DEPARTMENT OF OPERATIONS RESEARCH
UNIVERSITY OF AARHUS

Working Paper no. 2000/5

Capacitated Network Design with Uncertain Demand

Morten Riis
Kim Allan Andersen

ISSN 1398-8964

Department of Mathematical Sciences
Telephone: +45 8942 3188
E-mail: institut@imf.au.dk

Building 530, Ny Munkegade
DK-8000 Aarhus C, Denmark
URL: www.imf.au.dk

Capacitated Network Design with Uncertain Demand

Morten Riis*and Kim Allan Andersen

Department of Operations Research

University of Aarhus, building 530

Ny Munkegade

DK - 8000 Århus C

Denmark

Abstract

We consider a capacity expansion problem arising in the design of telecommunication networks. The problem is to install capacity on links of the network so as to meet customer demand while minimizing total costs incurred. When studying this and related problems it is customary to assume that point to point demands are given. This will not be the case in practice, however, since future demand is generally unknown and the decision must be based on uncertain forecasts. We develop a stochastic integer programming formulation of the problem and propose an L-shaped solution procedure based on well-known cutting plane procedures for the deterministic problem. The algorithm was tested on two sets of real life problem instances and we present results of our computational experiments.

Keywords: Stochastic programming, integer programming, telecommunication networks, capacity expansion.

1. Introduction

Capacity expansion problems is a very important class of problems arising in many contexts. Capacity expansion in telecommunication networks has been the center of particular attention due to the rapid increase in demand which network providers have been facing in recent years. In this paper we consider the design of a capacitated telecommunication network. The problem is to install additional capacity on the edges of the network and route traffic so as to meet customer demand while minimizing total costs incurred. We shall assume that two facilities with fixed capacities are available for installation but most of the results may be generalized if one wishes to consider the situation with several facilities.

*Corresponding author. Email: riis@imf.au.dk

The polyhedral structure of the two-facility capacitated network design problem has been studied by e.g. Bienstock and Günlük [2] and Günlük [5]. Here, facet defining inequalities are derived and used to solve the problem by cutting plane procedures. Similar results have been obtained for the closely related network loading problem by Magnanti, Mirchandani and Vachani [10], [11] and Mirchandani [12]. The network loading problem is a slight variation of the capacitated network design problem in which it is assumed that there is no existing capacity on the edges of the network and no cost of flow. The capacitated network design problem will generally be relevant for the network provider, whereas the network loading problem typically arises when customers wish to design a private line network by leasing transmission facilities from the network provider.

In the literature it has been customary to assume that point to point demands in the network are known when studying these problems. Typically this will not be the case, though, since actual demand is in general unknown at the time the decision is made. Instead, point to point demands should be thought of as depending on the outcome of some random variable. This means that the decision on capacity expansion cannot be based on actual demand. The only information that is available is the partial knowledge of demand conveyed through its distribution. (This may be thought of as some kind of forecast of demand.) It is well known (see e.g. Birge and Loveaux [3]) that the expected value problem, obtained by replacing random demand by its expected value, may not produce very good solutions to the problem. In this paper we will formulate the capacitated network design problem as a two-stage stochastic program with integer first stage and continuous second stage, hence explicitly taking uncertainty into account in the decision process.

Stochastic programming has previously been used as a modelling tool in telecommunications. Sen, Doverspike and Cosares [18] study a capacity expansion problem in which the expected number of unserved requests is minimized subject to limitations on the total capacity expansion. Riis and Andersen [15] use stochastic programming to solve the multiperiod capacity expansion problem of one connection in telecommunications. Finally, Dempster, Medova and Thompson [4] uses chance-constrained programming to solve a capacity expansion problem subject to certain grade of service constraints assuming that the arrival process of calls is known.

There is a vast amount of literature concerning the solution of stochastic programming problems. If no integrality restrictions are imposed on the variables, numerous powerful solution procedures for two-stage stochastic programs with recourse are available. Such procedures include the progressive hedging algorithm introduced by Rockafellar and Wets [16], regularized decomposition introduced by Ruszscynski [17] and stochastic decompostion introduced by Higle and Sen [7]. Regularized decomposition as well as stochastic decomposition may be viewed as specialized versions of the L-shaped procedure, introduced by Van Slyke and Wets [19], which is based on Bender's decomposition principle. The procedure takes advantage of the fact that the second stage value function is convex and piecewise linear on a convex polyhedral domain and hence may be replaced by a finite number of so-called feasibility cuts and optimality cuts.

Less has been said about stochastic integer programming problems which suffer from the hardships of both stochastic programming and integer programming. Still, when only first stage variables are restricted to integer values, as is the case in the problem considered in this paper, the second stage value function remains convex and piecewise linear. This means, that we may adapt the above-mentioned procedures to take integrality restrictions on first stage variables into account. Such an approach has previously been taken by Wollmer [20] who proposed an implicit enumeration scheme to solve two-stage stochastic programs with binary first stage and continuous second stage in an L-shaped algorithm. This approach was extended by Laporte and Loveaux [9] to problems with binary first stage and general, but easily computable, second stage problems.

This paper is organized as follows: in section 2 we will formulate the deterministic capacitated network design problem and present several classes of well-known valid inequalities. Next, in section 3 we formulate the capacitated network design problem as a two-stage stochastic program with integer first stage and continuous second stage. We discuss how the valid inequalities derived for the deterministic problem, generalize for the new formulation and under what conditions they are facet defining for the feasible region of the stochastic problem. An L-shaped solution procedure for the stochastic program is presented in section 4. The seminal idea is to project the feasible region of the problem onto the space of discrete first stage variables and hence the approach is closely related to that followed by Bienstock et al. [2] and by Mirchandani [12]. The projection is built in a master problem by imposing different kinds of cuts. In addition to the well-known optimality cuts and feasibility cuts which are generated through the solution of subproblems, the procedure uses heuristically generated facet defining inequalities as cutting planes in the master problem. Moreover, the L-shaped procedure is combined with a branch and cut scheme to solve the stochastic integer program. The algorithm was implemented in C++ using procedures from the callable library of CPLEX 6.6 and tested on two sets of real life problem instances, previously studied in Bienstock and Günlük [2] and Günlük [5]. The results of our computational experiments are reported in section 5. Finally, in section 6 we summarize our work and give some conclusions.

2. The Capacitated Network Design Problem

The network is modelled as a connected undirected graph $G = (V, E)$ in which directed point to point demands are to be routed. The existing capacity on an edge $\{i, j\} \in E$ is denoted by C_{ij}. Since we are modelling a telecommunication network with optical transmission systems we will assume that a given edge in the network can carry flow in either direction and, more importantly, that these flows do not interfere. Hence, each edge $\{i, j\} \in E$ conceptually corresponds to two directed edges (i, j) and (j, i), each with capacity C_{ij}. Additional capacity may be installed on the edges in multiples of two batch sizes corresponding to low capacity and high capacity facilities respectively. We will assume that the smaller batch size is equal to 1 which may be achieved by rescaling demand. The larger batch size will be denoted by λ and we will assume that λ is an integer.

Demand will be described by a set K of commodities and we let d_{ik} denote the net demand of commodity k at node i ($k \in K, i \in V$). Several possibilities exist for defining the commodities. One common approach is to define a commodity for every point to point demand resulting in a total of $O(|V|^2)$ commodities. We will, however, prefer to work with an aggregated formulation in which a commodity k corresponds to point to point demands with source node k ($k \in V$) resulting in a total of $O(|V|)$ commodities.

We are now ready to define the convex hull of feasible solutions for the capacitated network design problem:

$$\mathcal{P} = \mathrm{conv}\Bigg\{ (x, y, f) \in \mathbb{Z}_+^{|E|} \times \mathbb{Z}_+^{|E|} \times \mathbb{R}_+^{2|K||E|} \;\Bigg|$$

$$\sum_{j:\{i,j\}\in E} f_{jik} - \sum_{j:\{i,j\}\in E} f_{ijk} = d_{ik} \qquad i \in V,\ k \in K,\ i \neq k \tag{2.1}$$

$$\sum_{k\in K} f_{ijk} \leq C_{ij} + x_{ij} + \lambda y_{ij} \qquad \{i, j\} \in E \tag{2.2}$$

$$\sum_{k\in K} f_{jik} \leq C_{ij} + x_{ij} + \lambda y_{ij} \qquad \{i, j\} \in E \quad \Bigg\} \tag{2.3}$$

Here, x_{ij} (y_{ij}) denotes the number of low capacity (high capacity) facilities to be installed on edge $\{i, j\}$ while f_{ijk} and f_{jik} denote the flow of commodity k on the two conceptual edges (i, j) and (j, i) corresponding to edge $\{i, j\}$. Equation (2.1) is a flow conservation constraint while (2.2) and (2.3) are capacity constraints. In the following, whenever $e = \{i, j\} \in E$ we shall refer to the same variable by x_{ij}, x_{ji} and x_e interchangeably. A similar notation will be used for existing capacity and high capacity facilities.

The cost of installing a low capacity facility on the edge $\{i, j\}$ is denoted by a_{ij} and the corresponding cost of a high capacity facility is denoted by b_{ij}. Finally, we will use a flow cost $c_{ijk} = c_{jik}$ for one unit of commodity k on the edge $\{i,j\}$, knowing that this cost may well be zero in many real life applications. The deterministic capacitated network design problem may now be stated as:

$$z_{DP} = \min\Big\{ax + by + cf \;\Big|\; (x,y,f) \in \mathcal{P}\Big\} \tag{2.4}$$

where a, b and c are the cost vectors and transposes have been omitted for simplicity.

2.1 Metric Inequalities

The class of metric inequalities, originally introduced by Iri [8] and Onaga and Kakusho [14], can be used to project $\mathcal{P}$ onto the space of the discrete capacity variables. Günlük [5] briefly describes these inequalities for a general multicommodity flow problem but argues that it is only practical to use special subclasses of the inequalities in cutting plane procedures for problem (2.4). Bienstock et al. [1] use metric inequalities in a cutting plane procedure for a reformulation of the one facility capacitated network design problem using path variables rather than edge variables as in our formulation. For the stochastic programming problem that we are going to consider, the metric inequalities may be used as feasibility cuts in an L-shaped algorithm and hence they will be of some importance.

For fixed values of the capacity variables x and y problem (2.4) is a multicommodity flow problem. Associating dual variables ρ, σ and τ with constraints (2.1), (2.2) and (2.3) respectively, the dual of this problem is:

$$\begin{aligned}
\max \sum_{i\in V}\sum_{k\in K} d_{ik}\rho_{ik} - &\sum_{\{i,j\}\in E} (\sigma_{ij}+\tau_{ij})(C_{ij} + x_{ij} + \lambda y_{ij}) \\
\text{s.t. } \rho_{jk} - \rho_{ik} - \sigma_{ij} \le c_{ijk} \qquad & \{i,j\} \in E,\ k \in K \\
\rho_{ik} - \rho_{jk} - \tau_{ij} \le c_{ijk} \qquad & \{i,j\} \in E,\ k \in K \\
\rho_{kk} = 0 \qquad & k \in K \\
\rho \in \mathbb{R}^{|V||K|}, \quad \sigma, \tau \in \mathbb{R}_+^{|E|} &
\end{aligned} \tag{2.5}$$

The multicommodity flow problem is feasible if and only if the dual problem (2.5) is bounded. That is, if and only if:

$$\sum_{\{i,j\}\in E} (v_{ij} + w_{ij})(C_{ij} + x_{ij} + \lambda y_{ij}) \ge \sum_{i\in V}\sum_{k\in K} d_{ik}u_{ik} \qquad \forall (u,v,w) \in \mathcal{D}^+ \tag{2.6}$$

where $\mathcal{D}^+$ denotes the recession cone of the feasible region of the dual problem:

$$\begin{aligned}
\mathcal{D}^+ = \Big\{ &(u,v,w) \in \mathbb{R}^{|V||K|} \times \mathbb{R}_+^{|E|} \times \mathbb{R}_+^{|E|} \;\Big| \\
&u_{jk} - u_{ik} \le v_{ij},\ u_{ik} - u_{jk} \le w_{ij}\ , u_{kk} = 0,\ k \in K,\ \{i,j\} \in E \Big\}
\end{aligned} \tag{2.7}$$

Clearly, we may restrict attention to the extreme rays of the dual feasible region.

To understand why inequalities defined by (2.6) are called metric inequalities, consider a directed graph $\bar{G} = (V, A)$ constructed in the following way. For each undirected edge $\{i,j\} \in E$ let A contain the two directed edges (i,j) and (j,i). If we associate weights v_{ij} with edge $(i,j) \in A$ and w_{ij} with edge $(j,i) \in A$, we see that the right hand side of (2.6) is maximized if and only if u_{ik} is the length of a shortest (k,i)-path in $\bar{G}$. From now on we shall only be interested in vectors (u, v, w) satisfying this property.

Since the feasible region of the dual problem is a rational polyhedron, we may assume that such $(u, v, w) \in D^+$ are integral - this can be achieved by scaling. Hence (2.6) may be strengthened by rounding:

$$\sum_{\{i,j\}\in E} (v_{ij} + w_{ij})(x_{ij} + \lambda y_{ij}) \geq \left\lceil \sum_{i\in V} \sum_{k\in K} d_{ik} u_{ik} - \sum_{\{i,j\}\in E} C_{ij}(v_{ij} + w_{ij}) \right\rceil$$

The strengthened inequalities are referred to as integral metric inequalities. They are not necessarily facet defining but, as mentioned, they will be useful as feasibility cuts in the L-shaped algorithm.

2.2 Partition Inequalities

In this section we consider a special class of integral metric inequalities called partition inequalities. The reason why we consider these inequalities separately, is that some partition inequalities are facet defining under mild conditions. We will only consider partition inequalities obtained by assigning unit weights to some edges and zero weight to the remaining edges.

Let $\pi = (V_1, \ldots, V_l)$ be a partition of the node set into l subsets and let E_π denote the corresponding multicut, $E_\pi = \big\{\{i,j\} \in E \,\big|\, \exists\, r \in \{1, \ldots, l\} \;:\; |\{i,j\} \cap V_r| = 1\big\}$. Next, for any permutation $\alpha = (\alpha_1, \ldots, \alpha_l)$ of the sequence $(1, \ldots, l)$, we let $T(\pi, \alpha)$ denote the net traffic which must be routed across the multicut E_π from lower numbered subsets to higher numbered subsets when subsets are numbered with respect to α:

$$T(\pi, \alpha) = \sum_{r=1}^{l-1} \sum_{k\in V_{\alpha_r}} \sum_{i\in V^{\alpha_r}} d_{ik} - \sum_{e\in E_\pi} C_e$$

where $V^{\alpha_r} = \bigcup_{t=r+1}^{l} V_{\alpha_t}$. $T(\pi, \alpha)$ provides a lower bound on the capacity which needs to be installed across the multicut E_π. Taking the maximum over the $l!$ possible permutations of $(1, \ldots, l)$, we obtain a stronger lower bound $T(\pi)$ and the valid inequality:

$$x(E_\pi) + \lambda y(E_\pi) \geq \lceil T(\pi) \rceil \tag{2.8}$$

where we have defined the aggregate variables $x(E_\pi) = \sum_{e \in E_\pi} x_e$ and $y(E_\pi) = \sum_{e \in E_\pi} y_e$ for notational convenience.

Note that the valid inequality with left hand side as in (2.8) and right hand side equal to $\lceil T(\pi, \alpha) \rceil$ is an integral metric inequality obtained in the following way. For each undirected edge $\{i,j\} \in E_\pi$ let A_π contain the directed edge (i,j) or (j,i) going from a lower numbered subset to a higher numbered subset when subsets are numbered with respect to α. The desired inequality is now obtained by assigning unit weights to edges in A_π and zero weight to all other edges. Integral metric inequalities obtained in this fashion will be referred to as l-partition inequalities. 2-partition inequalities are usually referred to as cutset inequalities. A cutset inequality defines a facet of $\mathcal{P}$ provided that V_1 as well as $V_2 = V \setminus V_1$ are not empty and induce connected subgraphs and that $T(V_1, V_2)$ is not integer. A proof of this result may be found in e.g. Bienstock and Günlük [2] who also give several sufficient conditions for 3-partition inequalities to be facet defining. Finally, Bienstock et al. [1] consider a one facility capacitated network design problem and give sufficient conditions for a general l-partition inequality to be facet defining for the projection of the feasible region on the space of discrete capacity variables.

2.3 Mixed Integer Rounding Inequalities

By applying the mixed integer rounding procedure (see Nemhauser and Wolsey [13]) to the partition inequalities, we obtain a new class of valid inequalities referred to as mixed integer rounding inequalities. For notational convenience we let $\bar{T}_\pi$ denote the right hand side of the partition inequality (2.8) for a given partition π, that is $\bar{T}_\pi = \lceil T(\pi) \rceil$. Next we define $r_\pi = (\bar{T}_\pi \bmod \lambda)$ so that $\bar{T}_\pi = \lambda \lfloor \bar{T}_\pi / \lambda \rfloor + r_\pi$. Note that $0 \le r_\pi < \lambda$ and $r_\pi = 0$ if and only if $\bar{T}_\pi$ is a scalar multiple of λ.

In terms of the aggregate variables $\big(x(E_\pi), y(E_\pi)\big)$ the partition inequality (2.8) is tight at integer points $\big(\bar{T}_\pi, 0\big), \big(\bar{T}_\pi - \lambda, 1\big), \ldots, \big(r_\pi, \lfloor \bar{T}_\pi / \lambda \rfloor\big)$. Still, when $\bar{T}_\pi$ is not a scalar multiple of λ, new fractional extreme points with $x(E_\pi) = 0$ and $y(E_\pi) = \bar{T}_\pi / \lambda$ are induced. Such points may be cut off by including the following mixed integer rounding inequality in the constraint set:

$$x(E_\pi) + r_\pi y(E_\pi) \ge r_\pi \lceil \bar{T}_\pi / \lambda \rceil \tag{2.9}$$

In terms of the aggregate variables $\big(x(E_\pi), y(E_\pi)\big)$ this inequality is tight at integer points $\big(0, \lceil \bar{T}_\pi / \lambda \rceil\big)$ and $\big(r_\pi, \lfloor \bar{T}_\pi / \lambda \rfloor\big)$ assuming that r_π is not equal to zero in which case (2.9) is redundant. Hence, assuming that $r_\pi \neq 0$, the mixed integer rounding inequality is stronger than the corresponding partition inequality for points with $\lfloor \bar{T}_\pi / \lambda \rfloor \le y(E_\pi) \le \lceil \bar{T}_\pi / \lambda \rceil$.

It is possible to prove that if a partition inequality (2.8) is facet defining for $\mathcal{P}$, then the corresponding mixed integer rounding inequality (2.9) is also facet defining for $\mathcal{P}$ under mild conditions. (See e.g. Bienstock and Günlük [2].)

2.4 Mixed Partition Inequalities

Finally, we will consider the class of mixed partition inequalities introduced by Günlük [5]. Let π and π' be two distinct partitions of the node set V and consider the related partition inequalities (2.8). Once again we let $\bar{T}_\pi$ and $\bar{T}_{\pi'}$ denote the right hand sides and r_π and $r_{\pi'}$ the corresponding remainders by division with λ. We assume that $r_\pi > r_{\pi'}$. Applying a general mixing procedure for mixed-integer sets, introduced by Günlük and Pochet [6], to the two partition inequalities, we obtain a mixed partition inequality:

$$x(E_\pi \cup E_{\pi'}) + (r_\pi - r_{\pi'})y(E_\pi) + r_{\pi'}y(E_{\pi'}) \geq (r_\pi - r_{\pi'})\lceil \bar{T}_\pi/\lambda \rceil + r_{\pi'}\lceil \bar{T}_{\pi'}/\lambda \rceil$$

which is valid for $\mathcal{P}$ cf. Theorem 2.1 [6]. Günlük [5] presented several conditions for mixed partition inequalities to be facet defining for $\mathcal{P}$ but his computational experiments indicated that the inequalities only in rare cases had an effect when partition and mixed integer rounding inequalities were already included in the formulation. It turns out, however, that the mixed partition inequalities is the only class of inequalities presented so far which are able to combine information from different scenarios in the stochastic problem, and hence they turned out to be quite useful.

3. The Stochastic Programming Problem

So far, we have only considered the deterministic capacitated network design problem. As already mentioned, though, the assumption that demand is known, will generally not be justified since uncertainty is almost always an inherent feature of systems involving the assessment of future demand. In this section we explicitly take this uncertainty into account by formulating the capacitated network design problem as a two-stage stochastic program with integer first stage and continuous second stage.

The fact that demand is not known with certainty, at the time the decision on capacity expansion has to be made, is incorporated in the problem formulation by allowing demand to depend on the outcome of a random variable ξ defined on some probability space $(\Xi, \mathcal{F}, P)$. The number of low and high capacity facilities to install on each edge of the network must be decided upon well in advance of the point in time at which they are actually installed and operating. Also, it is quite likely that this capacity expansion is required to be sufficient for some period of time causing even more uncertainty about the size of demand that needs to be satisfied. The routing of traffic, on the other hand, is naturally postponed until the actual realization of demand is observed. Hence decisions can be split in two: first stage decisions x and y based solely on the information about demand conveyed through its distribution and second stage decisions $f(\xi)$ taken after demand $d(\xi)$ has been observed.

The following assumption about the distribution of ξ allows us to think of uncertainty in terms of scenarios:

Assumption 1. *The random variable ξ has a discrete distribution with finite support $\Xi = \{\xi^1, \ldots, \xi^S\}$ and corresponding probabilities $P(\xi^1) = p^1, \ldots, P(\xi^S) = p^S$.*

Associated with a scenario s is a realization of random demand $d(\xi^s)$ and a corresponding routing of traffic $f(\xi^s)$. The solution procedure, which we are going to propose, allows us to let the cost of flow c depend on the outcome of ξ too. One should be aware, though, that this drastically increases the number of scenarios needed to give an adequate description of the random vectors. For notational convenience we will refer to demand under scenario s simply as d^s and similar notation will be used for flows f^s and cost of flow c^s.

The feasible region of the stochastic capacitated network design problem, when only flow under scenario s is restricted, is:

$$\mathcal{R}^s = \Bigg\{ (x, y, f^1, \ldots, f^S) \in \mathbb{Z}_+^{|E|} \times \mathbb{Z}_+^{|E|} \times \mathbb{R}_+^{2|K||E|} \times \ldots \times \mathbb{R}_+^{2|K||E|} \;\Bigg|$$

$$\sum_{j:\{i,j\}\in E} f_{jik}^s - \sum_{j:\{i,j\}\in E} f_{ijk}^s = d_{ik}^s \qquad i \in V,\ k \in K,\ i \neq k \tag{3.1}$$

$$\sum_{k\in K} f_{ijk}^s \leq C_{ij} + x_{ij} + \lambda y_{ij} \qquad \{i,j\} \in E \tag{3.2}$$

$$\sum_{k\in K} f_{jik}^s \leq C_{ij} + x_{ij} + \lambda y_{ij} \qquad \{i,j\} \in E \quad \Bigg\} \tag{3.3}$$

and the feasible region of the stochastic programming problem is:

$$\mathcal{R} = \bigcap_{s=1}^{S} \mathcal{R}^s$$

Finally, we may state the capacitated network design problem as a two-stage stochastic programming problem in which total installment costs and expected flow costs are minimized subject to the usual flow conservation and capacity constraints:

$$z_{SP} = \min \Bigg\{ ax + by + \sum_{s=1}^{S} p^s c^s f^s \;\Bigg|\; (x, y, f^1, \ldots, f^S) \in \operatorname{conv} \mathcal{R} \Bigg\} \tag{3.4}$$

Since we are going to solve the problem by an L-shaped algorithm it will be convenient to reformulate it in terms of the capacity variables x and y only. To this end we define the projections of the sets $\mathcal{R}^s$ and $\mathcal{R}$ on the space of discrete capacity variables x and y:

$$\mathcal{R}^s_{x,y} = \Big\{(x,y) \in \mathbb{Z}_+^{|E|} \times \mathbb{Z}_+^{|E|} \;\Big|\; \exists f^1, \ldots, f^S \in \mathbb{R}_+^{2|K||E|} \;:\; (x,y,f^1,\ldots,f^S) \in \mathcal{R}^s\Big\}$$
$$\mathcal{R}_{x,y} = \bigcap_{s=1}^{S} \mathcal{R}^s_{x,y}$$

Problem (3.4) may now be stated as:

$$z_{SP} = \min\Big\{ax + by + \sum_{s=1}^{S} p^s Q^s(x,y) \;\Big|\; (x,y) \in \operatorname{conv} \mathcal{R}_{x,y}\Big\} \tag{3.5}$$

where the second stage value function $Q^s(x,y)$ is given by:

$$Q^s(x,y) = \min\{c^s f^s \mid (x,y,f^1,\ldots,f^S) \in \mathcal{R}^s\} \tag{3.6}$$

Before we proceed, it may be appropriate to pass a few remarks on the feasible region of the stochastic programming problem. In practice it is evident that demand for a commodity k at a node i cannot be thought of as a stationary variable d_{ik}. Usually we describe the arrival of demands/transmissions by a Poisson process, the length of a transmission by an exponentially distributed random variable and even the required bandwidth of a transmission may be a random variable. It is not (economically) feasible to construct a network in which no blocking will occur even in extreme peak situations. What the network provider usually does, is to select a grade of service (GoS) corresponding to a certain blocking probability that must not be exceeded. Given the blocking probalities and the distributions that describe demand, it is possible to determine the so-called *effective bandwidth* requirement, which can be thought of as the capacity needed to secure that the blocking probabilities are not exceeded. (See Dempster, Medova and Thompson [4].) The effective bandwidth requirement obtained in this fashion serves as demand input for our problem and a feasible solution is required to observe these requirements so that the blocking probabilities are not exceeded. Uncertainty in our formulation of the problem arises due to the fact that distributions describing future demand, and hence also the effective bandwidth requirements, are unknown.

3.1 Valid Inequalities

We will now generalize the valid inequalities derived for the deterministic problem to the new setting. Comparing the structure of the regions $\mathcal{P}$ and $\operatorname{conv} \mathcal{R}^s$ it is obvious that we may obtain integral metric inequalities, partition inequalities and mixed integer rounding inequalities for conv $\mathcal{R}^s$ in the exact same way as for the deterministic problem.

For any scenario s, consider the multicommodity flow problem obtained by fixing the values of the capacity variables x and y and minimizing the flow costs subject to (3.1), (3.2) and (3.3). Note that the recession cone $\mathcal{D}^+$ defined by (2.7) is the same for all of these problems. Hence for any extreme ray $(u, v, w) \in \mathcal{D}^+$ satisfying the property, that u_{ik} is the length of a shortest $(k\text{-}i)-$path using edge weights v_{ij} and w_{ij} as previously described, we obtain an integral metric inequality for conv $\mathcal{R}^s$:

$$\sum_{\{i,j\} \in E} (v_{ij} + w_{ij})(x_{ij} + \lambda y_{ij}) \geq \left\lceil \sum_{i \in V} \sum_{k \in K} d^s_{ik} u_{ik} - \sum_{\{i,j\} \in E} C_{ij}(v_{ij} + w_{ij}) \right\rceil \tag{3.7}$$

Similarly, for any partition $\pi = (V_1, \ldots, V_l)$ of the node set we may calculate the maximum net traffic which needs to be routed across the multicut E_π under scenario s:

$$T^s(\pi) = \max_{\alpha} \left\{ \sum_{r=1}^{l-1} \sum_{k \in V_{\alpha_r}} \sum_{i \in V^{\alpha_r}} d^s_{ik} \right\} - \sum_{e \in E_\pi} C_e$$

Letting $\bar{T}^s_\pi = \lceil T^s(\pi) \rceil$, we obtain an l-partition inequality for conv $\mathcal{R}^s$:

$$x(E_\pi) + \lambda y(E_\pi) \geq \bar{T}^s_\pi \tag{3.8}$$

Next, letting $r^s_\pi = (\bar{T}^s_\pi \bmod \lambda)$, we obtain a mixed integer rounding inequality for conv $\mathcal{R}^s$:

$$x(E_\pi) + r^s_\pi y(E_\pi) \geq r^s_\pi \lceil \bar{T}^s_\pi / \lambda \rceil \tag{3.9}$$

As pointed out these three classes of inequalities are all valid for conv $\mathcal{R}^s$ and hence for conv $\mathcal{R}$. Thus in principle we may generate cuts of each type from all of the S scenarios. It is easily seen, though, that similarities of cuts generated from different scenarios allow us to reduce the number of inequalities included in the formulation. We will denote by $\bar{T}_\pi$ the maximum of the right hand sides in (3.8) over all scenarios and by r_π the corresponding remainder by division with λ:

$$\bar{T}_\pi = \bar{T}^{s^*}_\pi \quad \text{where} \quad s^* \in \arg\max_{1 \leq s \leq S} \left\{ \bar{T}^s_\pi \right\}$$
$$r_\pi = \bar{T}_\pi \bmod \lambda$$

It is natural to consider the partition inequalities and mixed integer rounding inequalities generated by $\bar{T}_\pi$ and r_π.

Theorem 1. *For any scenario $s \in \{1,\ldots,S\}$ the partition inequality (3.8) is dominated by the stronger partition inequality:*

$$x(E_\pi) + \lambda y(E_\pi) \geq \bar{T}_\pi \tag{3.10}$$

Proof. The result is obvious. □

A similar result may be stated for the integral metric inequalities (3.7). When we turn to mixed integer rounding inequalities, on the other hand, a bit more care must be taken, since these cuts are not parallel for different scenarios.

Theorem 2. *For any scenario $s \in \{1,\ldots,S\}$ the mixed integer rounding inequality (3.9) is dominated by either non-negativity constraints, the partition inequality (3.10) or the mixed integer rounding inequality:*

$$x(E_\pi) + r_\pi y(E_\pi) \geq r_\pi \lceil \bar{T}_\pi/\lambda \rceil \tag{3.11}$$

Proof. Let $s \in \{1,\ldots,S\}$ and write the corresponding mixed integer rounding inequality as:

$$x(E_\pi) \geq r_\pi^s \big(\lceil \bar{T}_\pi^s/\lambda \rceil - y(E_\pi) \big) \tag{3.12}$$

Similarly, write (3.10) and (3.11) as:

$$x(E_\pi) \geq \bar{T}_\pi - \lambda y(E_\pi) \tag{3.13}$$

$$x(E_\pi) \geq r_\pi \big(\lceil \bar{T}_\pi/\lambda \rceil - y(E_\pi) \big) \tag{3.14}$$

First of all, note that unless $0 \leq y(E_\pi) \leq \lceil \bar{T}_\pi^s/\lambda \rceil$ the inequality (3.12) is dominated by nonnegativity constraints. Next, note that if $r_\pi^s \leq r_\pi$, the inequality (3.12) is dominated by (3.14) for $0 \leq y(E_\pi) \leq \lceil \bar{T}_\pi^s/\lambda \rceil$. So assume that $r_\pi^s > r_\pi$. Since $\lceil \bar{T}_\pi^s/\lambda \rceil = \lceil \bar{T}_\pi/\lambda \rceil$ implies $r_\pi^s \leq r_\pi$, we see that $\lceil \bar{T}_\pi^s/\lambda \rceil \leq \lfloor \bar{T}_\pi/\lambda \rfloor$ and (3.12) is dominated by:

$$x(E_\pi) \geq r_\pi^s \big(\lfloor \bar{T}_\pi/\lambda \rfloor - y(E_\pi) \big) \tag{3.15}$$

To see that this inequality is dominated by (3.13), we use the fact that $\bar{T}_\pi = r_\pi + \lambda(\lfloor \bar{T}_\pi/\lambda \rfloor)$ to write (3.13) as:

$$x(E_\pi) \geq r_\pi + \lambda \big(\lfloor \bar{T}_\pi/\lambda \rfloor - y(E_\pi) \big) \tag{3.16}$$

Since $r_\pi^s < \lambda$ and $r_\pi \geq 0$, we see that for $0 \leq y(E_\pi) \leq \lceil \bar{T}_\pi^s/\lambda \rceil$ the inequality (3.13) dominates (3.15) and with that also (3.12). □

Finally, we turn to the class of mixed partition inequalities. Unlike the previously considered inequalities we may derive inequalities of this type combining information from different scenarios. Thus consider two maximal partition inequalities (3.10) corresponding to two distinct partitions π and π'. Denote by $\bar{T}_\pi$, $\bar{T}_{\pi'}$, r_π and $r_{\pi'}$ the right hand sides and corresponding remainders by division with λ. Once again we assume that $r_\pi > r_{\pi'}$. Applying the mixing procedure of Günlük and Pochet [6] to these inequalities we obtain a mixed partition inequality:

$$x(E_\pi \cup E_{\pi'}) + (r_\pi - r_{\pi'})y(E_\pi) + r_{\pi'}y(E_{\pi'}) \geq (r_\pi - r_{\pi'})\lceil \bar{T}_\pi/\lambda \rceil + r_{\pi'}\lceil \bar{T}_{\pi'}/\lambda \rceil \tag{3.17}$$

which is valid for conv $\mathcal{R}$ cf. Theorem 2.1 [6].

The important thing to note at this point, is that the maximum right hand sides $\bar{T}_\pi$ and $\bar{T}_{\pi'}$ for the two partitions may very well be attained for different scenarios. Hence the mixed partition inequality (3.17) may combine demand information from distinct scenarios and for this reason this class of inequalities may have greater significance when solving the stochastic program than what was experienced by Günlük [5] for the deterministic problem. To test this conjecture we performed a series of preliminary computational testing. We used branch and cut to solve the problem AT13t (see section 5) with 10 scenarios. 10 independent runs were performed with and without the mixed partition inequalities. These test runs revealed a significant reduction in the CPU time as well as the number of nodes in the branching tree when the mixed partition inequaltities were included.

3.2 Facet Defining Inequalities

As previously mentioned, sufficient conditions for partition inequalities and mixed integer rounding inequalities to define facets of conv $\mathcal{R}^s$ (or conv $\mathcal{R}^s_{x,y}$) for some $s \in \{1, \ldots, S\}$ have been given by various authors. One should note, though, that even if such a facet also defines a facet of $\bigcap_{s=1}^S$ conv $\mathcal{R}^s$ (or $\bigcap_{s=1}^S$ conv $\mathcal{R}^s_{x,y}$), it does not necessarily define a facet of conv $\mathcal{R}$ (or conv $\mathcal{R}_{x,y}$) since, in general, we have:

$$\text{conv}\, \mathcal{R} \subseteq \bigcap_{s=1}^{S} \text{conv}\, \mathcal{R}^s$$

and consequently:

$$\text{conv}\, \mathcal{R}_{x,y} \subseteq \bigcap_{s=1}^{S} \text{conv}\, \mathcal{R}^s_{x,y}$$

and these inclusions may be strict.

In this section we provide sufficient conditions for (3.10) and (3.11), respectively, to define facets of conv $\mathcal{R}_{x,y}$. First, we note the following result:

Theorem 3. *conv* $\mathcal{R}_{x,y}$ *and conv* $\mathcal{R}^s_{x,y}$, $s = 1, \ldots, S$ *are full-dimensional polyhedrons.*

Proof. We only show that conv $\mathcal{R}_{x,y}$ is full-dimensional since the proof is exactly similar for the remaining sets. First of all note that conv $\mathcal{R}_{x,y}$ is non-empty. Let $(\bar{x}, \bar{y}) \in$ conv $\mathcal{R}_{x,y}$. Next, add to $(\bar{x}, \bar{y})$ each of the $2|E|$ unit vectors. The $2|E| + 1$ points obtained this way all belong to conv $\mathcal{R}_{x,y}$ and they are affinely independent. $\square$

Recall that given a partition inequality we let s^* denote the scenario for which the right hand side is maximized. We are now able to prove the following result:

Theorem 4. *Consider a partition* $\pi = \{V_1, \ldots, V_l\}$ *of the nodeset* V. *If the partition inequality (3.10) defines a facet of conv* $\mathcal{R}^{s^*}_{x,y}$ *then it also defines a facet of conv* $\mathcal{R}_{x,y}$.

Proof. We consider the two faces $F = \big\{(x,y) \in \mathcal{R}_{x,y} \;\big|\; x(E_\pi) + \lambda y(E_\pi) = \bar{T}_\pi\big\}$ and $F^{s^*} = \big\{(x,y) \in \mathcal{R}^{s^*}_{x,y} \;\big|\; x(E_\pi) + \lambda y(E_\pi) = \bar{T}_\pi\big\}$. Since F^{s^*} is a facet of conv $\mathcal{R}^{s^*}_{x,y}$, we know by Theorem 3 that we can find $2|E|$ affinely independent points $(x^1, y^1), \ldots, (x^{2|E|}, y^{2|E|}) \in F^{s^*}$. Now consider the points given by:

$$(\hat{x}^i_e, \hat{y}^i_e) = \begin{cases} (x^i_e, y^i_e) & \text{if } e \in E_\pi \\ (x^i_e + M, y^i_e) & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots, 2|E|$$

where M is some large number. By the definition of $\bar{T}_\pi$ and s^* we see that for any scenario the solution $(\hat{x}^i_e, \hat{y}^i_e)$, $e \in E$ allows a feasible routing of all demand across the cut E_π as well as all internal demand in each nodeset $V_1, \ldots, V_l$. Hence $(\hat{x}^1, \hat{y}^1), \ldots, (\hat{x}^{2|E|}, \hat{y}^{2|E|}) \in F$. Furthermore, these points are obtained by adding the same vector to each of the points $(x^1, y^1), \ldots, (x^{2|E|}, y^{2|E|})$ and hence they are affinely independent. $\square$

In the exact same way we may prove the following result:

Theorem 5. *Consider a partition* $\pi = \{V_1, \ldots, V_l\}$ *of the nodeset* V. *If the mixed integer rounding inequality (3.11) defines a facet of conv*$\mathcal{R}^{s^*}_{x,y}$ *then it also defines a facet of conv*$\mathcal{R}_{x,y}$.

Theorems 4 and 5 will prove extremely useful to us, since they allow us to use conditions derived for the deterministic capacitated network design problem to identify facet defining inequalities for the stochastic program.

4. An L-shaped Algorithm

Problem (3.4) is a large-scale mixed integer programming problem and may be solved as such by standard software packages. However, as always when working with stochastic programming problems one should exploit the special structure of the problem and hence we will use the formulation (3.5)-(3.6). We present a modified version of the L-shaped algorithm for continuous stochastic programming problems combining ordinary Bender's decomposition with a branch and cut scheme.

Since $\operatorname{conv} \mathcal{R}_{x,y}$ is a convex polyhedron, the condition $(x, y) \in \operatorname{conv} \mathcal{R}_{x,y}$ may be replaced by a finite number of *feasibility cuts* corresponding to the facets of $\operatorname{conv} \mathcal{R}_{x,y}$. In general, however, we cannot identify all of these cuts due to the integer requirements on the first stage variables. Still, we have shown that the integral metric inequalities (3.7), and in particular the partition inequalities (3.10), provide necessary conditions for feasibility of the second stage problems and hence we shall use these inequalities as feasibility cuts. The mixed integer rounding inequalities (3.11) and mixed partition inequalities (3.17) are not strictly necessary for second stage feasibility, but we will use them as a sort of feasibility cuts since they do define facets of $\operatorname{conv} \mathcal{R}_{x,y}$ under certain conditions.

Likewise, the convex and piecewise linear second stage value functions (3.6) may be represented by a number of linear models, referred to as *optimality cuts*. To be specific, we have by linear programming duality that:

$$Q^s(x,y) = \max_{l \in \{1,\ldots,L^s\}} \left\{ \sum_{i \in V} \sum_{k \in K} d^s_{ik} \rho^l_{ik} - \sum_{\{i,j\} \in E} (\sigma^l_{ij} + \tau^l_{ij})(C_{ij} + x_{ij} + \lambda y_{ij}) \right\}$$

where $(\rho^l, \sigma^l, \tau^l)$, $l \in \{1, \ldots, L^s\}$ are the dual extreme points of the s'th second stage problem. Hence we may replace each of the second stage value functions by a single variable θ^s and the constraints:

$$\theta^s \geq \sum_{i \in V} \sum_{k \in K} d^s_{ik} \rho^l_{ik} - \sum_{\{i,j\} \in E} (\sigma^l_{ij} + \tau^l_{ij})(C_{ij} + x_{ij} + \lambda y_{ij}) \qquad l = 1, \ldots, L^s$$

The algorithm progresses by sequentially solving a master problem and adding feasibility cuts or optimality cuts which are violated at the current solution. Violated optimality cuts as well as violated metric inequalities are identified by solving the second stage problems, thereby generating the needed dual extreme points and extreme rays. The separation problem for the integral metric inequalities, the partition inequalities, the mixed integer rounding inequalities and the mixed partition inequalities, on the other hand, is in general $\mathcal{NP}$-hard and we may have to resort to heuristics to identify violated cuts of these types -

an issue which we shall return to. By appropriately defining matrices $D = (D_1, D_2, d)$ and $E = (E_1, E_2, E_3, e)$ representing the feasibility cuts and optimality cuts which have been included, we may state the master problem as:

$$\begin{aligned} z_P = \min\ & ax + by + \sum_{s=1}^{S} p^s \theta^s \\ \text{s.t. } & D_1 x + D_2 y \geq d \\ & E_1 x + E_2 y + E_3 \theta \geq e \\ & x, y \in \mathrm{IR}^{|E|} \end{aligned} \tag{4.1}$$

Algorithm 1

Step 1 (*Initialization*) Set $\bar{z} = \infty$, choose an initial set of constraints represented by D and E and let $\mathcal{P}$ consist of problem (4.1).

Step 2 (*Termination/Node Selection*) If $\mathcal{P} = \emptyset$, stop; the solution which generated the current upper bound $\bar{z}$ is optimal. Otherwise, select and remove a problem P from $\mathcal{P}$.

Step 3 (*Master Iteration*) Solve the current master problem P and let $(\bar{x}, \bar{y}, \bar{\theta})$ be the optimal solution vector. Consider the following situations:

1. $z_P \geq \bar{z}$. The current problem is fathomed; go to step 2.
2. $z_P < \bar{z}$ and $(\bar{x}, \bar{y})$ is integral. Solve the second stage problem (3.6) for $s = 1, \ldots, S$ and update the upper bound if $a\bar{x} + b\bar{y} + \sum_{s=1}^{S} p^s Q^s(\bar{x}, \bar{y}) < \bar{z}$. If $\bar{\theta}^s = Q^s(\bar{x}, \bar{y})$ for all scenarios the current problem is fathomed; go to step 2. Otherwise; go to step 4.
3. $z_P < \bar{z}$ and $(\bar{x}, \bar{y})$ contains a fractional element. Decide whether to proceed by cutting (go to step 4) or branching (go to step 5).

Step 4 (*Cut Generation*) Identify a number of cuts which are violated at the current solution and augment D and E by appending the new rows to the appropriate matrix. Go to step 3.

Step 5 (*Branching*) Select an edge $\{i, j\}$ such that $\bar{x}_{ij}$ or $\bar{y}_{ij}$ is fractional and add two new problems to $\mathcal{P}$ by including either the bounds $x_{ij} \leq \lfloor \bar{x}_{ij} \rfloor$ and $x_{ij} \geq \lceil \bar{x}_{ij} \rceil$ or the bounds $y_{ij} \leq \lfloor \bar{y}_{ij} \rfloor$ and $y_{ij} \geq \lceil \bar{y}_{ij} \rceil$ in the formulation. Go to step 2.

5. Computational Experiments

We implemented Algorithm 1 in C++ using procedures from the callable library of CPLEX 6.6 to solve the master- and subproblems. A series of computational experiments was performed to test the practicability of the procedure and in particular the sensitivity of computation time with respect to the number of scenarios.

5.1 Implementational Details

The branch and cut segment of the algorithm was implemented in compliance with the guidelines provided by Günlük [5] to which we refer for a detailed description of this part of the algorithm. As previously discussed, feasibility cuts and optimality cuts were generated through the solution of the second stage problems. Integral metric inequalities, partition inequalities and mixed integer rounding inequalities, on the other hand, were generated heuristically using the procedures described in Günlük [5] and Bienstock et al. [1]. Mixed partition inequalities were only generated at initialization by mixing all tight partition inequalities at the root node. At each node of the branching tree we ran the heuristics until no more cuts could be identified in a fixed amount of time. That is, in step 3.3 of Algorithm 1 we chose to proceed by cutting whenever some second stage problem was infeasible or new violated cuts were identified by the heuristics in the last iteration. Optimality cuts were added whenever a second stage problem was feasible but the addition of new optimality cuts was not allowed to affect the decision whether to keep cutting or proceed by branching. Hence branching occurred whenever the current solution contained a fractional element, all second stage problems where feasible and no more cuts could be identified by the heuristics.

5.2 Problem Instances

The computational experiments were performed on two real-life instances previously studied in Bienstock and Günlük [2] and Günlük [5]. The first instance is a network representing the Atlanta area, containing 15 nodes and 22 edges. Since we are primarily interested in long-term planning where uncertainty is more significant, we chose as starting point the instance exhibiting the largest increase in demand, referred to as AT13t in the previous studies. The second instance is a denser network representing the New York area. This network contains 16 nodes and 49 edges and again we chose the instance with largest demand increase as starting point (NY17t). In the second instance there are no cost of flow and no existing capacity in the network. Both instances have fully dense traffic matrices.

For each network we performed a series of experiments with varying number of scenarios. Scenarios were generated randomly assuming some uncertainty in the overall demand level captured in a parameter μ as well as some regional (node dependent) fluctuations captured in the parameters ρ_i $(i \in V)$. The demand between nodes i and j under scenario s was calculated as:

$$D_{ij}^s = \mu^s \rho_i^s \rho_j^s D_{ij}$$

where D_{ij} is demand between nodes i and j in the deterministic problem and the random parameters μ^s and ρ_i^s $(i \in V)$ are sampled from the uniform distribution:

$$\mu^s \sim U(0.8, 1.2)$$
$$\rho_i^s \sim U(0.9, 1.1) \qquad \forall\, i \in V$$

5.3 Computational Results

For the Atlanta problem we first generated instances with 1, 5, 10, 50, 100 and 500 scenarios. For each number of scenarios we randomly generated ten independent instances and ran the algorithm. At termination we recorded the lower bound (LB) and upper bound (UB), the number of nodes in the branching tree (Nodes), the number of optimality cuts (OC), feasibility cuts (FC) and cuts generated by heuristics (HC), the total number of cuts generated (Total), the number of cuts remaining in the master problem (Active) and the CPU time spent by the procedure (CPU). CPU times are reported as minutes:seconds. The numbers reported in table 1 are all averages over the ten independent runs.

Table 1: Atlanta Problems

S	OC	FC	HC	Total	Active	Nodes	LB	UB	Gap	CPU
1	41	22	1606	1669	61	112	509068.0	509068.0	0%	0:13
5	161	43	1606	1810	69	502	561864.7	561864.7	0%	1:18
10	248	56	1611	1915	99	525	588765.7	588765.7	0%	2:24
50	438	149	1603	2190	172	558	622275.9	622275.9	0%	10:21
100	673	266	1603	2542	303	554	627343.4	627343.4	0%	19:27
500	1955	838	1612	4405	883	589	644081.3	644081.3	0%	111:08

First of all we note that the algorithm terminated with an optimal solution in every run performed on the Atlanta problems in this series of experiments. Also, we see that CPU time exhibits an approximately linear growth with respect to the number of scenarios. The large increase in the number of optimality cuts and the size of the master problem is only natural, since we chose to place disaggregate optimality cuts on the S second stage value functions seperately, and hence at least one active cut exists for each scenario. We also note that the number of cuts generated by the heuristics is fairly constant. This is due to the fact that the cuts generated at initialization were identical (as regards the left hand side) irrespective of the number of scenarios, and these cuts constitute the major part of the heuristically generated cuts. Finally, we observed that the increase in the average number of branching nodes when the number of scenarios is increased seemed to stem from a few "extreme" runs requiring a very large number of nodes, whereas the major part of the runs terminated after a few hundred nodes had been investigated. This tendency became even clearer when we ran the algorithm with 1000 scenarios in which case two of the ten runs did not terminate after more than eight hours of CPU time. Even in this situation, however, the algorithm is able to produce very good lower and upper bounds in a relatively short amount of time. Figure 1 shows the developement of the lower and upper bound for one of the extreme runs with 1000 scenarios.

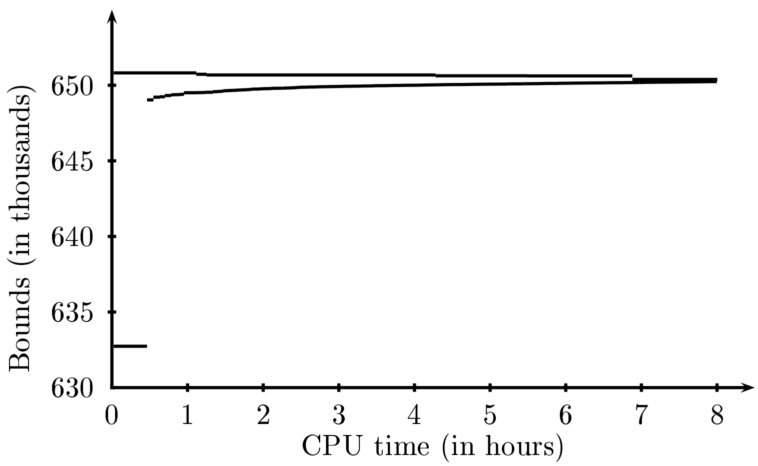

Figure 1: Atlanta problem, 1000 scenarios

We see that the gap between the lower and upper bound is narrowed very quickly. After 30 minutes of CPU time the relative gap is as small as 0.27%, whereas the remaining gap after eight hours of CPU time is 0.02%. The reason for the sudden large increase in the lower bound occurring after half an hour is that the lower bound is only updated when the branching tree is searched after a node has been fathomed. Hence, during the initial cutting phase and the first dive in the branching tree, the bound remains constant at the level reached after initialization. We should note that this initial lower bound was obtained after approximately 90 seconds of CPU time. We conclude that even with 1000 scenarios the algorithm usually terminated within a few hours of CPU time and when this was not the case, very good upper and lower bounds were provided in a reasonable amount of time.

Next, we turn to the New York problems. These problems are much harder than the Atlanta problems and the algorithm presented by Günlük [5] required more than one and a half hours of CPU time to solve the deterministic problem NY17t. Hence, our main interest lay in the quality of the bounds provided in a reasonable amount of time and we chose to stop all runs after three hours of CPU time. Since there is no cost of flow in these problems, we did not place any optimality cuts, but apart from this the statistics appearing in table 2 are the same as those recorded for the Atlanta problems.

Table 2: New York Problems

S	FC	HC	Total	Active	Nodes	LB	UB	Gap	CPU
1	26595	11363	37958	46	13741	3754.4	3789.6	0.92%	180:00
5	4745	11087	15829	46	2645	4116.4	4237.2	2.84%	180:00
10	2490	11437	13927	46	1350	4334.2	4447.0	2.79%	180:00
50	292	11964	12256	55	136	4444.1	4654.2	4.50%	180:00
100	277	11973	12249	61	61	4427.0	4738.3	6.53%	180:00

First of all we note the quality of the bounds provided by the algorithm. Thus, we see that the average gap is modest for all five series of experiments, even though a significant increase in the gap is observed when the number of scenarios increase. Naturally, the increased gap is caused by the drastic decrease in the number of feasibility cuts and the number of nodes investigated which result from the increased computation time required per iteration when the number of scenarios increase. On the other hand we see that the number of heuristically generated cuts is once again fairly constant due to the large number of cuts generated at initialization. The number of tight cuts in the master does not exhibit the same sensitivity with respect to the number of scenarios as for the Atlanta problems, since no optimality cuts were placed. We did observe, however, a slight increase in the number of active cuts which could not quite be accounted for. Since, on the average, only 61 nodes were investigated with 100 scenarios, we chose not to run the algorithm with more scenarios.

6. Conclusions

We have considered the capacitated network design problem. In specific it has been investigated how methods developed for the deterministic problem may be applied to a two-stage stochastic programming formulation arising when demand is assumed to be unknown at the point of decision. The algorithm which was elaborated was tested on two sets of real-life instances with promising results. For the smaller network the algorithm terminated with an optimal solution when it was applied to problems containing between 1 and 500 scenarios. When more scenarios were generated, the algorithm did not always terminate within a few hours of CPU time, but very good lower and upper bounds were relatively quickly available. For the larger network we never obtained an optimal solution, but again the quality of the bounds provided is good, though decreasing somewhat with the number of scenarios. We conclude that the method is certainly practicable and a valuable tool in the design of telecommunication networks under uncertainty.

References

[1] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81:177–200, 1998.

[2] D. Bienstock and O. Günlük. Capacitated network design. Polyhedral structure and computation. *INFORMS Journal on Computing*, 8(3):243–259, 1996.

[3] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming.* Springer-Verlag, Berlin Heidelberg New York, 1997.

[4] M.A.H. Dempster, E.A. Medova, and R.T. Thompson. A stochastic programming approach to network planning. *Teletraffic Contributions for the Information Age. Proceedings of the 15th International Teletraffic Congress - ITC 15*, 1:329–339, 1997.

[5] O. Günlük. A branch-and-cut algorithm for capacitated network design. *Mathematical Programming*, 86:17–39, 1999.

[6] O. Günlük and Y. Pochet. Mixing mixed-integer inequalities. CORE discussion paper. Available at http://www.ieor.columbia.edu/ oktay/mix.ps. 1996.

[7] J.L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16:650–669, 1991.

[8] M. Iri. On an extension of the max-flow min-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13:129–135, 1971.

[9] G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133–142, 1993.

[10] T.L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60:233–250, 1993.

[11] T.L. Magnanti, P. Mirchandani, and R. Vachani. Modelling and solving the two-facility capacitated network loading problem. *Operations Research*, 43:142–157, 1995.

[12] P. Mirchandani. Projections of the capacitated network loading problem. *European Journal of Operational Research*, 122:534–560, 2000.

[13] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley Interscience, New York, 1988.

[14] K. Onaga and O. Kakusho. On feasibility conditions of multicommodity flows in networks. *IEEE Transactions in Circuit Theory CT-18*, 4:425–429, 1971.

[15] M. Riis and K.A. Andersen. On using stochastic programming to plan the multiperiod capacity expansion of one connection in telecommunications. Working Paper 2000/2, University of Aarhus, Department of Operations Research, 2000.

[16] R.T. Rockafellar and R.J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[17] A. Ruszczynski. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[18] S. Sen, R.D. Doverspike, and S. Cosares. Network planning with random demand. *Telecommunication Systems*, 3:11–30, 1994.

[19] R.M. Van Slyke and R. J. Wets. L-shaped linear programs with applications to optimal control and stochastic linear programming. *SIAM Journal of Applied Mathematics*, 17:638–663, 1969.

[20] R.D. Wollmer. Two stage linear programming under uncertainty with 0-1 integer first stage variables. *Mathematical Programming*, 19:279–288, 1980.