SEPARABLE CONVEXIFICATION AND DCA TECHNIQUES FOR CAPACITY AND FLOW ASSIGNMENT PROBLEMS

P. Mahey[1,2], Thai Q. Phong[1,3] and H.P.L. Luna[4]

Abstract. We study a continuous version of the capacity and flow assignment problem (CFA) where the design cost is combined with an average delay measure to yield a non convex objective function coupled with multicommodity flow constraints. A separable convexification of each arc cost function is proposed to obtain approximate feasible solutions within easily computable gaps from optimality. On the other hand, DC (difference of convex functions) programming can be used to compute accurate upper bounds and reduce the gap. The technique is shown to be effective when topology is assumed fixed and capacity expansion on some arcs is considered.

Résumé. On étudie ici une version continue du problème de dimensionnement et routage dans un réseau de communications, dans lequel les coûts de conception sont combinés aux mesures de délai moyen d'acheminement, engendrant un problème de multiflots avec une fonction objectif non convexe. On propose un encadrement de la valeur optimale par convexification séparable sur les arcs, d'une part, et par calcul d'optima locaux issus d'un modèle DC (différence de fonctions convexes) des fonctions de coût. Cette dernière technique permet de réduire la distance à la valeur optimale et on illustre son efficacité sur des problèmes d'expansion de capacités.

Keywords: Network design, DC optimization, capacity and flow assignment.

1. Introduction

This paper presents an alternative approach for some models and algorithms that have been addressed to problems of telecommunication and computer network expansion and operation. Many such models are special issues of the general network design problem and lead to large-scale non convex or combinatorial optimization problems (see [2] for example). Approximation algorithms and heurisitics are frequently employed to treat these problems and the need for tight bounds of the global optimal value is a crucial issue we want to address here.

The motivation behind our modeling approach concerns the need to integrate the routing mechanisms (commodity flows) together with the control of bandwidth allocation (capacity assignment) in emerging technologies. The integrated problem (denoted hereafter by (CFA) for Capacity and Flow Assignment problem) is classical in network planning, but it has now a greater importance because capacity assignment can also be done in real time.

The (CFA) problem has been first considered by Gerla in his thesis [8] 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.* [4]). Most proposed algorithms in the literature treat alternatively the Capacity Assignment problem and the Flow Assignment problem like in Gerla and Kleinrock [9] or in successive papers by Gavish *et al.* [6, 7]. In [10], Gerla *et al.* 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 ([1, 7] and [19]). Gavish has also introduced Augmented Lagrangians to generate tight lower bounds in [5].

Recently, the (CFA) problem was solved by generalized Benders decomposition where the boolean variables represent the choice of capacities among a given set of facilities [14]. But we can avoid the use of discrete variables by combining the step increasing capacity expansion cost with the convex delay function, yielding a nonconvex but continuous increasing function on each arc. Continuous models have been proposed in the literature (see [21]), but they focus mainly on the effect of economies of scales which gives the concave-cost multicommodity flow problem. These latter problems are in general NP-hard, unless the number of nonlinear variables is kept fixed (see [20]). Further aspects concerning the global optimization of these difficult problems, which are not addressed here, can be found in [12].

We are not concerned here with pure topology design of the backbone network, and assume that an initial graph is given, such that there exists at least one path between each origin-destination pair and that in each arc is installed a given capacity which is supposed to be able to support a standard traffic load. The problem is formulated by minimizing the link expansion costs plus the congestion costs associated with the total flow circulating on each link.

We show in this paper how to compute sharp lower and upper bounds of the optimal value of the capacity expansion problem. In the next section we formulate the model and present the main result, related with a lower bound for a network optimization problem where the variables refer to routing and bandwidth allocation. In Section 3 we show how to solve efficiently the convex multicommodity flow subproblem by a proximal decomposition algorithm. In Section 4, we exploit the DC (difference of convex functions) structure of the arc costs to apply the DCA algorithm for computing local minima. The resulting upper bound is significantly better than the value of the feasible solution given by the convexification step. This allows to reduce the gap as shown by the numerical results in the last section.

2. A CONTINUOUS MODEL FOR NETWORK EXPANSION

We present in this section the network expansion model. The basic component of the model is a digraph $G = (V, E)$ with m nodes and n arcs representing a communication network. Any kind of traffic between a given pair of nodes (O_k, D_k) is treated as a separate commodity k. The kind of traffic indexed by k is also related to the parameter d_k, which represents consumers demand for commodity k. Given a commodity k, we consider a given set of directed paths P_{kh}, $h = 1, \ldots, N_k$ joining O_k and D_k. This set may be the set of all simple directed paths or a restricted set of feasible paths, for instance with a limited number of hops. Let ξ_{kh} be the amount of flow of commodity k through the path P_{kh} and a_{kh} its arc-path incidence vector defined by

$$a_{kh}^j = \begin{cases} 1 & \text{if arc } j \in P_{kh} \\ 0 & \text{otherwise.} \end{cases}$$

Each component x_j of the vector x denotes the total flow on arc j. We assume given a congestion plus leasing cost function $f_j(x_j)$ on each arc j, which is increasing on $[0, +\infty)$ and smooth. It aims at distributing the load among all capacitated arcs to reduce the total cost of leasing (or capacity investment) lines and congestion expressed in monetary values. Remark that the capacity in any arc may be possibly increased affording a given expansion cost, in such a case reducing the marginal time delay. We will not digress here neither on the question of the tradeoff between money and mean time delay, nor on the approximation in this case of any statistical congestion function like Kleinrock's delay function (see for example [2]). Each link j has an initial capacity $c_{0j} \geq 0$ and we assume a fixed cost π_j to expand the capacity from c_{0j} to $c_{1j} = c_{0j} + \delta_j$, $\delta_j > 0$. We can now state the Network

Expansion Problem (NEP) as follows:

$$\min_{\xi \geq 0, x \geq 0} \sum_{j=1}^{n} f_j(x_j) \tag{1}$$

subject to

$$d_k - \sum_{h=1}^{N_k} \xi_{kh} = 0 \qquad k = 1, ..., K \tag{2}$$

$$\sum_{k=1}^{K} \sum_{h=1}^{N_k} a_{kh}^j \xi_{kh} - x_j = 0 \qquad j = 1, \dots, n. \tag{3}$$

Constraints (2) ensures that the demand for each commodity is supplied by the path flows activated for the commodity, while constraints (3) impose that the total flow on each arc is equal to the sum of the individual path flows which use that arc. We seek for path flows, arc flows and implicit arc capacities in order to minimize a total cost that includes capacity and queuing components. We observe that each function f_j depends on the actual arc capacity c_{0j}, on the expanded arc capacity c_{1j} and on the associated expansion cost π_j:

$$f_j(x_j) = \min\{f_j^0(x_j), f_j^1(x_j) + \pi_j\} \tag{4}$$

where each $f_j^q, q = 0, 1$ is typically an increasing function which measures congestion on the arc when its capacity is equal to c_{qj}. For instance, Kleinrock's average delay function will give after an appropriate scaling:

$$f_j^q(x_j) = \frac{x_j}{c_{qj} - x_j} . \tag{5}$$

A typical combined function is represented in Figure 1 (arc index j is omitted on the figure). Indeed, we decide to expand the capacity when the flow reaches the value γ_j, $i.e.$ when the congestion cost is so high that it is worth investing the fixed expansion cost π_j.

Adding delays with expansion costs could sound strange and many approaches have introduced scaling factors to get a consistent cost function. These scaling factors are difficult to estimate, even if they have also an interpretation as a dual multiplier associated with a Quality of Service (QoS) constraint (see [14]), but the nonconvex nature of the model turns the use of marginal costs rather empirical. Another way to interpret the breakpoint γ_j on the graph of f_j (see Fig. 1) is to decide to expand the arc capacity when congestion reaches a given threshhold, $i.e.$ if α is the maximum tolerable percentage of the nominal capacity c_{0j} which can be reached by the total flow x_j, then the expansion cost π_j is such that:

$$\pi_j = f_j^0(\alpha c_{0j}) - f_j^1(\alpha c_{0j}).$$

This means of course that $\gamma = \alpha c_{0j}$.

The main result on which the present paper relies is based on the convexification of each arc cost function to yield a lower bound of the global minimal value. Indeed, these functions are likely to be non convex turning the search of a global optimum a very hard task. The motivation here is the possibility to use efficient convex multicommodity flow algorithms.

The following results are summarized from the technical note [13]. Let X be the non empty convex set of arc flow vectors for which we have feasible multicommodity flows, $i.e.$, there exists $x \in X$ and correspondent path flows ξ_{kh} satisfying constraints (2) and (3). We will denote by conv f the convex hull of a function f, $i.e.$ the greatest convex function majorized by f (to be consistent, that definition requires that f admits at least one minorizing affine function, see [11]).

Proposition 1. *Suppose that each function f_j is bounded below and that problem (1–3) has an optimal solution x^* with optimal value z^*. Then.*

$$\tilde{z} = \inf_{x \in X} \left\{ \sum_j \text{conv} f_j(x_j) \right\}$$

is a lower bound of the optimal value, i.e. $z^* \geq \tilde{z}$. Moreover,

$$z^* - \tilde{z} \leq \sum_j \max_{x_j} [f_j(x_j) - \text{conv} f_j(x_j)] = \Delta.$$

Proof. From the definition of $\tilde{z}$, we have:

$$\tilde{z} \leq \sum_j \text{conv} f_j(x_j), \ \forall x \in X.$$

Then, using the definition of the convex hull of each arc function, we obtain that $\tilde{z} \leq \sum_j f_j(x_j), \ \forall x \in X$, and thus $\tilde{z} \leq z^*$.

Let $\tilde{x} \in X$ be the vector with components $\tilde{x}_j$ for each arc j such that $\tilde{z} = \sum_j \text{conv} f_j(\tilde{x}_j)$.

$$
\begin{aligned}
z^* - \tilde{z} &= \sum_j f_j(x_j^*) - \sum_j \text{conv} f_j(\tilde{x}_j) \\
&\leq \sum_j [f_j(\tilde{x}_j) - \text{conv} f_j(\tilde{x}_j)] \\
&\leq \sum_j \max_{x_j} [f_j(x_j) - \text{conv} f_j(x_j)] = \Delta.
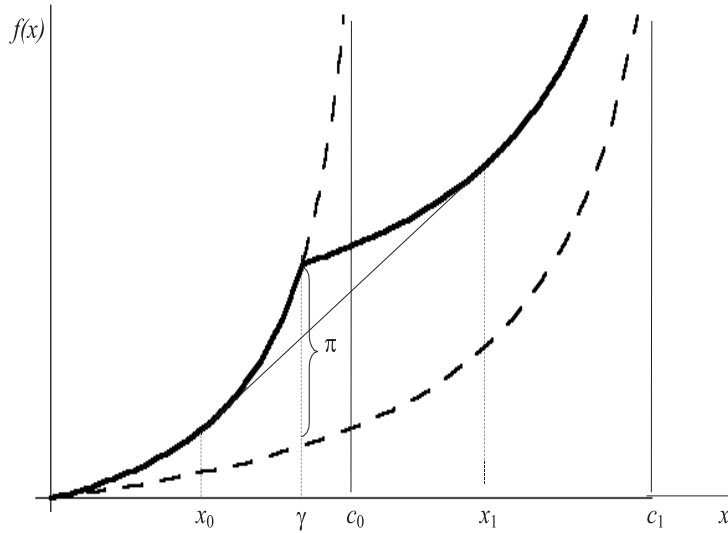\end{aligned}
\tag{6}
$$

□

FIGURE 1. The integrated function of congestion and expansion costs.

Observations:

1. The maximal gap Δ associated with the above lower bound is in general greater than zero as the convex hull of the sum of a set of functions is in general different from the sum of the convex hulls of each function. This is true even if the functions are separable, because we have the coupling constraints (3). These results can be seen as special cases of the analysis on convex hulls given by Falk [3].

2. The lower bound $\tilde{z}$ could have been computed by solving the Lagrangian dual problem associated with (NEP), but that approach would in general not lead to feasible solutions.

3. Our motivation here comes from the fact that the convex hull of a one-dimensional function is relatively easy to compute explicitly. In such a case, we can compute $\tilde{z}$ and the corresponding solution $\tilde{x}$ by solving the convex multicommodity flow problem by any efficient algorithm (see [16] for example). If $f_j(\tilde{x}_j) = \text{conv} f_j(\tilde{x}_j)$, $\forall j$, we can conclude that $\tilde{x}$ is an optimal solution and $\tilde{z}$ is the global optimal value of the problem. Otherwise, if the effective gap at $\tilde{x}$ is greater than a given tolerance, we can switch to a local improvement method, taking $\tilde{x}$ as a starting point, to reduce it gradually.

The convex hull of f_j is easily obtained by computing the tangent points at the flow values x_{0j} and x_{1j}. The calculus may be done explicitly in some simple cases as shown in [13], and we have observed there that the gap can be rather small in practical situations.

Another result which is of practical interest when Kleinrock's delay function (5) is used, is the following:

Proposition 2. *At the same marginal cost on a given link, the residual capacity of the expanded link is greater than the residual capacity of the existing link.*

Proof. Let $r_{qj} = c_{qj} - x_{qj}, j = 0, 1$ be the residual capacities; by computing the derivatives of f_j^q at two points $x_{qj}, j = 0, 1$ with the same marginal costs, we get:

$$\frac{c_{0j}}{r_{0j}^2} = \frac{c_{1j}}{r_{1j}^2}$$

which implies of course that $r_{1j} > r_{0j}$. $\qquad\square$

3. Solving the convex approximation

To solve the convex cost multicommodity flow subproblems, we can use the Proximal Decomposition algorithm described in Mahey *et al.* [15]. It is a primal-dual method for the decomposition of convex separable problems. It was initially designed to solve a generic convex constrained program: minimize a convex lower-semicontinuous F on a closed subspace A. If $A^\perp$ denotes the orthogonal subspace to A, an optimal primal-dual pair (x, z) must lie in the Cartesian product space $A \times A^\perp$. 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 corresponding subspaces.

Many distinct strategies are possible to put the multicommodity flow problem in the generic form. In [15], the arc-path formulation of the multicommodity flow problem was considered and a coupling subspace A were proposed which includes the equations (2) and (3). As the set of paths between o_k and d_k is not known *a priori*, the authors substitute it at each iteration $t = 0, 1, \dots$ by a subset which contains the previously generated paths. The proximal step consists of one-dimensional convex subproblems for each arc j to find aggregate flows x_j^{t+1}. Then, new paths are generated by shortest paths calculation with link costs $f_j'(x_j^{t+1})$ followed by a distributed updating for path flows and potentials. The whole algorithm 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 j, let η_j be the number of paths sharing j. For each path $h \in P_k^t$, let $|a_{kh}|$ denotes the number of arcs of the path. The residual vectors (violation of constraints (2) and (3)) associated with a multicommodity flow x^t are denoted by $r_k(x^t)$ and $r_j(x^t)$ respectively. The corresponding dual variables are denoted by $Z_k, \forall k$ and $z_j, \forall j \in E$.

To alleviate notations in the description of the algorithm below, we will denote by g_j the convex hull function on arc j, *i.e.* $g_j(x_j) = \text{conv } f_j(x_j)$:

Algorithm 1.

1. *Choose the convergence parameters $\varepsilon_1, \varepsilon_2, \lambda > 0$,. Set the iteration index $t = 0$. The initial vectors x^0, z^0, Z^0 may be chosen arbitrarily.*
2. *For each arc j compute*

$$x_j^{t+1} = \arg \min_{0 \le x_j < c_{1j}} \left\{ g_j(x_j) - z_j^t x_j + \frac{\lambda}{2} \left((x_j)^2 - 2 \left(x_j^t + \frac{r_j(x^t)}{\eta_j} \right) x_j \right) \right\}.$$

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 j used for this computation is $g_j'(x_j^{t+1})$. 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:*

$$\xi_{kh}^{t+1} = \max \left(0, \xi_{kh}^t + \frac{1}{\lambda(1 + |a_{kh}|)} \left(Z_k^t - \sum_{j \in kh} z_j^t \right) \right.$$
$$\left. + \frac{1}{1 + |a_{kh}|} \left(\frac{r_k(x^t)}{N_k} - \sum_{j \in kh} \frac{r_j(x^t)}{\eta_j} \right) \right).$$

4. *Update the dual variables*

$$z_j^{t+1} = z_j^t + \frac{\lambda}{\eta_j} r_j(x^{t+1}), \quad Z_k^{t+1} = Z_k^t + \frac{\lambda}{N_k} r_k(x^{t+1}).$$

5. *Test $(x_j^{t+1}, \xi_{kh}^{t+1}, z_j^{t+1}, Z_k^{t+1})$ for convergence and set $t \leftarrow t+1$ if one decides to continue the iteration.*

The Proximal Decomposition algorithm has been shown to be very effective on large multicommodity flow problems with convex costs (see [16]): problems with more than 100 nodes, 900 arcs and fully dense requirement matrices have been solved to ϵ-optimality (with $\epsilon < 10^{-6}$) in less than one hour cpu-time. In these tests, the inner iterations split among the arcs and the one-dimensional subproblems were solved by Newton method. As the convex hull functions g_j are no more twice differentiable, we have used a dichotomic search to solve them approximately and the overall average performance decayed in approximately 30 percent comparatively with the results reported in [16].

4. Computing a sharper upper bound of problem (NEP) by DCA

We will show in this section how to improve the convexification gap by computing good feasible solutions. Since each function defined by (4) is *DC* (Difference of Convex functions), we can compute an approximate local minimum by applying

available methods for DC Optimisation. Recently, the so-called DC Algorithm (DCA), developed in [17], has been reported to be efficient for various large DC optimization problems. It has been applied to the network design problem with staircase cost functions in [18]. In this section we provide some preliminary numerical results for DCA applied to the capacity expansion model (NEP) in comparison with solutions obtained for the convexified problem.

Algorithm DCA is proposed for solving the general DC programming problem

$$(DCP) \qquad \min\{g(x) - h(x) : x \in R^n\}$$

where g, h are l.s.c convex functions on R^n. In its simplified form, DCA consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$ such that

$$x^0 \in \text{Dom}\, \partial h, \quad y^k \in \partial h(x^k)$$

and x^{k+1} is a solution of the problem

$$\min\{g(x) - [h(x^k) + < x - x^k, y^k >] : x \in R^n\}.$$

It has been shown that the sequence $\{(g-h)(x^k)\}$ is decreasing and converges to a limit whenever problem (NEP) is bounded below. Moreover, under certain assumptions, one can show that $\|x^{k+1} - x^k\| \to 0$ and every cluster point x^* of $\{x^k\}$ is a local minimizer. In what follows we give a DC formulation of Problem (NEP) and details of the calculation that we have to perform at each iteration by DCA.

4.1. DC DECOMPOSITION

It is well known that there are infinitely many DC decompositions for a given DC function and the choice of a DC decomposition may have a decisive influence on the performance of DCA for a given problem. When a function is the infimum of two convex functions g_1, g_2 we have the following DC decomposition on the intersection of their domains:

$$\min\{g_1, g_2\} = (g_1 + g_2) - \max\{g_1, g_2\}$$

for all $x \in \text{Dom}\,(g_1) \cap \text{Dom}\,(g_2)$.

In the present case, f_j^0 is not defined on $[c_{0j}, c_{1j}]$, so that we need to add a slight modification of the component functions to get a feasible DC decomposition. Using the specific structure of function $f_j(x_j)$, as represented in Figure 1, we propose the following DC decomposition $f_j = g_j - h_j$ where

$$g_j(x_j) = \begin{cases} f_{0j}(x_j) + f_{1j} + \pi_j & x_j \leq \gamma_j \\ a_j x_j + b_j + f_{1j} + \pi_j & \text{otherwise} \end{cases}$$

$$h_j(x_j) = \begin{cases} f_{1j} + \pi_j & x_j \leq \gamma_j \\ a_j x_j + b_j & \text{otherwise} \end{cases}$$

and

$$a_j = f'_{0j}(\gamma_j), b_j = f_{0j}(\gamma_j) - a_j x_j.$$

Proposition 3. $g_j(x_j) - h_j(x_j)$ *is a D.C. decomposition of* $f_j(x_j)$ *on* $[0, c_{1j})$, *i.e.*

(i) g_j, h_j *are convex functions;*

(ii) $f_j(x_j) = g_j(x_j) - h_j(x_j)$ *for all* $x_j \in [0, c_{1j}]$.

Proof. (ii) is immediate. It is clear that g_j is convex on each interval $[0, \gamma_j)$ and (γ_j, c_{1j}). At the value γ_j ,we have

$$g'_j(\gamma_j - 0) = f'_{0j}(\gamma_j) + f'_{1j}(\gamma_j) = a_j + f'_{1j}(\gamma_j) = g'_j(\gamma_j + 0)$$

hence the graph of g_j is lying above the tangent plane at the point $(\gamma_j, g_j(\gamma_j))$ and so g_j is convex on the whole interval $[0, c_{1j})$. Analogously for h_j we have

$$h'_j(\gamma_j - 0) = f'_{1j}(\gamma_j) < f'_{0j}(\gamma_j) = a_j = h'_j(\gamma_j + 0)$$

which completes the proof.

$\square$

4.2. DCA for problem (NEP)

Denote by S the feasible region of (NEP) and define

$$g(x) = \sum_j g_j(x_j) + \chi_S, \quad h(x) = \sum_j h_j(x_j)$$

(NEP) can be then rewritten as

$$\min\{g(x) - h(x) : x \in R^n\}$$

where χ_S denotes the indicator function of S. The DC-Algorithm applied to (NEP) is summerized as follows

- take $x_j^0 \in [0, c_{1j})$;
- compute

$$y_j^k = \begin{cases} f'_{1j}(x_j) & x_j \leq \gamma_j \\ a_j & \text{otherwise;} \end{cases}$$

- solve the problem

$$\min_{\xi \geq 0, x \geq 0} \sum_{j=1}^{n} g_j(x_j) - y_j^k x_j \tag{7}$$

subject to

$$d_k - \sum_{h=1}^{N_k} \xi_{kh} = 0, \quad k = 1, \dots, K \qquad (8)$$

$$\sum_{k=1}^{K} \sum_{h=1}^{N_k} a_{kh}^j \xi_{kh} - x_j = 0, \quad j = 1, \dots, n; \qquad (9)$$

- stopping criterion:

$$|(g - h)(x^{k+1}) - (g - h)(x^k)| \leq \varepsilon.$$

The problem (7–9) is in fact a separable convex multicommodity flow problem that can be solved efficiently by the Algorithm 1.

4.3. Preliminary results

Tables 1 and 2 give some preliminary results for a network of 9 nodes, 64 expanded arcs and 30 commodities with the same demand $b_k = 1$.

c_0, c_1 are the initial and expanded capacities on all arcs, z_{conv} is the optimal value of the convexified problem and $\bar{z}_{\text{conv}}$ is the objective value of Problem (NEP) at this solution. The values of c_0, c_1 are chosen so that its ratio c_0/c_1 is $1/3$, $1/2$ and $2/3$, respectively. For the DCA, we first take as starting point the zero flow, *i.e.* $x_j = 0$ for all $j = 1, \dots, n$ and the result is given by the optimal value z_{DCA}^1 and the number N_1 of convex multicommodity flow problems to be solved. Then we take as starting point the solution of convexified problem that provides z_{DCA}^2 and N_2 respectively. We run two test examples for different values of α: $\alpha = 50\%$ (Tab. 1) and $\alpha = 70\%$ (Tab. 2).

TABLE 1. $\gamma = 50\%, b_k = 1$.

c_0	c_1	z_{conv}	$\bar{z}_{\text{conv}}$	z_{DCA}^1	N_1	z_{DCA}^2	N_2	Gap (%)
3	9	28.776497	36.977814	33.560944	6	33.799976	3	16.62
4	12	21.470337	28.505762	27.488590	4	26.995285	3	25.73
5	15	17.078259	22.020987	22.015285	3	21.271002	2	24.55
3	6	37.605270	40.163151	39.910885	4	39.122307	3	4.03
4	8	26.797176	29.077398	28.813215	5	28.508305	3	6.38
5	10	20.561415	22.065268	22.271173	4	21.639713	3	5.24
3	4.5	45.107479	45.775230	45.646366	7	45.490089	3	0.84
4	6	29.967743	30.704700	30.565926	4	30.519283	5	1.84
5	7.5	22.221663	22.489975	22.650450	4	22.360434	3	0.62

Theoretically, DCA provides an upper bound for the optimal value of Problem (NEP). As can be seen from Tables 1 and 2, this upper bound is better then

TABLE 2. $\gamma = 70\%, b_k = 1$.

c_0	c_1	z_{conv}	$\bar{z}_{\text{conv}}$	z_{DCA}^1	N_1	z_{DCA}^2	N_2	Gap (%)
3	9	40.560810	52.714867	50.327148	4	49.301311	4	21.54
4	12	28.431997	35.871189	34.849823	3	32.951809	4	15.89
5	15	21.534222	24.789640	23.493114	2	23.566057	2	9.09
3	6	48.937592	53.924282	51.028286	6	52.082657	4	4.27
4	8	32.151947	34.862839	34.848812	3	34.101494	3	6.06
5	10	23.097792	23.949223	23.493397	3	23.493670	3	1.71
3	4.5	55.122379	55.841873	55.381149	5	55.382584	5	0.47
4	6	33.839134	34.338093	34.849529	4	34.124561	3	0.84
5	7.5	23.474411	23.538622	23.493290	3	23.493807	3	0.08

the upper bound $\bar{z}_{\text{conv}}$ provided by the value of the feasible solution obtained by the convexification procedure. We can observe too that DCA terminates rapidly. Indeed, with a stopping tolerance of $\epsilon = 10^{-4}$, we need to solve in average only 4 convex multicommodity flow subproblems. The optimal value by DCA depends on starting points but, in most cases, the solution of the convexified problem is a starting point better than the zero flow. The relative gap $(z_{\text{DCA}} - z_{\text{conv}})/z_{\text{conv}}$ (in percentage) between the computed lower bound and upper bound for the optimal value of Problem (NEP) is shown in the last column of tables. We observe that it is reasonable for the ratios 1/2 and 2/3 and seems not very good when the expended capacity is more larger than its initial value.

5. CONCLUSIONS

The combination of convexification which leads to the best dual lower bound and DC programming which can computes sharp upper bounds is a first step towards the global optimization of large design problems. This paper has proposed a modeling framework to integrate design and operation in modern backbone networks. In practice, the capacity of an arc may be expanded from c_0 to any value of an ordered set of standard capacity levels $c_1, c_2, \dots, c_N$. Let π_n be the fixed cost of expanding the arc from the capacity c_0 to the capacity c_n, for $n = 1, 2, \dots, N$. Let f^0 be the congestion cost function before expansion, and let f^n, $n = 1, 2, \dots, N$ be the congestion cost function after the correspondent expansion. Then, instead of (4), the integrated cost function is defined on $[0, c_N]$ by:

$$f(x) = \min\{f^0(x), f^1(x) + \pi_1, f^2(x) + \pi_2, \dots, f^N(x) + \pi_N\}$$

and a sequential reduction on the number of pieces of the convex hulls can invoke the approach studied here for this general case.

Our analysis fits in the framework referred as *topology tuning* by Gerla *et al.* [10], since we address the issue of bandwidth allocation together with the conventional use of routing strategies. Indeed, our procedure may run as part of network

management for large networks for which the dynamic adjustement of bandwith is complex to be carried out manually. We have shown how to consider discrete changes of capacity within a continuous modeling framework, and we have provided theoretical means in order to evaluate heuristic solutions with respect to global optimality in this class of problems.

References

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

[2] D.P. Bertsekas and R.G. Gallager, *Data Networks*. Prentice-Hall (1987).

[3] J.E. Falk, Lagrange multipliers and nonconvex programs. *SIAM J. Control Optim.* **7** (1969) 534-545.

[4] L. Fratta, M. Gerla and L. Kleinrock, The flow deviation method: an approach to store-and-forward communication network design. *Networks* **3** (1973) 97-133.

[5] B. Gavish, Augmented Lagrangian based bounds for centralized network design. *IEEE Trans. Comm.* **33** (1985) 1247-1257.

[6] B. Gavish and K. Altinkemer, Backbone network design tools with economic tradeoffs. *ORSA J. Comput.* **2/3** (1990) 236-252.

[7] B. Gavish and I. Neuman, A system for routing and capacity assignment in computer communication networks. *IEEE Trans. Comm.* **37** (1989) 360-366.

[8] M. Gerla, *The Design of Store-and-forward Networks for Computer Communications*. Ph.D. Thesis, UCLA (1973).

[9] M. Gerla and L. Kleinrock, On the topological design of distributed computer networks. *IEEE Trans. Comm.* **25** (1977) 48-60.

[10] M. Gerla, J.A.S. Monteiro and R. Pazos, Topology design and bandwith allocation in ATM nets. *IEEE J. Selected Areas in Communications* **7** (1989) 1253-1261.

[11] J.B. Hiriart–Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*. Springer-Verlag (1993).

[12] H. Konno, P.T. Thach and H. Tuy, *Optimization on Low Rank Nonconvex Structures*. Kluwer Academic Publishers, Dordrecht (1997).

[13] H.P.L. Luna and P. Mahey, Bounds for global optimization of capacity expansion and flow assignment problems. *Oper. Res. Lett.* **26** (2000) 211-216.

[14] P. Mahey, A. Benchakroun and F. Boyer, Capacity and flow assignment of data networks by generalized Benders decomposition. *J. Global Optim.* (to appear).

[15] P. Mahey, A. Ouorou, L. LeBlanc and J. Chifflet, A new proximal decomposition algorithm for routing in telecommunications networks. *Networks* **31** (1998) 227-238.

[16] A. Ouorou, P. Mahey and J.P. Vial, A survey of algorithms for convex multicommodity flow problems. *Management Sci.* **46** (2000) 126-147.

[17] P.D. Tao and L.T.H. An, Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Math. Vietnam.* **22** (1997) 289-355.

[18] N.T. Quang, *Une approche dc en optimisation dans les réseaux. Algorithmes, codes et simulations numériques*. Doct. Thesis, Univ. Rouen (1999).

[19] B. Sanso, M. Gendreau and F. Soumis, An algorithm for network dimensioning under reliability considerations. *Ann. Oper. Res.* **36** (1992) 263-274.

[20] H. Tuy, S. Ghannadan, A. Migdalas and P. Varbrand, A strongly polynomial algorithm for a concave production-transportation problem with a fixed number of nonlinear variables. *Math. Programming* **72** (1996) 229-258.

[21] R. Wong, Introduction and recent advances in network design: Models and algorithms, in *Transportation Planning Models*, edited by M. Florian. Elsevier-North-Holland Publ. (1984).