Proximal Decomposition Method for solving Global Survivability in Telecommunication Network

J. Frederic Bonnans, Mounir Haddou, Abdel Lisser, Raja Rébaï

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Proximal Decomposition Method for solving Global Survivability in Telecommunication Network

J. Frédéric Bonnans , Mounir Haddou , Abdel Lisser, Raja Rébaï

N ° 4055

Novembre 2000

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4055--FR+ENG

Proximal Decomposition Method for solving Global Survivability in Telecommunication Network

J. Frédéric Bonnans* , Mounir Haddou† , Abdel Lisser‡, Raja Rébaï§

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet MOCOA

Rapport de recherche n°4055 — Novembre 2000 — 23 pages

Abstract: In this paper, we are concerned with the global survivability in telecommunication networks. The objective consists in finding the optimal routing and the least cost investment in base and reserve capacities. The routing and the base capacity insure nominal traffic and the reserve capacity guarantees survivability of the traffic against any arc failure (using a global rerouting strategy).
In our model we consider that routings and capacities can be fractional. So the Global Survivability Problem (GSP) can be formulated as a large-scale linear program. Its special structure favours the use of decomposition algorithms.
We propose a method using columns generation and proximal decomposition techniques. The main task of this algorithm consists in solving independent quadratic subproblems.
We report some numerical results obtained by testing these algorithms with data from the France-Telecom Paris district transmission network.

Key-words: Proximal methods, network survivability, large-scale linear programming, decomposition, multicommodity network flow models.

(Résumé : tsvp)

* INRIA, B.P. 105, 78153 Rocquencourt, France. Email: Frederic.Bonnans@inria.fr.
† UMR 6628 MAPMO, Université d'Orléans, France. Email: haddou@labomath.univ-orleans.fr
‡ France Télécom R & D, 38-40 Rue du Général Leclerc, 92794 Issy Les Moulineaux Cedex 9, France. Email: abdel.lisser@francetelecom.fr.
§ France Télécom R & D, 38-40 Rue du Général Leclerc, 92794 Issy Les Moulineaux Cedex 9, France. Email: Raja.Rebai@francetelecom.fr.

Unit´e de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
T´el´ephone : 01 39 63 55 11 - International : +33 1 39 63 55 11
T´el´ecopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Sécurisation globale de réseaux de télécommunications par une méthode de décomposition proximale

Résumé : Cet article discute le problème de sécurisation globale dans un réseau de télécommunication. L'objectif est d'obtenir le routage optimal et l'investissement de moindre coût pour les capacités des réseaux nominal et de réserve. Ces capacités permettent l'acheminement du trafic et garantissent sa survie en cas de panne simple d'arcs (avec une stratégie de reroutage global).
Notre modèle suppose que les capacités et routages peuvent être fractionnés. Le problème de sécurisation globale (PSG) peut alors être formulé comme un programme linéaire de très grande taille. Cette structure particulière facilite l'emploi d'algorithmes de décomposition.
Nous proposons une méthode basée sur la génération de colonnes et les techniques de décomposition proximale. L'effort principal de l'algorithme porte sur la résolution de programmes quadratiques indépendants. Nous présentons des résultats numériques obtenus en testant ces algorithmes avec des données réelles fournies par France Télécom R & D.

Mots-clé : Méthodes proximales, sécurisation de réseaux, programmation linéaire de grande taille, décomposition, modèles multiflots.

AMS subject classifications 90C08, 90C05, 90C06.

Référence France Telecom de ce document : NT/FTR&D/7074

1 Introduction

Survivability is one of the most important concern in telecommunication industry nowadays. Telecommunication networks are complex transmission systems made up of nodes and links. These systems are devoted to transmit informations between pairs of nodes. In case of failure of network components, the interrupted traffic must be quickly restored. The network is said to be survivable if it remains "operational" even when failures occur. In [1], Alveras et al. survey the development of models for network survivability used in practice in the last years. There are several ways to design a survivable network depending on the sense of "operational" and on the defensive properties required. One important defensive property is to keep the network connected in case of failure. This concerns topological survivability, one of the first studied network design problem [29, 3]. Stoer dedicated his book [30] to this kind of survivability problem. In order to guarantee the topological survivability of a network, most of models invoke a connectivity constraints, which require a certain number of paths between every distinct pair of nodes. This concept is called k-connectivity [1]. Several authors have considered this problem: Monma and Shallcross [23], Grötschel and Monma [13] and Grötschel et al [14, 15, 12].
An other important defensive property is the resistance of routing to failures. To keep the network alive in failure situation, we distinguish three different concepts: the diversification, the reservation and the rerouting of affected demands. The concept of diversification require that for every pair origin-destination nodes, no path carries more than a certain percentage of the total traffic between the nodes. Using the reservation concept, for every pair origin-destination nodes and for every failure situation, a certain percentage of the traffic demand between the node pair can be routed. The third concept consider the rerouting of the affected demands only. Survivability model rerouting the affected demands is according to [1], the best compromise between cost and maintenance effort. Minoux was among the first authors to consider survivability in communication networks using the latter concept with continuous capacities in [22]. Dahl and Stoer [4, 5] were the first to study survivability with diversification and reservation with discrete capacities. Lisser, Sarkissian and Vial [18] develop two survivability models where the interrupted traffic is rerouted in a separate network, called reserve network. Their models are very close to our. In our model a telecommunication network is survivable if, following an arc failure, the interrupted traffic can be redirected through the network via existing excess capacity. A network failure is said to be basic if it affects a single arc or a single node. We do not consider simultaneous multiple basic failures. We are interested only in arc basic failures. Henceforth, each failure is associated to an arc. A failure is active if it disturbs routing. In our model, we consider that the network topology is given and fixed. We assume that the former is at least 2-connected. An active failure creates new demands. In order to satisfy these demands, we consider a

second network, called reserve network, with the same topological support than the basic one. The nominal network will be only devoted to the nominal traffic. In case of failure, the interrupted traffic must be rerouted in the reserve network going round the failed arc.
We distinguish two rerouting approachs, local and global. The local approach considers that the failure of an arc creates, at its endpoints, a demand equal to the total flow which transited through the arc. Then, each failure creates a single commodity requirement. The global approach recognizes the different fractions of demands affected by the failure. Each fraction must be rerouted in the reserve network between its initial origin-destination nodes. Consequently, a link rupture creates, in the case of global rerouting, a multiple commodity requirement. Even if the global approach is more difficult to be brought into play, it is economically preferable than the local one.
In our model we consider that routings and capacities can be fractional. In spite of these simplifications, the survivability problem is still difficult since it is a large scale linear programming problem.
The GSP amounts to finding nominal and reserve capacities and a multiflow vector such that:

- the multiflow satisfies all the demands,
- the multiflow is compatible with nominal network capacity,
- the multiflow can be secured by global rerouting against any arc failure,
- and the total investment cost is minimized.

GSP can be formulated as a large-scale linear programming problem with several coupling levels.
The model described above has been developed according to France Telecom networks topologies.In [19], Lisser, Sarkissian and Vial propose a method for finding the least investment cost on the reserve network to secure a given traffic using the global strategy for rerouting. The same authors consider in [18] the more involved problem of simultaneously designing the nominal traffic and the reserve capacity investment. Their model is similar to ours except that they use the local rerouting strategy. In [18] Lisser, Sarkissian and Vial formulate this problem as a large scale linear program and solve it with cutting plane algorithm based on the concept of the analytic center (ACCPM) [9]. They first propose in [19] a method for finding the least investment cost on the reserve network to secure a given traffic using the global strategy fo rerouting. The method allow them to secure with a local approach networks up to 60 nodes and 120 arcs with an accuracy of 7.10^{-5}.
Our approach is different and none of the coupling levels of GSP is omitted. The paper is organized as follows. In section 2 we give a mathematical formulation of GSP. We focus on arc-path formulation and give a decomposition approach in section 3. In section 4 we give a short presentation of the proximal decomposition method and specialize this method for solving GSP. Section 5 will be devoted to path generation method. In section 6 we present optimality guarantee. Finally, we report some numerical results.

2 The global survivability problem

GSP is characterized by a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ and a set $\mathcal{K}$ of K commodities. The graph $\mathcal{G}$ is 2-connected and defined by $\mathcal{V}$, a set of p nodes, and $\mathcal{E}$, a set of n arcs. This graph represents the topological support of the nominal and reserve networks. Capacities are maps from $\mathcal{E}$ into $\mathbb{R}$ and are represented as an $\mathbb{R}^n$ vector.
We designate by y a nominal capacity vector and by z the reserve one.
A commodity k in $\mathcal{K}$ is defined by an origin-destination pair (o^k,p^k) and a demand value, $r^k \in \mathbb{R}$.
We associate to $k \in \mathcal{K}$ the sets $I^{0,k}, I^{1,k}, \cdots, I^{n,k}$ defined by:

- $I^{0,k}$: set of distinct elementary nominal paths between o^k and p^k.
- $I^{e,k}$: set of distinct elementary reserve paths between o^k and p^k in case of failure of the e^{th} arc.

For $j \in \{0,\cdots,n\}$, the set $I^{j,k}$ is characterized by a $n \times n^{j,k}$ matrix $\Pi^{j,k}$, where $n^{j,k} = |I^{j,k}|$. Each column $\Pi_i^{j,k}$, characterizes a path from $I^{j,k}$ and satisfies:

$$(\Pi_i^{j,k})_e = \begin{cases} 1 \text{ if } e \text{ belongs to the path } i, \\ 0 \text{ else.} \end{cases} \tag{2.1}$$

We denote by $x_i^{j,k}$ the component of the flow k carried by the path i characterized by $\Pi_i^{j,k}$. The sum over i of all the flow $x_i^{0,k}$ is equal to the demand r^k:

$$\sum_{i \in I^k} x_i^{0,k} = r^k. \tag{2.2}$$

In order to simplify the GSP formulation, we set $x^{0,k} = x^k$, $\Pi^{0,k} = \Pi^k$, $I^{0,k} = I^k$ and $n^{0,k} = n^k$ for all $k \in \mathcal{K}$.
A link break creates at most K commodities. We designate by $\mathcal{K}^e$ the subset of commodities affected by the failure of the e^{th} arc. We associate to each $k \in \mathcal{K}^e$, for $e \in \mathcal{E}$, a value $r^{e,k}$ of the fraction of the flow to be rerouted :

$$r^{e,k} = \sum_{i \in I^k} x_i^k (\Pi_i^k)_e. \tag{2.3}$$

We denote by c and d the $\mathbb{R}^n$ vectors cost per unit low of respectively the nominal and the reserve networks. GSP can be formulated as follows :

$$
\left\{
\begin{array}{l}
\underset{x^1,\cdots,x^K,x^{1,1},\cdots,x^{n,K},z}{\text{Min}} \; c^T \sum_{k=1}^{K} \Pi^k x^k + d^T z \\
\\
k \in \mathcal{K} \left\{ \begin{array}{ll} \text{(i.k)}\ \omega^k x^k = r^k, & (\alpha^k) \\ \text{(ii.k)}\ x^k \geq 0, & (s^k) \end{array} \right. \\
\\
e \in E \left\{
\begin{array}{l}
\text{(iii.e)}\ x^{e,0} + \sum_{k=1}^{K} \Pi^{e,k} x^{e,k} = \hat{\Delta}_e z, \qquad (\lambda^e) \\
\text{(iv.e)}\ x^{e,0} \geq 0, \qquad (s^{e,0}) \\
\\
k \in \mathcal{K}^e \left\{ \begin{array}{ll} \text{(v.e.k)}\ \omega^{e,k} x^{e,k} = \Pi^k_e x^k, & (\beta^k_e) \\ \text{(vi.e.k)}\ x^{e,k} \geq 0, & (s^{e,k}) \end{array} \right.
\end{array}
\right. \\
\\
\text{(vii)}\ z \geq 0, \qquad (t)
\end{array}
\right. \tag{2.4}
$$

where

- $x^{e,0}$ is the residual reserve capacity vector in case of the e^{th} failure.
- ω^k is the transpose of the $\mathbb{R}^{n^k}$ unit vector.
- $\omega^{e,k}$ is the transpose of the $\mathbb{R}^{n^{e,k}}$ unit vector.
- $\hat{\Delta}_e$ is the n order diagonal matrix defined by

$$\hat{\Delta}_e = diag(\mathbf{1} - \delta_e)$$

We also introduce the Lagrange multipliers:

- α^k: associated with demand constraint of commodity k (2.4.i.k) ($\alpha^k \leq 0$),
- s^k: dual variable of x^k ($s^k \geq 0$),
- λ^e: associated with reserve capacity constraints in the e^{th} failure configuration (2.4.iii.e) ($\lambda^e \geq 0$),

- $s^{e,0}$: dual variable of $x^{e,0}$ ($s^{e,0} \geq 0$),
- β_e^k: associated to rerouting constraint of flow k in case of the e^{th} failure (2.4.v.e.k) ($\beta_e^k \leq 0$),
- $s^{e,k}$: dual variable of $x^{e,k}$ ($s^{e,k} \geq 0$),
- t: dual variable of z ($t \geq 0$).

The above formulation associates to each commodity $k \in \mathcal{K}$ the constraints for routing under normal operating state which are demand (2.4.i.k) and non-negativity (2.4.ii.k) constraints. For each failure configuration e, capacity (2.4.iii.e), rerouting (2.4.v.e.k) and non-negativity (2.4.iv.e.k) (2.4.vi.e.k) constraints must be satisfied. The capacity constraints imply that for each edge the flow through may not exceed its capacity in case of any failure operating state. (2.4.vii) corresponds to non-negativity constraints.
The size of GSP can be significant even for networks of moderate size. Let $\widehat{A}$ be the $\hat{p} \times \hat{n}$ constraint matrix, i.e., $\hat{A} =$

$$\begin{pmatrix}
\omega^1 & & & & & & & & & & & & \\
 & \ddots & & & & & & & & & & & \\
 & & \omega^K & & & & & & & & & & \\
 & & & \Pi^{1,1} & \cdots & \Pi^{1,K} & Id(n) & & & & & & -\hat{\Delta}_1 \\
\Pi_1^1 & & & \omega^{1,1} & & & & & & & & & \\
 & \ddots & & & \ddots & & & & & & & & \\
 & & \Pi_1^K & & & \omega^{1,K} & & & & & & & \\
\vdots & & & \vdots & & & & \ddots & & & & \ddots & \vdots \\
 & & & & & & & & \Pi^{n,1} & \cdots & \Pi^{n,K} & Id(n) & -\hat{\Delta}_n \\
\Pi_n^1 & & & & & & & & \omega^{n,1} & & & & \\
 & \ddots & & & & & & & & \ddots & & & \\
 & & \Pi_n^K & & & & & & & & \omega^{n,K} & &
\end{pmatrix}$$

For instance, if we have:

- $p = 20$,
- $n = 190$,
- $K = 190$,

the constraints matrix in the case of node-arc formulation has the following size:

- $\hat{p} = n + n^2 + K \times p \times (n+1)$=762.090,
- $\hat{n} = 2 \times [n + n^2 \times (K+1)]$= 13.790.580.

For arc-path formulation, we can estimate the size of $\hat{A}$:

- $\hat{p} = K + K \times n + n \times n$=72.390,
- $\hat{n} = \sum_{k=1}^{K} n^k + \sum_{e \in E^k} n^{e,k} + n(n+1) \ >> 762.090.$

The arc-path formulation associates with each elementary path a variable. The number of variables may then increase exponentially with the size of the graph. This constitutes the main shortcoming for this formulation. However, in practice, an optimal solution is carried by a relatively small number of paths. Indeed the formulation (2.4) has only $\hat{p}$ constraints. This implies that (if (2.4) has a solution) an optimal solution that has at most $\hat{p}$ strictly nonzero variables exists. Then no more than $\hat{p}$ different paths are needed to satisfy all the requested flow.
We propose to take advantage of this property by limiting GSP to a subset of elementary paths generated by an iterative process. Indeed, the path matrices $\pi^{j,k}$ are only known implicitly. However, the form of such columns is known, and they can be generated as needed during the course of the algorithm- hence the name column generation [7, 16]. The scheme of column generation algorithm requires one to solve successive linear programs of smaller size. We use proximal decomposition algorithm for solving these programs.

3 Decomposition approach

The difficulty of GSP comes from not only its large scale but also from the coupling constraints. These constraints tie together routing and rerouting variables, failures and commodities. In order to simplify GSP model, the decomposition approach must reduce the coupling levels.
We note that routing $x = (x^1, \ldots, x^k)$ and reserve capacities z are coupling variables in (2.4). We copy these variables as many as they are able to link the problem constraints. Let $z(e)$ denotes the copy of z associated with failure e. The reserve capacity constraints in the e failure configuration (2.4.iii.e) concern only the copy z(e). For routing copies the choice of constraints is less obvious. According to (2.4) $x = (x^1, \ldots, x^k)$ must meet two types of constraints: routing (2.4.i.k) and rerouting (2.4.v.e.k) constraints. We choose to copy both of these constraints for each routing copy $x(e)$, for $e \in \mathcal{E}$. Using this copy strategy, we obtain the following formulation:

$$
\left\{
\begin{array}{l}
\underset{x^1(1),\cdots,x^K(n),x^{1,0},\cdots,x^{n,K},z(1),\cdots,z(n)}{\text{Min}} \displaystyle\sum_{e=1}^{n} C^T \sum_{k=1}^{K} \Pi^k x^k(e) + \sum_{e=1}^{n} D^T \hat{\Delta}_e z(e); \\
\\
e \in E \left\{
\begin{array}{ll}
x^{e,0} + \displaystyle\sum_{k=1}^{K} \Pi^{e,k} x^{e,k} = \hat{\Delta}_e z(e), & (\lambda^e) \\
x^{e,0} \geq 0, & (s^{e,0}) \\
z(e) \geq 0, & (s_z(e)) \\
\\
k \in \mathcal{K} \left\{
\begin{array}{ll}
\omega^k x^k(e) = r^k, & (\alpha_e^k) \\
\omega^{e,k} x^{e,k} = \Pi_e^k x^k(e), & (\beta_e^k) \\
x^k(e) \geq 0, & (s^k(e)) \\
x^{e,k} \geq 0, & (s^{e,k})
\end{array}
\right.
\end{array}
\right. \\
\\
z(1) = \cdots = z(n), \quad (T) \\
x(1) = \cdots = x(n), \quad (Y)
\end{array}
\right. \tag{3.5}
$$

where

- $D_i = \dfrac{d_i}{n-1}$.
- $C_i = \dfrac{c_i}{n}$.
- for $e = 1, \cdots, n$,
  - $t(e)$ is dual variable of the e^{th} copy of reserve capacity.
  - $s^k(e)$ is dual variable of $x^k(e)$ for $k \in \mathcal{K}$.
  - α_e^k is the Lagrange multipliers associated with routing constraint satisfied by $x^k(e)$.
- $Y = (y(1), \ldots, y(n))$ is the Lagrange multipliers associated with the equality routing copies constraints.
- $T = (t(1), \ldots, t(n))$ is the Lagrange multipliers associated with the equality reserve capacity copies constraints.

The other multipliers $(\lambda^e, s^{e,0}, s^{e,k}, \beta_e^k)$ have the same role as in formulation (2.4).
The formulation (3.5) is equivalent to:

$$\begin{cases} \underset{x(1),\cdots,x(n),x^{1,0},\cdots,x^{n,K},z(1),\cdots,z(n)}{\text{Min}} \sum_{e=1}^{n} f_e(x(e),x^e,z(e)); \\ \\ z(1)=\cdots=z(n), \qquad (T) \\ x(1)=x(1)=\cdots=x(n), \qquad (Y) \end{cases} \tag{3.6}$$

where $f_e(x(e),x^e,z(e)) =$

$$\begin{cases} C^T \sum_{k=1}^{K} \Pi^k x^k(e) + D^T \hat{\Delta}_e z(e) \text{ if } \begin{cases} x^{e,0} + \sum_{k=1}^{K} \Pi^{e,k} x^{e,k} = \hat{\Delta}_e z(e), \\ x^{e,0} \geq 0, \\ z(e) \geq 0, \\ \\ k \in \mathcal{K} \begin{cases} \omega^k x^k(e) = r^k, \\ \omega^{e,k} x^{e,k} = \Pi_e^k x^k(e), \\ x^k(e) \geq 0, \\ x^{e,k} \geq 0, \end{cases} \end{cases} \\ \\ +\infty \text{ elsewhere.} \end{cases} \tag{3.7}$$

Note that the functions f_e for $e = 1,\ldots,n$ are independent. We denote by X^0 the routing copy vector $(x(1),\cdots,x(n))$. Let $Z = (z(1),\cdots,z(n))$ the $\mathbb{R}^{n\times n}$-vector of reserve capacity copies. We associated to each $e \in \{1,\cdots,n\}$ a rerouting vector $X^e = (x^{e,1},\cdots,x^{e,K},x^{e,0})$. The vector $(X^0,\cdots,X^n,Z)$ belongs to $\mathcal{H}$ defined as follows:

$$\mathcal{H} = \mathbb{R}^{(n)\times n_n} \times \mathbb{R}^{n_r+n^2} \times \mathbb{R}^{n^2},$$

where n_n is the number of total nominal paths and n_r the one of reserve paths. We refer to the feasibility sub-space associated with(3.6) by $\mathcal{A}$:

$$\mathcal{A} = \{(X,Z) \in \mathcal{H};\ x(1)=\cdots=x(n) \text{ and } z(1)=\cdots=z(n)\}.$$

Formulation (3.6) is advantageous to the use of the proximal decomposition method to be presented next.

4 Proximal decomposition method

4.1 Motivation

The method used here is based on the proximal decomposition on the graph of a maximal monotone operator. We aim to

$$\text{find } (x,y) \in \mathcal{A} \times \mathcal{A}^{\perp} \text{ such that } y \in T(x), \tag{4.8}$$

where $\mathcal{A}$ is a subspace of a finite dimensional vector space $\mathcal{H}$ and T a maximal monotone operator. The proximal decomposition method on the graph of a maximal monotone operator (PDM) was introduced by Mahey et al. in [21].

There are several ways to transform a monotropic problem into the format of (4.8). The general idea is to represent coupling effects between subsystems with a product of subspaces corresponding to copies of primal and dual variables.
The proximal decomposition algorithm is a specialized version of the partial inverse method due to Spingarn [28] for convex separable problems. The algorithm performs two distinct steps at each iteration :

1. A proximal step that regulates the objective function by adding a quadratic term depending on the previous primal-dual pair of solutions,

2. A projection step on the corresponding subspaces.

In [21], Mahey and al. show that the algorithm may be accelerated by a convenient choice of a scaling parameter. Numerical results performed on real data network confirm this behaviour in [2].
Ouorou et al. [25], present a survey of existing algorithms for convex multicommodity flow problems motivated by message routing problem. The proximal decomposition algorithm, applied to convex multicommodity problem of [25], gives good results compared with algorithm ACCPM, for which we refer to [9]. Despite the efficiency of PDM for solving multicommodity problems, it may converge slowly if the number of copies is important. One of the reason for the slow convergence lies in the repetitive solving of quadratic multicommodity subproblems.

For the routing problem as for the survivability one, paths associated with flows are not unique. Path-arc formulation needs the use of column (path) generation method. Consequently, solutions with a large number of paths are less attractive than the ones using few paths. From this point of view, PDM is well adapted to find solutions with a small number of paths. The reason why PDM uses a few paths can be explained intuitively by the update procedure that tightens to force several paths to zero. This intuition is confirmed empirically by numerical results of [25].

4.2 Generic proximal decomposition algorithm

We aim to solve problem (4.8) using the proximal decomposition algorithm.
We call proximal decomposition of $z \in \mathcal{H}$ on the graph of the maximal monotone operator T, the unique couple (u, v) such that $z = u + v$ and $v \in T(u)$.
The uniqueness of the proximal decomposition follows from the maximality of T, since we have

$$\begin{cases} u = (I + T)^{-1}(z), \\ v = (I + T^{-1})^{-1}(z). \end{cases} \tag{4.9}$$

The proximal decomposition algorithm alternates a proximal decomposition on the graph of T with a projection on $\mathcal{A} \times \mathcal{A}^{\perp}$. We state below the proximal decomposition algorithm:

Algorithm PDA

1. Let $(x^0, y^0) \in \mathcal{A} \times \mathcal{A}^{\perp}$, $\quad j := 0$,

2. $u^j := (I+T)^{-1}(x^j + y^j)$,
   $v^j := x^j + y^j - u^j$,
   If $(u^j, v^j) \in \mathcal{A} \times \mathcal{A}^{\perp}$ then **Stop**.

3. $x^{j+1} := u^j_{/\mathcal{A}}, \qquad y^{j+1} := v^j_{/\mathcal{A}^{\perp}}; \qquad j := j+1, \quad$ return to **2** .

The convergence of PDA was proved in [21] without using the concept of partial inverse.
We call the scaled proximal decomposition, with scale factor $\Lambda > 0$, of $(x, y) \in \mathcal{H} \times \mathcal{H}$ on the graph of T, the unique couple $(u, v) \in \mathcal{H} \times \mathcal{H}$ such that: $x + \Lambda y = u + \Lambda v$ and $(u, v) \in Gr(T)$. The proximal decomposition with scale factor Λ of $(x, y) \in \mathcal{H} \times \mathcal{H}$ can also be defined as a couple (u, v) such that $\begin{cases} u = (I + \Lambda T)^{-1}(x + \Lambda y), \\ v = (I + \Lambda^{-1} T^{-1})^{-1}(\Lambda^{-1} x + y). \end{cases}$
The following algorithm is a scaled and relaxed version of PDA and has the same properties and convergence results:

Algorithm PDΛ

1. Let $(x^0, y^0) \in \mathcal{A} \times \mathcal{A}^{\perp}$, $\quad j := 0$, $\Lambda > 0$, $\rho \in]0, 2[$.

2. $u^j := (I + \Lambda T)^{-1}(x^j + \Lambda y^j)$,

   $$v^j := \frac{x^j + \Lambda y^j - u^j}{\Lambda},$$

   If $(u^j, v^j) \in \mathcal{A} \times \mathcal{A}^{\perp}$ then **Stop**.

3. $x^{j+1} := \rho u^j_{/\mathcal{A}} + (1-\rho) x^j, \qquad y^{j+1} := \rho v^j_{/\mathcal{A}^{\perp}} + (1-\rho) y^j;$

   $j := j+1, \quad$ return to **2** .

Remark 1 *The speed ratio of PDΛ convergence is sensitive to the choice of scaling parameter Λ. A value of Λ may be good or bad according to problem data. The relaxation parameter seems to be inefficient and numerous experiments prove that we can choose it equal to* 1.

In the case where T is both strongly monotone with modulus α and Lipschitz with constant L, Mahey and al have proved that the optimal scaling parameter Λ is equal to L^{-1} [21]. The convergence of the sequence $\{(x^j, \Lambda y^j)\}$, generated by PDΛ is in this case linear with speed ratio: $\sqrt{1 - \dfrac{2\Lambda\alpha}{(1+\Lambda L)^2}}$.

4.3 Convex optimisation

When the operator T is the subdifferential ∂f of a lower semicontinuous proper convex function f on a Hilbert space $\mathcal{H} = \mathbb{R}^n$ with values in $]-\infty, +\infty]$, problem (4.8) is equivalent to the optimisation problem:

$$\begin{array}{l} \mathrm{Min}_x\, f(x) \\ x \in \mathcal{A} \end{array} \tag{4.10}$$

Indeed, a $\mathbb{R}^n$ vector is an optimal solution of (4.10) if and only if $0 \in \partial(\mathcal{X}_{\mathcal{A}} + f)(x)$, where $\mathcal{X}_{\mathcal{A}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{A}, \\ +\infty & \text{elsewhere.} \end{cases}$

According to theorem 23.8 in [27], $0 \in \partial\mathcal{X}_{\mathcal{A}}(x) + \partial f(x)$. Since $\partial\mathcal{X}_{\mathcal{A}}(x) = \mathcal{A}^{\perp}$, $x\ (\in \mathcal{A})$ is an optimal solution of (4.10) if it exists $y \in \mathcal{A}^{\perp}$ such that $y \in \partial f(x)$. Problem (4.10) is then equivalent to finding $x \in \mathcal{A}$ and $y \in \mathcal{A}^{\perp}$ such that $y \in \partial f(x)$.

Recall that ∂f is maximal monotone if f is a lower semicontinuous proper convex function. The proximal step (corresponding to step **2** of PDΛ algorithm) in the case of convex optimization corresponds to solving:

$$0 \in -x^j - \Lambda y^j + (I + \Lambda\partial f)u^j. \tag{4.11}$$

This step amounts to finding u^j that minimizes $f(u) + \dfrac{1}{2\Lambda}\|u - (x^j + \Lambda y^j)\|^2$.

The proximal decomposition algorithm for convex optimization problem proceeds as follows:

Algorithm PDA

1. Let $(x^0, y^0) \in \mathcal{A} \times \mathcal{A}^{\perp}$, $\quad j := 0$, $\Lambda > 0$, $\rho \in]0, 2[$.

2. $u^j = \mathrm{argmin}_u \quad f(u) + \dfrac{1}{2\Lambda}\|u - (x^j + \Lambda y^j)\|^2, \qquad v^j := \dfrac{x^j + \Lambda y^j - u^j}{\Lambda},$

   If $(u^j, v^j) \in \mathcal{A} \times \mathcal{A}^{\perp}$ then **Stop**.

3. $x^{j+1} := \rho u^j_{/\mathcal{A}} + (1-\rho)x^j, \qquad y^{j+1} := \rho v^j_{/\mathcal{A}^{\perp}} + (1-\rho)y^j; \qquad j := j+1,$ return to **2:** .

Remark 2 *Algorithm PDA converges even if the proximal step calculates approximate solution u^j with an accuracy α^j such that $\sum_j \alpha_j < +\infty$. That is to say:*

$$u^j \in \alpha_j - \mathrm{argmin}_u\, f(u) + \frac{1}{2\Lambda}\|u - (x^j + \Lambda y^j)\|^2, \text{ and } \sum_{j=1}^{+\infty} \alpha_j < +\infty.$$

If $f(u) = \sum f_i(x_i)$, then the proximal step consists in solving several independent subproblems (for more details see [24]).

4.4 Using proximal decomposition algorithm

4.4.1 Proximal step

The proximal step in case of problem (3.6) consists in finding u^j defined by

$$u^j = \underset{u}{\operatorname{argmin}} \quad \sum_{e=1}^{n} f_e(u(e), u^e, u_z(e)) + \frac{1}{2\Lambda}\|u - ((X,Z)^j + \Lambda(Y,T)^j)\|^2,$$

where f_e is given in (3.7) and the vectors $(X,Z)^j$ and $(Y,T)^j$ are the orthogonal projections of u^{j-1} on respectively $\mathcal{A}$ and $\mathcal{A}^\perp$. The algorithm solves for each failure e a quadratic problem subject to routing and rerouting constraints:

$$\left\{\begin{array}{l}
\underset{u(e),u^e}{\mathrm{Min}} \displaystyle\sum_{k=1}^{K} C^T\Pi^k u^k(e) + \frac{1}{2\Lambda}\left\|u^k(e) - x^{kj}(e) + \Lambda y^{kj}(e))\right\|^2 + \\
D^T(u^{e,0} + \sum_{k=1}^{K}\Pi^{e,k}u^{e,k}) + \dfrac{1}{2\Lambda}\left\|u^{e,0} + \displaystyle\sum_{k=1}^{K}\Pi^{e,k}u^{e,k} - (z^j(e) + \Lambda t^j(e))\right\|^2; \\
k \in \mathcal{K}^e \left\{\begin{array}{ll} \omega^k u^k(e) = u^k, & (\alpha_e^k) \\ \omega^{e,k}u^{e,k} = \Pi_e^k u^k(e), & (\beta_e^k) \\ u^k(e) \geq 0, & \\ x^{e,k} \geq 0, & \end{array}\right. \\
k \notin \mathcal{K}^e \left\{\begin{array}{ll} \omega^k u^k(e) = r^k, & (\alpha_e^k) \\ u^k(e) \geq 0, & \\ u^{e,k} = 0, & \end{array}\right.
\end{array}\right. \tag{4.12}$$

We recall that we write the Lagrange multipliers in brackets in front of the corresponding constraints.
The first block of constraints in (4.12) concerns the flow affected by the e^{th} failure. It contains constraints of routing and rerouting associated with each affected flow.
The second block corresponds to other flows (not affected by the e^{th} failure). A such flow must satisfy routing and non-negativity constraints. The rest of constraints of this block ($u^{e,k} = 0$) comes from the fact that no path associated with those flows contains the arc e ($\Pi_e^k = 0$).
In practice, we do not consider the second block because its variables are not coupling.
We propose to solve n independent sub-problems at the proximal step (4.12) by a reduced gradient method [20].
We refer by $A^{e,K}$ the two line constraint matrix associated with $k \in \mathcal{K}^e$ in the e^{th} failure configuration.
We have:

$$A^{e,k} = \begin{pmatrix} \omega^k & O(1, n^{e,k}) \\ -\Pi_e^k & \omega^{e,k} \end{pmatrix} \tag{4.13}$$

For $k \in \mathcal{K}^e$, $e \in \mathcal{E}$, the basic matrices $B^{e,k}$ associated with $A^{e,k}$ are defined by:

- if a given basic variable corresponds to rerouting variable, two basic matrices are possible:
  - $B^{e,k} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, if the routing basic variable is associated to a nominal path that does not contain the arc e.
  - $B^{e,k} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$, else.
- if the two basic variables correspond to routing variables, necessary one and only one of them is associated with a path containing arc e, and hence, $B^{e,k} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$.

Denote by $L(e)$ and $Q(e)$, respectively, the linear and the quadratic parts of the objective function (4.12) associated with copy e. Their expression is

- $L(e) = \sum_{k=1}^{K} \left[C^T \Pi^k - \frac{1}{\Lambda}(x^k(e) + \Lambda y^k(e)) \right] u^k(e) +$
  $\sum_{k=1}^{K} \left[D^T \Pi^{e,k} - \frac{1}{\Lambda}(z(e) + \Lambda t(e)) \Pi^{e,k} \right] u^{e,k} +$
  $\left[D^T - \frac{1}{\Lambda}(z(e) + \Lambda t(e)) \right] u^{e,0}.$
- $Q(e) = \frac{1}{2}(u(e), u^e)^T H(u(e), u^e)$, where H is defined by

$$H = \frac{1}{\Lambda} \begin{pmatrix} Id(n_n, n_n)) & O(n_n, n^e) & O(n_n, n) \\ O(n^e, n_n) & L(n^e, n^e) & \delta(n^e, n) \\ O(n, n_n) & \delta(n^e, n)^T & Id(n) \end{pmatrix}, \tag{4.14}$$

with

- $n^e = \sum_{k=1}^{K} n^{e,k}$,
- $n_n = \sum_{k=1}^{K} n^k$,
- $O(i, i')$ and $Id(i, i')$ are respectively $i \times i'$ null and identity matrix.

4.4.2 Projection step

The projection step boils down to computation of mean value of components of a vector. At the j^{th} projection step, we obtain for each $e \in \{1, \cdots, n\}$:

$$\begin{cases} x^{j+1}(e) = \dfrac{\sum_{e=1}^{n} u^j(e)}{n}, \\ z^{j+1}(e) = \dfrac{\sum_{e=1}^{n} u_z^j(e)}{n-1}, \end{cases} \tag{4.15}$$

5 Path generation

The master program consists in solving a global survivability problem limited to nominal and reserve path sub-sets. In order to generate this new paths, the algorithm calls at each major iteration oracles. Each oracle is connected to a flow k and computes a shortest nominal path and shortest reserve paths associated with failures affecting k. The costs per unit of flow used by the oracle are given by the dual solution of the master program. We use Dijkstra algorithm [10] for solving those shortest path problems . To each pair commodity-outer iteration (k, j), we associate the following sets:

- I_j^k: sub-set of I^k on which the j^{th} master program limits the routing of commodity k.
- $I_j^{e,k}$: sub-set of $I^{e,k}$ on which the j^{th} master program limits the rerouting of commodity k in the e failure configuration.

At each outer iteration, the algorithm starts by solving the master program and gives a solution $(x_j, x_j^1, \cdots, x_j^n, z_j)$. The components of this solution which are associated to paths belonging to I^k/I_j^k or $I^{k,e}/I_j^{k,e}$ are, by construction, set to 0.
$(x_j, x_j^1, \cdots, x_j^n, z_j)$ is solution of GSP (2.4) means that there exist Lagrange multipliers $(\alpha^{1\star}, \cdots, \alpha^{K\star}, \beta^{1\star}, \cdots, \beta^{K\star}, \lambda^{1\star}, \cdots, \lambda^{n\star})$ such that optimality system associated with GSP is confirmed.
That is to say, $(x_j, x_j^1, \cdots, x_j^n, z_j)$ is optimal if and only if we have, for each commodity k:

$$\begin{cases} (i)\ \Pi_i^{k\,T}(c - \beta^k) + \alpha^k \geq 0,\ i \in I^k, \\ e \in \mathcal{E} \begin{cases} (ii.1)\ \beta_1^k + \Pi_i^{1,k\,T}\lambda^1 \geq 0,\ i \in I^{1,k}, \\ \vdots \\ (ii.e)\ \beta_e^k + \Pi_i^{e,k\,T}\lambda^e \geq 0,\ i \in I^{e,k}, \\ \vdots \\ (ii.n)\ \beta_n^k + \Pi_i^{n,k\,T}\lambda^n \geq 0,\ i \in I^{n,k}, \end{cases} \end{cases} \tag{5.16}$$

For $(\lambda^1, \cdots, \lambda^n)$, Lagrange multipliers dual associate with the master program, we have $d \geq \sum_{e=1}^{n} \hat{\Delta}_e^T \lambda^e$.

The first equation (5.16.i) is satisfied by paths belonging to I_j^k.
For each e and each commodity k, the equation (5.16.ii.e) is satisfied by paths belonging to $I_j^{e,k}$.
A necessary and sufficient condition for optimality of $(x_j, x_j^1, \cdots, x_j^n, z_j)$ is that equations (5.16.i), (5.16.ii,1),$\cdots$, (5.16.ii,n) are expanded to the rest of, respectively, sets I^k, $I^{1,k}$, $\cdots, I^{n,k}$.
We denote by:

- Π_{jk}^k a shortest nominal path between o_k and p_k calculated with $c - \lambda^k$ as unitary cost vector.
- Π_{jke}^k a shortest reserve path between o_k and p_k in case of failure e, calculated with λ^e as unitary cost vector.

We have the following classical result:

Lemma 5.1 $(x_j, x_j^1, \cdots, x_j^n, z_j)$ *is optimal for GSP* (2.4) *if for each commodity* $k \in 1, \cdots, K$ *:*

- Π_{jk}^k *satisfies* (5.16.i).
- *and for all failure* $e \in E$ Π_{jke}^k *satisfies* (5.16.ii,e).

Remark 3 *In order to put into practice the path generation principle described above, we need to have access to Lagrange multipliers values. All this multipliers are provided by the master program except* λ^e *for* $e \in E$*. We point out the fact that* λ^e *is nothing else but the dual variable associated with the residual reserve capacity vector in case of the* e^{th} *failure,* $x^{e,0}$*. By using the equivalence at optimality between the two formulations (2.4) and (4.12), we obtain:*

$$\lambda^e = D + t(e) \tag{5.17}$$

If the master program computes an approximate solution, we can also deduce the value of λ^e*:*

$$\lambda^e = D + (1/\Lambda)(u_z(e) - z(e) - \Lambda t(e)).$$

6 Optimality guarantee

Since the master program may have an approximate solution, we propose a method for computing an upper bound. At each iteration, the algorithm may compute a feasible point of GSP. It suffices to consider a feasible routing $\tilde{x} = (\tilde{x}^1, \cdots, \tilde{x}^n)$ satisfying to constraints

(2.4.i.k) and (2.4.ii.k), and then to compute the reserve capacity that assures its survivability against any arc failure:

$$\left\{\begin{array}{l} \underset{x^{1,0},\cdots,x^{n,K},z}{\text{Min}}\ d^T z; \\ e\in\mathcal{P}\left\{\begin{array}{l} x^{e,0}+\sum_{k=1}^{K}\Pi^{e,k}x^{e,k}=\hat{\Delta}_e z, \\ \omega^{e,k}x^{e,k}=\Pi_e^k\tilde{x}^k,\ k=1,\cdots,K, \\ x^{e,k}\geq 0,\ k=0,\cdots,K, \end{array}\right. \\ z\geq 0. \end{array}\right. \tag{6.18}$$

The problem (6.18) was studied in [18, 8].
Solving this problem gives certainly a good upper bound but it is very expensive. We must make compromise between the bound quality and the computing effort to obtain it.
We propose to compute at each inner iteration a feasible point to (6.18). This point can be obtained by an algorithm inspired by PDΛ. We are concerned by solving the $|\mathcal{P}|$ following problems:

$$\left\{\begin{array}{l} \underset{x^{e,0},\cdots,x^{e,K},z(e)}{\text{Min}}\ D^T z(e)+(1/2\Lambda)\|z(e)-z^j(e)|^2; \\ x^{e,0}+\sum_{k=1}^{K}\Pi^{e,k}x^{e,k}=z(e), \\ \omega^{e,k}x^{e,k}=\Pi_e^k\tilde{x}^k,\ k=1,\cdots,K, \\ x^{e,k}\geq 0,\ k=0,\cdots,K, \\ z(e)\geq 0, \end{array}\right. \tag{6.19}$$

where

- z^j is the reserve capacity, solution of the j^{th} master problem,
- and $\tilde{x}$ is the feasible routing:

$$\tilde{x}^k = x^{k^j}(1)=\cdots=x^{k^j}(n).$$

We use the reduced gradient for solving (6.19). The scaling parameter Λ is not necessarily the same as the one of the master program.
The cost of the reserve capacity z defined by: $z_i=\max_e(z_i(e))$ for $i=1.\cdots,n$ is an upper bound of the solution of GSP (2.4).
In this section we have described the essential features of the optimality guarantee approach, for more details we refer the reader to Rébaï [26].

7 numerical results

The proximal decomposition method takes full advantage of our problem structure. In this section we aim to evaluate numerically the performance of this method. With this intention,

we propose to compare PDA with adaptation of ready-to-use interior point code called HOPDM (Higher Order Primal Dual Method) [11]. Among the advantages of using this code we mention the fast sparsity-exploiting Cholesky decomposition and the presolve routine incorporated into HOPDM. The algorithm that uses HOPDM to solve master program is refered by DIP. We specify that we have implemented PDA in FORTRAN 77. The code DIP is also entirely written in FORTFRAN 77. All the computational tests were performed on a DEC ALPHA 3000 machine. We refer by PBi, the randomly generated network with i nodes and $\frac{i\times(i-1)}{2}$ arcs and crossed by $\frac{i\times(i-1)}{2}$ flows. We report in the following some numerical results obtained with these algorithms.

7.1 Solution accuracy and sensivity to the Λ value

We consider a fully meshed network PB12 with 12 nodes and 66 arcs and fully dense requirement matrix. We call relative optimality guarantee (or accuracy) the ratio of the difference between lower and upper bound values over the upper bound value for the optimum of GSP (i.e. $\frac{(upperbound-lowerbound)}{upperbound}$). PDA is stopped when this relative optimality guarantee is lower than 10^{-6} or when the number of outer iterations is greather than 400. In this section we aim to study the sensitivity of PD2 accuracy to the choice of the proximal parameter Λ. For all tested values of Λ, the code PDA stops computing when the outer iterations number exceeds 400. We choose Λ as follows

$$\Lambda^{-1} = \theta \times c_{max},$$

with $c_{max} = \max\left[\max\{D_i,\ i \in \{1,\ldots,n\}\}, \max\{C_i, i \in \{1,\ldots,n_n\}\}\right]$.

$10^3 \times \theta$	Λ	**accuracy**
3	33.3	0.115
3.5	28.57	0.101
4	25	8.06×10^{-2}
4.5	22.2	6.10×10^{-2}
5	20	3.7×10^{-2}
5.5	18.18	1.26×10^{-2}
6	16.66	1.43×10^{-5}
6.5	15.38	1.24×10^{-5}
7	14.28	1.62×10^{-5}
8	12.5	1.01×10^{-5}
9	11.11	1.29×10^{-5}

Table 7.1: Accuracy

The table 7.1 illustrates very well the proximal parameter's importance. In [21] an estimation of a good choice of the proximal parameter was given when the objective function is strictly convex with a Lipschitzian gradient.

Remark 4 *The same definition of accuracy is used for PDA and DIP for all test problems.*

7.2 Path dispersion

The path dispersion gives the maximum number of paths used. We notice that PDA uses a few paths to find solution. In [25] Ouorou, Mahey and Vial have observed this behaviour of proximal decomposition method.
In table 7.2, we collect path dispersion for problems PB8, ..., PB16.

PB	DIP	PDA
8	623	196
9	1043	607
10	1485	5167
11	2270	878
12	4001	726
13	4295	936
14	3765	1183
15	> 4797	2809
16	X	1803

Table 7.2: Path dispersion

7.3 Iterations and computing time

The table 7.3 summarizes the numerical results obtained with the instances PBi, for $i \in \{8, \ldots, 16\}$.

PB	p.unm.1	p.unm.2	cpu.1	cpu.2	prec.1	prec.2
8	623	196	132.315	25.253	1.82E-009	9.31E-007
9	1043	607	893.001	255.916	1.37E-009	9.83E-005
10	1485	5167	3190.309	2870.345	1.19E-011	1.59E-006
11	2270	878	15882.353	199.788	3.47E-009	1.59e-06
12	4001	726	28934.772	171.917	1.35E-009	1.01E-005
13	4295	936	30360.794	231.636	7.16E-012	1.16E-005
14	3765	1183	23216.124	156.055	7.77E-010	5.78E-006
15	> 4797	2809	54402.129	353.194	0.0002530792	7.49E-006
16	X	1803	X	353.194	X	8.02E-006

Table 7.3

where,

- p.num.1 : number of generated paths using DIP.
- p.num.2 : number of generated paths using PDA.
- cpu.1 : computing time of DIP.
- cpu.2 : computing time of PDA.
- prec.1 : accuracy obtained by DIP.
- prec.2 : accuracy obtained by PDA.

7.4 Conclusion

We solved in this paper the GSP problem based on the global strategy. This problem is solved for meduim size networks corresponding to metropolitain areas. In order to solve GSP, we developped a new approach based on the proximal decomposition scheme and test it on networks with up to 16 nodes, 140 edges and 140 commodities. Our approach is also compared with a cutting plane algorithm (DIP) using HOPDM for solving the master program. Although DIP is more accurate, our approach outperforms it as chown in table 7.3. Moreover, due to the needs in telecommunication planning area highly accurate results are not helpfull while computing perfomances are more requested. Finally, the solutions provided by our approach are fractional. In order to find out integer feasible solutions, heuristics (Kernighan and Lin [17], GRASP [6], ...) could be applied.

References

[1] D. Alevras, M. Grötschel, and R. Wessäly. Capacity and survivability models for telecommunication networks. Technical Report SC-97-24, Konard-Zuse-Zentrum für Informationstechnik, Berlin, June 1997.

[2] J. Chifflet, P. Mahey, and V. Reynier. Proximal decomposition for multicommodity flow problems with convex costs. *Telecommunication Systems*, 3:1–10, 1994.

[3] N. Christophides and C.A. Whitlock. Network synthesis with connectivity constraint: A survey. *Operational Research*, pages 705–723, 1981.

[4] G. Dahl and M. Stoer. Multisun- mathematical model and algorithms. Technical Report TF R 46/92, Televerkets Forskningsinstitutt, 1992.

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

[6] T.A. Feo and M.G.C. Resende. Greedy randomized adptive search procedures. *Math. Programming*, 6:109–133, 1995.

[7] L.R. Ford and D.R. Fulkerson. A suggested computation flor maximal multicommodity network flow. *Manag.Sci*, 5:97–101, 1958.

[8] A. Frangioni. *Dual ascent methods and multicommodity flows.* Dip. di informatica, Univ. di Pisa, 1997.

[9] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38:284–302, 1992.

[10] M. Gondran and M. Minoux. *Graphes et Algorithmes.* Eyrolles, 1979.

[11] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.

[12] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. In *Network models*, pages 617–672. North-Holland, Amsterdam, 1995.

[13] M. Grötschel and C.L. Monma. Integer polyhedra arising from certain network design problems with connectivity constraints. *SIAM J. Discrete Math*, 3:502–524, 1990.

[14] M. Grötschel, C.L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40(2):1992, 1992.

[15] M. Grötschel, C.L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization*, 2(3):474–504, 1992.

[16] W.S. Jewell. Optimal flow through networks. Technical report, Massachussets Institue of Technology, 1958.

[17] B. Kernighan and S. Lin. An efficient heuristic procedure for graph partitioning. *Bell Systems Technical J.*, 49:671–680, 1970.

[18] A. Lisser, R. Sarkissian, and J.P. Vial. Optimal joint synthesis of base and reserve telecommunication networks. Technical report, CNET, Novembre 1995.

[19] A. Lisser, R. Sarkissian, and J.P. Vial. Survivability in transmission telecommunication networks. Technical report, CNET, April 1995.

[20] David G. Luenberger. *Introduction to linear and nonlinear programming.* Addison-Wesley., 1984.

[21] P. Mahey, S. Oualibouch, and P.D. Tao. Proximal decomposition on the graph of a maximal monotone operator. *SIAM J. Optimization*, 5(2):454–466, May 1995.

[22] M. Minoux. Synthèse optimale d'un réseau de télécommunication avec contraintes de sécurité. *Annales des télécommunications*, 36:211–230, 1981.

[23] C.L. Monma and D.F. Shallcross. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research*, 37(4):153–171, 1989.

[24] A. Ouorou. *Décomposition proximale des problèmes de multiflot à critère convexe. Application aux problèmes de routage dans les réseaux de communication.* PhD thesis, Université Blaise Pascal, Ecole doctorale des sciences pour l'ingenieur de Clermont-Ferrand, Clermont-Ferrand, 1995.

[25] A. Ouorou, P. Mahey, and J.Ph. Vial. A survey of algorithms for convex multicommodity flow problems. Technical report, Laboratoire inter-universitaire d'informatique, de modélisation et d'optimisation des systèmes, Université Blaise Pascal, France, 1997.

[26] R. Rébaï. *Optimisation de réseaux de télécommunications avec sécurisation.* PhD thesis, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny 75775 Paris cedex 16, France, 10 Février 2000.

[27] R.T. Rockafellar. *Convex Analysis.* Princeton Univ. Press, 1970.

[28] J.E. Spingarn. Partial inverse of a monotone operator. *Appl. Math. Optim*, 10:247–265, 1983.

[29] K. Steiglitz, P. Weiner, and D. J. Kleitman. The design of minimum-cost survivable networks. *IEEE Trans. Circuit Theory*, CT-16:455–460, 1969.

[30] M. Stoer. *Design of Survivable Networks*, volume Lecture Notes in Mathematics. Springer, 1992.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399