Solving LP Relaxation for Survivability Problems in Telecommunications Networks*

A. Lisser[1] *R. Sarkissian*[2] *J. P. Vial*[2]

[1]FRANCE TELECOM
CNET/DAC/OAT, 38-40, Rue du General Leclerc
92131, Issy les Moulineaux Cedex, France

[2]LOGILAB, HEC-Genève
Université de Genève
102 Bd Carl Vogt, CH-1211 Genève 4, Suisse

Abstract

It is proposed that traffic in a telecommunications network be secured in the event of a node or link failure by the rerouting of traffic over a reserve network. The problem consists of two related parts: the dimensioning of a reserve network, and the re-allocation, or rerouting of traffic. We formulate the problem as a linear programming problem of huge size which we solve using a cutting plane algorithm based on the concept of an analytic center. The method enables the solution of the survivability problem for networks with up to 60 nodes and 120 links, which allows a realistic modelling of France Telecom's Main Interconnection Network.

Keywords: Survivability in telecommunication networks, cutting plane methods, interior point methods, decomposition.

1 Introduction

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

*This work was financed by contract No 94 5B 022 of CNET.

The concerns of clients include the areas of network availability, tariffs levels, and finally network reliability. If the rupture of a single telephone line is not a catastrophic event, for a commercial enterprise, the inability to transfer data, even for a limited time, could be seen as unacceptable by the user. Even a small risk of a failure in the operator's network could discourage a client from making use of the services of that operator. 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 operator in charge of real time network management must, as quickly as possible, see to the rerouting of all the transmission demands which were being served by the failed elements. This is only possible if the remainder of the network has excess capacity. To ensure that sufficient capacity exists for any possible degree of failure, the current practice is to install, in parallel with the base network, a second network of the same type and with identical topology. This network is known as the reserve network, and its dimensioning is the subject of this paper.

A telecommunication network is a complex web of transmission links and nodes. So much so that one cannot speak of a single network, but rather of a hierarchy. In the context of such a hierarchy the survivability problem is an extremely complex one. One can nevertheless conceive of a Main Interconnection Network at its apex, which can be modelled with the aid of a graph. Transmission demands are defined by the volume of information to be sent and the origin-destination pairs. A routing table is established in advance by the network operators, according to certain technical criteria.

Basic network failures are of two types: arc failure (e.g., severing of an optical cable), which also affects the corresponding reserve arc, and node failure (e.g., failure of electronic equipment), which affects all adjacent arcs from both the base and the reserve network. The fundamental hypothesis in network survivability problems is that only one basic failure can be handled at once. The possibility of multiple basic failures occurring simultaneously is considered extremely improbable.

There exists two types of rerouting, local and global. In the local case, it is considered that the rupture of a link creates, at its endpoints, a demand equal to the total flow which transitted through the link. This demand must be rerouted through the reserve network. It can be described as a single commodity requirement. In the global case, 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 through the reserve network. The demand requirement may now be described as one of multiple commodities. The flow analysis can be taken further: those flows interrupted release capacity in the network. If the operating conditions allow it, this released capacity can be used to advantage in the rerouting. The global approach, 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 can be stated as follows:

Determine the capacities to be installed in the reserve network, and the required rerouting of demands such that

i) for each basic failure, all demands are satisfied,

ii) the total cost of investment be minimized.

In this problem, each failure creates a single (in case of local rerouting) or multiple (in case of global rerouting) commodity requirement. Since our formulation considers at most one failure at a time, the problem can be described [26] as "the minimum cost network synthesis under single or multiple commodity requirement".

Under the hypotheses of linear capacity installation costs and divisible flows, the survivability problem can be formulated as a linear programming problem. See [26]. Unfortunately the LP problem is of such a large size, even for networks of moderate dimensions, that it challenges the capabilities of the most advanced 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.

There are at least two ways of implementing decomposition. The first one consists of separating the capacity design issue (the master program) and the nonsimultaneous multiflow requirements (the subproblems). The master program selects a tentative capacity design. The subproblems test whether this proposal meets the multiflows 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 [27]. This approach is usually named Benders decomposition [7]. In [28, 29], the authors use a subgradient optimization technique to solve (approximately) the subproblems. They report results on small size networks (12 nodes, 25 arcs and 66 demands). Different approaches for solving survivability problems in telecommunications networks are reported in [1, 6, 12, 18, 19, 20, 23, 24, 25, 33, 34].

The other approach, which we advocate in this paper, 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 of maximizing the Lagrangian in the dual variables. The subproblems tests whether for a given set of dual variables there exist 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 sharp 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.

The classical approach in decomposition is to solve the master program to optimality. The rate of convergence of this method is then known to be extremely slow in some instances. Many alternatives have been proposed to achieve more stable convergence rates. For a review see [11]. In this paper we use the analytic center cutting plane method (ACCPM) [15]. This technique has been tested on a wide variety of problems [15, 4, 5].

The paper is organized as follows. In section 2 we give a mathematical formulation of the problem. In section 3 we offer a succinct exposition of the method of cutting planes. In the following section we set out the variant of this method based on the analytic center: we outline briefly the interior point algorithm used to calculate approximations to analytic centers. In section 5 we review some important points concerning the implementation, whereas section 6 reports on the numerical results.

Notation: Given a vector x, we shall denote by X the diagonal matrix whose diagonal components are equal to x. We shall also denote $\mathbf{1}$ to be the vector with each component equal to one; its dimension is to be inferred from the context.

2 A Mathematical Model of the Dimensioning of the Reserve Network

The telecommunications network is represented by a base graph $G = (V, E)$. In the language of graph theory this graph is simple, that is undirected, without loops and with a maximum of one edge connecting any two vertices. The flows in the network represent packets of information, or messages, which are transferred between nodes in order to satisfy their demands.

2.1 The Multicommodity Netflow Problem

We begin by recalling the basic problem, that of satisfaction of the total demand in the nominal network. The real flows are directed, as they correspond to a routing of messages away from origins toward destinations. When one wishes to measure the utilization levels of network links however, absolute flow values need to be added. To take this fact into account, we let $G = (V, E)$ be undirected and define for each directed edge $a = (j, k) \in E$ two types of flows: a positive flow $(x_a)^+$ which transits from j to k and a negative flow $(x_a)^-$ which transits in the opposite direction. The total flow over the edge is then:

$$x_a = (x_a)^+ + (x_a)^-.$$

These flows, positive and negative, are themselves superpositions of other flows, corresponding to specific demands called commodities between origin-destination pairs somewhere in the network. We index these flows by commodity. Thus for an edge a:

$$x_a = \sum_{i \in I} \Big((x_a^i)^+ + (x_a^i)^- \Big),$$

where I is the set of all commodities.

To define the demand and flow conservation constraints we introduce a convenient notation. For each vertex $j \in V$, or more generally for any subset of vertices $A \subset V$, we define $\omega_+(A)$ as the set of edges terminating in A, and $\omega_-(A)$ the set with origins in A. We can then write for each vertex j, and commodity i with origin s and destination t, the flow conservation equations.

$$\sum_{a \in \omega_+(j)} \big((x_a^i)^+ - (x_a^i)^- \big) - \sum_{a \in \omega_-(j)} \big((x_a^i)^+ - (x_a^i)^- \big) = \begin{cases} -f_i, & \text{if } j = s, \\ f_i, & \text{if } j = t, \\ 0, & \text{otherwise.} \end{cases} \qquad (1)$$

We define the *load* of an edge $a \in E$ by the commodity i to be the sum of the two directed flows

$$x_a^i = (x_a^i)^+ + (x_a^i)^-. \qquad (2)$$

The vector $x^i = \{x_a^i\}$ is a feasible load for the commodity i if there exist flows $(x^i)^+$ and $(x^i)^-$ satisfying (2) for all $a \in E$ and (1) for all $j \in V$. The set of feasible loads for commodity i is denoted by F_i. By introducing the load vector $x = (x^1, \ldots, x^{|I|})$ over the set of all commodities, we can define the set of feasible loads as the set product

$$F = \prod_{i \in I} F_i.$$

Finally, if K is the capacity of the graph, the flow equations must be complemented by the capacity constraints for each of the edges $a \in E$:

$$\sum_{i \in I} x_a^i \le K_a. \qquad (3)$$

The Multicommodity Network Flow Problem is a LP problem with flow constraints (1) on each commodity, and capacity constraints (3) on each arc. It is possible to add the cost of routing to this base model. If the unit cost is zero, we need only solve the feasibility problem.

In practice, it is often required that flows be modular. In this case the problem loses its linearity property, and we enter the domain of integer programming.

2.2 The Failure Problem

Failure management cannot be defined independently of the flow routing in place under normal operating conditions. In fact, in the event of a failure, only those flows that transitted the failed elements require rerouting. This will be achieved according to certain rules. The most restrictive is that rerouted flows use the reserve network only. A wider concept of failure management is that these flows can also be carried by the nominal network, to the extent that there is unutilized capacity.

More formally, we say that the failures p form a set P. These failures will modify the topology of the base graph and give rise, via a deletion of elements, to the partial-subgraphs $G^{(p)} = (V^{(p)}, E^{(p)})$, which we refer to as failure networks.

Each failure p disables a set of paths in the base graph. As a result there are unsatisfied demands between certain origin-destination pairs, and a corresponding set of commodities $I^{(p)}$. Each commodity $i \in I^{(p)}$ of the failure network is defined by a demand of magnitude f_i directed away from an origin s_i toward a destination t_i. These unsatisfied demands are to be rerouted over the failure network $G^{(p)}$. So that the rerouted flows can be managed in practice, we may need to constrain the flows x^i, $i \in I^{(p)}$ to be integer valued.

The capacity of the failure network $G^{(p)}$ can be divided into two parts: the capacity of the edges in the reserve network, and the spare capacity in the nominal network. In the failure p configuration, the spare capacity $K^{(p)}$ is made up of the spare capacity in the base network under normal operating conditions and the additional capacity generated by the interrupted flows under failure p. It is important to realize that the spare capacity $K^{(p)}$ can be considered as fixed for any given failure condition, whereas the capacity of the reserve network is a decision variable.

Let $K^{(p)}$ be the spare capacity of the failure network $G^{(p)}$ and $y = (y_a)_{a \in E}$ the capacity to be installed. The installation unit cost is $c = (c_a)_{a \in E}$. We set an upper bound $\bar{y} = (\bar{y}_a)_{a \in E}$ on it. The survivability problem (SP_{INT}) can be formulated as follows:

$$\min \quad c^{\mathrm{T}} y$$

$$\text{s.t.} \quad \sum_{i \in I^{(p)}} x_a^{(i)} \leq K_a^{(p)} + y_a, \quad \forall a \in E^{(p)},\ \forall p \in \mathcal{P},$$
$$x^i \in F^{(p)}, \quad \forall i \in I^{(p)},\ \forall p \in \mathcal{P},$$
$$0 \leq y_a \leq \bar{y}_a, \quad x,\ y \text{ integers.}$$

Constraining the flows to be integer valued greatly complicates the problem. A lower bound to the optimal solution can be obtained by dropping this condition. We consider then the relaxed problem (SP)

$$\min \quad c^{\mathrm{T}} y \tag{4}$$
$$\text{s.t.} \quad \sum_{i \in I^{(p)}} x_a^{(i)} \leq K_a^{(p)} + y_a, \quad \forall a \in E^{(p)},\ \forall p \in \mathcal{P}, \tag{5}$$
$$x^i \in F^{(p)}, \quad \forall i \in I^{(p)},\ \forall p \in \mathcal{P}, \tag{6}$$
$$0 \leq y_a \leq \bar{y}_a. \tag{7}$$

The problem (SP) is linear, but its size can be significant even for a nominal network of moderate size.

3 The Method of Cutting Planes

The size of the survivability problem is almost certainly beyond any direct application of a linear programming algorithm. The alternative is to use the principle of decomposition, which converts the original problem into a smaller nondifferentiable problem in convex optimization. This is achieved by partial dualization. In this section we recall this technique, show how to construct the elements of the sub-differential of the function thus obtained, and present a generic cutting planes algorithm [22].

3.1 The Dual Problem

3.1.1 The Lagrangian and Partial Dualization

Consider the Lagrangian obtained by the dualization of the coupling constraints (5) of (SP). The dual vector associated with the constraints (5) is $v = (v^1, \ldots, v^p, \ldots, v^{|\mathcal{P}|})$. We obtain:

$$L(x, y; v) = c^{\mathrm{T}} y + \sum_{p \in \mathcal{P}} \sum_{a \in E^{(p)}} v_a^p \Big(\sum_{i \in I^{(p)}} x_a^i - K_a^{(p)} - y_a \Big). \tag{8}$$

To simplify what follows we introduce the set product of the feasible flows:

$$F = \prod_{p \in \mathcal{P}} \prod_{i \in I^{(p)}} F_i^{(p)}.$$

From (8) we define the following two functions

$$\overline{L}(x,y) := \sup_{v \geq 0} L(x,y;v), \tag{9}$$

and

$$\underline{L}(v) := \min \left\{ L(x,y;v) \,:\, 0 \leq y \leq \bar{y},\, x \in F \right\}. \tag{10}$$

These two functions are respectively convex and concave. The former yields an alternative formulation of the primal problem. The latter allows us to introduce the dual problem.

$$\max \, \{ \underline{L}(v) \,:\, v \geq 0 \}. \tag{11}$$

By the minmax theorem of convex programming, there exist vectors v^* and (x^*, y^*) such that

$$\underline{L}(v^*) = \max \{ \underline{L}(v) \,:\, v \geq 0 \},$$

$$\overline{L}(x^*, y^*) = \max \left\{ \overline{L}(x,y) \,:\, 0 \leq y \leq \bar{y},\, x \in F \right\},$$

and

$$\underline{L}(v^*) = \overline{L}(x^*, y^*).$$

The dual problem is concave but piecewise linear. It is thus nondifferentiable.

3.1.2 Calculating Values of the Dual Function

It is relatively easy to calculate $\underline{L}(v)$ for a given v. In fact, using the definition (8) we can write

$$\underline{L}(v) = \gamma(v) + \varphi(v) - \sum_{p \in P} \sum_{a \in E^{(p)}} v_a^p K_a^{(p)},$$

where γ and φ are given by

$$\gamma(v) = \min_{x \in F} \sum_{p \in \mathcal{P}} \sum_{i \in I^{(p)}} \Big(\sum_{a \in E^{(p)}} v_a^p x^i_a \Big), \tag{12}$$

and

$$\varphi(v) = \min_{0 \leq y \leq \bar{y}} \left\{ \sum_{a \in E} y_a c_a - \sum_{p \in \mathcal{P}} \sum_{a \in E^{(p)}} y_a v_a^p \right\}. \tag{13}$$

At the risk of an abuse of notation, we extend the vector v with arbitrary components v_a^p for each $a \notin E^{(p)}$ and introduce the parameters

$$\delta_a^p = \begin{cases} 1 & \text{if } a \in E^{(p)}, \\ 0 & \text{otherwise.} \end{cases}$$

the functions γ and φ can be decomposed into simple functions

$$\gamma_i(v) = \min_{x^i \in F^{(p)}} \sum_{a \in E^{(p)}} v_a^p x_a^i, \qquad i \in I^{(p)} \tag{14}$$

and

$$\varphi_a(v) = \min_{0 \le y_a \le \bar{y}_a} y_a \Big(c_a - \sum_{p \in \mathcal{P}} \delta_a^p v_a^p\Big). \tag{15}$$

Thus we have

$$\gamma(v) = \sum_{p \in \mathcal{P}} \sum_{i \in I^{(p)}} \gamma_i(v)$$

and

$$\varphi(v) = \sum_{a \in E} \varphi_a(v).$$

The elementary functions $\gamma_i(v)$ and $\varphi_a(v)$ 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. The functions γ are the optimal values of simple flow problems corresponding to the shortest path problem over a graph with non-negative costs. The functions φ take either the value zero or $(c_a - \sum_{p \in \mathcal{P}} \delta_a^p v_a^{(p)})\bar{y}_a$ according to the sign of $(c_a - \sum_{p \in \mathcal{P}} \delta_a^p v_a^{(p)})$. Thus $\bar{y}_a$ essentially takes only two values, 0 and $\bar{y}_a$.

3.1.3 Calculation of the Sub-Differential

To determine an element of the sub-differential problem, it suffices to express the dependence of the optimal values of γ and φ 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 $\hat{v}$ be a point at at which we calculate $\gamma_i(\hat{v})$ and $\varphi_a(\hat{v})$, and let $\hat{x}^i$ and $\hat{y}_a$ be the values respectively where these functions attain their minima. Let v be an arbitrary point. Using the fact that, for a given v, γ_i is the minimum of $\sum_{a \in E^{(p)}} v_a^p x_a^i$ for $x^i \in F^{(p)}$, we obtain for $i \in I^{(p)}$,

$$\begin{aligned} \gamma_i(v) &\le \sum_{a \in E^{(p)}} v_a^p \hat{x}_a^i \\ &= \gamma_i(\hat{v}) + \sum_{a \in E^{(p)}} \hat{x}_a^i (v_a^p - \hat{v}_a^p). \end{aligned} \tag{16}$$

Inequality (16) defines a support of the concave function γ_i, $i \in I^{(p)}$ at the point $\hat{v}$. The coefficients $(v_a^p - \hat{v}_a^p)$ of $\hat{x}_a^i$ are the components of the sub-differential of the γ_i.

In the same way, we construct the sub-differential of φ_a by the inequality

$$\begin{aligned}\varphi_a(v) &\leq \hat{y}_a \left(c_a - \sum_{p\in\mathcal{P}} \delta_a^p v_a^{(p)} \right) \\ &= \varphi_a(\hat{v}) - \sum_{p\in\mathcal{P}} \delta_a^p \hat{y}_a (v_a^{(p)} - \hat{v}_a^{(p)}). \end{aligned} \tag{17}$$

Inequality (17) defines a support of the concave function φ_a at the point $\hat{v}$. The sub-gradient is therefore a null vector, excepting those components corresponding to edges in the failure network, which have value $\hat{y}_a$.

3.2 General Cutting Planes Algorithm

To summarize, we have shown that the survivability problem can be written as a nondifferentiable convex programming problem, of type

$$\min \{ f(v) \, : \, v \geq 0 \}$$

where $f(v) = -\underline{L}(v)$. The cutting planes method can be applied successfully to problems of this type.

Let $f_1, f_2, ..., f_N : \mathbb{R}_+^m \to \mathbb{R}$ be convex functions and b a vector in $\mathbb{R}_+^m$. Now define the functions

$$f(v) = \sum_{i=1}^{N} f_i(v) + b^{\mathrm{T}} v.$$

We consider the problem

$$z^* = \min\{ f(v) : 0 \leq v \leq h \}. \tag{18}$$

This is the same problem as before; the constraint $v \leq h$ has been added simply to ensure the compactness of the feasible domain. We call the inequalities box constraints. Given that the bound h is arbitrary, for convenience we choose it to be large. In the survivability problem the box constraints on v may be given a specific value in the case where there is no capacity constraint $y \leq \bar{y}_a$ (or equivalently, the capacity constraint is known to be inactive at the optimum). From (15) we can see that as $\bar{y}_a \to +\infty$, the minimum of φ_a is finite if and only if $c_a - \sum_{p\in\mathcal{P}} \delta_a^p v_a^{(p)} \geq 0$. Hence we have the box constraint $0 \leq v_a^{(p)} \leq c_a$.

We will use the notation

$$\partial f(v) = \{\xi \in \mathbb{R}_+^m : f(w) \geq f(v) + \xi^T(w-v), \forall w \in \mathbb{R}_+^m\}$$

to describe the set of sub-gradients of f as functions of v. The procedure used to calculate $f(v)$ and elements of its sub-differential will henceforth be referred to as *oracle*.

Finally we recall that the epigraph of f is defined by

$$\text{epi} f = \{(z,v) \,:\, z \geq f(v)\}\,.$$

The aim of the method of cutting planes is to construct increasingly sharp approximations to the epigraph of f. Note that if we use the form of f with the summation, we can write

$$\text{epi} f = \Big\{(z,v) \,:\, z = \sum_{i=1} z_i + b^{\mathrm{T}} v, \quad (z_i, v) \in \text{epi} f_i\Big\}.$$

3.2.1 The Polyhedral Approximation

Assume that a sequence of vectors $\{v^k\}_{k=1,\dots\kappa}$ has been generated by some procedure. From this the oracle then generates $2N$ new sequences : the values f_i^k and the subgradients $\xi_i^k \in \partial f_i(v^k)$, $i=1,\dots,N$,$k=1,\dots,\kappa$. From these we can form a linear approximation to the epigraphs of f_i using the inequalities

$$z_i - (\xi_i^k)^{\mathrm{T}} v \geq f_i(v^k) - (\xi_i^k)^{\mathrm{T}} v^k, \quad k = 1,\dots,\kappa. \tag{19}$$

We obtain an approximation of epi$f$ upon adding to (19) the equation

$$z = \sum_{i=1}^{N} z_i + b^{\mathrm{T}} v.$$

The linear program

$$\begin{array}{ll} \min & z = \displaystyle\sum_{i=1}^{N} z_i + b^{\mathrm{T}} v \\ \text{s.t.} & z_i - (x_i^k)^{\mathrm{T}} v \geq f_i(v^k) - (x_i^k)^{\mathrm{T}} v^k, \quad k=1,\dots,\kappa, \quad i=1,\dots,N, \\ & h \geq v \geq 0 \end{array}$$

provides a lower bound z_l^κ to the optimal solution z^* of problem (18), whereas

$$z_u^\kappa = \min_{k=1,\dots,\kappa} \left\{ \sum_i f_i(v^k) + b^{\mathrm{T}} v^k \right\}$$

provides an upper bound.

When a new point $v^{\kappa+1}$ is obtained, the oracle generates more inequalities following (19) which are added to the existing set. The result is a refinement of the approximation to the epigraph of f. At step k the accuracy of the approximation to the optimal solution is given by the duality gap

$$\Delta^\kappa = z_u^\kappa - z_l^\kappa.$$

3.2.2 The Localization Set

We make use of the upper bound z_u^κ to construct a compact set containing the optimal solution (z^*, v^*). This set, known as the localization set and denoted by $\mathcal{L}(z_u)$, is defined by the inequalities

$$\begin{array}{rcl} z_u^\kappa & \geq & \sum_{i=1}^{N} z_i + b^{\mathrm{T}} v \\ z_i - (\xi_i^k)^{\mathrm{T}} v & \geq & f_i(v^k) - (\xi_i^k)^{\mathrm{T}} v^k, \quad i = 1,\dots,N, \quad k = 1,\dots,\kappa \\ l & \leq & v \leq h. \end{array}$$

3.2.3 A Generic Cutting Plane Algorithm

We present here a generic algorithm for the case $N = 1$ in the definition of f. In this case only a single subgradient is returned by the oracle, that is a single cut across the localization set.

To simplify notation we omit the iteration index k. At the start of the iteration are given a lower bound z_l, an upper bound z_u, and a localization set $\mathcal{L}(z_u)$. The iteration steps are:

1. Choose $(\bar{z}, \bar{v}) \in \mathcal{L}(z_u)$.
2. Calculate a lower bound $\underline{z}$ for each z such that $(z, v) \in \mathcal{L}(z_u)$.
3. Calculate $f(\bar{v})$ and $\xi \in \partial f(\bar{v})$.
4. Update the bounds:
   (a) $z_u := \min\{f(\bar{v}), z_u\}$
   (b) $z_l := \max\{\underline{z}, z_l\}$.

5. Update the localization set by adding

$$z - \xi^{\mathrm{T}} v \geq f(\bar{u}) - \xi^{\mathrm{T}} \bar{v}$$

to the set of inequalities defining $\mathcal{L}(z_u)$.

The algorithm halts when the duality gap $\Delta = z_u - z_l$ falls below the required precision.

To implement the generic cutting plane algorithm, the way in which the point $(\bar{z}, \bar{v})$ is chosen in the localization set must be specified. Several choices are possible, each defining a particular version of the algorithm. For example, in the decomposition algorithm of Dantzig-Wolfe [9] (see also [22]), $(\bar{z}, \bar{v})$ is a point which minimizes z in $\mathcal{L}(z_u)$. In the following section, we propose an alternative, that of the analytic center of the localization set.

It is convenient to use the terminology of Dantzig-Wolfe regardless of the version of the algorithm used. The procedure which selects $(\bar{z}, \bar{v})$ in the first stage of the algorithm will be called the master program. The oracle, which calculates $f(\bar{v})$ and $\xi \in \partial(f(\bar{v}))$, will be called the subproblem.

If there are more than one subproblem, the oracle may generate more than one cut at a time.

4 The ACCPM

Having already stated that step 1 of the algorithm, the choosing of the point in the localization set, is that which differentiates between the different versions, we now specify the method which defines the ACCPM. For a detailed description the reader is referred to [13, 15]; for an analysis of convergence to [3, 31, 16]; and for the implementations to [4, 5].

4.1 Definition of the Analytic Center

The linear program (20) can be written in the following way:

$$\begin{array}{rl} \min & z_1 + \ldots + z_N + b_0 v \\ \text{s.t} & G_i^{\mathrm{T}} v + z_i \geq \gamma_i, \quad i = 1, \ldots, N, \\ & h \geq v \geq 0, \end{array}$$

where we have put

$$\begin{array}{rcl} (G_i)_k & = & -\xi_i^k \\ (\gamma_i)_k & = & f_i(v^k) - {\xi_i^k}^{\mathrm{T}} f_i(v^k), \end{array}$$

for all $\quad k = 1, \ldots, \kappa$ et $i = 1, \ldots, N$.

Now define

$$u = \begin{pmatrix} v \\ z_1 \\ \vdots \\ z_N \end{pmatrix}, \; c = \begin{pmatrix} 0 \\ -h \\ \gamma_1 \\ \vdots \\ \gamma_N \end{pmatrix}, \; b = \begin{pmatrix} b_0 \\ \mathbf{1} \end{pmatrix}, \; G = \begin{pmatrix} I & -I & G_1 & G_2 & \ldots & G_\kappa \\ 0 & 0 & E_1 & E_2 & \ldots & E_\kappa \end{pmatrix},$$

where the sub-matrices E_k of G are null except for the k ith row which is the unit vector $\mathbf{1}^{\mathrm{T}}$ of the appropriate dimension.

We can now express the problem (20) in the condensed form

$$\begin{aligned} \min \quad & b^T u \\ G^T u - s &= c \\ s &\geq 0. \end{aligned} \tag{20}$$

The analytic center for the localization set $\mathcal{L}(z_u)$ is defined as the point which minimizes the product of the deviations from the constraints defining the localization set. In other words it is the solution of the problem

$$\begin{aligned} \min \quad & \psi(s, u; z_u) = -\ln(z_u - b^T u) - \sum_{j=1}^{n} \ln s_j \\ \text{s.t.} \quad & G^T u - s = c, \quad s > 0 \\ & z_u - b^T u > 0. \end{aligned} \tag{21}$$

It can be verified that the conditions for first order optimality are

$$\begin{aligned} Gy &= b, \quad y > 0 \\ G^T u - s &= c, \quad s > 0 \\ Ys &= (z_u - b^T u)\mathbf{1} > 0. \end{aligned} \tag{22}$$

Here Y is the diagonal matrix with elements equal to those of the vector y. It can be shown [30], under the hypothesis that the sets

$$S_{\mathcal{P}} = \{y : Gy = b, \, y > 0\}$$

and

$$S_{\mathcal{D}} = \{u : G^T u > c\}$$

are non empty, that the first order conditions (22) can always be satisfied.

4.2 The Projective Algorithm

The analytic center of the polytope $\mathcal{L}(z_u)$ can also be defined with respect to the dual problem (20)

$$\begin{aligned} \min \quad & c^T y \\ Gy &= b \\ y &\geq 0. \end{aligned} \tag{23}$$

Consider

$$\varphi(y; z_u) = (n+1)\ln(z_u - c^T y) - \sum_{j=1}^{n} \ln y_j$$

the Karmarkar potential function [21]. It can be shown that the first order optimality conditions associated with the problem

$$\begin{aligned} \min \quad & \varphi(y; z_u) = (n+1)\ln(z_u - c^T y) - \textstyle\sum_{j=1}^{n} \ln y_j \\ \text{s.t.} \quad & Gy = b, \quad y > 0 \\ & z_u - c^T y > 0 \end{aligned} \tag{24}$$

are the same as for (21).

To solve (24) we use the version of the projective algorithm of Karmarkar [21] described in [10]. This involves embedding the problem in a projective space by the addition of the variable $y_0 \geq 0$. If we set $\tilde{G} = (-b\ G)$, $\tilde{c} = (z_u, -c^{\mathrm{T}})$ and $\tilde{y}^{\mathrm{T}} = (1, y^{\mathrm{T}})$ we obtain from (24) the equivalent formulation

$$\begin{aligned} \min \quad & \tilde{\varphi}(\tilde{y}; z_u) = (n+1)\ln(\tilde{c}^T \tilde{y}) - \textstyle\sum_{j=0}^{n} \ln \tilde{y}_j \\ \text{s.t.} \quad & \tilde{G}\tilde{y} = 0, \quad \tilde{y} > 0 \\ & \tilde{c}^T \tilde{y} > 0. \end{aligned}$$

This is a positively homogeneous problem of degree zero. By dividing by $\tilde{y}_0$ we can recover the original formulation (24).

4.2.1 The Direction of Displacement

The minimum of $\tilde{\varphi}$ is calculated using the projected Newton's method. The direction is $\tilde{Y}q$, where

$$q = \mathbf{1}_{n+1} - \frac{\tilde{c}^T \tilde{y}}{\|P_{\mathcal{N}(\tilde{G}\tilde{Y})} \tilde{Y}\tilde{c}\|^2} P_{\mathcal{N}(\tilde{G}\tilde{Y})} \tilde{Y}\tilde{c}, \tag{25}$$

and where $P_{\mathcal{N}(\tilde{G}\tilde{Y})}\tilde{Y}\tilde{c}$ denotes the projection of $\tilde{Y}\tilde{c}$ onto the null space of $\tilde{G}\tilde{Y}$. The presence of the box constraints ensures that $\tilde{G}$ is of full rank. The projection can then be calculated by the following explicit formulae:

$$P_{\mathcal{N}(\tilde{G}\tilde{Y})}\tilde{Y}\tilde{c} = \tilde{Y}\tilde{s}, \tag{26}$$

together with

$$\tilde{s} = \tilde{c} + \tilde{G}^T u, \tag{27}$$

and

$$u = -(\tilde{G}\tilde{Y}^2\tilde{G}^T)^{-1}\tilde{G}\tilde{Y}^2\tilde{c}. \tag{28}$$

It can be verified that the value u^c calculated using (28) is the analytic center of $\mathcal{L}(z_u)$ if and only if $q = 0$. For a proof, we refer the reader to [14].

4.2.2 The Algorithm

The projective algorithm begins with a feasible interior point solution for which $y > 0$ and $Gy = b$, and maintains the interiority throughout.

1. Calculate the direction q using (25).
2. If $\|q\| \leq \eta < 1$, the vector u^c, calculated from (28), is close to an analytic center $\mathcal{L}(z_u)$.
3. Find an approximate solution to
$$\min_{\alpha>0}\{\tilde{\varphi}(\tilde{y} + \alpha\tilde{Y}q) \ : \ \tilde{y} + \alpha\tilde{Y}q > 0\}.$$
4. Calculate the new point $\bar{y} = \tilde{y} + \bar{\alpha}\tilde{Y}q$, and normalize $\tilde{y} := \frac{\bar{y}}{\bar{y}_0}$ so as to satisfy $\tilde{y}_0 = 1$.

4.2.3 Addition of Cutting Planes

The addition of a cutting plane corresponds to the appending of a new column to G. The vector y is extended by an additional component. If this component is zero we have a solution to $Gy = b$ immediately, but one which violates the interiority condition. There is a very efficient technique, utilized in [15] and presented in detail in [4, 5], which circumvents this problem. We employ this technique but do not describe it here.

5 Implementation

5.1 The oracle

The oracle is a procedure in the ACCPM which evaluates the objective function at a point and generates an element of its subdifferential. It is the only part of the method where the objective function appears explicitly. We have seen that to solve the survivability problem, we must minimize the objective function:

$$f(v) = -\underline{L}(v) = -\sum_{p\in\mathcal{P}} \sum_{i\in I^{(p)}} \gamma_i(v) - \sum_{a\in E} \varphi_a(v) + \sum_{p\in P} \sum_{a\in E^{(p)}} v_a^p K_a^{(p)},$$

where

$$\gamma_i(v) = \min_{x^i \in F^{(p)}} \sum_{a\in E^{(p)}} v_a^p x_a^i, \qquad i \in I^{(p)}$$

and

$$\varphi_a(v) = \min_{0\le y_a \le \bar{y}_a} \; y_a\Big(c_a - \sum_{p\in\mathcal{P}} \delta_a^p v_a^p\Big).$$

The function $\gamma_i(v)$ reduces to the calculation of a shortest path problem in the subgraph $G^{(p)}$. In addition, from the values of γ and φ the oracle generates the corresponding elements of the subdifferential, as explained in section 3. Recall that the subgradients of γ correspond to shortest paths. We can store this information economically in a sparse binary matrix to which we append an element equal to the flow on the path. As for the subgradients of φ , they are determined by the two extremal solutions $y_a = 0$ and $y_a = \bar{y}_a$. They need only be generated once at the beginning of the algorithm.

5.2 The Projective Method

5.2.1 Matrix Structure

For the survivability problem the ACCPM generates a master program for which the matrix $\tilde{G} = (-b\; G)$ has a block structure. We label these blocks as follows

$$G = \begin{pmatrix} G_{11} & G_{12} & 0 & G_{14} \\ 0 & G_{22} & 0 & 0 \\ 0 & 0 & G_{33} & G_{34} \end{pmatrix}. \tag{29}$$

Each bloc has its own structure. To illustrate this, in figure (1), we have put the vector b and G into tabular form.

We begin by describing the vertical bands $G_{.1}$, $G_{.2}$, $G_{.3}$ and $G_{.4}$. The first band is associated with the box constraints. The matrix G_{11} is composed of the identity matrix and its negation.

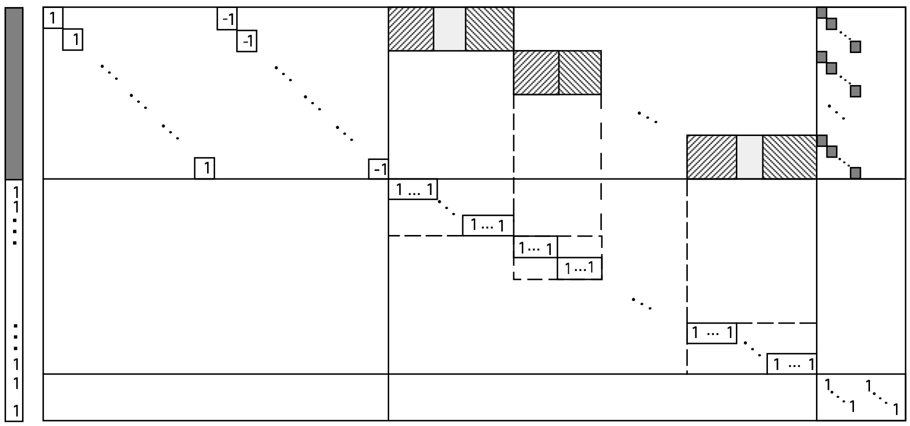

Fig. 1: Master program matrix: vector b and matrix G

The second band concerns the subproblem γ. Block G_{12} contains the subgradients and G_{22} the convexity constraints. These two blocks have themselves a block-diagonal structure of $|P|$ blocks: each subblock corresponds to a subprogram γ. The mutually disjoint nature of the different failure networks induces the block-diagonal structure. The submatrix G_{22} contains the row vectors associated with the convexity constraints, each row corresponding to a failure network. This is therefore the same block-diagonal structure as G_{12}, which has subblocks which are row vectors. Note that trivially, these rows are mutually orthogonal.

Band $G_{\cdot 2}$ has a dynamic structure, growing with the contribution of new subgradients as the algorithm progresses. Also note that each sub-block is of the type seen in multicommodity problems: in fact each failure network must route several commodities.

The two final bands $G_{\cdot 3}$ and $G_{\cdot 4}$ hold the information associated with the subproblem φ. Each function φ_a has two extreme subgradients: the null vector, and another whose elements equal $\bar{y}_a$ when the edge $a \in E$ is present in the failure network $E^{(p)}$, else zero. Block G_{14} is therefore a concatenation of pseudo diagonal blocks (one per failure network).

5.2.2 Dealing with Sparse Matrices

The most expensive operation in the projective method is the solution of system (28)

$$u = -(\tilde{G}\tilde{Y}^2\tilde{G}^T)^{-1}\tilde{G}\tilde{Y}^2\tilde{c}.$$

The solution is obtained using the Cholesky factorization of $(\tilde{G}\tilde{Y}^2\tilde{G}^T)$. The operation consists of two main stages, the formation of the product $(\tilde{G}\tilde{Y}^2\tilde{G}^T)$ itself, and its factorization. These stages have a comparable theoretical complexity. Note that the presence of the diagonal matrix $\tilde{Y}$ is equivalent to a normalization (or, more accurately, a rescaling) of $\tilde{G}$. At each iteration of the projective method $\tilde{Y}$ is normalized so that its first component is 1.

The master program matrix is extremely sparse and highly structured, whereas b is dense. On examination of the contribution of each block of $\tilde{G}$ in the product $(\tilde{G}\tilde{Y}^2\tilde{G}^T)$, it can be seen that, if it were not for b and and the block G_{14}, there would be a diagonal block structure. To exploit this, we calculate separately the contribution of b and G_{14} in the inversion of $(\tilde{G}\tilde{Y}^2\tilde{G}^T)$, using the Shermann-Morrison-Woodburry formula. First let us denote $\bar{G}$ the scaled matrix G. Let $U = (\bar{G}_{.1}\ \bar{G}_{.2}\ \bar{G}_{.3})$ be a submatrix of $\bar{G}$ and let $V = (b\ \bar{G}_{.4})$. Clearly $(\tilde{G}\tilde{Y}^2\tilde{G}^T) = (UV)(UV)^T$. Using the Shermann-Morrison-Woodburry formula, one can therefore write

$$\begin{aligned} \Big((U\ V)(U\ V)^T\Big)^{-1} &= (UU^T + VV^T)^{-1} \\ &= (UU^T)^{-1} - (UU^T)^{-1}V(I + V^T(UU^T)^{-1}V)^{-1}V^T(UU^T)^{-1}. \end{aligned}$$

We now show how to exploit the structure of (UU^T). Given the structure of U, we have

$$H = (UU^T) = \begin{pmatrix} H_{11} & H_{12} & 0 \\ H_{12}^T & H_{22} & 0 \\ 0 & 0 & H_{33} \end{pmatrix}.$$

The submatrices $H_{22} = \bar{G}_{22}\bar{G}_{22}^T$ and $H_{33} = \bar{G}_{33}\bar{G}_{33}^T$ are diagonal. This property follows from the fact that $\bar{G}_{22}$ and $\bar{G}_{33}$ are made of simple convexity constraints. This structure is common to all decomposition problems (multiregional decomposition, stochastic linear programming, etc.) which involve several subprograms. The exploitation of this structure has been described in [5]. This consists essentially of using the Schurr complement

$$F = H_{11} - H_{12}H_{22}^{-1}H_{12}^T. \tag{30}$$

The dimension of F is the number of the coupling constraints in the decomposition problem. This number can be very large (over 20,000 in some of the cases detailed in the next section). To cope with this problem we exploit the block-structure of F inherited from H_{11}.

The factorization is therefore best performed block by block. The advantages of this approach are significant, both in terms of the complexity of the calculation and in memory requirements.

5.3 Reconstruction of the Primal Solution

When the algorithm stops we readily have the optimal value of the objective function and the vector of dual variables associated with the coupling constraints. The primal variables, which are the capacities installed and the rerouting lists, are not given directly. Nevertheless we can easily reconstruct them from the columns of G and the primal variables of problem (24). In fact, the value of these variables gives us the convex combination of the columns of G, i.e., the mix of paths for rerouting. The rerouting list is therefore easily constructed. As for the capacities, they are given by the corresponding variables (24) as a fraction of the maximal capacity $\bar{y}_a$.

6 Numerical results

The survivability algorithm for failures was tested on two types of problem: a problem reported in [8] (the JLLGV) and several randomly generated problems. For the latter a random problem generator was developed [32]. In this generator, the problems are characterized by the number of links and nodes in the nominal network and the initial number of active routes. These parameters can be fixed by the user. In principle this is all the data we need to define the problem. However to provide a self contained formulation, the number of demands to be rerouted in the event of a basic failure should be calculated. The size of this quantity impacts on the difficulty of the problem. It can not be fixed in advance by the user.

The survivability problem can be formulated as a single linear programming problem of huge size. The capacity constraints for each failure configuration form the coupling constraints. The routing constraints are simple flow constraints in the network. Let n be the number of nodes, m the number of links, and p the number of demands to be rerouted. Each link failure generates $m-1$ coupling constraints. A node failure generates $n-d$ coupling constraints, where d is the degree of the node. We denote the average degree by $\bar{d}$. The number of coupling constraints can be estimated to be $m(m-1)+n(m-\bar{d})$. There are n flow constraints for each rerouting, thus np in total[1]. There are mp variables for the flows and m for the capacities. We arrive finally at a problem of size $(m(m-1)+n(m-\bar{d})+np)\times(m(p+1))$.

The parameters of the test problems are given by the following table.

The dimension of the extensive formulation for the test problems are given in table 2.

With the exception of JLLGV, all of these problems consider both node and link failures. In the case of JLLGV there exists a node failure which separates the network into two connected components. It is thus impossible to ensure the survivability of the network with respect to such a failure. Thus only link failures are considered. The numerics were performed on an HP 735/125 with 180 megabytes of RAM. The code was written in C, and compiled with HP's cc compiler with options $+O2$.

[1] This true only for the case of a link failure. For node failure the number is $n-1$ because of the loss of a node in the reserve network. We do not take this distinction into account.

Tests	Arcs	Nodes	Routing	Failures	Rerouting
JLLGV	42	26	264	42	786
P1	57	30	150	81	1100
P2	71	37	300	102	3266
P3	77	40	200	109	1942
P4	87	45	300	123	3658
P5	127	65	400	177	4612

Tab. 1: Dimensioning of the base networks

tests	Coupling Constraints	flow Constraints	Variables
JLLGV	5280	20436	33012
P1	4465	39816	62700
P2	7048	120842	231886
P3	8191	77680	149634
P4	10472	164610	318246
P5	22153	299780	585724

Tab. 2: Dimension of the extensive formulation

To evaluate the performance of the ACCPM, the following measures were calculated.

1. **Gap:** the duality gap;
2. **Iter:** the number of calls to the oracle;
3. **Newton:** the total number of iterations in the projective module of the master program;
4. **Cuts:** the number of independent columns introduced after elimination of copies;
5. **CPU:** the CPU time in seconds on the dedicated machine.

Table 3, page 42, summarizes the results obtained.

Tests	Iter	Newton	Cuts	GAP	CPU
JLLGV	31	205	4333	7.10^{-7}	293
P1	41	525	5494	6.10^{-7}	2487
P2	29	709	20008	7.10^{-7}	11297
P3	38	663	11924	3.10^{-7}	10038
P4	37	1022	22401	9.10^{-7}	29290
P5	21	612	60389	7.10^{-5}	80541

Tab. 3: The ACCPM solution

A precision of 6 digits was obtained for every problem except for *P5*. For it we contented ourselves with 3 digits, a precision considered acceptable for the majority of practical applications.

To analyze the behavior of the algorithm, we have extracted statistics from *P1* to plot the graphs in figure 2. Comparable results were obtained from *P3* using the same analysis.

The first graph shows the number of columns introduced at each exterior iteration. Recall that when the oracle generates a column which is already in the master program, that column is not appended. Note that after 20 exterior iterations no further columns are introduced. All the linear forms necessary to the definition of the optimal solution are present.

The second graph gives the CPU time as a function of the number of interior iterations (or iterations of the projective method). We see an initial growth which then levels off. This corresponds to the addition of columns which ceases after approximately 20 exterior iterations.

The third graph shows the evolution of the duality gap as a function of the number of exterior iterations, using a logarithmic scale. It can be seen that after an initial phase the drop off is log-linear. This behaviour is typical of cutting plane methods. A graph of exactly the same form was obtained in [4] in the very different context of geometric programming.

From this analysis one sees that a large proportion of time is spent in locating the optimum after all the necessary cuts have been generated. We made a similar observation on the other problems of the test set. vIt would definitely be profitable in problem P1 to use a mixed algorithm that would switch from the analytic center of the localisation set to the lower point of the set, the Dantzig-Wolfe point, after the 22nd outer iteration.

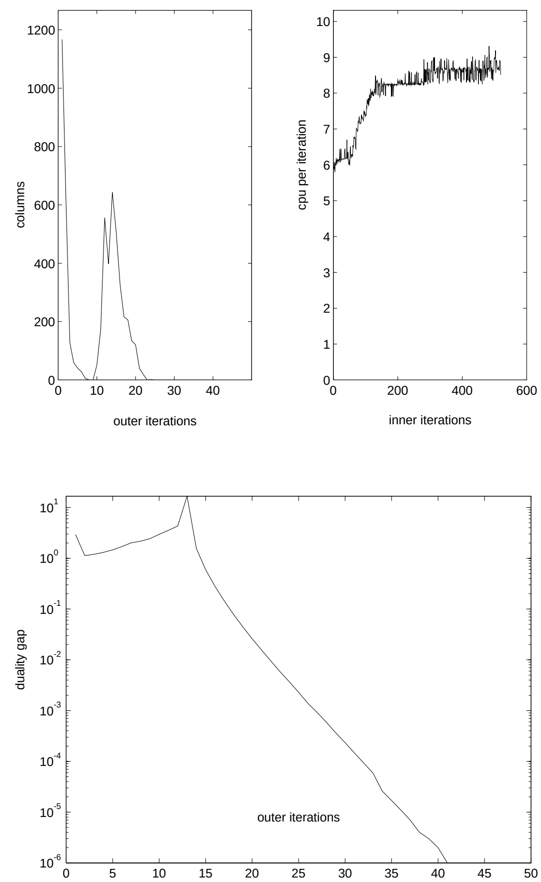

Fig. 2: Statistics for the problem P1.

References

[1] M. Abou-Sayed, J. Kennington and S. Nair, "Joint working and spare capacity assignment in a link restorable mesh network", Report 96-CSE-16, Department of Computer Science and Engineering, Southern Methodist University.

[2] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows*, Prentice-Hall, 1993.

[3] D. S. Atkinson and P. M. Vaidya, "A cutting plane algorithm that uses analytic centers", University of Illinois at Urbana–Champaign, Urbana, IL 61820, USA, June 1992, to appear in "Nondifferentiable and Large Scale Optimization", *Mathematical Programming, Series B*, J.-L. Goffin and J.-P. Vial, eds.

[4] O. Bahn, J.-L. Goffin, J.-P. Vial and O. du Merle, "Implementation and behavior of an interior point cutting plane algorithm for convex programming: An application to geometric programming", *Discrete Applied Mathematics* **49**, 1994, 3–23.

[5] O. Bahn, O. du Merle, J.-L. Goffin, and J.-P. Vial "A cutting plane method from analytic centers for stochastic programming", to appear in "Nondifferentiable and Large Scale Optimization", *Mathematical Programming, Series B*, J.-L. Goffin and J.-P. Vial, eds.

[6] A. Balakrishnan, T.L. Magnanti, A. Shulman and R.T. Wong, " Models for planning capacity extension in local access netwoks", *Annals of Operations Research*, **33**, 1991, 239–284.

[7] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems", *Numerische Mathematik* **4**, 1962, 238–252.

[8] B. A. Coan, W. E. Leland, M. P. Vecchi and L. T. Wu, "Using distributed topology update and preplanned configurations to achieve trunk network survivability", *IEEE Transactions on Reliability*, **40**, No 4, 1991.

[9] G. B. Dantzig and P. Wolfe, "The decomposition algorithm for linear programming", *Econometrica* **29**, 1961, 767–778.

[10] G. de Ghellinck and J.-P. Vial, "A polynomial Newton method for linear programming", *Algorithmica* **1**, 1986, 425–453.

[11] O. du Merle, *Points intérieurs et plans coupants: mise en œuvre et développement d'une méthode pour l'optimisation convexe et la programmation linéaire structurée de grande taille*, PhD Thesis, HEC-Section of Management Studies,

University of Geneva, Switzerland, March 1995.

[12] B. Gavish, P. Trudeau, M. Dror, M. Gendreau and L. Masson "Fiberoptic circuit network design under reliability constraints", *IEEE Journal on Selected Areas in communications* **7**, 1989,1181–1187.

[13] J.-L. Goffin and J.-P. Vial, "Cutting planes and column generation techniques with the projective algorithm", *Journal of Optimization Theory and its Applications*, **65**, 1989, 409–429.

[14] J.-L. Goffin, and J.-P. Vial, "On the computation of weighted analytic centers and dual centers ellipsoids with the projective algorithm". *Mathematical Programming* **60**, 1993, 93–113.

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

[16] J.-L. Goffin, Z.-Q, Luo, Y. Ye, "Further Complexity Analysis of a Primal-Dual Column Generation Algorithm for Convex or Quasiconvex Feasibility Problems", Manuscript, McGill University, Canada, 1993, to appear in *SIAM Journal on optimization.*

[17] G. H. Golub and C. Van Loan, *Matrix Computations*, (2nd ed.) The Johns Hopkins University Press, Baltimore and London, 1989.

[18] M. Grotschel, C.L. Monma and M. Stoer, "Design of survivable networks", 1995, 617–672, in "Network Models, Handbooks in Operations Research and Management Science", M.O. Ball, T.L. Magnanti and G.L. Nemhauser, eds.

[19] W.D. Grover, T.D. Bilodeau and B.D. Venables, "Near optimal spare capacity planning in a mesh restorable network", *Proceedings of IEEE GLOBECOM'91*, 1991, 2007-2012.

[20] W.D. Grover, R.R. Iraschko and Y. Zheng, "Comparative methods and issues in design of mesh-restorable STM and ATM networks", in B. Sanso and P. Soriano (eds), T̂elecommunications Network Planning, 1998, 169–200, Kluwer, Norwell, MA.

[21] N. Karmarkar, "A new polynomial time algorithm for linear programming", *Combinatorica* 4, 1984, 373–395.

[22] J. E. Kelley, "The cutting plane method for solving convex programs", *Journal of the SIAM* **8**, 1960, 703–712.

[23] A. Lisser, R. Sarkissian and J.P. Vial, "Optimal joint synthesis of base and reserve telecommunications Networks", 1995, Research Report NT/PAA/ATR/ORI4491, Centre National D'études des Télécommunications, Issy les Moulineaux, France.

[24] A. Lisser, R. Sarkissian and J.P. Vial, "Mid range planning of survivable telecommunications networks: Joint optimal synthesis of base and spare network capacities", Research Report Logilab/HEC, University of Geneva, 1998.

[25] A. Lisser, R. Sarkissian and J.P. Vial, "Survivability in transmission telecommunications networks", 1995, Research Report NT/PAA/ATR/ORI4230, Centre National D'études des Télécommunications, Issy les Moulineaux, France.

[26] M. Minoux, *Programmation Mathématique: Théorie et algorithmes*, Dunod (Bordas), Paris 1983.

[27] M. Minoux, "Network Synthesis and Optimum network Design Problems: Models, Solution Methods and applications", *Network*, **19**, 1989, 313–360.

[28] M. Minoux, J. Y. Serreault, "Synthèse optimale d'un réseau de télécommunication avec contraintes de sécurité", *Annales des Télécommunications*, **36**, 1981, pp. 211–230.

[29] M. Minoux, J. Y. Serreault, "Subgradient optimization and large scale programming: application to multicommodity network synthesis with security constraints", *R.A.I.R.O.*, **15**, 1981, pp. 185–203.

[30] N. Megiddo, "Pathways to the Optimal Set in Linear Programming", in N. Megiddo, ed., *Progress in Mathematical Programming: Interior Point and Related Methods*, Springer–Verlag, 131–158, New York 1989.

[31] Y. Nesterov, "Cutting plane algorithms from analytic centers: efficiency estimates", 1992 to appear in "Nondifferentiable and Large Scale Optimization",

Mathematical Programming, Series B, J.-L. Goffin and J.-P. Vial, eds.

[32] R. Sarkissian, "Un générateur aléatoire de réseaux de télécomunication pour le problème de sécurisation", LOGILAB, HEC-Section of Management Studies, University of Geneva, Switzerland, 1994.

[33] P. Soriano, C. Wynants, R. Seguin, M. Labbé, M. Gendreau and B. Fortz, "Design and dimensioning of survivable SDH/SONET networks", in B. Sanso and P. Soriano (eds), T̂elecommunications Network Planning, 1998, 169–200, Kluwer, Norwell, MA.

[34] T.H. Wu, D.J. Kolar and R.H. Cardwell, "Survivable network architectures for broadband fiber optic networks: models and performance comparison. ÎEEE Journal of Lightwave technology, **6**, 1698–1709.