000); Lightpath assignment for multifibers WDM networks with wavelength translators (Coudert and Rivano, 2002); A family of algorithms for network reliability problems (Shaio, 2002); A procedure for resource allocation in switchlet networks (Fonseca et al., 2002); A simple polynomial time framework for reduced-path decomposition in multi-path routing (Mirrokni et al., 2004); Optimized routing adaptation in IP networks utilizing OSPF and MPLS (Riedl, 2003); Design of a meta-mesh of chain subnetworks: Enhancing the attractiveness of mesh-restorable WDM networking on low connectivity graphs (Grover, 2002); and Simultaneous routing and resource allocation via dual decomposition (Xiao et al., 2004).

In this chapter, we present two formulations of MCNF. These formulations are used for the routing of different traffic demands in telecommunications. Several techniques have been applied to solve multicommodity flow problems and their extensions. These techniques exploit the special structure of MCNF to reduce the problem to a sequence of solutions of smaller problems. We will describe here different decomposition schemes illustrated on minimax congestion problems, routing with problems with QoS criterion and survivability problems. The chapter is organized as follows. In Section 1.2, we present different formulations of MCNF. We use the nonoriented multicommodity flow model for the flow formulation. Congestion problems and their equivalent counterpart, the maximum concurrent flow problem are presented in Section 1.3. In Section 1.4, we discuss and formulate survivability problems in telecommunications networks. Two decomposition techniques applied to spare capacity problem are presented in Section 1.4.3.

1.2 MCNF FORMULATIONS

In formulating MCNF, we can adopt either of two equivalent modeling approaches: the flow formulation, where flows are defined on arcs, and the path formulation, where flows are defined on paths (Ahuja et al., 1993).

Let $G = (V, E)$ be the graph representing the underlying telecommunications network with $|V| = n$ nodes and $|E| = m$ edges. Denote by $x^k_{ij}$ the decision flow variable corresponding to the flow fraction of the commodity $k$ on the arc $(i, j)$ and $c^k_{ij} > 0$ be its routing cost. Let $u_{ij}$ be the capacity of the edge $(i, j)$.

The flow formulation can be expressed as follows:

$$
\begin{align*}
\text{min} & \sum_{k \in K} \sum_{(i,j) \in E} c^k_{ij} |x^k_{ij}| \\
\text{s.t.} & \quad Nx^k = r^k, \quad \forall k \in K, \\
& \quad \sum_{k \in K} |x^k_{ij}| \leq u_{ij}, \quad \forall (i,j) \in E.
\end{align*}
$$

where:

- $N$ is the node-arc incidence matrix corresponding to an arbitrary orientation of the network graph (i.e. $N_{ij} = +1$ if arc $j$ is directed away from node $i$, $N_{ij} = -1$ if arc $j$ is directed towards node $i$, $N_{ij} = 0$ otherwise),
**K** is a set of commodities. \( k \) is a commodity characterized by a demand \( d^k \), in number of traffic units, to be routed through the network between a given pair of nodes, \( s^k \) and \( t^k \).

- \( r^k \) is the requirement vector for commodity \( k \) (i.e. \( r^k = d^k \) if \( k = s^k \), \( r^k = -d^k \) if \( i = t^k \) and \( r^k = 0 \) otherwise).

- \( x^k \) is the flow vector for a given commodity \( k \) (i.e. \( x^k_{ij} \) is the flow on edge \((i, j)\) with \( x^k_{ij} > 0 \) if the flow goes from \( i \) to \( j \) and \( x^k_{ij} < 0 \) if the flow goes from \( j \) to \( i \)).

An equivalent linear programming formulation is obtained by introducing nonnegative variables, \( x^k_{ij}^+ \) and \( x^k_{ij}^- \), such that \( x^k_{ij} = x^k_{ij}^+ - x^k_{ij}^- \) and \( |x^k_{ij}| = x^k_{ij}^+ + x^k_{ij}^- \), see Minoux (1986),

\[
\begin{align*}
\text{min} & \quad \sum_{k \in K} \sum_{(i,j) \in E} c^k_{ij} (x^k_{ij}^+ + x^k_{ij}^-) \\
\text{s.t.} & \quad A \left( x^k_+ - x^k_- \right) = r^k, \quad \forall k \in K, \\
& \quad \sum_{k \in K} \left( x^k_{ij}^+ + x^k_{ij}^- \right) \leq u_{ij}, \quad \forall (i,j) \in E, \\
& \quad x^k_{ij}^+, x^k_{ij}^- \geq 0, \quad \forall k \in K, \quad \forall (i,j) \in E. 
\end{align*}
\] (1.2)

Constraints (1.2b) express the flow conservation for each node \( i \in V \) and for each \( k \in K \), whereas constraints (1.2c) express that for each \((i, j) \in E \) the flow routed on \((i, j)\) must not exceed the capacity of edge \((i, j)\). The standard form of this model has \( M = |K||V| + |E| \) constraints and \( N = 2|K||E| \) variables. This problem is large even for networks with a small number of nodes as shown in Table 1.1. It can be solved directly by LP packages or by specialized interior point and simplex based algorithms (Chardaire and A.Lisser, 2002a;b). The size of (1.2) can also be reduced by aggregating the commodities by origins or by destinations (Chardaire and A.Lisser, 2002b).

1.2.1 Path formulation

The path formulation starts with an enumeration of all paths \( p \) between any commodity pair of nodes. Denote by \( P^k \), the set of routing paths associated with commodity \( k \), i.e. the set of all simple paths in \( G \) that connect \( s^k \) and \( t^k \). For each path, \( p \in P^k \), denote by \( X^k_p \) the fraction of the demand, \( d^k \), routed on \( p \). For each edge, \( e \in E \), denote by \( S^k_e \) the set of paths that contain \( e \). Let \( C^k_p = \sum_{e \in P^k} c^k_e \) be the cost of the path \( p \). The path formulation problem can then be written as
Table 1.1 Size of MCNF problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>m</th>
<th>n</th>
<th>K</th>
<th>Constraints</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>JLL-split</td>
<td>26</td>
<td>42</td>
<td>257</td>
<td>6981</td>
<td>34994</td>
</tr>
<tr>
<td>A-500-split</td>
<td>119</td>
<td>302</td>
<td>500</td>
<td>59802</td>
<td>302302</td>
</tr>
<tr>
<td>A-900-split</td>
<td>119</td>
<td>302</td>
<td>900</td>
<td>108302</td>
<td>758102</td>
</tr>
<tr>
<td>A-1000-split</td>
<td>119</td>
<td>302</td>
<td>1000</td>
<td>119302</td>
<td>604302</td>
</tr>
<tr>
<td>A-1885-split</td>
<td>119</td>
<td>302</td>
<td>1885</td>
<td>224617</td>
<td>1138842</td>
</tr>
<tr>
<td>B-5800-split</td>
<td>119</td>
<td>302</td>
<td>5800</td>
<td>835801</td>
<td>4240986</td>
</tr>
<tr>
<td>B-7021-split</td>
<td>119</td>
<td>302</td>
<td>7021</td>
<td>842822</td>
<td>5911984</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\min & \sum_{k \in K} \sum_{p \in P_k} c_p x_p \\
\text{s.t.} & \sum_{p \in P_k} x_p = d_k, \quad \forall k \in K, \\
& \sum_{p \in \delta^+ e} x_p \leq u_e, \quad \forall e \in E, \\
& x_p \geq 0, \quad \forall p \in P_k, \forall k \in K.
\end{align*}
\]

Constraints (1.3b) specify that for each \( k \in K \) demand \( d_k \) has to be routed, whereas constraints (1.3c) express that for each \( e \in E \), the flow routed on arc \( e \) must not exceed its capacity. The number of constraints, i.e. \(|K| + |E|\), is much smaller than in the flow formulation, but the number of variables is exponential in the number of nodes of \( G \). Problem (1.3) can be solved using decomposition methods. Amongst all, Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1961) was originally created for solving such problems. In Chardaire and A.Lisser (2002a), specialized algorithms for problem (1.3) based both on the simplex method and interior point methods are proposed and numerical results provided on large size instances. The simplex based method corresponds to Dantzig-Wolfe decomposition while the interior point methods are used within the Analytic Center Cutting Plane Method also called ACCPM (Chardaire and A.Lisser, 2002a; Goffin et al., 1992). It is also shown in (Chardaire and A.Lisser, 2002a) that the path formulation outperforms the node arc formulation and Dantzig-Wolfe decomposition outperforms ACCPM for the set of tested instances.

Numerical results for the flow and path formulations are displayed in Table 1.2, see Chardaire and A.Lisser (2002b). Concerning the flow formulation, columns “iter” and “CPU secs” report the number of iterations of specialized simplex algorithm and the computational time, respectively. The first column under the path formulation gives the number of generated paths, whereas the other columns give the number of iterations of the master program, the number of iterations...
of the subproblems and the computational time, respectively. It is easy to see that the path formulation requires less computation time than the flow formulation.

We will consider now a specialized version of the multicommodity flow problem, the optimal routing problem with a congestion cost function.

1.3 MINIMIZING CONGESTION ON THE NETWORK

1.3.1 Minimal congestion and the maximum concurrent flow problem

The minmax congestion network design problem consists in determining an optimal topology supporting a given traffic (with many origin-destination pairs) which minimizes the maximal congestion on the arcs of the network.

We will consider a network routing problem where we want to minimize the maximal relative congestion on the arcs of the network. The problem of minimizing the flow on the most congested link when we not only have to route some traffic in a network, but also have to decide the topology of the network, is of valuable interest for the design of data communication networks (see Bertsekas and Gallager (1987)).

The basic problem of minimizing the more congested arc in the routing of a multicommodity flow will be described below as:

\[
(MINCONG) \quad \min \quad z \quad \text{s.t.} \quad \begin{cases} \ x_e - u_e z \leq 0 & \forall e \in E \\ N x^k = r^k & \forall k \in K \\ x \in X \end{cases}
\]

Table 1.2 Performance of flow and path formulations using specialized simplex algorithms

<table>
<thead>
<tr>
<th>Problem</th>
<th>Flow Formulation (1.2)</th>
<th>Path Formulation (1.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Itr</td>
<td>CPU (secs)</td>
</tr>
<tr>
<td>JLL-split</td>
<td>23661</td>
<td>217.7</td>
</tr>
<tr>
<td>A-5000-split</td>
<td>207</td>
<td>16.68</td>
</tr>
<tr>
<td>A-900-split</td>
<td>7110</td>
<td>805.0</td>
</tr>
<tr>
<td>A-1000-split</td>
<td>10138</td>
<td>1285.11</td>
</tr>
<tr>
<td>A-1885-split</td>
<td>77892</td>
<td>17261.88</td>
</tr>
<tr>
<td>B-5800-split</td>
<td>†</td>
<td>†</td>
</tr>
<tr>
<td>B-7021-split</td>
<td>†</td>
<td>†</td>
</tr>
</tbody>
</table>

† Memory limit exceeded
where $x_e = \sum_{k \in K} |x^k_e|$ and $X$ represents eventual topology constraints. For instance, some authors have considered constraints on the degrees of the node in the topology (see Labourdette and Acampora (1991)), and they have stressed that even very small instances on an 8-node mesh and with in- and out-degrees forced to 2 are very hard to solve with the best available Branch-and-cut codes (Bienstock and Günlük, 1995).

We will focus first on the simplest case of (MINCONG) without topology constraints. Even if it is a linear multicommodity flow problem, thus an LP which can be solved by standard decomposition techniques exploiting the underlying flow structure, it is generally considered to be a hard problem.

The first reason why this occurs is that the objective function (in the min-max formulation) is convex piecewise linear and not separable with respect to arcs. The second reason is that it produces optimal solutions with a huge number of active paths, being in that sense equivalent to the Maximum Concurrent Flow problem as shown below. Bienstock (2003) reported a set of numerical experiments on very large maximum concurrent flow problems (with up to $4 \times 10^5$ rows and $2 \times 10^6$ columns) exhibiting abnormal cubic growth of the CPU time to solve them with the CPLEX dual code.

Observe that the minmax objective function implies that we can solve (MINCONG) without capacity constraints and test feasibility afterwards by comparing the optimal value of $z$ with the value 1. (MINCONG) itself is feasible if and only if there exists at least one path linking each origin to its destinations. Let $z^*$ be an optimal value for (MINCONG); there exists a feasible multicommodity flow satisfying the capacity constraints if and only if $z^* \leq 1$.

The path structure of an optimal solution of (MINCONG) is a very important issue for the traffic manager who needs to implement a given routing table in a given period. Too many paths will penalize the protocols which take charge of the packets at each switching equipment, and too few paths can make the solution too sensitive to failures.

We will call active a path that carries some positive flow and critical an arc $e$ such that $x_e = u_e z$. An active path containing critical edges will be called a critical path. Suppose now that some commodity is routed on a critical path at the optimal solution and that there exists a second path supporting that commodity which is not critical. Both paths form a cycle, so that one can modify the solution deviating a small quantity from the critical path $p$ to the noncritical one $p'$. A new basic optimal solution should be obtained when $p'$ turns out to be critical. Critical edges appear thus as bottlenecks for some “critical” commodities:

**Proposition 1** There always exists an optimal solution of (MINCONG) such that, if any commodity is routed on a critical path, all paths used to route that commodity are critical.

Analyzing the basic structure of the arc-path formulation of (MINCONG), an optimal solution will contain at most $|K| + \sigma$ active paths (and at least $|K|$), where $\sigma$ is the number of critical edges.

As observed in Shahrokhi and Matula (1990), (MINCONG) is also strongly related to the Maximum Concurrent Flow problem (MAXCONCUR), where one wants to maximize the throughput of the network for a given set of capacities. The throughput of the network is a load factor that multiplies the traffic matrix.

---

1Abdel & Philippe: Is this sentence correct? I changed it a little because it didn’t make sense. Do you mean the protocols which control the packets?
HANDBOOK OF OPTIMIZATION IN TELECOMMUNICATIONS

\[
\text{(MAXCONCUR)} \quad \max Z \\
\text{s.t.} \quad \begin{cases} 
  x_e & \leq u_e & \forall e \in E \\
  N x^k - Z r^k & = 0 & \forall k \in K
\end{cases}
\]

One can easily verify that the optimal solutions \( x^* \) and \( \tilde{x}^* \) of (MINCONG) and (MAXCONCUR), respectively, satisfy \( x^* = z^* \tilde{x}^* \) and that \( z^* = 1/Z^* \).

Both problems (MINCONG) and (MAXCONCUR) have received a lot of attention in the past decade, especially since the seminal paper by Shahrokhi and Matula (1990) who first proposed a fully polynomial approximation scheme to solve (MAXCONCUR) with uniform capacities. They showed that the minimization of a separable exponential penalty function on the arcs yields a flow with a nearly maximal throughput. They chose the following penalty function:

\[
\phi_e(x_e) = \exp\left(\frac{2|E|^2}{\varepsilon} x_e^2\right)
\]

where \( \varepsilon \) is a positive parameter which defines the approximation. They found a complexity bound of \( O(|V||E|^7/\varepsilon^5) \) and showed that the number of active paths is \( O(|E|^3/\varepsilon^2) \). Faster algorithms based on refinements of that exponential penalty function have been since proposed and extended to nonuniform capacities and to other packing and covering problems (see Garg and Könemann (1998); Grigoriadis and Khachiyan (1994); Leighton et al. (1995)).

An interesting link between the Flow Deviation algorithm described in Section 1.2 and the resolution of (MINCONG) has been analyzed recently by Bienstock and Raskina (2002). They proposed an algorithm to solve (MINCONG) which alternates between a magnification step where the throughput is increased for a fixed congestion and a potential reduction step where congestion is minimized for a fixed throughput. That latter step is realized by performing flow deviation inner steps with the Kleinrock’s congestion function \( \phi_e(x_e) = \frac{x_e}{u_e - x_e} \) (proportional to the queueing delay under Poissonian hypotheses for the entering traffic routed on arc \( e \)). These are the crucial calculations as they will introduce new active paths in the solution. Bienstock and Raskina suggested to substitute the shortest-path calculations by min-cost flow subproblems. This comes naturally in the linearization procedure of Frank-Wolfe’s algorithm if we add redundant capacity constraints for each commodity, i.e. \( x^k_e \leq u_e, \forall e \). The higher cost of the subproblems is compensated by better bounds in the complexity estimates and that trick is also used in other papers (see Leighton et al. (1995) or Grigoriadis and Khachiyan (1994)). Recently, some authors modified the main step of the general algorithm to get back to shortest-path calculations without affecting the complexity bound (see Garg and Könemann (1998)).

Thus, the interesting idea behind these contributions brought by the “Approximation Algorithms” community is the possibility to approximate the nonseparable congestion problem (MINCONG) by a separable multicommodity flow model with convex increasing costs on the arcs which can be resumed in the following problem (without topology constraints):

\[
\text{(MINDEL)} \quad \min \sum_{e \in E} \phi_e(x_e) \\
\text{s.t.} \quad \begin{cases} 
  x_e & \leq u_e & \forall e \in E \\
  N x^k = r^k, & \forall k \in K
\end{cases}
\]

We will analyze below two algorithmic approaches to solve (MINDEL) when the arc cost functions are given by the average delay suffered by a data packet on arc \( e \). Indeed, independence assumptions with Poissonian arrival on the queueing network lead to Kleinrock’s delay function which is proportional to \( \phi_e(x_e) = \frac{x_e}{u_e - x_e} \). The total cost is simply \( \Phi(x) = \sum_{e} \phi_e(x_e) \).

Observe that the objective function acts as a barrier function on the capacity constraints which can thus be ignored in the model.
We will begin with the Flow Deviation method which was used very early to solve capacity and flow assignment problems on packet-switched networks (see Fratta et al. (1973)). The second approach is the Proximal Decomposition method which has been tested on large routing problems for broadband data networks (see Ouorou et al. (2000)).

1.3.2 The flow deviation method

The Flow Deviation method (FD) is an adaptation of the classical linearization algorithm of Frank and Wolfe (1956) to solve multicommodity flow problems with convex costs. It was first proposed by Fratta et al. (1973) in the context of design of packet-switched networks and has been widely used by the community of transportation (see LeBlanc et al. (1985)) and in telecommunications networks (see Bertsekas and Gallager (1987)). We recall below the formal ideas behind Frank-Wolfe’s algorithm for general nonlinear programs with linear constraints:

Minimize $\Phi(x)$

s.t. $Ax = b$

$x \geq 0$

The method procedes by successive linearization for solving LP subproblems at each iteration $t$ where the gradient $\nabla \Phi(x^t)$ has been computed. Let $\bar{x}^t$ be the optimal solution of the subproblem

$$\bar{x}^t = \text{Argmin}_{x \in P} \nabla \Phi(x^t)x$$

where $P = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ is the polyhedron of feasible solutions supposed bounded. Thus, we can suppose that $\bar{x}^t$ is an extreme point of $P$. Then the new iterate is obtained by carrying out a line search on the segment $[x^t, \bar{x}^t]$ with the nonlinear function $\Phi$. Convergence results have been obtained in the strictly convex case (see Bertsekas (1995) for example) but the method suffers from very slow convergence tail which is evidenced by the following fact. The solution of the nonlinear program is in general not a vertex of the feasible polyhedron so that, as the sequence of feasible solutions tends to a point on a face of the polyhedron, the angle between successive descent directions tends to 180° (see Figure 1.1). It is shown in Bertsekas (1995) how typical sublinear convergence rate can be exhibited in very simple and low-dimensional situations.

On the other hand, the positive aspect of the implementation of Frank-Wolfe’s method to multicommodity flow problems is the simplicity of the subproblem resolution which reduces to shortest-path computations and the solution update which consists in deviating flows fairly from the active paths towards the new ones without explicitly computing the individual commodity flows as we can see below.

Now, we will apply the Flow Deviation method to (MINDEL) with the Kleinrock’s delay function, a smooth convex function, separable with respect to arcs.

Assume that we get a strictly feasible multicommodity flow at iteration $k$, i.e. such that $x_e < u_e, \forall e$. Then, the linearization step at iteration $t$ of (FD) reduces to separate shortest-path computations for each commodity $k$ with arc lengths $\nabla \phi_e(x_e^t)$. The solution $\bar{x}^t$ corresponds to the situation where all demands $d_k$ are routed separately on these new paths. Observe that $\bar{x}^t$ may not satisfy the capacity constraints. The new solution $x^{t+1}$ is obtained by carrying out a line search on the segment $[x^t, \bar{x}^t]$. Observe again that the new solution will be forced strictly feasible by the barrier function.

The main point in the procedure above is the fact that all computations are performed on the total flow variables. However, in some situations, one may be interested in computing the optimal path flows. An arc-path formulation is then necessary and the (FD) iterations can be carried out on the path flow variables as explained in Bertsekas and Gallager (1987). The first
Figure 1.1 Zigzagging convergence of Frank and Wolfe’s method

step of the procedure is unchanged so that, if \( \tilde{p} \) is the shortest path with the first-derivative arc lengths for a given commodity \( k \), then we set \( X_p = d_k \) and, denoting by \( P^k \) the subset of paths carrying the current solution, the flow deviation step can be computed in the following way:

\[
X^t_{p+1} = X^t_p + \theta_t(\tilde{X}_p^t - X^t_p), \forall p \in P^k_{t+1} = P^k_t \cup \{ \tilde{p} \}
\]

where \( \theta_t \) is the optimal step size which minimizes the objective function over all \( \theta \in [0, 1] \) (observe that the feasible direction whose components are the \( \tilde{X}_p^t - X^t_p \) is a descent direction for \( \Phi \).

Thus an equal fraction of the flow on the nonshortest paths is shifted to the shortest path. One may observe that this update of path flows will tend to increase monotonically the number of active paths. Variants of the basic Flow Deviation method can be built by modifying one of the inner steps of the algorithm, i.e.

- either by modifying the search direction, for instance by substituting the shortest-path calculations by min-cost flow subproblems for each commodity,
- or by deviating nonuniform flow proportions on the newly generated paths.

To understand the first strategy, one can add the redundant constraints \( x^k_e \leq u_e, \forall e, k \) to the model (MINDEL). After the linearization step, the direction-finding subproblem of (FD) decomposes now in \( K \) minimum-cost flow problems. The computational overhead of these subproblems compared to the original shortest-path calculations is compensated by the choice of feasible paths (considering one commodity at a time). But the drawback is that more paths are likely to be generated which is exactly what we do not wish, and the path flow updates are not so straightforward as in the original method. Observe nevertheless that some authors have chosen to implement min-cost flow computations instead of shortest-path calculations to improve worst-case behaviour of some approximation algorithms (see Plotkin et al. (1995)).
There are many different strategies to modify the search direction which result in nonuniform deviations from the current active paths to the shortest one. Second-order information can be used to yield Newton or Quasi-Newton directions as suggested in Bertsekas and Gallager (1987). Conjugate gradient strategies have also been tested in earlier works (see LeBlanc et al. (1985)). Again, the expected gain in convergence rate is overtaken by the extra work of performing each iteration and reconstituting the path support.

1.3.3 The Proximal Decomposition method for convex cost multicommodity flow problems

The Proximal decomposition method (PDM) is an adaptation of the Alternating Direction Method of Multipliers originally proposed by Glowinski and Marocco (1975) and further developed by Eckstein and Fukushima (1993) and Mahey et al. (1995). It can be viewed as a separable Augmented Lagrangian method and it is best understood when applied to the following generic separable convex program with coupling constraints:

\[
\begin{align*}
\text{Minimize} & \quad \Phi(x) = \sum_{i=1}^{p} \phi_i(x_i) \\
\text{subject to} & \quad \sum_{i=1}^{p} A_i x_i = b \\
& \quad x_i \in S_i, i = 1, \ldots, p
\end{align*}
\]

where the variables are partitioned into \( p \) blocks, all functions being convex on the compact convex sets \( S_i \); suppose also that there exists \( x_I \in S_i, i = 1, \ldots, p \) such that \( \sum_{i=1}^{p} A_i x_i = b \) so that the problem has an optimal solution with value \( v^* \).

The key idea is to add primal allocation variables \( y_i, i = 1, \ldots, p \) to get the equivalent formulation (where \( \sum_i b_i = b \)):

\[
v^* = \min_{x,y} \{ \Phi(x) \mid A_i x_i + y_i = b_i, x_i \in S_i, i = 1, \ldots, p, \sum_i y_i = 0 \}
\]

which is itself equivalent to:

\[
v^* = \min_{x,y} \{ \Phi(x) + \frac{\lambda}{2} \sum_i \|A_i x_i + y_i - b_i\|^2 \mid A_i x_i + y_i = b_i, x_i \in S_i, i = 1, \ldots, p, \sum_i y_i = 0 \}
\]

The Lagrangian dual with respect to the local constraints in \((x_i, y_i)\) will induce the following subproblem :

\[
v(u_1, \ldots, u_p) = \min_{x, y \in \mathcal{Y}} \{ \Phi(x) + \frac{\lambda}{2} \sum_i \|A_i x_i + y_i\|^2 + \sum_i \langle u_i, A_i x_i + y_i - b_i \rangle \}
\]

where \( \mathcal{Y} = \{ (y_1, \ldots, y_p) \mid \sum_i y_i = 0 \} \) is the primal subspace. To decompose that subproblem, the trick is to apply some kind of Gauss-Seidel scheme to alternate minimizations with respect to \( x \) and \( y \). Indeed, the minimization w.r.t. \( y \) can be solved explicitly.

To sum up, we first solve the inner subproblem with respect to \( x \) for a fixed \( u' \) and fixed allocations \( y'^{-1} \in Y \); observe that the subproblem decomposes in \( p \) subproblems, where the \( i \)-th subproblem is:

\[
\text{Minimize}_{x_i \in S_i} \{ \phi_i(x_i) + \frac{\lambda}{2} \|A_i x_i + y_i'^{-1} - b_i\|^2 + \langle u'_i, A_i x_i + y_i'^{-1} - b_i \rangle \}
\]

Let \( x'^{i} \) be the optimal solution. We must then solve the primal allocation subproblem, i.e. the following quadratic program with linear equality constraints:

\[
\inf_{y \in \mathcal{Y}} \sum_i \frac{1}{2} \|A_i x'^{i} + y_i - b_i\|^2 + \langle u'_i, A_i x'^{i} + y_i - b_i \rangle
\]
$y^\prime$ is an optimal solution of that quadratic program if and only if there exists $v^\prime \in \mathbb{R}^m$ such that:

$$u^\prime_i + A_i x_i^\prime + y_i^\prime - b_i = v^\prime_i \sum_i y_i^\prime = 0$$

We can easily solve the linear system in $\mathbb{R}^m$ to get $v^\prime_t = \frac{1}{p} \sum_i u_i^\prime + \frac{1}{p} r^\prime$ where $r^\prime = \sum_i (A_i x_i^\prime - b_i)$ is the residual of the coupling constraint, and substitute above to get $y^\prime$. Finally, the update of $u^\prime$ will be simply $u_i^\prime+1 = \cdots = u_p^\prime+1 = v^\prime$ (i.e. $u^\prime \in Y^\perp$).

Observe that, even if the method may be interpreted as a separable Augmented Lagrangian technique, the parameter $h$ is a scaling parameter and not a penalty one, which must be estimated to drive the two primal and dual sequences at the same pace towards the optimal fixed point. Eckstein (1994) has shown the potential of that algorithm for massively distributed computing and different versions have been shown to be very efficient to solve multicommodity flow problems with linear or convex costs (Eckstein and Fukushima, 1993; Ouorou et al., 2000).

When applied to the convex multicommodity flow problem (MINDEL), the coupling constraints are not the capacity constraints as in the Dantzig-Wolfe’s method (see Section 1.2 and 1.4.3.1) as they are already included in the delay function. Indeed, $\Phi(x)$ is finite whenever $x_e < u_e, \forall e$. We can apply the decomposition scheme to the multicommodity constraints

$$x_e - \sum_K x_{e_k} = 0, e \in E$$

(1.4)

to split the problem into $K + |E|$ subproblems, i.e. one quadratic subproblem for each commodity and one convex one-dimensional subproblem for each arc. It is shown in Ouorou et al. (2000) how quadratic flow problems can be avoided by including the $K$ individual demand satisfaction constraints

$$\sum_p X_p = d_k, k \in K$$

(1.5)

in the coupling constraints to yield a completely distributed decomposition algorithm. Moreover, the arc-path formulation is used to induce the generation of supporting paths by successive shortest-paths calculations. These calculations may be performed by sweeping all possible destinations for a given origin, avoiding the explosion of the computational time when the requirement matrix is dense.

Numerical experiments with algorithm (PDM) have been performed by Ouorou et al. (2000) on medium-size networks but with dense requirement matrices (100 nodes, 900 arcs and 10,000 commodities). We limit ourselves, in Table 1.3, to only the comparisons made by the authors, who also illustrated the performance of ACCPM and a Projected Newton algorithm) on the Flow Deviation method versus the Proximal Decomposition algorithm. For a given tolerance on the final residuals of $10^{-4}$, (PDM) performs better than (FD) on standard traffic demand matrices (where “scale” represents a load factor $Z$ – the one used in model (MAXCONCUR), which varies from 1 to 2.5) and the average path dispersion is lower which makes its implementation on real routers much easier.

1.4 SURVIVABILITY ISSUES

Network survivability has always been a major preoccupation of telecommunication operating companies. The explosion of new services is attracting a new, more demanding customer. The intense competition between operating companies, due to market deregulation, demands that an ever increasing attention be given to the quality and cost of services.

The main concern of customers include the availability of network connections, which is the key component of quality based network services. Telecommunication companies are faced
Table 1.3  Comparison of Proximal Decomposition with Flow Deviation

<table>
<thead>
<tr>
<th>Scale</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>33.502997</td>
<td>52.292984</td>
<td>72.696367</td>
<td>94.984274</td>
<td>119.474206</td>
</tr>
<tr>
<td>CPU time (sec) PDM</td>
<td>76.93</td>
<td>83.88</td>
<td>156.49</td>
<td>161.54</td>
<td>712.04</td>
</tr>
<tr>
<td>Numb. of iter. PDM</td>
<td>24</td>
<td>36</td>
<td>84</td>
<td>192</td>
<td>494</td>
</tr>
<tr>
<td>Path dispersion for PDM</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>CPU time (sec) FD</td>
<td>355.8</td>
<td>362.7</td>
<td>359.4</td>
<td>347.6</td>
<td>361.3</td>
</tr>
<tr>
<td>Numb. of iter. FD</td>
<td>219</td>
<td>223</td>
<td>221</td>
<td>213</td>
<td>222</td>
</tr>
</tbody>
</table>

with the challenge of reducing the risk of an unpredictable failure while at the same time maximizing network efficiency.

Failures however cannot be entirely eliminated, whether it be a question of equipment (switching, transmission, . . .) at a network node, or a physical break in a connecting link. In the event of such failures, the network operator should have planned an excess capacity in his network in order to reroute all demands which were being served by the failed network components. To ensure that sufficient capacity exists for any possible degree of failure, the current practice is to install an optimized spare capacity.

Basic network failures are of two types: arc and node failures. The former affects the traffic transiting by the failed arc while the latter affects all adjacent arcs traffic. The fundamental hypothesis in network survivability problems is that only one basic failure can be handled at once.

There exists two approaches to modelling and rerouting interrupted traffic demands: link restoration and path restoration. Link restoration considers that the rupture of a link creates, at its endpoints, a demand equal to the total flow which transited through the link. This demand must be rerouted by using the spare capacity. It can be described as a single commodity requirement. In the path restoration, the interrupted flow is analyzed and the fraction of demands in the nominal network affected by the failure is found. In this way, a set of demands is generated to be routed using the spare capacity. The demand requirement may now be described as one of multiple commodities. The path restoration, although more difficult to put into effect, is economically preferable. It is the approach used in practice and the one we will study here.

The survivability problem consists in determining the excess capacities to be installed in the network and the required rerouting of demands in order to satisfy all demands and to minimize the total investment cost.

Under the hypotheses of linear capacity installation costs and divisible flows, the survivability problem can be formulated as a linear special case of MCNF problems using either flow formulation or path formulation.
1.4.1 Survivability model using flow formulation

In this section, we consider the global survivability model as we optimize network capacity for both routing and rerouting requirements. We do not consider any reuse of capacity released by any failure. Denote by $E_f \subseteq E$ the subset of edges with positive flows that can fail.

Suppose edge $e \in E_f$ fails. Let $(x^k_e)^+$ and $(x^k_e)^-$ denote the flows on edge $(i, j)$ used to restore the demands affected by this failure, with corresponding vectors $x^k_+$ and $x^k_-$. By definition, $(x^k_e)^+ = (x^k_e)^- = 0$, i.e. when an edge fails both its working and spare capacities are unusable.

Therefore, at most $K$ flows routed through edge $(i, j)$ will be interrupted. Let $K_e$ be the subset of such commodities. Let $y \in \mathbb{R}^{|E|}$ be the capacity to be installed and $q_{ij} > 0, \forall (i, j) \in E$ the installation unit cost. We can formulate the survivability model as follows:

$$
\begin{align*}
\min & \quad \sum_{(i,j) \in E} q_{ij}(\sum_{k=1}^{K} ((x^k_{ij})^+ + (x^k_{ij})^-) + y_{ij}) \\
\text{s.t.} & \quad N x^k_+ - N x^k_- = d^k, \quad \forall k \in K \quad (1.6a) \\
& \quad \sum_{e \in E_f} \sum_{k \in K_e} ((x^k_e)^+ + (x^k_e)^-) \leq y, \quad (1.6b) \\
& \quad N x^k_+ - N x^k_- = r^{ek}, \quad \forall k \in K_e, \forall e \in E_f \quad (1.6c) \\
& \quad (x^k_e)^+ = (x^k_e)^-, \quad \forall k \in K_e, \forall e \in E_f, \quad (1.6d) \\
& \quad x^k_+, \ x^k_- \geq 0, \quad \forall k \in K, \quad (1.6e) \\
& \quad y_{ij} \geq 0, \quad \forall (i, j) \in E, \quad (1.6f)
\end{align*}
$$

where

- $N$ is the $n \times m$-node-arc incidence matrix of the network as in Section 1.2.

- $r^{ek} = \sum_{k \in K} x^k_{ij}, \quad \forall (i, j) = e \in E_f$ is the flow interrupted by failure $e$ to be rerouted.

The objective function (1.6a) concerns both working and spare capacities which have in this case the same cost. This formulation allows us also to assign different costs for the working and spare capacities (Lisser et al., 1998). Constraints (1.6c) ensure that sufficient capacity is installed on each edge to meet rerouting requirements over all edge failures. Constraints (1.6b) and (1.6d) are flow conservation constraints for the working and restoration flows respectively. Constraints (1.6e) prevent using the capacity of a failed edge for restoration needs. Some transmission and/or switching technologies require distinct working and spare capacities and thus consider two distinct networks, i.e. working network and reserve network (Lisser et al., 2000). In this case, problem (1.6) can be adapted easily by setting different capacity constraints for the working and protection networks and updating the objective function accordingly.

The main problem studied in the literature concerns only the spare capacity planning problem. In this case, the working flow and its routing are given. The problem consists in finding the minimum amount of spare capacity so that the network can survive the failure of an edge. This
The path formulation of problem (1.7) was studied by Kennington and Lewis (2001) while the integer path restoration variant of (1.7) was studied by Balakrishnan et al. (2001).

1.4.2 Survivability model using path formulation

We consider the path formulation of problem (1.7). Each commodity \( k \in K \) is routed on a set of working paths \( P^k \) between the source and the destination of the commodity. Each path \( p \) is characterized by a Boolean vector \( \pi_p \) on the set of edges with \( (\pi_p)_{ij} = 1 \) if path \( p \) intersects edge \( (i,j) \in E \), and 0 otherwise. To each path \( p \) in the set \( P^k \), we shall assign a variable \( X_p \) to represent the flow component of commodity \( k \) routed on \( p \). The flow of demand \( k \) is split on different routes: its total value \( \sum_{p \in P^k} X_p \), must equal the demand \( d_k \).

We shall additionally consider the set of restoration paths \( P^{ek}_k \) for the demand \( k \) when \( e \in E_f \) fails. Each restoration path \( p \) is characterized by a Boolean vector \( \pi_p^e \) on the set of edges with \( (\pi_p^e)_{ij} = 1 \) if path \( p \) intersects arc \( (i,j) \in E \), and 0 otherwise. If edge \( e \) fails, then at most \( K \) commodities will be interrupted on all paths that contain \( e \). A variable \( X_p^e \) is introduced for each path for the flow component.

The problem can be written as

\[
\begin{align*}
\text{min} & \quad \sum_{(i,j) \in E} q_{ij} y_{ij} \\
\text{s.t.} & \quad \sum_{k \in K} \sum_{p \in P^k} X_p (\pi_p)_{ij} + y_{ij} \\
& \quad \sum_{k \in K} \sum_{p \in P^k} X_p^e (\pi_p^e)_{ij} \leq y_{ij}, \quad \forall (i,j) \in E, \forall e \in E_f, \\
& \quad \sum_{p \in P^k} X_p^e = r^e_k, \quad \forall k \in K, \forall e \in E_f, \\
& \quad X_p \geq 0, \quad \forall p \in P^k, \forall k \in K, \forall e \in E_f, \\
& \quad X_p^e \geq 0, \quad \forall p \in P^{ek}, \forall k \in K, \forall e \in E_f, \\
& \quad y_{ij} \geq 0, \quad \forall (i,j) \in E.
\end{align*}
\]

where \( r^e_k = \sum_{p \in P^{ek}} X_p (\pi_p^e) \), \( \forall k \in K, \forall e \in E_f \).

Problem (1.8) models the global survivability problem by using the path formulation. It is based on the path restoration. Constraints (1.8b) ensure that sufficient capacity is installed on
each edge to meet rerouting requirements over all edge failures. Constraints (1.8a) and (1.8c) ensure that total flow routed or rerouted on several paths correspond to total flow of given commodity. As for the flow formulation, we can deduce the spare capacity planning from (1.8) as follows:

\[
\begin{align*}
\min & \quad \sum_{(i,j) \in E} q_{ij} y_{ij} \\
& \quad \sum_{k \in K_p \in P_k} \sum_{e \in E_f} X_{e}^{k} (\pi_{e}^{k})_{ij} \leq y_{ij}, \quad \forall (i, j) \in E, \forall e \in E_f, \\
& \quad \sum_{p \in P_k} X_{e}^{p} = r^{k}, \quad \forall k \in K, \forall e \in E_f, \\
& \quad X_{e}^{p} \geq 0, \quad \forall p \in P^{k}, \forall k \in K, \forall e \in E_f, \\
& \quad y_{ij} \geq 0, \quad \forall (i, j) \in E.
\end{align*}
\] (1.9a, 1.9b, 1.9c, 1.9d)

This spare capacity planning problem is based on the path formulation. As for (1.7), we can consider link restoration by considering \( r^{k} \) as a single commodity and reroute it as a whole, see Lisser et al. (1998) for more details.

Whatever the problem considered, routing or rerouting, MCNF and its extensions are so large scale linear programs, even for networks of moderate dimensions, that they challenge the capabilities of the most advanced commercial LP codes. The alternative is to turn to the principle of decomposition whose effect is to break down the huge initial problem into interconnected problems of much smaller dimensions.

1.4.3 Decomposition approaches for treating survivability models

There are at least two ways of implementing decomposition. The first one consists in separating the capacity design issue (the master program) and the nonsimultaneous multicommodity flow requirements (the subproblems). The master program selects a tentative capacity design. The subproblems test whether this proposal meets the multicommodity flow requirements: If the proposal is not feasible, the subproblems return a cut, or constraint, to the master program. The merit of this first approach is that the master program is of moderate size. The difficulty lies in the subproblems, i.e., the constraint generation scheme, that requires the solution of a nonsmooth optimization problem, see Minoux (1986). This approach is usually named Benders decomposition (Benders, 1962).

The other approach uses Lagrangian relaxation. An extensive formulation of the problem includes capacity constraints on each arc flow (one per arc and per failure configuration) and flow constraints (one set per commodity and per failure configuration). The idea is to dualize the capacity constraints and construct a Lagrangian in the space of the corresponding dual variables. The master program consists in maximizing the Lagrangian in the dual variables. The subproblems tests whether for a given set of dual variables there exists more profitable reroutings of the commodities. The information that is sent back to the master takes the form of a column generation scheme. In this decomposition mode, the subproblems are very simple: they are just shortest path problems. In contrast with Benders decomposition the master program can be very large. However this program is sparse and structured. It can be solved using appropriate techniques for exploiting sparsity.

As an illustration of those two decomposition principles, we apply Lagrangian relaxation and Benders decomposition techniques to the spare capacity planning problem (1.9). A positive upper bound \( \bar{y}_{ij} \) is set on the edge capacity variable \( y_{ij}, \forall (i, j) \in E. \)

In Lisser et al. (2000; 1998), both Lagrangian relaxation and Benders decomposition are applied to problem (1.7) and also to problem (1.6) based on link restoration. The path restora-
tion global survivability problem (1.8) is studied in Bonnans et al. (2000) using the proximal decomposition method described in Section 1.3.3.

1.4.3.1 Lagrangian Relaxation. Decomposition methods consist in converting the problem (1.9) into a smaller nondifferentiable problem in convex optimization. This is achieved by partial dualization. In this section, we apply Lagrangian relaxation to problem (1.9) and show how to construct the elements of the subdifferential of the function thus obtained.

Consider the Lagrangian obtained by dualizing the coupling constraints (1.9a). The dual vector associated with (1.9a) is \( v_i^e, \forall (i, j) \in E, \forall e \in E_f \). The partial Lagrangian function is

\[
L(X, y; v) = \sum_{(i, j) \in E} c_{ij} y_{ij} + \sum_{e \in E_f} \sum_{(i, j) \in E} v_i^e \left( \sum_{k \in K_e} \sum_{p \in P_{sk}} X_p^e (\pi_p^e)_{ij} - y_{ij} \right).
\]

Problem (1.9) can be formulated as the minmax problem

\[
\min_{X \geq 0, y \geq 0} \{ \max_{v \geq 0} L(X, y; v) : \sum_{p \in P_{sk}} X_p^e = r^e, \; \forall k \in K_e, \; \forall e \in E_f \}.
\]

By the minmax theorem, this problem has the same optimal value as

\[
\max_{v \geq 0} L(v).
\]

where

\[
L(v) = \min_{X \geq 0, y \geq 0} \{ L(X, y; v) : \sum_{p \in P_{sk}} X_p^e = r^e, \; \forall k \in K_e, \; \forall e \in E_f \}.
\]

This problem is convex, nondifferentiable, but of small size. To compute \( L(v) \), we disaggregate it into a sum of elementary functions

\[
L(v) = \gamma(v) + \varphi(v).
\]

where

\[
\gamma(v) = \min_X \sum_{e \in E_f} \sum_{k \in K_e} \sum_{p \in P_{sk}} (\pi_p^e)_{ij} \sum_{(i, j) \in E} v_i^e X_p^e,
\]

and

\[
\varphi(v) = \min_{0 \leq y \leq S_{ij}} \{ \sum_{(i, j) \in E} c_{ij} y_{ij} - \sum_{e \in E_f} \sum_{(i, j) \in E} v_i^e y_{ij} \}.
\]

To decompose the function \( \varphi(v) \), we extend the vector \( v \) with arbitrary components \( v_i^e \) for each \( (i, j) \neq e \in E_f \) and introduce some new notations: \( \delta_{ij}^e = 1 \) if \( (i, j) = e \in E_f \) and 0 otherwise.

Functions \( \gamma \) and \( \varphi \) can be decomposed into simple functions

\[
\gamma_k(v) = \min_{y_k} \sum_{(i, j) \in E} v_i^e X_p^e (\pi_p^e)_{ij}, \forall k \in K_e, \forall e \in E_f.
\]

and

\[
\varphi_{ij}(v) = \min_{0 \leq y \leq S_{ij}} y_{ij} (c_{ij} - \sum_{e \in E_f} \delta_{ij}^e v_i^e).
\]

Thus we have

\[
\gamma(v) = \sum_{e \in E_f} \sum_{k \in K_e} \gamma_k(v),
\]

and

\[
\varphi(v) = \sum_{(i, j) \in E} \varphi_{ij}(v).
\]
The elementary functions \( \gamma_k(v) \) and \( q_{ij}(v) \) each have the form of a minimum of functions linear in \( v \). Hence, they are concave. In addition, their values at a given point \( v \) are easily calculated. Functions \( \gamma_k \) are the optimal values of simple flow problems corresponding to the shortest path problem over a graph with nonnegative costs. Functions \( q_{ij} \) take either the value zero or \( (c_{ij} - \sum_{e \in E} \delta c_{ij}^{(e)} \tilde{y}_{ij}) \) according to the sign of \( (c_{ij} - \sum_{e \in E} \delta c_{ij}^{(e)} \tilde{y}_{ij}) \). Thus \( \tilde{y}_{ij} \) essentially takes only two values, 0 and \( \tilde{y}_{ij} \).

To determine an element of the sub-differential problem, it suffices to express the dependence of the optimal values of \( \gamma \) and \( q \) on \( v \). Since this involves only simple linear expressions, the components of the sub-gradients are just the coefficients of \( v \).

To be more precise, let \( \tilde{v} \) be a point at which we calculate \( \gamma_k(\tilde{v}) \) and \( q_{ij}(\tilde{v}) \), and let \( \hat{X}_P^e \) and \( \hat{y}_{ij} \) be the values respectively where these functions attain their minima. Let \( v \) be an arbitrary point. Using the fact that, for a given \( v \), \( \gamma_k \) is the minimum of \( \sum_{(i,j) \in E} v_{ij}^e X_p^e(\pi_p^e)_{ij} \) for \( X_p^e \in P^k \), we obtain for \( k \in P^k \),

\[
\gamma_k(v) \leq \sum_{(i,j) \in E} v_{ij}^e \hat{X}_P^e(\pi_P^e)_{ij} = \gamma_k(\tilde{v}) + \sum_{(i,j) \in E} \hat{X}_P^e(\pi_P^e)_{ij} (v_{ij}^e - \hat{v}_{ij}^e). \tag{1.21}
\]

Inequality (1.21) defines a support of the concave function \( \gamma_k, k \in K_e \) at the point \( \tilde{v} \). The coefficients \( (v_{ij}^e - \hat{v}_{ij}^e) \) of \( \hat{X}_P^e \) are the components of the sub-differential of the \( \gamma_k \).

In the same way, we construct the sub-differential of \( q_{ij} \) by the inequality

\[
q_{ij}(v) \leq \hat{y}_{ij} \left( c_{ij} - \sum_{e \in E} \delta c_{ij}^{(e)} \right) = q_{ij}(\tilde{v}) - \sum_{e \in E} \delta \hat{y}_{ij} (v_{ij}^e - \hat{v}_{ij}^e). \tag{1.22}
\]

Inequality (1.22) defines a support of the concave function \( q_{ij} \) at the point \( \tilde{v} \). The sub-gradient is therefore a null vector, excepting those components corresponding to edges in the failure network, which have value \( \tilde{y}_{ij} \).

1.4.3.2 Benders Decomposition. The formulation of the survivability problem (1.9) suggests the following remark: an instantiation, \( y \), of the design variables gives rise to elementary nonsimultaneous multicommodity flow problems. We apply Benders decomposition on the variables \( y \). Since there is no cost associated with the routings, the optimal value of the routing problems is either 0 or +\( \infty \) depending on whether the point \( y \) belongs to the domain \( D \) defined by the set of capacities \( y \in \mathbb{R}^{|E|} \). It is described by linear inequalities associated with the following degenerate linear program

\[
\text{min } 0
\]

\[
\sum_{k \in K_e} \sum_{p \in P^k} X_p^e(\pi_p^e)_{ij} \leq y_{ij}, \forall (i,j) \in E, \tag{1.23a}
\]

\[
\sum_{p \in P^k} v_{ij}^e = v_{ij}, \quad \forall k \in K_e, \tag{1.23b}
\]

\[
X_p^e \geq 0, \quad \forall p \in P^k, \forall k \in K_e, \tag{1.23c}
\]

\[
y_{ij} \geq 0, \quad \forall (i,j) \in E. \tag{1.23d}
\]
Let us consider the dual of (1.23) where \( v_{ij} \) and \( u_k \) are the dual variables associated with the constraints (1.23a) and (1.23b) respectively. This problem is written as follows:

\[
\begin{align*}
\text{max} & \quad \sum_{(i,j) \in E} y_{ij} v_{ij} - \sum_{k \in K} r^k u_k \\
& \quad \sum_{(i,j) \in E} (\pi^*_p)_{ij} v_{ij} + u_k \leq 0, \forall p \in P^k \forall k \in K_e, \\
& \quad u_k \geq 0, \forall p \in P^k, \forall k \in K_e.
\end{align*}
\]

Problem (1.24) is homogenous and the trivial solution \((0, 0)\) is feasible. It has an optimal value which takes either 0 or \(+\infty\). By the strong duality theory, problem (1.23) has no feasible solution if and only if (1.24) is unbounded. In order to check unboundness, we solve problem (1.24) with the additional box constraints \( v_i \leq \bar{v}_i \), where \( \bar{v}_i \) is an arbitrary positive vector:

\[
\begin{align*}
\text{max} & \quad \sum_{(i,j) \in E} y_{ij} v_{ij} - \sum_{k \in K} r^k u_k \\
& \quad \sum_{(i,j) \in E} (\pi^*_p)_{ij} v_{ij} + u_k \leq 0, \forall p \in P^k \forall k \in K_e, \\
& \quad 0 \leq v_{ij} \leq \bar{v}_{ij}, \forall (i,j) \in E.
\end{align*}
\]

1.5 CONCLUSION

Multicommodity flow formulations are very common in network design and routing problems. Unfortunately, they are much more difficult to solve than single commodity flow problems even in the continuous case. Their large scale combined with the underlying structure induces different decomposition schemes and they have greatly contributed to the development of that area since the pioneer works of Dantzig, Wolfe, and Benders. We have seen how path generation is a crucial issue to get efficient algorithms. This is the reason why the Flow Deviation algorithm has been very popular to solve practical flow and capacity assignment problems with congestion costs. More recently, the Proximal Decomposition method has been shown to be very competitive to solve convex cost multicommodity flow problems as it exploits the monotropic structure of the network model and allows path generation with fewer supporting paths than Flow Deviation.

Network survivability is a key component of the quality of based network services. We have given an illustration of network design problems with survivability constraints by extending
different MCNF formulations. We have applied two different decomposition approaches to the spare capacity problem. Finally, survivability models are of higher degree of magnitude due to their huge size. However, they represent new exciting challenges for the mathematical programming community.
Bibliography


