

AN APPROACH TO ROBUST NETWORK DESIGN IN TELECOMMUNICATIONS

GEORGIOS PETROU¹, CLAUDE LEMARÉCHAL² AND
ADAM OUOROU¹

Abstract. In telecommunications network design, one of the most frequent problems is to adjust the capacity on the links of the network in order to satisfy a set of requirements. In the past, these requirements were demands based on historical data and/or demographic predictions. Nowadays, because of new technology development and customer movement due to competitiveness, the demands present considerable variability. Thus, network robustness w.r.t demand uncertainty is now regarded as a major consideration. In this work, we propose a min-max-min formulation and a methodology to cope with this uncertainty. We model the uncertainty as the convex hull of certain scenarios and show that cutting plane methods can be applied to solve the underlying problems. We will compare Kelley, Elzinga-Moore and bundle methods.

Keywords. Telecommunications network design, robust optimization, min-max-min problems, cutting plane methods.

Mathematics Subject Classification. 65K05, 90C26, 90B12.

1. INTRODUCTION

In the present competitive markets, robustness is regarded as a major consideration in telecommunication networks. The demand presents considerable variability

Received February 9, 2006. Accepted February 26, 2007.

¹ France Télécom Division R&D, MCN-OTT, 38-40 rue du Général Leclerc, 92794 Issy-Les-Moulineaux Cedex 9, France; {georgios.petrou, adam.ouorou}@orange-ftgroup.com

² Inria, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France;
Claude.Lemarechal@inrialpes.fr

© EDP Sciences, ROADEF, SMAI 2007

because of customers movement, introduction of new services and product development. Historical data for predicting the future is inadequate since the area is constantly changing or because there is no history for new services. In this work, we study the problem of assigning capacity to the links of a network in order to satisfy a set of requirements. We consider that the demand is an uncertain parameter and propose a new formulation and a methodology to cope with this uncertainty. To model the uncertainty, the demand is assumed to belong to an uncertainty set and we seek for a capacity assignment of the links that is *good enough* for every possible demand in the uncertainty set. Thus, our approach fits in the framework of robust optimization as termed in [2, 16]. We analyze the case where the uncertainty set is the convex hull of given demand scenarios. Our approach can be viewed as worst-case oriented since we consider a min-max-min criterion which will result in conservative decisions, with the purpose of contributing to the learning process of decision makers by seeking solutions that hedge against demand uncertainty.

The paper is organized as follows. After reviewing the literature in Section 2, we present our min-max-min formulation for the general case in Section 3. In Section 4, we introduce the resolution procedures when the uncertainty set is described by the convex hull of a reasonable number of scenarios. The computational results are reported in Section 5, and Section 6 concludes this study.

2. RELATED WORKS

Stochastic programming and robust optimization are mathematical tools to deal with uncertainty. They have been both used for telecommunication network design under uncertainty.

A stochastic approach for the network design problem with uncertain demand is presented in [23]. Sen *et al.* deal with the problem under the budget constrained approach. They formulate a two-stage stochastic linear program and they solve it with the use of *stochastic decomposition* [10]. This method is based on sampling which has asymptotic properties and a statistically motivated stopping rule. The first stage problem minimizes the expected unserved demands and provides the capacity of the graph, while the second stage problem strives for an efficient use of the capacity. The model is validated with the use of simulation, and it is also noted that the proposed solution is better than replacing the uncertain variables by their expected values.

Another stochastic optimization method is presented by Lisser *et al.* [19]. Scenario representation is used in order to formulate the uncertainty of the demand, and every scenario corresponds to a given probability for the possible realization. The resulting non-smooth problem is solved with the *analytic center cutting plane* method [9]. The subproblem is decomposed into multicommodity flow problems by demand scenarios. The authors take advantage of this formulation by using parallel computing.

Italiano *et al.* [12] are concerned with the problem of reserving capacity in a network in order to satisfy pairwise demands. To remove uncertainty of the demands they use terminal nodes and bounds for the traffic between the terminals. This model is known as *hose model*. The solution has to satisfy any traffic matrix for the demand, and the capacity vector should form a tree. Hence, the flows are not split, and for any traffic matrix the flow of every demand uses the same path.

Duffield *et al.* [6] have studied a similar problem, but they concentrate on Virtual Private Networks (VPN) based on the hose model. Under this concept they take into account only the ingress and the egress traffic for each endpoint of the VPN, which is more easily specified compared to the conventional point-to-point approach, and Steiner trees are employed in order to connect the VPN endpoints. However, they consider the capacity assignment on a pre-existing network, which is dynamically resized.

Another study in the framework of the hose model is done by Kumar *et al.* [17]. They propose two new methods for solving the problem of the VPN tree computation: the *breath-first search* algorithm (BFS) and a primal-dual algorithm. They compare these two methods in terms of cost and computational time, with the Steiner tree which is given as solution in [6]. The BFS algorithm was developed initially for the case of symmetric ingress and egress bandwidths and it computes the optimal tree in polynomial time. In the general case the problem is NP-hard, and both algorithms outperform the Steiner tree method.

Ben-Ameur and Kerivin [1] consider VPN networks with a polyhedral model for the demand uncertainty. Their mathematical formulation is built on the arc-path flow formulation and a cutting plane method is devised to deal with the large size of the problem. There are two procedures which identify violated inequalities for the master problem. These inequalities are based on unsatisfied demands and improvement of the value of the objective function by selecting sequentially different paths. The final solution specifies a capacity vector, the supporting paths of the demands, and the splitting coefficients which are independent of the possible values of the demands.

In [22] Ouorou proposes three models for a robust capacity assignment in telecommunication networks with demand uncertainty in the framework of robust optimization as defined by Kouvelis and Yu [16]. The algorithmic solutions are based on cutting plane methods. Some computational experiments indicate that the Elzinga-Moore cutting plane method [7] can be a more valuable choice when compared with Kelley's method [13]. Since different possible uncertainty sets may exist in some circumstances, a generalization of these models is proposed in order to cope with a finite number of plausible uncertainty sets, and a weight is associated with each uncertainty set to determine its relative importance or worth.

3. PROBLEM FORMULATION

We follow the approach introduced in [20] by Lucertini and Paletta, who studied a similar problem in which the investment for installing the capacity is limited by

a fixed budget. Suppose that we have a telecommunication network represented by a graph $G = (V, E)$ where V is the set of nodes (*e.g.* terminals) and E is the set of edges (*e.g.* optical links). We also have a set K of pairs of nodes, which are defined as origin-destination pairs (OD-pairs). The commodities that have to be transferred between the OD-pairs are not given but they belong to an uncertainty set Ω . Let $\Gamma(x)$ be the set of all possible demand vectors which can be satisfied given a capacity vector x . A robust decision can be obtained by minimizing a function $\phi(\Omega, \Gamma(x))$ with respect to the capacity vector x , where ϕ is an appropriate measure of the set of unsatisfied demands. In this study we suppose that the feasible capacities are continuous, as in [23]. If $c \in \mathbb{R}_+^{|E|}$ is the cost vector, then

$$X := \left\{ x \in \mathbb{R}_+^{|E|} : c^T x \leq B \right\} \quad (1)$$

will denote the set of feasible capacities available for the network, under a given budget B . The problem we consider is as follows:

$$\min_{x \in X} \phi(\Omega, \Gamma(x)).$$

To define the measure ϕ , we introduce a penalty function $f(\omega, \gamma)$, which expresses the gap between a demand vector ω that the graph *has to* support and a demand vector γ that the graph *can* support. For any capacity vector $x \in X$, we set

$$\phi(\Omega, \Gamma(x)) := \max_{\omega \in \Omega} \min_{\gamma \in \Gamma(x)} f(\omega, \gamma)$$

and we formulate the problem as

$$\min_{x \in X} \phi(\Omega, \Gamma(x)), \quad \text{i.e.} \quad \min_{x \in X} \max_{\omega \in \Omega} \min_{\gamma \in \Gamma(x)} f(\omega, \gamma). \quad (2)$$

There are several ways to choose the function f . In this study, we consider that it is suitable to take as penalty function the weighted sum of unsatisfied commodities. So, if π_k is the penalty for not satisfying one unit of demand for the k^{th} OD-pair, then we set f to be

$$f(\omega, \gamma) := \sum_{k \in K} \pi_k (\omega_k - \gamma_k)^+,$$

where $(y)^+ := \max(0, y)$. As for $\Gamma(x)$, we introduce the following notation: P_k is the set of the possible paths considered to support the demand between the k^{th} OD-pair, $z_{k,p}$ is the flow of path $p \in P_k$, γ is the vector of demands which can be carried through the graph given the capacity vector x . Then,

$$\Gamma(x) = \left\{ \gamma \in \mathbb{R}_+^{|K|} : \exists z \in \mathbb{R}_+^{|P|}, \sum_{k \in K} \sum_{p \in P_k : j \in p} z_{k,p} \leq x_j, \forall j \in E, \sum_{p \in P_k} z_{k,p} = \gamma_k, \forall k \in K \right\}.$$

The first constraint implies that the total flow on every path does not exceed the corresponding capacity, and the second constraint indicates that the vector of demands γ is supported from the flow z .

Introducing $\delta_k := (\omega_k - \gamma_k)^+$ for every $k \in K$, problem (2) can be rewritten as:

$$\begin{aligned} \min_{x \in X} \max_{\omega \in \Omega} \min_{\gamma, \delta, z} \quad & \sum_{k \in K} \pi_k \delta_k \\ \text{s.t.} \quad & \sum_{k \in K} \sum_{p \in P_k: j \in p} z_{k,p} \leq x_j, \quad j \in E, \\ & \sum_{p \in P_k} z_{k,p} = \gamma_k, \quad k \in K, \\ & \delta_k \geq \omega_k - \gamma_k, \quad k \in K, \\ & \gamma, \delta, z \geq 0, \end{aligned}$$

or in a compact form as

$$\begin{aligned} \min_{x \in X} \max_{\omega \in \Omega} \min_{\gamma, \delta, z} \quad & \pi^T \delta \\ \text{s.t.} \quad & Az \leq x, \\ & Fz = \gamma, \\ & \delta \geq \omega - \gamma, \\ & \gamma, \delta, z \geq 0, \end{aligned} \tag{3}$$

where A and F are suitable 0-1 matrices. This formulation is defined as *arc-path formulation*.

Remark. Since we assume that the supporting paths for the commodities are not restricted, we consider in our implementation the following equivalent formulation:

$$\begin{aligned} \min_{x \in X} \max_{\omega \in \Omega} \min_{\gamma, \delta, X} \quad & \sum_{k \in K} \pi_k \delta_k \\ \text{s.t.} \quad & \sum_{k \in K} (X_j^{k+} + X_j^{k-}) \leq x_j, \quad j \in E, \\ & \mathcal{A}(X^{k+} - X^{k-}) = \gamma_k l_k, \quad k \in K, \\ & \delta_k \geq \omega_k - \gamma_k, \quad k \in K, \\ & \gamma, \delta, X \geq 0, \end{aligned}$$

where \mathcal{A} is the node-arc incidence matrix of G , X^{k+} and X^{k-} are the “positive” and the “negative” flow vectors – respectively – which are used to satisfy commodity k , and l_k is the vector of $\mathbb{R}^{|V|}$ defined by

$$(l_k)_i = \begin{cases} -1, & \text{if } i \text{ is the } k^{\text{th}} \text{ origin node,} \\ 1, & \text{if } i \text{ is the } k^{\text{th}} \text{ destination node,} \\ 0, & \text{otherwise.} \end{cases}$$

This formulation is defined as *node-arc formulation*.

4. SOLUTION METHODS

Problem (3) is difficult because of non-convexity (due to the internal min) and nonsmoothness (due to the intermediate max). We refer to [5] for a discussion

of these difficulties. However, an algorithm can be constructed for the solution of (3) when Ω is a bounded polyhedron with a limited number of extreme points. Thus we limit our study to the case where Ω is defined as the convex hull of $|S|$ scenarios $\omega^1, \dots, \omega^{|S|}$ of possible demands; then the extreme points of Ω will be among these scenarios.

For fixed x and ω in (3), the inner linear minimization problem is compactly written as

$$\begin{aligned} \min_{\delta, z} \quad & \pi^T \delta \\ \text{s.t.} \quad & Az \leq x, \\ & \delta \geq \omega - Fz, \\ & \delta, z \geq 0, \end{aligned}$$

whose dual problem is

$$\begin{aligned} P(x, \omega) := \max_{u, v} \quad & \omega^T u - x^T v \\ \text{s.t.} \quad & F^T u \leq A^T v, \\ & u \leq \pi, \\ & u, v \geq 0. \end{aligned} \tag{4}$$

We have substituted Fz for γ and eliminated the resulting constraint $Fz \geq 0$ since z and F are positive by definition. By LP duality the above two programs have the same optimal value. Thus (3) is equivalent to solving

$$\min_{x \in X} \max_{\omega \in \Omega} P(x, \omega).$$

Note that $P(x, \omega)$ is easily computed for every (x, ω) through a linear maximization program.

Setting

$$Q(x) := \max_{\omega \in \Omega} P(x, \omega)$$

we have to minimize Q over X and the next result gives crucial properties of function Q .

Theorem 1. *Function Q is convex and we have*

$$Q(x) = \max_{i=1,2,\dots,|S|} \mathcal{P}_i(x), \quad \text{where } \mathcal{P}_i(x) := P(x, \omega^i), \quad i = 1, 2, \dots, |S|. \tag{5}$$

Proof. Let $g_{u,v} : X \times \Omega \mapsto \mathbb{R}$ be the linear function $g_{u,v}(x, \omega) = u^T \omega - v^T x$, where

$$(x, \omega) \in X \times \Omega \quad \text{and} \quad (u, v) \in U \times V = \{(u, v) : F^T u \leq A^T v, u \leq \pi, u, v \geq 0\}.$$

We have that $X \times \Omega \neq \emptyset$ for every $(u, v) \in U \times V$, so $g_{u,v}$ is well-defined. Moreover $g_{u,v}$ is convex as a linear function, and for an arbitrary $(x', \omega') \in (X, \Omega)$ we have:

$$\max_{(u,v) \in U \times V} g_{u,v}(x', \omega') \leq \sum_{k \in K} \pi_k \omega'_k < +\infty.$$

So, if we rewrite P as

$$P = \max_{(u,v) \in U \times V} g_{u,v},$$

then we deduce that P is convex as the maximum of a family of convex functions (see [11], Prop. IV.2.1.2).

Now, the convex function $P(x, \cdot)$ attains its maximum at some extreme point of the set Ω (see [11], Prop. III.2.4.6), and these extreme points are among the $|S|$ possible scenarios. Hence:

$$\max_{\omega \in \Omega} P(x, \omega) = \max \left\{ P(x, \omega^1), P(x, \omega^2), \dots, P(x, \omega^{|S|}) \right\}.$$

This establishes (5), and also shows that $Q(x) < +\infty$ for all $x \in X$. Thus Q is convex for the same reason as P . \square

Then the convex nonsmooth problem

$$\min_{x \in X} Q(x) \tag{6}$$

is equivalent to (3).

We will analyze three cutting plane algorithms for the solution of (6). The basic idea underlying these algorithms can be described as follows. Having computed t sets of $|S|$ values $\mathcal{P}_i(x_k)$, $i = 1, \dots, |S|$ and corresponding subgradients $g_{ki} = -v_{ki}$ (see Th. 2 below) for $i = 1, \dots, |S|$ and $k = 1, \dots, t$, the true function Q is approximated (from below) by the polyhedral function

$$x \mapsto \max \{ \mathcal{P}_i(x_k) + g_{ki}^T(x - x_k) : i = 1, \dots, |S|, k = 1, \dots, t \}. \tag{7}$$

A cutting plane method uses this function to select a new point x_{t+1} , so as to improve the current approximation (7). The way this point is chosen determines whether and how fast the algorithm converges.

Theorem 2. For fixed $x_0 \in X$ and $i \in \{1, \dots, |S|\}$, let (u_{0i}, v_{0i}) be an optimal solution of $\mathcal{P}_i(x_0)$. Then $-v_{0i} \in \partial \mathcal{P}_i(x_0)$.

Proof. The theorem results from elementary convex calculus, but a simple direct proof can be given:

If (u_{0i}, v_{0i}) is an optimal solution of the mathematical programming problem $P(x_0, \omega^i)$ of (4), then $\mathcal{P}_i(x_0) = u_{0i}^T \omega^i - v_{0i}^T x_0$. Let (u_{si}, v_{si}) be the optimal solution of $\mathcal{P}_i(x)$ for an arbitrary $x \in X$. Since the constraints of (4) do not depend on x , (u_{0i}, v_{0i}) is feasible for problem $P(x, \omega^i)$ and

$$\begin{aligned} \mathcal{P}_i(x) = u_{s_i}^T \omega^i - v_{s_i}^T x &\geq u_{0_i}^T \omega^i - v_{0_i}^T x \\ &= \mathcal{P}_i(x_0) - v_{0_i}^T(x - x_0), \quad \forall x, x_0 \in X, i \in \{1, \dots, |S|\}. \end{aligned}$$

Hence $-v_{0i} \in \partial \mathcal{P}_i(x_0)$. \square

We consider and compare the three following cutting plane methods, which differ in the way they construct the trial points x_k . They all use an *oracle* which,

at a given x , solves (4) $|S|$ times to compute the $\mathcal{P}_i(x)$'s and then $Q(x)$, as well as the optimal v_i 's to provide the subgradients stipulated in Theorem 2. The optimal v_i 's computed at $x = x_k$ will be denoted by v_{ki} for $i = 1, \dots, |S|$.

4.1. KELLEY CUTTING PLANE METHOD

This is one of the first cutting plane methods proposed in the literature. The next trial point x_{t+1} is obtained by solving the following relaxed problem (Master Problem) of (6):

$$\begin{aligned} \min_{x, \underline{Z}} \quad & \underline{Z} \\ \text{s.t.} \quad & \underline{Z} \geq \mathcal{P}_i(x_k) - v_{ki}^T(x - x_k), \quad i = 1, 2, \dots, |S|, \quad k = 0, 1, \dots, t, \\ & c^T x \leq B, \\ & x \geq 0. \end{aligned} \quad (8)$$

The method iterates until a “good” approximation of the solution is found, and the resulting algorithm can be stated as follows:

KELLEY ALGORITHM (KA)

Step 0 Initialize $x_0 = 0$, $\varepsilon > 0$, $\bar{Z} = +\infty$ and $t = 0$.

Step 1 Call the oracle at x_t to obtain $\mathcal{P}_i(x_t)$ and v_{ti} for $i = 1, \dots, |S|$, and $Q(x_t)$.

Step 2 Let $\bar{Z} = \min\{\bar{Z}, Q(x_t)\}$.

Step 3 Solve (8) and let (x_{t+1}, \underline{Z}) be its optimal solution.

Step 4 If $\bar{Z} - \underline{Z} \leq \varepsilon(1 + |\bar{Z}|)$, then stop. Otherwise set $t = t + 1$ and loop to Step 1.

This procedure converges, and the proof is given in [13]. Note that X is bounded, so (8) has always an optimal solution. This substantially simplifies the issue (see [11], Th. XII.4.2.3). However, if convergence is slow, then the Master Problem can be uncomfortably large. For further discussion see Sections 5 and 6.

4.2. ELZINGA-MOORE CUTTING PLANE METHOD

This method is based on the following result by Nemhauser and Widhelm [21].

Theorem 3. *Given a bounded, non-empty polyhedron described from a set of linear inequalities,*

$$\alpha_j^T x \geq \beta_j, \quad j = 1, \dots, p,$$

the optimal solutions in σ and x of the problem

$$\begin{aligned} \max_{\sigma, x} \quad & \sigma \\ \text{s.t.} \quad & \alpha_j^T x - \|\alpha_j\| \sigma \geq \beta_j, \quad j = 1, \dots, p, \end{aligned}$$

are respectively the radius and the center of the largest sphere inscribed in the polyhedron.

Proof. See [21]. □

Naturally, this result has a meaning only for a bounded polyhedron. Now consider in the (x, \mathcal{Z}) -space the polyhedron defined by the constraints in (8) – *i.e.* the epigraph of the current approximation (7). It is unbounded but it can be truncated from above by appending the constraint $\mathcal{Z} \leq \overline{\mathcal{Z}}$. The polyhedron thus obtained clearly contains any optimal solution $(x^*, Q(x^*))$ of (6): the center revealed by Theorem 3 may be deemed an adequate approximation of such an optimal solution. This is the rationale for Elzinga-Moore cutting plane method, which computes the next trial point by solving

$$\begin{aligned}
 \max_{x, \sigma, \mathcal{Z}} \quad & \sigma \\
 \text{s.t.} \quad & \mathcal{Z} + \sigma \leq \overline{\mathcal{Z}} \\
 & \mathcal{P}_i(x_k) - v_{ki}^T(x - x_k) - \mathcal{Z} + (\|v_{ki}\|^2 + 1)^{\frac{1}{2}}\sigma \leq 0, \\
 & \quad i = 1, 2, \dots, |S|, k = 0, 1, \dots, t, \\
 & c^T x \leq B, \\
 & x \geq 0.
 \end{aligned} \tag{9}$$

Note again that this linear program has a non-empty bounded feasible set, and therefore an optimal solution $(x_{t+1}, \sigma_{t+1}, \mathcal{Z}_{t+1})$ (with $\sigma_{t+1} > 0$). This method has little usage in the literature even though it is not much harder to be implemented than Kelley’s method. It was shown to be more efficient than Kelley’s for some min-max problems, see [22]. Elzinga-Moore cutting plane algorithm for the solution of (6) is as follows:

ELZINGA-MOORE ALGORITHM (EMA)

Step 0 Initialize $x_0 = 0$, $\varepsilon > 0$, $\overline{\mathcal{Z}} = +\infty$ and $t = 0$.

Step 1 Call the oracle at x_t to obtain $\mathcal{P}_i(x_t)$ and v_{ti} for $i = 1, \dots, |S|$, and $Q(x_t)$.

Step 2 Let $\overline{\mathcal{Z}} = \min \{\overline{\mathcal{Z}}, Q(x_t)\}$.

Step 3 Solve (9) to get x_{t+1} and σ_{t+1} .

Step 4 If $\sigma_{t+1} \leq \varepsilon$, then stop. Otherwise set $t = t + 1$ and loop to Step 1.

For further details about this procedure and the proof of its convergence, see [7].

4.3. BUNDLE METHOD

Kelley cutting plane method is known to be unstable. Bundle methods [11, 14, 18] aim at overcoming this instability by computing the Moreau-Yosida regularization of the piecewise linear approximation of (7). More precisely, the master problem to consider is obtained by adding a quadratic term to (8) as follows:

$$\begin{aligned}
 \min_{x, \mathcal{Z}} \quad & \mathcal{Z} + \frac{\mu_t}{2} \|x - \hat{x}_t\|^2 \\
 \text{s.t.} \quad & \mathcal{Z} \geq \mathcal{P}_i(x_k) - v_{ki}^T(x - x_k), i = 1, 2, \dots, |S|, k = 0, 1, \dots, t, \\
 & c^T x \leq B, \\
 & x \geq 0,
 \end{aligned} \tag{10}$$

where \hat{x}_t is the *stability center*: a point which is known to be relatively good; μ_t is a positive parameter which controls the tradeoff between minimizing \mathcal{Z} and staying close to \hat{x}_t . The stability center is updated if x_{t+1} is significantly better than \hat{x}_t in the sense that

$$Q(x_{t+1}) \leq Q(\hat{x}_t) - \kappa(Q(\hat{x}_t) - \mathcal{Z}_{t+1}), \quad (11)$$

where $(x_{t+1}, \mathcal{Z}_{t+1})$ is the solution of (10), and $\kappa \in]0, 0.5[$. If (11) holds, then we have a *descent step* and set $\hat{x}_{t+1} = x_{t+1}$, otherwise we have a *null step* and the stability center is left as it is. The resulting cutting planes are added to (10) in both cases. The bundle algorithm (BA) stops when

$$Q(\hat{x}_t) - \mathcal{Z}_{t+1} \leq \varepsilon(1 + |Q(\hat{x}_t)|), \quad (12)$$

where ε is the desired accuracy.

BUNDLE ALGORITHM (BA)

Step 0 Initialize $x_0 = \hat{x}_0 = 0$, $\varepsilon > 0$, $\kappa \in]0, 0.5[$ and $t = 0$. Call the oracle at x_0 to obtain $\mathcal{P}_i(x_0)$ and v_{0i} for $i = 1, \dots, |S|$, and $Q(x_0)$.

Step 1 Solve (10) and let $(x_{t+1}, \mathcal{Z}_{t+1})$ be its optimal solution.

Step 2 Call the oracle at x_{t+1} to obtain $\mathcal{P}_i(x_{t+1})$ and $v_{t+1,i}$ for $i = 1, \dots, |S|$, and $Q(x_{t+1})$.

Step 3 If (11) is true then set $\hat{x}_{t+1} = x_{t+1}$, otherwise set $\hat{x}_{t+1} = \hat{x}_t$.

Step 4 Compute μ_{t+1} .

Step 5 If (12) is true then stop. Otherwise set $t = t + 1$ and loop to Step 1.

Remark. Step 3 is defined as *weight updating* and it is crucial for the speed of convergence of the algorithm. We use a procedure described in [14], where the weight μ_t is decreased if the approximation of Q is close to Q at x_{t+1} , and it is increased if the errors of the new linearizations are greater than a variation estimate of $Q(\hat{x}_t)$. See [14] for further details.

5. RESULTS AND NUMERICAL EXPERIMENTS

We consider the above three methods for the following reasons. Kelley's method is one of the most methods used in nonsmooth optimization while Elzinga-Moore algorithm has been, to the best of our knowledge, used rarely even if it is not much more hard to implement than the former. It outperformed Kelley's method on some min-max problems arising in robust capacity planning problems, see [22]. Our first aim was to confirm this efficiency. We also consider the proximal bundle method because it is known as one of the best methods for nonsmooth optimization and we think it is extremely informative to compare it with the two above methods. Its subproblem is a quadratic problem which is considered to more difficult than the linear subproblems of Kelley and Elzinga-Moore methods.

We have written an experimental code based on the above development, under Eclipse Platform 2.1.1; we have used Java 1.4.2 and ILOG Cplex 9.0 for the solution

TABLE 1. Graphs specifications.

	Nodes	Edges	OD-pairs	Scenarios
Graph 1	12	25	50	21
Graph 2	19	34	50	21
Graph 3	26	30	100	21
Graph 4	16	49	89	10
Graph 5	26	53	100	21
Graph 6	60	40	140	11

of the subproblems (4), (8), (9) and (10). All the runs are performed on a two-processor Xeon Intel server of 2.4 GHz CPU speed and 1.5 GB of RAM memory, running under Linux.

The networks of our experiments derive from actual networks with given nominal demands. Their sizes are given in Table 1, as well as the numbers of OD-pairs and the number $|S|$ of scenarios considered in each case. Those scenarios have been generated in two ways as follows:

- (1) From the available nominal vector of demands ω and a positive parameter ρ , we have generated random $\epsilon_k^s \in [-\rho, \rho]$ for $s = 1, \dots, |S|$ and $k = 1, \dots, |K|$ and set

$$\omega_k^s = \epsilon_k^s \omega_k, \quad k = 1, 2, \dots, |K|.$$

- (2) From the nominal vector of demands ω we have generated random $\epsilon_k^s \in [0, 1]$ for $s = 1, \dots, |S|$ and $k = 1, \dots, |K|$, and set

$$\omega_k^s = \epsilon_k^s \frac{\sum_j \omega_j}{\sum_j \epsilon_j^s}, \quad k = 1, 2, \dots, |K|.$$

The first case results in scenarios that are variations around the nominal demand, and we use it for graphs 1, 2, 3, 4 and 6. In the second case the sum of the demands is the same for every scenario, and we use it for graphs 5 and 6. Finally, the nominal demand is included in the set of scenarios.

In Table 2 we report the results obtained by the three algorithms presented in Section 4. The column headed “Budget” specifies the parameter B in each test problem, see (1). This budget is derived arbitrarily, having as indicators the solutions of the multicommodity flow problems, formed by every graph and its corresponding scenarios. Different values for the budget are tested, but it is always requested that finally at least one objective value equals to 0 (zero) and various positive solutions for every graph are acquired. Then, for every method we report the value of the objective function, the number of oracle calls and the CPU time in seconds. Graph 6 appears twice because both scenario generation methods 1 and 2 were used for it (the notations 6a and 6b correspond to the first and second method respectively).

TABLE 2. Results obtained by the different algorithms.

Budget	KA				EMA				BA			
	obj. value	# oracle calls	CPU time	obj. value	# oracle calls	CPU time	obj. value	# oracle calls	CPU time			
Graph 1	612.3	63.33	90	18.45	63.33	85	23.63	63.33	16	5.21		
	614.0	44.44	94	21.47	44.44	79	20.07	44.44	15	4.21		
	616.0	22.22	85	17.26	22.22	71	17.23	22.22	15	4.45		
618.0	0.00	94	22.12	0.00	75	16.23	0.00	14	3.67			
Graph 2	190.0	94.44	189	239.33	94.44	120	159.69	94.44	17	35.24		
	194.0	50.00	189	177.94	50.00	113	114.14	50.00	17	29.26		
	197.0	16.67	180	151.40	16.67	95	84.29	16.67	20	35.31		
	199.0	0.00	165	159.94	0.00	87	68.72	0.00	13	18.26		
Graph 3	12165.0	97.22	179	280.11	97.25	138	221.64	97.22	39	93.48		
	12175.0	41.67	181	274.68	41.67	144	240.34	41.67	41	103.16		
	12182.0	2.78	202	307.01	2.79	137	222.82	2.78	47	112.26		
12187.0	0.00	173	269.13	0.00	128	201.06	0.00	43	102.38			
Graph 4	7380.0	389.76	265	2049.12	389.77	167	1167.71	389.76	35	289.49		
	7395.0	202.26	181	1224.03	202.28	161	1120.62	202.26	39	335.48		
	7408.0	39.76	172	1201.59	39.78	160	1038.69	39.76	36	311.50		
	7415.0	0.00	244	2244.69	0.00	130	1025.08	0.00	38	303.00		
Graph 5	8150.0	1930.00	687	15131.10	1930.01	664	16826.14	1930.00	60	2266.53		
	8300.0	612.50	479	8628.33	612.51	288	5238.21	612.50	49	922.21		
	8375.0	46.43	425	7437.13	46.43	257	3785.32	46.43	52	815.31		
8390.0	0.00	281	3180.40	0.00	254	3882.62	0.00	37	506.80			
Graph 6a	40000.0	152.91	1176	22616.64	152.91	581	14216.79	152.91	216	16438.04		
	40400.0	82.65	1111	23615.07	82.65	533	11110.81	82.65	198	13079.14		
	40800.0	17.10	1064	19324.76	17.11	541	10882.63	17.10	167	9633.43		
	40900.0	2.13	1155	23065.01	2.14	562	9485.44	2.13	160	9878.38		
41000.0	0.00	1157	22113.83	0.00	525	7134.68	0.00	144	8663.29			
Graph 6b	45000.0	121.38	659	24628.16	121.38	325	8103.27	121.38	125	13348.54		
	45600.0	39.44	633	19361.97	39.44	345	7021.62	39.44	93	8893.14		
	45800.0	13.68	643	24012.30	13.68	355	7260.20	13.68	106	9383.37		
	45900.0	0.87	720	25288.75	0.87	371	12712.30	0.87	95	8986.39		
45920.0	0.00	609	16217.29	0.00	330	7134.05	0.00	100	9657.57			

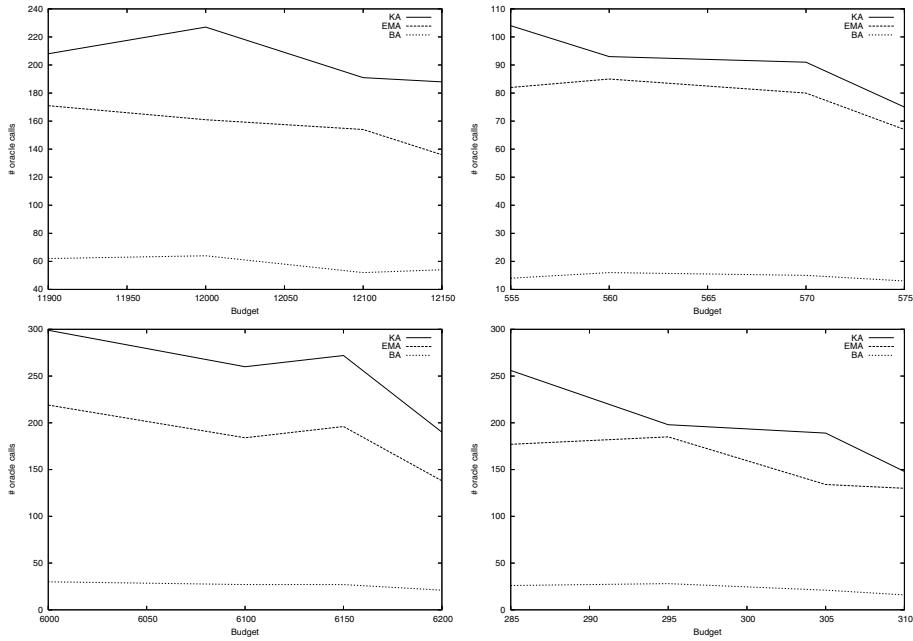


FIGURE 1. Illustration of the number of oracle calls for the first 4 graphs.

It appears from these results that, in terms of number of calls to the oracle, the bundle algorithm outperforms the other two methods: it divides the number of Kelley’s [resp. Elzinga-Moore] oracle calls by a factor of 4.2 to 12.7 [resp. 2.6 to 11.1]. However, this reduction is not reflected in the CPU times: in particular for large problems, the bundle algorithm can become substantially slower than Elzinga-Moore. The reason is that the quadratic subproblem in our implementation becomes time consuming when the sizes of the problem and of the bundle increase. This behaviour contradicts the experiments of [4], which clearly show that quadratic programming is hardly more expensive than linear programming, and thus demonstrates the usefulness of specialized algorithms such as [8, 15]. On the other hand, Elzinga-Moore method behaves efficiently when compared to Kelley’s, as reported in [22]. We hope that the readers will be encouraged to test this method when only a linear solver is at their disposal. Figure 1 illustrates in another way the above observations, where we plot the number of oracle calls vs different budget for the first four graphs using other 20 traffic scenarios in each case.

Using the first four graphs, we conduct some numerical experiments with the proximal bundle algorithm, to analyse the model in terms of number of scenarios and budget. The number of scenarios to be considered in the model is itself a difficult issue which must be considered at a first stage before applying the model. Interesting studies about the demand modeling exist in the literature, (see for instance [3]) and can be used to simulate the demand and provide the scenarios.

TABLE 3. Results with different number of scenarios.

	Budget	obj. value	# oracle calls	CPU time
Graph 1	5 scenarios			
	535	234.54	22	2.79
	540	156.21	20	2.81
	550	13.36	19	2.46
	555	0.00	14	1.92
	10 scenarios			
	550	217.79	17	4.65
	555	154.37	15	3.00
	565	37.83	14	2.71
	570	0.00	13	2.17
	20 scenarios			
	555	199.39	14	4.40
	560	128.23	16	4.89
	570	8.22	15	4.11
575	0.00	13	3.10	
Graph 2	5 scenarios			
	280	288.92	30	8.66
	290	164.68	26	7.59
	300	50.56	29	8.58
	305	0.00	27	7.47
	10 scenarios			
	285	235.79	21	13.53
	295	116.86	18	11.57
	305	5.74	20	8.93
	310	0.00	17	6.77
	20 scenarios			
	285	276.43	26	27.70
	295	151.78	28	34.57
	305	34.82	21	17.42
310	0.00	16	10.80	
Graph 3	5 scenarios			
	11650	1300.07	65	73.96
	11750	704.29	59	70.05
	11850	132.36	51	49.61
	11900	0.00	48	38.66
	10 scenarios			
	11700	1337.42	50	103.86
	11800	730.56	43	77.27
	11900	151.83	48	82.02
	11950	0.00	38	52.13
	20 scenarios			
	11900	1368.16	62	262.72
	12000	743.16	64	245.15
	12100	180.17	52	104.06
12150	0.00	54	101.67	
Graph 4	5 scenarios			
	5900	2776.79	29	103.03
	6000	1178.93	40	160.77
	6050	552.19	31	120.31
	6100	0.00	33	101.92
	10 scenarios			
	6000	2158.93	29	164.43
	6100	853.38	26	139.90
	6150	228.38	27	126.68
	6200	0.00	25	97.01
	20 scenarios			
	6000	2224.75	30	427.17
	6100	835.63	27	315.88
	6150	210.63	27	272.93
6200	0.00	21	181.50	

The greater the number of scenarios is, the better the uncertainty set minimize the forecast error. However, some traffic scenario may dominates others. One of the main features of the proximal bundle method is its small number of oracle calls that do not depend on the number of scenarios in hand.

6. CONCLUSIONS

We have proposed a new approach for the network design problem in telecommunications under demand uncertainty. The uncertainty is modelled as a convex set of a moderate number of demand's scenarios, which results in a convex non-smooth problem. For its solution, we considered and compared three algorithms: the proximal bundle method and the cutting plane algorithms by Kelley and Elzinga-Moore. The proximal bundle method appears to be the most efficient in terms of number of calls to the oracle, but is impeded by the use of a general-purpose quadratic solver. An extension of our proposed methodology to the general case of a polyhedral uncertainty set is currently under study.

REFERENCES

- [1] W. Ben-Ameur and H. Kerivin, Routing of uncertain traffic demands. *Optim. Eng.* **3** (2005) 283–313.
- [2] A. Ben-Tal and A. Nemirovski, Robust convex optimization. *Math. Oper. Res.* **23** (1998).
- [3] M. Bonatti, A. Gaivoronski, A. Lemonche and P. Polese, Summary of some traffic engineering studies carried out within RACE project R1044. *European Transactions on Telecommunications* **5** (1994) 79–90.
- [4] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot and F. Vanderbeck, Technical Report 5453, Inria, 2005. Accepted for publication at *Math. Program.*
- [5] A.V. Demyanov, V.F. Demyanov and V.N. Malozemov, Minimaxmin problems revisited. *Optim. Methods Softw.* **17** (2002) 783–804.
- [6] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan and J.E. van der Merwe, Resource management with hoses: Point-to-cloud services for virtual private networks. *IEEE/ACM Trans. Networking* **10** (2002) 679–692.
- [7] J. Elzinga and T.J. Moore, A central cutting plane algorithm for the convex programming problem. *Math. Program.* **8** (1975) 134–145.
- [8] A. Frangioni, Solving semidefinite quadratic problems within nonsmooth optimization algorithms. *Comput. Oper. Res.* **23** (1996) 1099–1118.
- [9] J.-L. Goffin, A. Haurie and J.-Ph. Vial, Decomposition and nondifferentiable optimization with the projective algorithm. *Manage. Sci.* **38** (1992) 284–302.
- [10] J.L. Higle and S. Sen, Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Math. Oper. Res.* **16** (1991) 650–669.
- [11] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*. Springer-Verlag, Berlin (1993).
- [12] G.F. Italiano, S. Leonardi and G. Oriolo, Design of networks in the hose model, in *2nd International Workshop on Approximation and Randomization Algorithms in Communication Networks*, Carleton Scientific Press (2002) 65–76.
- [13] J.E. Kelley, The cutting plane method for solving convex programs. *J. Appl. Math. SIAM* **8** (1960) 703–712.
- [14] K. Kiwiel, Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Program.* **46** (1990) 105–122.

- [15] K. Kiwiel, A cholesky dual method for proximal piecewise linear programming. *Numer. Math.* **68** (1994) 325–340.
- [16] P. Kouvelis and G. Yu, *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers (1997).
- [17] A. Kumar, R. Rastogi, A. Silberschatz and B. Yener, Algorithms for provisioning virtual private networks in the hose model. *IEEE/ACM Trans. Networking* **10** (2002) 565–578.
- [18] C. Lemaréchal, Lagrangian relaxation, in *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions [based on a Spring School]* **2241**, London, UK . Springer-Verlag (2001) 112–156.
- [19] A. Lisser, A. Ouorou, J.-Ph. Vial and J. Gondzio, *Capacity planning under uncertain demand in telecommunication networks*. Technical Report 99.13, Logilab, Department of Management Studies, University of Geneva, Switzerland, October 1999.
- [20] M. Lucertini and G. Paletta, A class of network design problems with multiple demand: Model formulation and an algorithmic approach. *Math. Program. Stud.* **26** (1986) 225–228.
- [21] G.L. Nemhauser and W.B. Widhelm, A modified linear program for columnar methods in mathematical programming. *Oper. Res.* **19** (1971) 1051–1060.
- [22] A. Ouorou, Robust capacity assignment in telecommunications. *Comput. Manage. Sci.* **3** (2006) 285–305.
- [23] S. Sen, R.D. Doverspike and S. Cosares, Network planning with random demand. *Telecommun. Syst.* **3** (1994) 11–30.