Journal of Global Optimization **20:** 173–193, 2001.

Capacity and flow assignment of data networks by generalized Benders decomposition

PHILIPPE MAHEY[1], ABDELHAMID BENCHAKROUN[2] and FLORENCE BOYER[3]

[1]*Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes, CNRS, Campus des Cézeaux, 63173 Aubière, France (e-mail: philippe.mahey@isima.fr)*

[2]*Département de Mathématiques et d'Informatique, Université de Sherbrooke, Quebec, Canada*

[3]*France Telecom R&D, Sophia-Antipolis, France*

Abstract. A mixed-integer non-linear model is proposed to optimize jointly the assignment of capacities and flows (the CFA problem) in a communication network. Discrete capacities are considered and the cost function combines the installation cost with a measure of the Quality of Service (QoS) of the resulting network for a given traffic. Generalized Benders decomposition induces convex subproblems which are multicommodity flow problems on different topologies with fixed capacities. These are solved by an efficient proximal decomposition method. Numerical tests on small to medium-size networks show the ability of the decomposition approach to obtain global optimal solutions of the CFA problem.

Key words: Network design, multicommodity flow networks, Benders decomposition

1. Introduction

Network design is a fundamental problem with a large scope of applications which have given rise to many different models and solution approaches (Mac Gregor Smith and Winter, 1991; Ferreira and Galvão, 1994; Minoux, 1989). In data networks and more precisely, in the design of packet-switched networks with high grade of service constraints, the design of the topology at lowest cost, the dimensioning of the links to accept given demands between each pair of nodes and the computation of optimal routes with the smallest packet delay, are so closely related problems that it is largely justified to try to treat them within a common model, the Capacity and Flow Assignment problem, denoted hereafter by the (**CFA**) problem. To be more precise, the (CFA) problem can be formulated as follows: given a basic topology and a requirement matrix, determine the capacity and flow variables which satisfy the capacitated multicommodity flow constraints and minimize the total design cost. The trade-off between investment costs and congestion is the central feature of the proposed model. In other words, we will search an equilibrium between a low cost topology which tends toward a tree-like arc structure, and a low delay multicommodity flow which tends to use multiple routes between each origin–destination pair with a higher QoS (Quality of Service). The resulting problem is very hard to solve when discrete capacities are considered, mainly because it contains as a subproblem a general network design problem which is known to be NP-hard in most situations (Johnson et al., 1978). We will propose in the present paper a new approach based on generalized Benders decomposition, which aims at solving the (CFA) problem to optimality. We wish to point out that the difficulty we intend to overcome with that computationally efficient algorithm is the competition between dimensioning and routing in data networks and not the combinatorial complexity of the network design problem itself.

The (CFA) problem has been first considered by Gerla in his thesis (1973) and the early approaches used Kleinrock's delay function and linear design costs, allowing the application of the Flow Deviation algorithm to solve the corresponding convex multicommodity flow problem (see Fratta et al., 1973). Most proposed algorithms in the literature treat alternatively the Capacity Assignment problem and the Flow Assignment problem like in Gerla and Kleinrock (1977) or in successive papers by Gavish and collaborators (1989, 1990). Gerla et al. (1989) proposed to embed the packet-switched network into a given backbone facility network and they obtained local optimal solutions to the non convex design and routing model. Lagrangian relaxation has been quite often used to split the problem into separate design and routing (Gavish and Neuman, 1989; Balakrishnan and Graves, 1989; Sanso et al., 1991). Gavish has also introduced Augmented Lagrangians to generate tight lower bounds in Gavish (1985).

On the other hand, the literature on the Capacity Assignment problem (without routing costs or delay bounds) is very large and still growing as that problem is by itself NP-hard and very important in practice. The present approach is complementary with polyhedral approaches which lead to branch-and-cut algorithms and we will show in particular that many results on valid inequalities for these related multicommodity flow formulations are of high interest to enhance the formulation of our master program. These cuts, often called 'metric inequalities' (see Onaga and Kakusho, 1971, or Bienstock and Günlük, 1996), define the convex hull of the discrete set of capacities for which there exist feasible multicommodity flows for a given requirement matrix.

Different kinds of capacity modularities are considered in the literature like two-facility network design problems where small capacities can be grouped in batches (see Magnanti et al., 1995), but we do not address these special structures here. The explicit modelling of capacities inside a non convex cost function has been proposed in the literature, mostly using a concave cost which represents economies of scale (see Minoux, 1989). Gabrel and Minoux showed recently (1997) how generalized linear programming can generate good lower bounds for the Capacity Assignment problem with step increasing cost functions. Finally, separable continuous and non convex cost functions are considered in Luna and Mahey (2000) to integrate QoS measures with fixed costs for capacity expansion problems.

After introducing the (CFA) model, where boolean variables are associated with each type of capacity on each link to get a large-scale mixed-integer non-linear multicommodity flow problem, we discuss the application of generalized

Benders decomposition. Resuming the main features of the method, the master program, itself a linear program in 0-1 variables, adjusts capacities by adding cuts derived from the successive solutions of convex routing subproblems. The detailed algorithm is shown in Sections 3 and 4, followed by some numerical results which confirm the potential of the decomposition approach as an exact method to solve (CFA).

2. Modelling the (CFA) problem

We assume given the traffic requirements matrix and a basic network topology, i.e., a set of nodes and arcs where some capacities could be installed. It is clear enough that a positive capacity will be installed if and only if a positive flow must be routed on that arc, but it is possible that a potential link will not be used in the optimal network, so that our (CFA) model includes the topology design problem.

In computer data networks, messages must be routed 'simultaneously' between each pair of nodes. When the link capacities are fixed, the resulting routing problem (the Flow Assignment problem) can be modelled with continuous flow variables. In this routing problem, packets can take any number of different routes between each source and destination; this is referred to as `bifurcation`. This modelling is appropriate to either datagram or virtual circuit packet-switched networks.

Let $G = (X, U)$ be the directed graph supporting the network, where X is a fixed set of nodes ($|X| = n$) and U is the set of candidate links ($U \subset X \times X$). For each possible arc $u \in U$, we consider a set of feasible capacities $\mathcal{C}_u$, finite and with positive values. In practice, the same capacity is assigned to both directions associated with a link, which will imply additional constraints in the model.

2.1. FLOW CONSTRAINTS

We use a multicommodity flow formulation with an implicit arc-path representation of the network. Let $\mathcal{K}$ be the set of commodities (origin–destination pairs).

Let f_u be the total flow flowing on arc u and x_{kp} be the rate of messages associated with commodity k flowing on path p between corresponding origin–destination pair (o_k, d_k). The set of possible paths between these nodes is denoted by $\mathcal{P}_k$ and the flow requirement by b_k. Thus, a multicommodity flow satisfying flow requirements is given by the following set of equations:

$$f_u = \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k} \pi_{kpu} x_{kp} \tag{2.1}$$

where $\pi_{kpu} = 1$ if path p for commodity k uses arc u and 0 elsewhere.

$$\begin{aligned} \sum_{p \in \mathcal{P}_k} x_{kp} &= b_k \\ x_{kp} &\geqslant 0 \end{aligned} \tag{2.2}$$

2.2. MODELLING DISCRETE CAPACITIES

Discrete capacities or step increasing design cost functions as well as fixed costs have been considered in the literature about network design (see Gabrel and Minoux, 1997, for example), but the corresponding models do not include delay costs or total average delay bounds.

To model the choice of optimal capacities among the finite sets $\mathcal{C}_u$, we create multiple arcs for each link $u = (i, j) \in U$ and each capacity $c \in \mathcal{C}_u$. Let $\mathcal{E} = \{(u, c) | u \in U, c \in \mathcal{C}_u\}$ be the set of virtual arcs in the resulting multigraph $\mathcal{G}$. The cost of installing capacity c on arc u is denoted by $\mathcal{K}_{uc}$ for $(u, c) \in \mathcal{E}$ and let finally introduce a boolean variable y_{uc} for each $(u, c) \in \mathcal{E}$ to model the choice of the capacity type on a link.

$$y_{uc} = \begin{cases} 1 & \text{if capacity } c \text{ is assigned to arc } u \\ 0 & \text{else} \end{cases} \tag{2.3}$$

Then, we must add the following constraints :

$$\sum_{c \in \mathcal{C}_u} y_{uc} \leqslant 1 \, , \; \forall u \in U \tag{2.4}$$

$$y_{ij,c} = y_{ji,c} \, , \forall (u, c) \in \mathcal{E}, u = (i, j) \tag{2.5}$$

to retrieve a capacitated network on topology G.

If we denote by f_{uc} the total flow assigned to the directed link u with capacity c, the multicommodity flow constraints can be written on $\mathcal{G}$ as:

$$f_{uc} - c y_{uc} \leqslant 0 \, , \; \forall (u, c) \in \mathcal{E} \tag{2.6}$$

REMARK 1. *(1) Observe that it is possible that $y_{uc} = 0$, $\forall c \in \mathcal{C}_u$, so that the link is not used.*

(2) No doubt that the multigraph model will tend to increase the number of boolean variables. On the other hand, we will see in section 3 that it induces separability between continuous and discrete variables which is strongly needed in the implementation of generalized Benders decomposition.

2.3. THE AVERAGE DELAY FUNCTION

Many different routing costs can be introduced to measure the grade of service or the reliability of the network. We have chosen to model explicitly the total average delay on the network. The average delay on arc u with capacity c_u when a flow f_u circulates on it is given by Kleinrock's law, assuming classical hypotheses on the arrival of packets at each node, i.e., the computer network is modelled as a Jackson network of queues in which each queue behaves as an independent $M/M/1$ queue. Thus, the average delay on a link is proportional to :

$$T_u(f_u, c_u) = \frac{f_u}{c_u - f_u}$$

Expressions that are more accurate than the above can be found in Gerla and Kleinrock (1977) and in Bertsekas and Gallager (1987), but, for most design purpose, the Kleinrock's formula is sufficiently accurate.

REMARK 2. *(1) For a fixed capacity c_u, the delay function T_u is a strictly convex increasing barrier function of the flow, but for variable capacities, it is no more jointly convex in (c_u, f_u). It is moreover discontinuous in $(0, 0)$.*

(2) To define an optimization problem, one should minimize the design cost subject to a total delay constraint. Relaxing that constraint would indeed lead to the same combined cost we are going to define hereafter, but the non convexity of the entire model would also impede to interpret the corresponding Lagrange multiplier as the correct marginal price for one unit of delay, as we will observe in the last section below.

2.4. (CFA) MODEL

The global (CFA) problem can thus be modelled on the formerly defined multigraph $\mathcal{G}$ by :

$$(P)\quad \begin{cases} \text{Minimize} & \sum_{(u,c)\in\mathcal{E}}[\mathcal{K}_{uc}y_{uc} + \gamma\frac{f_{uc}}{c-f_{uc}}] & \\ \text{subject to} & f_{uc} - cy_{uc} \leqslant 0, & \forall (u,c)\in\mathcal{E} \\ & \sum_{c\in\mathcal{C}_u} f_{uc} = \sum_k \sum_{p\in\mathcal{P}_k} \pi_{kpu}x_{kp}, & \forall u \in U \\ & \sum_{p\in\mathcal{P}_k} x_{kp} = b_k & \\ & x_{kp} \geqslant 0 & \\ & \sum_{c\in\mathcal{C}_u} y_{uc} \leqslant 1\ , & \forall u\in U \\ & y_{ij,c} = y_{ji,c}\ , & \forall (u,c)\in\mathcal{E}, u=(i,j) \\ & y_{uc}\in\{0,1\} & \end{cases}$$

where γ is a positive weight representing the scale between cost and delay.

If we denote by $\mathcal{F}$ the set of multicommodity flows $\{f_u\}_{u\in U}$ satisfying (2.1) and (2.2) and by $\mathcal{Y}$ the set of $y_{uc}\in\{0,1\}$ satisfying (2.4) and (2.5), then the model (P) can be written as :

$$(P)\quad \begin{cases} \text{Minimize} & \sum_{(u,c)\in\mathcal{E}}[\mathcal{K}_{uc}y_{uc} + \gamma\frac{f_{uc}}{c-f_{uc}}] \\ \text{subject to} & f\in\mathcal{F},\ y\in\mathcal{Y} \\ \text{and} & f_{uc} - cy_{uc} \leqslant 0,\ \forall (u,c)\in\mathcal{E} \end{cases}$$

REMARK 3. *(1) Like in most approaches dealing with the (CFA) problem, the aim is to separate the Capacity Assignment problem (CA) from the Flow Assignment problem (FA). But, we must recall here that this is not an easy task as, in general, if f^* solves (FA) with $c = c^*$ and c^* solves (CA) with $f = f^*$, this does not mean that (f^*, c^*) is even a local minimum for the (CFA) problem.*

(2) The flow constraints $f \in \mathcal{F}$ should be developed using the multiple arc flow variables f_{uc}. There is indeed no need to transpose these notations, as the flow subproblems will work directly with the f_u variables on a specific topology associated with a fixed y.

In the following, we will denote, for a given $y \in \mathcal{Y}$, by $U(y)$ the set of arcs in U with positive capacities, i.e., such that $y_{uc} = 1$ for one c. Let too $G(y) = (X, U(y))$ be the corresponding partial graph.

3. A solution approach using generalized Benders decomposition

Benders decomposition (see Benders, 1962) has been mainly introduced to treat mixed integer programs with underlying separable convex structure and it was applied to network design in different situations such as concentrator location problems (Geoffrion and Graves, 1974), optimal topology design (Magnanti and Wong, 1981) or network design with underlying tree structure (Benchakroun et al., 1997). Hoang has applied the generalized Benders decomposition to some nonlinear model for network design (Hoang, 1982). On the other hand, the use of Benders decomposition to treat discontinuous cost functions has been considered by Holmberg for facility location problems (Holmberg, 1994).

Generalized Benders decomposition (see Geoffrion, 1972) is motivated by the fact that, fixing the complicating variables, i.e., the boolean variables y, problem (P) reduces to a convex cost multicommodity flow problem on a given topology with fixed capacities. Indeed, fixing $y \in \mathcal{Y}$ is equivalent to install capacities c_u on each arc u of the graph $G = (X, U(y))$.

In what follows, we summarize the approach and describe its application to our model which can be written in compact form as:

$$\text{(P)} \quad \begin{cases} \text{Min} & \mathcal{K}(y) + \mathcal{T}(f) \\ \text{subject to} & g(f, y) \leqslant 0 \\ & f \in \mathcal{F}, \ \ y \in \mathcal{Y} \end{cases}$$

where $g(f, y)$ is the vector function with components $f_{uc} - cy_{uc}$.

To solve (P) with the generalized Benders decomposition, we first complete the projection of (P) onto the space of the complicating variables y. This projection is defined as follows:

$$\text{(PP)} \quad \begin{cases} \text{Min} & \mathcal{K}(y) + v(y) \\ \text{subject to} & y \in \mathcal{Y} \bigcap \mathcal{V} \end{cases}$$

where
$\mathcal{V} = \{y \mid g(f, y) \leqslant 0 \quad \text{for some} \quad f \in \mathcal{F}\}$.

and
$v(y)$ is the optimal value of the subproblem,

$$(\mathrm{SP}-y)\quad \begin{cases} \min & \mathcal{T}(f) = \gamma \sum_{u\in U(y)} T_u(f_u) \\ \text{subject to} & f \in \mathcal{F}(y) \end{cases}$$

where $\mathcal{F}(y)$ is the set of multicommodity flows with respect to the restricted topology $U(y)$.

It is easy to see that if $y \in \mathcal{Y}\bigcap\mathcal{V}$ then $v(y) = \sup_{\mu\geqslant 0}\ [\inf_{f\in\mathcal{F}} \mathcal{T}(f) + \mu g(f, y)]$. Furthermore, $y \in \mathcal{Y}$ is also in the set $\mathcal{V}$ if and only if y satisfies

$$\inf_{f\in\mathcal{F}}\ \nu.g(f, y) > 0,\ \text{ for all } \nu \geqslant 0 \text{ such that } \sum_{(u,c)\in\mathcal{E}} \nu_{uc} = 1$$

These results are used to specify the following master problem (MP) which is equivalent to (PP):

$$(\mathrm{MP})\quad \begin{cases} \min & \mathcal{K}(y) + t \\ \text{subject to} & \mathrm{Inf}_{f\in\mathcal{F}}\mathcal{T}(f) + \mu g(f, y) \leqslant t, \quad \forall\mu \geqslant 0 \\ & \mathrm{Inf}_{f\in\mathcal{F}}\nu.g(f, y) \leqslant 0 \qquad\qquad \forall\nu \geqslant 0, \sum_{(u,c)\in\mathcal{E}} \nu_{uc} = 1 \\ & y \in \mathcal{Y} \end{cases}$$

The difficulty to approximate the implicit function v and the related implicit set $\mathcal{V}$ induces an iterative procedure where relaxed master programs are built by adding constraints, the Benders cuts, which correspond to successive linearizations (sometimes called cutting planes) of these implicit but convex objects.Therefore, the relaxed master problem (RMP) takes the following form :

$$(\mathrm{RMP})\quad \begin{cases} \min & \mathcal{K}(y) + t \\ \text{subject to} & \mathrm{Inf}_{f\in\mathcal{F}}\mathcal{T}(f) + \mu^k g(f, y) \leqslant t, \quad 1 \leqslant k \leqslant p \\ & \mathrm{Inf}_{f\in\mathcal{F}}\nu^r.g(f, y) \leqslant 0 \qquad\qquad 1 \leqslant r \leqslant q \\ & y \in \mathcal{Y} \end{cases}$$

Now, suppose we have computed an optimal solution $(\bar{y}, \bar{t})$ of the relaxed problem (RMP); we can now solve the subproblem $(SP - \bar{y})$.

If $(SP - \bar{y})$ is feasible and $v(\bar{y}) \leqslant \bar{t}$, then it follows from Lagrangean duality that $(\bar{y}, \bar{t})$ is an optimal solution of (MP). Otherwise, if $(SP - \bar{y})$ is feasible and $v(\bar{y}) > \bar{t}$, then using the vector of optimal multipliers $\bar{\mu}$ associated with constraints $g(f, \bar{y}) \leqslant 0$, a cut of type I

$$B_1(y) = \mathrm{Inf}_{f\in\mathcal{F}}\mathcal{T}(f) + \mu g(f, y) \leqslant t$$

is generated and introduced in (RMP) to specify a new relaxation of (MP).

Finally, if $(SP - \bar{y})$ is not feasible, then a cut of type II

$$B_2(y) = \mathrm{Inf}_{f\in\mathcal{F}}\nu.g(f, y) \leqslant 0$$

is identified and added to (RMP).

REMARK 4. *(1) We will assume standard hypotheses to get finite convergence of Benders decomposition, i.e., $\mathcal{F}$ is non empty and the transformed (CFA) problem is feasible.*

(2) To solve Problem (SP-y), we decided to use a specially tailored distributed algorithm based on the Proximal Decomposition method which was shown to be very performant on large congested networks with a huge number of commodities (see Mahey et al., 1995*). It is a primal-dual decomposition technique which induces distributed computations among arcs and paths of the network and is further described in the next section.*

*(3) Another important feature to understand the application of Benders decomposition to (P) is the nature of the coupling between the continuous flow variables and the boolean decision variables. We will not recall here the convergence results of generalized Benders decomposition which can be found in Geoffrion's seminal paper (*Geoffrion, 1972*), but an important practical aspect to generate explicit cuts is the following: there are two kinds of cuts which involve the minimization with respect to the flow variables f of linear combinations of the objective function of (P) and the coupling constraint. As the objective function $\mathcal{K}(y) + \mathcal{T}(f)$ and the coupling constraint $g(f, y) = f - cy$ are linearly separable in f and y, the minimization can be performed independently of y and explicit cuts are then easily obtained. As commented by Geoffrion (*1972*, p. 251), the separability hypothesis is not necessary for convergence, but it is certainly strongly desired to get implementable algorithms.*

4. Application to the model

4.1. SOLVING THE CONVEX COST MULTICOMMODITY FLOW SUBPROBLEMS

We have used the Proximal Decomposition algorithm described in Mahey et al (1995). It is a primal–dual massively distributed method which can be seen to work like a *separable Augmented Lagrangian* method. The algorithm performs two distinct steps at each iteration: a proximal step which regularizes the objective function by adding a quadratic term depending on the previous primal–dual pair of solutions, and a projection step on the coupling subspaces (associated with the multicommodity flow constraints (2.1) and the copies of the dual variables for each arc and each commodity). The Proximal Decomposition algorithm is closely related to the Alternate Direction Method of Multipliers which have been applied to convex multicommodity flow problems by different authors (see Eckstein and Fukushima, 1993, for example).

As the set of paths between o_k and d_k is not known a priori, it is shown in Mahey et al. (1995) how to substitute it at each iteration $t = 0, 1, \ldots$ by a subset which contains the previously generated paths. The proximal step consists of one-dimensional convex subproblems for each arc u to find aggregate flows f_u^{t+1}. Then, new paths are generated by shortest paths calculation with link costs $T_u^{'}(f_u^{t+1})$ followed by a distributed updating of path flows and potentials. The whole algorithm (applied to a topology $U(y)$) is represented below with the following notations:

P_k^t will denote the set of paths between o_k and d_k already generated at iteration t and let $N_k = |P_k^t|$. For each arc u, let $d(u)$ be the number of paths sharing arc u. For each path $p \in P_k^t$, let $|\pi_{kp}|$ denotes the number of arcs of the path. The residual vectors (violation of constraints (2.1) and (2.2)) associated with a multicommodity flow f^t are denoted by :

$$r_u(f^t) = \sum_k \sum_p \pi_{kpu} x_{kp}^t - f_u^t \ \text{ and } \ r_k(f^t) = b_k - \sum_p x_{kp}^t$$

The dual variables are thus associated with equations (2.1) and (2.2) and will be denoted by $z_u, \forall u \in U(y)$ and $Z_k, \forall k$, respectively. The algorithm is stopped when the commodity residuals r_k are less than ε_1 and the optimality conditions (Kuhn-Tucker conditions) are satisfied within a tolerance of ε_2.

algorithm (subproblem)

(1) Choose the convergence parameters $\varepsilon_1, \varepsilon_2, \lambda > 0$,. Set the iteration index $t = 0$. The initial vectors f^0, z^0, Z^0 may be chosen arbitrarily.

(2) For each arc u compute

$$f_u^{t+1} = \arg \min_{0 \leqslant f_u < c_u} \{T_u(f_u) - z_u^t f_u + \frac{\lambda}{2}((f_u)^2 - 2(f_u^t + \frac{r_u(f^t)}{d(u)}) f_u)\}$$

(3) For each commodity k, compute the shortest path that joins the origin o_k and the destination d_k. The length for each arc u used for this computation is $T_u'(f_u^{t+1})$. This shortest path is added to P_k^t and N_k is incremented, if it is not already there. Then, the path flows are updated according to the following rule:

$$x_{kp}^{t+1} = \max \left(0, x_{kp}^t + \tfrac{1}{\lambda(1+|\pi_{kp}|)} \left(Z_k^t - \sum_{u \in kp} z_u^t\right) + \tfrac{1}{1+|\pi_{kp}|} \left(\tfrac{r_k(f^t)}{N_k} - \sum_{j \in kp} \frac{r_u(f^t)}{d(u)}\right)\right)$$

(4) Update the dual variables

$$z_u^{t+1} = z_u^t + \frac{\lambda}{d(u)} r_u(f^{t+1}), \quad Z_k^{t+1} = Z_k^t + \frac{\lambda}{N_k} r_k(f^{t+1})$$

(5) Test $(f_u^{t+1}, x_{kp}^{t+1}, z_u^{t+1}, Z_k^{t+1})$ for convergence and set $t \leftarrow t+1$ if one decides to continue the iteration.

4.2. CUTS GENERATION

The Benders cuts are of two types :

(1) Benders cuts of type I :

$$B_1(y) = \text{Inf}_{f\in\mathcal{F}} \mathcal{T}(f) + \mu g(f, y) \leqslant t$$

for all $\mu \geqslant 0$, the dual vectors defined on $\mathcal{E}$; t is an auxiliary variable such that $t = v(y) = \sup_{\mu\geqslant 0} B_1(y)$ at optimality.

For a given $\bar{y}$, the dual variables μ_{uc} are generated by computing the optimal multipliers associated with the coupling constraints (2.6) in the solution of (SP-$\bar{y}$). Optimality conditions for (SP-$\bar{y}$) are :

$$\begin{cases} \dfrac{\gamma c}{(c - f_{uc})^2} + \beta_j - \beta_i + \mu_{uc} = 0 \;\; \forall (u, c) \in \mathcal{E}, u = (i, j) \\ \mu_{uc} \geqslant 0\ ,\ \mu_{uc}(f_{uc} - c\bar{y}_{uc}) = 0 \quad \forall (u, c) \in \mathcal{E} \end{cases}$$

where β_i is the optimal potential at node i.

The computation is now straightforward as $\mu_{uc} = 0$ if $\bar{y}_{uc} = 1$:

optimal multipliers

For each arc $u = (i, j)$ in the original topology U :

– Either $\exists \bar{c} \in \mathcal{C}_u$ such that $\bar{y}_{u\bar{c}} = 1$ (i.e. $u \in U(\bar{y})$), then:
  $\mu_{u\bar{c}} = 0$
  $\mu_{uc} = \dfrac{\gamma \bar{c}}{(\bar{c} - f_{u\bar{c}})^2} - \dfrac{\gamma}{c}\ ,\ \forall c \in \mathcal{C}_u \quad \text{and} \quad c \neq \bar{c}$
– Or $\sum_{c\in\mathcal{C}_u} \bar{y}_{uc} = 0$ (i.e., $u \notin U(\bar{y})$), then:
  $\mu_{uc} = \Gamma_u - \dfrac{\gamma}{c}$ with $\Gamma_u \geqslant \dfrac{\gamma}{c_{\min}}, \quad \forall c \in \mathcal{C}_u$ with $c_{\min} = \inf_{c\in\mathcal{C}_u} \{c\}$
  In practice, we will always use the lowest value : $\Gamma_u = \frac{\gamma}{c_{\min}}$.

The cuts of type I are thus of the following form :

$$t + \sum_{(u,c)\in\mathcal{E}} \mu_{uc} c y_{uc} \geqslant v(\bar{y}) \tag{4.7}$$

(2) Benders cuts of type II

$$B_2(y) = \text{Inf}_{f\in\mathcal{F}} \nu.g(f, y) \leqslant 0$$

for all $\nu \geqslant 0, \sum_{(u,c)\in\mathcal{E}} \nu_{uc} = 1$. These cuts are essentially feasibility cuts for (SP-y). They are indeed equivalent to the following feasibility condition for the multicommodity flow problem :

THEOREM 1. *A necessary and sufficient condition to exist a feasible multicommodity flow on topology $U(\bar{y})$ is :*

$$\forall \nu \geqslant 0 : \quad \sum_k l_k b_k \leqslant \sum_{(u,c)\in\mathcal{E}} \nu_{uc} c \bar{y}_{uc}$$

where l_k is the length of the shortest path between o_k and d_k using arc lengths ν_{uc}.

Proof. The conditions, sometimes referred as the 'japanese theorem', were first developed by Onaga, Kakusho and Iri (see Onaga and Kakusho, 1971). The resulting cuts, the so called 'metric inequalities', are a direct consequence of the application of Farkas' lemma to the arc-path formulation of the flow constraints (2.1) and (2.2). Indeed, for given capacity decision variables $\bar{y}_{uc}$, there exist non negative path flows x_{kp} such that

$$\begin{aligned} \sum_{p\in\mathcal{P}_k} x_{kp} &= b_k, \ \forall k \\ -\sum_k \sum_{p\in\mathcal{P}_k} \pi_{kpu} x_{kp} &\geqslant -c\bar{y}_{uc}, \ \forall u \end{aligned}$$

if and only if $\forall l_k, k \in \mathcal{K}$ and $\forall \nu_{uc} \geqslant 0, (u,c) \in \mathcal{E}$, such that $l_k - \sum_u \pi_{kpu}\nu_{uc} \leqslant 0, \forall p \in \mathcal{P}_k$, we get :

$$\sum_k b_k l_k - \sum_u c\bar{y}_{uc}\nu_{uc} \leqslant 0$$

But the above conditions on the multipliers l_k and ν_{uc} mean, as each b_k is non negative, that $l_k = \sum_{u\in\bar{p}} \nu_{uc}$ is the length of the shortest path $\bar{p}$ between origin o_k and destination d_k with arc lengths ν_{uc}. □

Observe that these conditions authorize that some arc saturates when congested, which is not allowed by our subproblem objective function, so that, in practice, we must substitute c by $c_\epsilon = c(1-\epsilon)$, where ϵ is a small positive tolerance.

A way to compute optimal multipliers ν and the corresponding shortest paths consists in verifying the existence of a feasible multicommodity flow for the topology proposed by the master problem (i.e. for given $\bar{y}_{uc}$), by maximizing the dual function

$$\theta(\nu) = \sum_k b_k l_k - \sum_u c_\epsilon \bar{y}_{uc} \nu_{uc}$$

where l_k are the corresponding shortest path lengths as said before. If the maximum value θ^* is nonnegative, there is no feasible multicommodity flow and we obtain the corresponding feasibility cut by forcing the metric inequality in the master. Maximizing θ, which is a non smooth concave function, is not an easy task. A subgradient algorithm was used in our basic implementation to solve that problem.

Indeed, for any non negative ν, we can compute the shortest paths for each commodity with these weights and the partial derivative of the dual function θ with respect to ν_{uc} is :

$$r_{uc} = \sum_{k \in K_u} b_k - c_\epsilon \bar{y}_{uc} \tag{4.8}$$

where K_u is the subset of commodities which use arc u when routed on the shortest paths. The vector r is thus a subgradient of θ at the current solution ν. Observe that $-r_u$ is the residual capacity of arc u when all commodities are routed on the shortest paths. If that partial derivative is positive, the capacity c of arc (u, c) is violated and we must increase ν_{uc} to reduce the number of shortest paths which concurrently use that arc. Any stepsize rule to update the solution in the subgradient direction can be used (see Held et al., 1974, for classical update formulae). We can stop the iterations as soon as some solution with $\theta(\nu) > 0$ is obtained. When the installed capacities are close to support a feasible multicommodity flow, it could be necessary to perform many iterations to converge and the typical slow convergence rate of subgradient algorithms may turn the procedure rather time consuming. We show in Section 5.2 how an alternative method based on the max-cut problem can be used to yield efficient cuts.

4.3. SUMMARIZING THE GENERAL ALGORITHM (MASTER PROGRAM)

(1) Set $\tau = 0$ and choose the convergence tolerance ε_3; initialize (RMP) (cf. Section 5.1);
(2) Solve (RMP). Let (y^τ, t^τ) the optimal solution ;
(3) Test the feasibility of G^τ by the subgradient procedure; if there exist no feasible multicommodity flow, add the corresponding cut of type II and return to 4.3 ;

Else, solve subproblem (SP $-y^\tau$), let $v(y^\tau)$ its optimal value :

If $v(y^\tau) \leqslant t^\tau + \varepsilon_3$ then STOP (y^τ, t^τ) is ε_3-optimal for (P),

Else, compute the optimal multipliers [section 4.2] and add the corresponding cut of type I to (RMP) and return to 4.3 ;

We will discuss in the next section the implementation aspects which are strongly needed to get a computationally efficient algorithm to solve the (CFA) problem.

5. Adding valid inequalities to the master

5.1. INITIALIZATION

In order to reduce the number of iterations, Geoffrion and Graves (1974), and Magnanti et al. (1986) advise to enrich as much as possible the initial master program with valid cuts (see too the use of Pareto-optimal cuts in Magnanti and Wong, 1981).

Benders decomposition exhibits typical slow convergence and its efficiency depends highly on its ability to avoid infeasible topologies (the ones which imply Benders cuts of type II). It is then particularly fruitful to introduce a priori cuts which will help, but won't guarantee, to find feasible topologies. Following that objective and aware that the problem is harder at the beginning of the iterative process, we have introduced the following constraints in the initial (RMP) :

$$\sum_{j \in X} c y_{ij,c} \geqslant \psi_i^+ \quad \forall i \in X \tag{5.9}$$

and

$$\sum_{i \in X} c y_{ij,c} \geqslant \psi_j^- \quad \forall j \in X \tag{5.10}$$

with

$$\psi_i^+ = \sum_{k \in \Psi_i^+} b_k \qquad \psi_j^- = \sum_{k \in \Psi_j^-} b_k$$
$$\Psi_i^+ = \{k \;/\; o_k = i\} \qquad \Psi_j^- = \{k \;/\; d_k = j\}$$

These constraints simply state that the total capacity of the arcs outflowing an origin node or inflowing into a destination node can support the corresponding offer or demand of flow. We will discuss in the remainder of this section some more specific cuts which can be found in the literature under the spelling of cut-set or cut-capacity inequalities for many distinct network design problems. They are all special cases of metric inequalities and some of them can be separated in polynomial time (see Bienstock et al., 1998, for more polynomially separable metric inequalities). We will show too that deeper cuts can be obtained by solving a max-cut problem, which is indeed NP-hard, but can be approximately solved by good heuristics.

5.2. CONNEXITY CUTS AND CUT CRITERION

If, for a given y in $\mathcal{Y}$, $G(y)$ is not sufficiently connex (we mean here that there exist at least one path between each origin–destination pair), the feasibility cut (of type II) discussed in section 3 is a weak cut. Deeper cuts can be generated by sorting the connex components of the graph $G(y)$:

– *Generate the L connex components,* $S_1, \ldots, S_{\mathcal{L}}$.
Let:

$$\omega^+(S_l) = \{(u, c) \in \mathcal{E}, u = (i, j) / i \in S_l \text{ and } j \notin S_l\}$$
$$\omega^-(S_l) = \{(u, c) \in \mathcal{E}, u = (i, j) / j \in S_l \text{ and } i \notin S_l\}$$
$$\Psi_l^+ = \{k / o_k \in S_l \text{ and } d_k \notin S_l\}$$
$$\Psi_l^- = \{k / d_k \in S_l \text{ and } o_k \notin S_l\}$$
$$\psi_l^+ = \sum_{k \in \Psi_l^+} b_k$$
$$\psi_l^- = \sum_{k \in \Psi_l^-} b_k$$

– For $l = 1, \ldots, \mathcal{L}$, add to (RMP) the constraints :

$$\sum_{(u,c) \in \omega^+(S_l)} c.y_{uc} \geqslant \psi_l^+ \tag{5.11}$$

$$\sum_{(u,c) \in \omega^-(S_l)} c.y_{uc} \geqslant \psi_l^- \tag{5.12}$$

More efficient feasibility cuts can be obtained by searching a cut of maximum weight in a weighted multigraph. The multiple arcs are defined by each pair (u, c) associated with the weight $-c_\epsilon y_{uc}$ with, again, $c_\epsilon = (1 - \epsilon)c$ where ϵ is a small positive tolerance to avoid saturation of the arcs. In addition, we add an arc (o_k, d_k) with weight b_k to each origin–destination pair. The maximum weight cut on that graph will generate a feasibility cut if its weight is positive. Indeed, a positive cut in that multigraph implies that, keeping the precedent notations $\omega^+(S)$ for the arcs flowing out of the cut set S and $\Psi^+(S)$ for the set of commodities with origins in S and destinations outside S, we have:

$$- \sum_{(u,c) \in \omega^+(S)} c.y_{uc} + \sum_{k \in \Psi^+(S)} b_k > 0$$

which means that the cut constraint of type (5.11) is violated.

Computing a maximum weight cut is again NP-hard, but good heuristics have been proposed in the literature (see Barahona, 1996, for instance). In Section 5.3 below, these cuts will be referred as ‘cut criterion’.

5.3. SPANNING TREE CUTS

When the y decision variables describe a tree structure (i.e. a connex graph with $n-1$ arcs), the feasibility test is very simple as each commodity can use a single path. When it is infeasible, the added constraints have the same form as the connexity cuts described in the precedent section.

Suppose indeed that $G(y)$ defined by y is a tree and denote by the index $k1$ the unique path associated with each commodity k. Observe that a feasible multicommodity flow on $G(y)$ is uniquely determined with individual values $x_{k1} = b_k$ and total value on each arc u determined by :

$$f_u = \sum_k \pi_{k1u} b_k$$

Let $V_0 = \{u \in U(y) \mid f_u > cy_{uc}\}$. If $V_0 = \emptyset$, the tree is feasible and we need only to compute the corresponding delay value.

If $V_0 \neq \emptyset$, feasibility cuts can be easily obtained as we show below:

PROPOSITION 1. *Let $u = (i, j), u \in V_0$ be an infeasible arc and let S_u and $\mathcal{X} \setminus S_u$ be the two connex components associated with the deletion of arc u from the tree $G(y)$. If δ_u is the set of arcs in $\mathcal{X}$ such that $i \in S_u$ and $j \in \mathcal{X} \setminus S_u$, and γ_u is the set of commodities with source node in S_u and sink node in $\mathcal{X} \setminus S_u$, then a valid inequality is given by :*

$$\sum_{v \in \delta_u} c_v y_{vc} \geqslant \sum_{k \in \gamma_u} b_k \tag{5.13}$$

Proof. For all arc (u, c) of the tree $G(y)$, we have:

$$\sum_{v \in \delta_u} c_v = c_u y_{uc}$$

and

$$\sum_{c \in \mathcal{C}_u} f_{uc} = f_u = \sum_{k \in \gamma_u} b_k$$

Moreover, if $u \in V_0$, the cut associated with S_u is violated, i.e.:

$$f_u > cy_{uc} \implies \sum_{k \in \gamma_u} b_k > \sum_{v \in \delta_u} cy_{vc}$$

The valid cut is no more than part of necessary conditions for a feasible multicommodity flow on $G(y)$ for one given disconnecting arc. □

6. Numerical tests

We have tested the decomposition algorithm for the (CFA) problem on small and medium size networks with different characteristics. In all our tests performed on a Sun Sparc 10 workstation with 32 Mb RAM, the master program has been solved using CPLEX and the subproblems were solved by the Proximal Decomposition method described in Mahey et al. (1995). The test-beds are relatively small networks (up to 25 nodes and 62 arcs) but with dense requirement matrices (up to 500 commodities). These figures reflect however the complexity of some real situations, for instance when dealing with the design of private networks. We will show at the end of this section some experimentation on larger networks, but where initial capacities have been already settled to support the traffic load (see the Network Expansion problem and Table 5). All networks and data come from real-world situations which have been selected with the agreement of France Telecom; more details about their structure can be found in Boyer (1997) (Networks 1–6). The following precisions have been used in all tests: each restricted master is solved by CPLEX with a 10^{-3} precision and the convex subproblems used a 10^{-2} tolerance (the same for ε_1 and ε_2); finally, the final convergence test to stop Benders iterations was taken as $\varepsilon_3 = 10^{-2}$.

Solving much larger design problems with an exact method is probably hopeless as long as global optimality is pursued, but we can expect to be able to lower the cpu times by reducing the computational complexity of the successive master programs. Indeed, big efforts have been made to design fast convergent algorithms for the convex cost multicommodity flow subproblems (see the compared performance of various algorithms among which stands the Proximal Decomposition method in Ouorou et al., 2000). No comparable effort has been done to reduce the master computational cost, and this explains partly why most of the cpu time is spent with CPLEX iterations solving the master. But the key point to explain the variations of the relative weight in cpu time of the master problem with respect to the subproblems is the effect of congestion. It can be illustrated in two distinct situations : the example of network 2 with increasing arc densities (see last column of Table 4); we observe that the master wastes always more than 80% of the total cpu time and that the ratio reaches 99.9% when density is greater than 60%. On the other hand, the example of network 6 where most of the arcs capacities are held fixed, reducing the possibilities to spread congestion among many paths, the situation is reversed and the subproblem is much harder to solve (see Table 5). The congestion effect is here steered by the parameter γ which, as will be explained below, tends to reduce the delay, thus the congestion, when increased.

There are many directions to alleviate the master complexity, some of which have been discussed in many places in the literature. A common feature is the relaxation of the master program which must be controlled to maintain overall convergence (see Geoffrion and Graves, 1974, for example). As our internal solver is CPLEX, we have decided to investigate the possibilities to reduce the number of iterations. A crucial point in most implementations of Benders decomposition is the efficient use of feasibility cuts, either a priori or iteratively generated cuts. The introduction of specific cuts has been described in the previous section where different kinds of constraints have been proposed, i.e.:

– Connexity cuts

– Spanning tree cuts
– Feasibility cuts (cut criterion)

We have compared the relative efficiency of adding the three types of cuts which we described in the precedent section. Let A, B, C and D represent the implementations of the master problem respectively without a priori cuts, with connexity cuts only, with both connexity and tree cuts, and finally with all cuts including cut criterion computed by the max cut heuristic. Algorithm A could solve only a small number of problems, for which the number of iterations was reduced to 22% in average by algorithm B. Algorithm C improves algorithm B by a factor 2 and the cut criterion could still reduce the average number of iterations to 70% of the performance of algorithm C.

Implementation issues resulting from the specific nature of the (CFA) model are concerned too with the choice of the delay/cost parameter γ. We observe on Tables 1 and 2 how sensitive is the convergence with respect to increasing values of that parameter. Recall that different values of γ correspond to different (CFA) problems and, as the general problem is not convex, we cannot drive it to an optimal value (i.e., one which could force the average delay to be feasible with respect to some given upperbound λT). Intuitively, the delay will tend to decrease if we increase γ. On the other hand, the design cost will increase to allow a higher quality of service. The tests reported in Table 1 confirm this intuition. We observe that only four distinct pairs (cost, delay) are available among the 10 tests.

A less intuitive effect is the growth of the number of iterations. Table 2 shows that cpu times, the number of iterations and the number of Benders cuts of type I increase with γ.

Tables 3 and 4 resume the general behaviour of the algorithm with respect to problem parameters, mainly the size of the graph (Tables 3 and 4), the density (Table 4) and the number of different capacities on each arc (Table 3).

The final tests are slightly different from the above as they consider a network expansion problem where the topology is initially fixed such that a given traffic load can flow through the network. To improve the global quality of service, the arcs of that topology can be resized but not eliminated from the network. This means that exactly one positive capacity has to be installed on each link (i.e. constraint (2.4) is forced to an equality). Table 5 shows the impact of the delay/cost parameter on the results and the increasing weight of the convex subproblems in the cpu time. This last fact is due to the increasing difficulty to solve the subproblems with highly congested networks.

7. Conclusion

We have shown an exact method based on Benders decomposition to solve nonlinear mixed-integer models of the (CFA) problem. The application of the decomposition scheme is not straightforward and we have focussed on the use of feasibility

Table 1. Influence of the γ parameter on the cost

Network 2					
Problem size	n (nb of nodes)	m (nb of arcs)	Density $\frac{m}{n(n-1)}$	nb of commod.	nb of capacities
	10	30	0.33	78	4
Tests	γ	Design cost	Delay value	Delay cost	Total cost
Test 1	1.0	947.42	7.53	7.53	954.95
Test 2	2.0	947.42	7.53	15.06	962.48
Test 3	4.0	947.42	7.53	30.13	977.55
Test 4	6.0	953.53	5.64	33.85	987.38
Test 5	8.0	953.53	5.64	45.12	998.65
Test 6	10.0	956.56	5.20	51.99	1008.55
Test 7	12.0	956.56	5.20	62.39	1018.95
Test 8	14.0	959.71	5.06	70.83	1030.54
Test 9	16.0	959.71	5.06	80.96	1040.67
Test 10	18.0	959.71	5.06	91.07	1050.78

Table 2. Influence of the γ parameter on the performance

Network 2					
Problem size	n (nb of nodes)	m (nb of arcs)	Density $\frac{m}{n(n-1)}$	nb of commod.	nb of capacities
	10	30	0.33	78	4
Tests	γ	Iteration number	CPU time (s)	Type I cuts	Total nb of cuts
Test 1	1.0	5	13.9	2	22
Test 2	2.0	5	13.9	2	22
Test 3	4.0	6	19.2	3	23
Test 4	6.0	8	36.3	5	25
Test 5	8.0	14	85.3	11	31
Test 6	10.0	24	243.9	21	41
Test 7	12.0	33	634.5	30	50
Test 8	14.0	47	1367.2	44	64
Test 9	16.0	72	2901.4	69	89
Test 10	18.0	72	2935.7	69	89

Table 3. Twenty-five nodes network with one to three capacities

Network 5					
Problem	n (nb of nodes)	m (nb of arcs)	Density $\frac{m}{n(n-1)}$	nb of commod.	γ
	25	62	0.103	298	1.0
Tests	Nb of capacities	Nb of integer var.	Nb of iterations	CPU I time (s)	Total nb of cuts
Test 1	1	62	23	41.1	198
Test 2	2	124	24	225.9	208
Test 3	3	186	76	24549.7	260

Table 4. Ten nodes network with increasing densities

Network 2					
Problem	n (nb of nodes)	nb of capacities	nb of commod.	max nb of arcs	γ
	10	4	78	90	1.0
Tests	Nb of arcs	Density	Nb of iterations	CPU I time (s)	Master rel. weight
Test 1	18	0.20	4	1.56	85.3
Test 2	24	0.27	4	3.27	94.2
Test 3	30	0.33	5	14.4	98.2
test 4	36	0.40	6	30.98	99.0
test 5	48	0.53	22	1261.3	99.8
test 6	60	0.67	25	4157.48	99.9
test 7	72	0.80	25	8729.5	99.9
test 8	90	1.00	26	24418.3	99.9

cuts which tend to reduce the number of iterations of the master program. We believe that the combination of these cuts with the relaxation of the global optimality of the master will help to solve much larger instances of the network design problem.

The numerical results are restricted to some real communication networks as our aim was mainly to justify the use of exact methods for the design of private communication networks. More tests on different, randomly generated networks should be performed to definitively state the potential of generalized Benders de-

Table 5. Network expansion problem

Network 6					
Problem size	n (nb of nodes)	m (nb of arcs)	Density $\frac{m}{n(n-1)}$	nb of commod.	nb of capacities
	30	72	0.08	433	9
Tests	γ	Iteration number	CPU time (s)	Type I cuts	Master rel. weight
Test 1	0.5	5	673.2	4	0.004
Test 2	0.75	9	1248.6	9	0.007
Test 3	1.0	17	2392.6	15	0.025
Test 4	1.5	48	8092.5	47	0.198

composition for the nonlinear (CFA) problem. The numerical performance of the algorithm is largely dependent on the capacity to solve the successive master programs. We believe that substituting the Cplex solver by an ad hoc algorithm for topology optimization under feasibility constraints will reduce the computational burden of the master problems.

8. Acknowledgements

Research supported by France Telecom under grant 96-1B021

References

Balakrishnan, A. and Graves, S. C. (1989) A composite algorithm for a concave-cost network flow problem. *Networks* 19: 175–202.

Barahona, F. (1996) Network design using cut inequalities. *SIAM J. on Optimization*, 6: 822–837.

Benchakroun, A., J. Ferland and Gascon, V. (1997) Benders decomposition for network design problems with underlying tree structure. *Investigacion Operativa* 6: 165–180.

Benders, J. F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4: 238–252.

Bertsekas, D. P. and Gallager, R. G. (1987) *Data Networks.* Prentice-Hall, Englewood Cliffs, NJ.

Bienstock, D. and Günlük, O. (1996) Capacitated network design—Polyhedral structure and computation. *INFORMS J. of Computing*, 8: 243–259.

Bienstock, D., Chopra, S., Günlük, O. and Tsai, C. Y. (1998) Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming B* 81: 177–199.

Boyer, F. (1997) *Conception et Routage des Réseaux de Télécommunications.* PhD Thesis, Université Blaise Pascal, Clermont-Ferrand.

Eckstein, J. and Fukushima, M. (1993) Some reformulations and applications of the alternating direction method of multipliers, in W. W. Hager, D. W. Hearn and P. M. Pardalos (eds.), *Large-scale Optimization: State of the Art*, Kluwer Academic Publishers, Dordrecht.

Ferreira Filho, V. J. M. and Galvão, R. D. (1994) A Survey of Computer Network Design Problems. *Investigacion Operativa*, 4: 183–211, 1994.

Fratta, L., M. Gerla, and L. Kleinrock. The flow deviation method : an approach to store-and-forward communication network design. *Networks* 3: 97–133.

Gabrel, V. and Minoux, M. (1997) LP relaxations better than convexification for multicommodity network optimization problems with step increasing cost functions. *Acta Mathematica Vietnamica* 22: 128–145.

Gavish, B. (1985) Augmented Lagrangian based bounds for centralized network design. *IEEE Transactions on Communications*, COM-33: 1247–1257.

Gavish, B. and Altinkemer, K. (1990) Backbone network design tools with economic tradeoffs. *ORSA J. on Computing* 2/3: 236–252.

Gavish, B. and Neuman, I. (1989) System for routing and capacity assignment in computer communication network. *IEEE Transactions on Communications* COM-37: 360–366.

Geoffrion, A. M. (1972) Generalized Benders decomposition. *J. of Optimization Theory and Applications* 10: 237–260.

Geoffrion, A. M. and Graves, G. W. (1974) Multicommodity distribution system by Benders decomposition. *Management Science* 20: 822–844.

Gerla, M. (1973) *The Design of Store-and-forward Networks for Computer Communications.* PhD Thesis, UCLA.

Gerla, M. and Kleinrock, L. (1977) On the topological design of distributed computer networks. *IEEE Transactions on Communications* COM-25: 48–60.

Gerla, M., Monteiro, R. and Pazos, R. (1989) Topology design and bandwith allocation in ATM nets. *IEEE Transactions on Selected Area in Communications* SAC-7: 1253–1262.

Held, M., Wolfe, P. and Crowder, H. P. (1974) Validation of subgradient optimization. *Mathematical Programming* 6: 62–88.

Hoang, H. H. (1982) Topological Optimization of networks: A nonlinear mixed integer model employing generalized Benders decomposition. *IEEE Transaction on Automatic Control* AC-27: 164–169.

Holmberg, K. (1994) Solving the staircase cost facility location problem with decomposition and piecewise linearization. *European J. of Operations Research* 75: 41–61.

Johnson, D. S., Lenstra, J. K. and Rinnoy Kan, A. H. G. (1978) The complexity of the network design problem. *Networks* 8: 279–285.

Kleinrock, L. (1972) *Communications, Nets, Stochastic Message Flow and Delay.* Dover.

Luna, H. P. L. and Mahey, P. (2000) Bounds for global optimization of capacity expansion and flow assignment problems. *Operations Research Letters* 26: 211–216.

Mac Gregor Smith, J. and Winter, P. (eds.), (1991) *Topological Network Design. Annals of Operations Research*, 33.

Magnanti, T. L. and Wong, R. T. (1981) Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research*, 29: 464–484.

Magnanti, T. L., Mireault, P. and Wong, R. T. (1986) Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study* 26: 112–154.

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

Mahey, P., Ouorou, A., LeBlanc, L. B. and Chifflet, J. (1995) A new proximal decomposition algorithm for routing in telecommunications networks. *Networks* 31: 227–238.

Minoux, M. (1989) Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks* 19: 313–360.

Onaga, K. and Kakusho, O. (1971) On feasibility conditions of multicommodity flows in networks. *IEEE Trans. on Circuit Theory*, CT-18: 425–429.

Ouorou, A., Mahey, P. and Vial, J. P. (2000) A survey of algorithms for convex multicommodity flow problems. *Management Science* 46: 126–147.

Sanso, B., Soumis, F. and Gendreau, M. (1991) On the evaluation of telecommunication networks reliability using routing models. *IEEE Trans. Communications* COM-39: 1494–1501.