Journal of Global Optimization **22:** 205–232, 2002.

D.C. programming approach for multicommodity network optimization problems with step increasing cost functions

LE THI HOAI AN and PHAM DINH TAO
Modelling, Applied Optimization and Operations Research Group, Laboratory of Mathematics, National Institute for Applied Sciences-Rouen, BP 8, F 76 131 Mont Saint Aignan Cedex, France (E-mail: lethi@insa-rouen.fr / pham@insa-rouen.fr)

Abstract. We address a class of particularly hard-to-solve combinatorial optimization problems, namely that of multicommodity network optimization when the link cost functions are discontinuous step increasing. Unlike usual approaches consisting in the development of relaxations for such problems (in an equivalent form of a large scale mixed integer linear programming problem) in order to derive lower bounds, our d.c.(difference of convex functions) approach deals with the original continuous version and provides upper bounds. More precisely we approximate step increasing functions as closely as desired by differences of polyhedral convex functions and then apply DCA (difference of convex function algorithm) to the resulting approximate polyhedral d.c. programs. Preliminary computational experiments are presented on a series of test problems with structures similar to those encountered in telecommunication networks. They show that the d.c. approach and DCA provide feasible multicommodity flows x^* such that the relative differences between upper bounds (computed by DCA) and simple lower bounds $r := (f(x^*) - LB)/f(x^*)$ lies in the range [4.2 %, 16.5 %] with an average of 11.5 %, where f is the cost function of the problem and LB is a lower bound obtained by solving the linearized program (that is built from the original problem by replacing step increasing cost functions with simple affine minorizations). It seems that for the first time so good upper bounds have been obtained.

Key words: Multicommodity network flows, Step increasing cost function, Capacity assignment problem, Flow assignment problem, Mixed-integer linear program, d.c. (difference of convex functions) program, Polyhedral d.c. program, Relaxation techniques, DCA (d.c. algorithm)

Dedicated to Professor R. Horst on the occasion of his 60th birthday

1. Introduction

Minimum cost multicommodity network optimization problems are basic models in the context of many applications such as: telecommunication networks, transportation networks and traffic analysis, logistic, etc. Many such problems lead to large-scale nonconvex or combinatorial optimization problems (see e.g. [4]). These applications have risen many different solutions approaches (see [6, 28, 35]).

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 pairs of nodes and the computation of optimal routes with the smallest packet delay have been treated within a common model, the Capacity and Flow Assignment problem (CFA). This 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 difficulty in computing exact optimal solutions or good approximate solutions (with guaranteed quality) depends very much on the structure and/or mathematical properties of the cost functions on the links. The easiest known special case is when all cost functions are linear (with nonnegative cost per unit flow) since this reduces to shortest path computations. The case of separable convex cost functions (whether differentiable or nonsmooth) may also be considered as a well-solved class, by means of linearization and decomposition techniques (see [7, 36]).

The case of concave (differentiable) cost functions and the linear with fixed costs case have also been studied by various authors (for a survey see [35]). However no practically efficient exact algorithms are known for such problems, at least for solving practical size instances, only good approximate solution algorithms (without a priori quality guarantee) are available (see [1, 35]). One of the reasons for this situation is the lack of lower bounds of reasonable quality to direct tree search in branch and bound approaches (even in the linear with fixed cost case, known lower bounds may be as poor as 40–60 % off the exact optimum value).

The (CFA) problem has been first considered by Gerla in his thesis [14] and the early approaches used Kleinrock's delay and linear design costs, allowing the application of the Flow Deviation algorithm to solving the corresponding convex multicommodity flow problem (see [7]). Most proposed algorithms treat alternatively the Capacity Assignment problem (CA) and the Flow Assignment problem (FA) like in Gerla and Kleinrock [15] or in successive papers by Gavish and collaborators ([11, 12]). In [16], Gerla et al. proposed to embed the packet-switched network into a given backbone facility network and they obtained local optimal solutions to the nonconvex design and routing model. Lagrangian relaxation has been quite often used to split the problem into separate design and routing ([3, 12, 41]). Gavish [10] also introduced Augmented Lagrangians to generate tight lower bounds.

On the other hand, the literature on the (CA) problem (without routing costs or delay bounds) is vast and still growing, for the problem is by itself NP-hard and very important in practice. The explicit modelling of capacities inside a nonconvex cost function has been proposed in the literature, mostly using a concave cost which represents economies of scales [35]. Gabrel and Minoux showed recently [8, 9] how generalized linear programming can generate good lower bounds for the (CA) problem with step increasing cost functions (problem (P_0) defined in Subsection 2.1). More precisely, in order to get better lower bounds, they proposed a relaxation of an equivalent mixed integer linear program (problem (P_2) defined in Subsection 2.3) in terms of a large scale LP model, the exact solution of which may be obtained by combining both column generation and constraint generation. Computational results given in [9] show that, in 90 % of the examples treated, their relaxation is always better: the improvement over the optimal solution to the convexified problem lies in the range 5–25 % (the average improvement is equal to 13.3 %).

Mahey and Luna [32, 33] have studied a continuous version of the (CFA) problem where the design cost is combined with an average delay measure to yield a nonconvex objective function coupled with the multicommodity flow. This problem, called the Network Expansion Problem (NEP), can be reformulated as mixed integer programs with the convex objective function where the boolean variables represent the choice of capacities among a given set of facilities.

Our approach to multicommodity flow problems with step increasing cost functions (P_0) in their continuous framework is based on the d.c. programming and DCA. More precisely, step increasing functions will be approximated as closely as desired by differences of polyhedral convex functions and DCA can then be applied to the approximated polyhedral d.c. program. D.c. programming and DCA were extensively developed by Pham Dinh Tao and Le Thi Hoai An during the last years for solving nonconvex and nonsmooth programming problems (see [23–27, 37, 38] and references therein). Despite its local character, in practice DCA has been successfully applied to many and various nonconvex optimization problems, especially for those in large scale setting, because found local solutions are quite often global ones and DCA proved to be efficient with respect to standard methods. Another important point to be pointed out is that unlike most proposed algorithms in the literature (for nonconvex optimization problems) which compute lower bounds via relaxation techniques, DCA provides upper bounds which are quite often optimal values. This fact is crucial for the use of branch and bound techniques since DCA may considerably reduce the number of branching and bounding and so makes it possible to treat problems of real dimension.

Note that most real world optimization problems are d.c. programs, in particular, the (CA), (FA) and (CFA) problems can be solved in the d.c. programming framework. DCA is actually one of a few algorithms (in the convex analysis approach to d.c. programming) which allows to solve large-scale d.c. programs.

Preliminary computational experiments are presented on a series of test problems with structures similar to those encountered in telecommunication networks. They show that the d.c. approach and DCA provide feasible multicommodity flows x^* such that: $r := (f(x^*) - LB)/f(x^*)$ lies in the range [4.2 %, 16.5 %] with an average of 11.5 %, where f is the cost function of (P_0) and LB is a lower bound obtained by solving the linearized program ($\mathbf{LP}_0$) (that is (P_0) in which the step increasing cost functions f_u are replaced by simple affine minorizations l_u. Evidently r may be better estimated with better estimations of lower bounds LB. As far as we know that is the first time so good upper bounds have been obtained.

The paper is organized as follows: the presentation of multicommodity network optimization problems with step increasing cost functions, its combinatorial optimization problem and the equivalent mixed integer linear program are given in Section 2 with relationships between their optimal solutions. Section 3 is devoted to d.c. programming and DCA for general d.c. programs with a short background indispensable for a good understanding of our approach. The DCA for solving multicommodity network optimization problems with step increasing cost functions is described in Section 4. Finally computational results are reported in the last section.

2. Problems statements and formulations

The network structure is given as a non-directed graph $G = (V, U)$ where S is the set of nodes and U is the set of (non-directed) edges. We denote $|V| = m$ and $| U |= n$.

The problem to be considered is to decide the amount of capacity $x_u \geqslant 0$ to install on each edge u of the network in order to

- satisfy a given set of multicommodity flow requirements: there are K source-sink pairs, and for each $k \in [1, K]$ a given requested flow value d_k has to be routed between the source node $s(k)$ and the sink node $t(k)$,
- satisfy upper bound constraints:

$$0 \leqslant x_u \leqslant \beta_u \qquad \forall u \in U,$$

- minimize the total cost of the network which, in terms of given individual link cost functions $f_u(x_u)$ $(u = 1, \ldots, n)$ may be written as:

$$z = \sum_{u \in U} f_u(x_u).$$

Minimum cost multicommodity flow problems have been extensively studied in the special cases where the cost functions $f_u(x_u)$ are linear [22], linear with fixed cost or nonlinear but continuous or differentiable (see, e.g., [35]).

We are concerned here with the multicommodity network optimization problem in the case of discontinuous step-increasing cost functions. To the best of our knowledge, no systematic study has been carried out before the works in [8, 9] to build relaxations of this problem and to solve relaxed nonconvex programming problems.

The multicommodity network optimization problem with step increasing costs function is in fact of combinatorial nature. We will give below two equivalent formulations, as a minimization of z over a finite set of $x = (x_u)_{u \in U}$ and as a mixed integer linear programming problem.

2.1. MULTICOMMODITY NETWORK OPTIMIZATION PROBLEMS WITH STEP INCREASING COST FUNCTIONS

This problem is formulated as

$$(P_0) \qquad \begin{cases} \min f(x) = \sum_{u \in U} f_u(x_u) \\ x \in \mathcal{X} \cap \mathcal{D} \end{cases}$$

in which, for $u \in U$, the individual link cost function $f_u(x_u)$ is defined on $[0, \beta_u]$ by

$$f_u(x_u) = \begin{cases} \gamma_u^0 = 0 & \text{if } x_u = 0 & i = 0 \\ \gamma_u^i & \text{if } v_u^{i-1} < x_u \leqslant v_u^i & i = 1, .., q(u) - 1 \\ \gamma_u^{q(u)} & \text{if } v_u^{q(u)-1} < x^u \leqslant v_u^{q(u)} = \beta_u & i = q(u) \end{cases}$$

where

- $V_u = \{0 = v_u^0 < v_u^1 < ... < v_u^{q(u)} = \beta_u\}$ is a finite set of values representing the discontinuity points of the $f_u(x_u)$ functions and $0 = \gamma_u^0 < \gamma_u^1 < ... < \gamma_u^{q(u)}$. The function $f_u(x_u)$ has exactly $q(u)$ steps increasing.
- For a given set of commodity flow requirements defined by a list of source-sink pairs $s(k), t(k)$ $(k = 1, ..., K)$ and a list of requirements d_k (amount of the k^{th} flow to be routed between $s(k)$ and $t(k)$, we denote by $\mathcal{X} \subset \mathbb{R}^n$ the set of all feasible multicommodity flows. Thus $x = (x_u)_{u \in U}$ belongs to $\mathcal{X}$ if and only if a feasible multicommodity flow exists when, on each edge $u \in U$, the total capacity installed is x_u.
- $\mathcal{D} = \{x = (x_u)_{u \in U} : x_u \in [0, \beta_u]\ \forall u \in U\}$.

Let us now describe formally the feasible multicommodity flow polyhedron $\mathcal{X}$ within the directed framework. We first define the corresponding directed multi-graph $\mathbf{G} = (\mathbf{V}, \{\mathbf{U}^+, \mathbf{U}^-\})$ from the non-directed graph $G = (\mathbf{V}, U)$ by duplicating each edge $u = (s, t) \in U$ in two arcs $u^+ = (s, t) \in \mathbf{U}^+$ and $u^- = (t, s) \in \mathbf{U}^-$. It follows that

$$(x_u)_{u \in U} = (x_{u^+})_{u^+ \in \mathbf{U}^+} + (x_{u^-})_{u^- \in \mathbf{U}^-},$$

with

$$x_{u^+} = \sum_{k=1}^{K} \xi_{u^+}^k, \; x_{u^-} = \sum_{k=1}^{K} \xi_{u^-}^k,$$

where

$$\begin{pmatrix} \xi_{u^+}^1 \\ \xi_{u^-}^1 \end{pmatrix}_{u \in U} \cdots \begin{pmatrix} \xi_{u^+}^K \\ \xi_{u^-}^K \end{pmatrix}_{u \in U}$$

are K positive simple flows circulating simultaneously on the directed multigraph $\mathbf{G} = (\mathbf{V}, \{\mathbf{U}^+, \mathbf{U}^-\})$.

For $k = 1, .., K$, $\xi^k \in \mathbb{R}_+^{2n}$ is a feasible simple flow for a given requested flow value $d_k \in \mathbb{R}_+$ to be rooted between the source node $s(k)$ and $t(k)$. In other words, ξ^k is solution of the familiar flow conservation equations

$$\mathbf{A}\xi^k = d_k \delta^{k,}$$

where $\mathbf{A}$ is the node-arc incidence matrix of the directed multigraph $\mathbf{G} = (\mathbf{V}, \{\mathbf{U}^+, \mathbf{U}^-\})$ and $\delta^k (\in \mathbb{R}^n)$ is the requirement vector taking 1 at $s(k)$, -1 at $t(k)$ and 0 elsewhere. The matrix $\mathbf{A}$ can be written in the form $[A^+ A^-]$ where the submatrix A^+ (resp. A^-), is the node-arc incidence matrix of the directed graph $(\mathbf{V}, U^+)$, (resp. $(\mathbf{V}, U^-)$). Since $A^+ = -A^-$, we denote the former matrix by A for simplicity. The linear representation as a plyhedral convex set of the set $\mathcal{X}$ of all feasible multicommodity flows x is given below in the node-arc formulation

$$\begin{cases} \mathbf{A}\xi^k = d_k \delta^k & k = 1, .., K \\ x = \sum_{k=1}^{K} (\xi_+^k + \xi_-^k) & , \\ \xi^k \geqslant 0 & \end{cases}$$

where

$$\xi^k = (\xi_u^k)_{u \in U} = \begin{pmatrix} \xi_{u^+}^k \\ \xi_{u^-}^k \end{pmatrix}_{u \in U} = \begin{pmatrix} \xi_+^k \\ \xi_-^k \end{pmatrix}.$$

That system of linear constraints has the usual block diagonal structure

$$\begin{bmatrix} \mathbf{A} & & & & \\ & .. & & & \\ & .. & \mathbf{A} & & \\ & & & .. & \\ & & & \mathbf{A} & \\ \mathbf{II} & .. & \mathbf{II} & .. \ \mathbf{II} & -\mathbf{I}_n \end{bmatrix} \begin{pmatrix} \xi^1 \\ : \\ \\ \xi^i \\ : \\ \xi^K \\ x \end{pmatrix} = \begin{pmatrix} d_1 \delta^1 \\ .. \\ d_i \delta^i \\ .. \\ d_K \delta^K \\ 0 \end{pmatrix}, \tag{1}$$

where $\mathbf{A} = [\ A \ \ -A\ ]$, $\mathbf{II} = [\mathbf{I}_n \ \mathbf{I}_n]$, $\mathbf{I}_n$: the $n \times n$ identity matrix.
Another representation of the polyhedron $\mathcal{X}$ involving the x variables only is given by:

For any $\lambda = (\lambda_1, ..., \lambda_n) \in \mathbb{R}_+^n$, let $\theta(\lambda)$ denote the quantity

$$\theta(\lambda) = \sum_{k=1}^{K} d_k \times l_k^*(\lambda)$$

where $l_k^*(\lambda)$ is the length of the shortest chain joining $s(k)$ and $t(k)$ in G, when each edge $u \in U$ is given length $\lambda_u \geqslant 0$ (note that $\theta(\lambda)$ may be interpreted as the value of the minimum cost multicommodity flow solution when, on each edge u, the cost function $f_u(x_u)$ is linear of the form $\lambda_u x_u$).

Then $x = (x_u)_{u\in U}$ belongs to $\mathcal{X}$ if and only if, for all $\lambda = (\lambda_1, ..., \lambda_n) \in \mathbb{R}^n_+$, we have (see e.g. [17])

$$\sum_{u\in U} \lambda_u x_u \geqslant \theta(\lambda).$$

This result is generally used in relaxation techniques for computing lower bounds of minimum cost multicommodity flow problems.

2.2. DISCRETE MULTICOMMODITY NETWORK OPTIMIZATION PROBLEMS WITH STEP INCREASING COST FUNCTIONS

The 'continuous' multicommodity network optimization with step increasing cost functions (P_0) can be considered as being of discrete nature, thanks to the following result [8, 9]:

PROPOSITION 1. *Problem (P_0) is equivalent to*

$$(P_1) \qquad \begin{cases} \min f(x) = \sum_{u\in U} f_u(x_u) \\ x \in \mathcal{X} \\ x_u \in V_u \quad \forall u \in U \end{cases}$$

where $V_u = \{0 = v_u^0 < v_u^1 < ... < v_u^{q(u)} = \beta_u\}, \quad \forall u \in U.$

Proof. (P_0) is a relaxation of (P_1) since $V_u \subset U$ and $v_u^{q(u)} = \beta_u, \forall u \in U$, so it is enough to prove that any solution $x^0 = (x_u^0)_{u\in U}$ may be converted into a solution to (P_1) having the same objective function value.

For any $u \in U$ we have either $x_u^0 = 0$ or there is $i \in [1, q(u)]$ such that $v_u^{i-1} < x_u^0 \leqslant v_u^i$ and $f_u(x_u^0) = f_u(v_u^i)$.

It follows that the corresponding $x^1 = (x_u^1)_{u\in U}$, defined by $x_u^1 = x_u^0$ in the first case and v_u^i in the second case, belongs to $\mathcal{X}$ (since $x^1 \geqslant x^0$) and $f(x^1) = f(x^0)$. Hence x^1 is a solution to (P_1).

2.3. THE LARGE SCALE MIXED INTEGER LINEAR PROGRAMMING REFORMULATION

In order to globalize DCA it is important to present below the usual reformulation of problem (P_1) as a large scale mixed integer linear programming problem (P_2) (see [9]). These authors proposed an alternative relaxation of (P_2) in terms of a large scale LP model, which can be solved by a generalized linear programming approach combining both column generation and constraint generation. The resulting optimal solutions provide lower bounds to the exact solutions to the minimum cost multicommodity flow problem to be solved.

Consider any node $i \in V$ and r_i the amount of flow requirements to be routed between node i to all other nodes in the network. So

$$r_i = \sum_{k \in K_i} d_k$$

where

$$K_i = \{k \in [1, K] : s(k) = i \text{ or } t(k) = i\}.$$

Clearly, a necessary condition for $x = (x_u)_{u \in U}$ to be feasible solution to (P_1) is that

$$\sum_{u \in w(i)} x_u \geqslant r_i,$$

where $\omega(i)$ denotes the subset of edges having i as an endpoint.

Also, for $t := 1, ..., \delta(i) = |w(i)|$ (degree of node i), define $u = \alpha_i(t)$ as being the index number of the t^{th} edge of $w(i)$. The finiteness of the sets V_u, $u \in U$, implies that of the solution set of the system

$$(\mathbf{I}) \begin{cases} \sum_{u \in w(i)} x_u \geqslant r_i \\ x_u \in V_u \quad \forall u \in w(i) \end{cases}.$$

Each solution may be described as a vector with $\delta(i)$ components.

Denote by A^i the matrix, the columns of which are the various vectors solving $(\mathbf{I})$. It has $\delta(i)$ rows indexed by $t = 1, ..., \delta(i)$, and $P(i)$ columns, indexed by $p = 1, ..., P(i)$. $A^i_{t,p}$ denotes the entry of A^i in the t^{th} row and the p^{th} column. From the definition $A^i_{t,p} \in V_u$, where $u = \alpha_i(t)$.

Associate with each column p of the matrix A^i $(p \in [1, ..., P(i)])$ a cost $\gamma^i_p = \sum_{t=1}^{\delta(i)} f_{\alpha_i(t)}(A^i_{t,p})$ (this is the part of the objective function value corresponding to the edges in $w(i)$ only, when the capacities installed on those edges are the ones appearing as the components of the p^{th} column of A^i).

Now, associated with each column p of the matrix A^i, we consider a $0 - 1$ decision variable y^i_p expressing the selection ($y^i_p = 1$) or not ($y^i_p = 0$) of the p^{th} column of A^i to be part of the solution. The corresponding cost of selecting one and only one of the column vectors of A^i to be part of the solution is

$$\sum_{p=1}^{P(i)} \gamma^i_p y^i_p,$$

the binary variables being constrained to satisfy

$$\sum_{p=1}^{P(i)} y_p^i = 1.$$

Let $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ be any solution of (P_1). Then clearly the subvector of $\overline{x}$ formed by the components $\overline{x}_u$ with $u \in w(i)$ satisfies (**I**). So there exists a $0-1$ y^i vector, denoted by $\overline{y}^i$, such that

$$A^i \overline{y}^i = \begin{pmatrix} \overline{x}_{\alpha_i(1)} \\ \overline{x}_{\alpha_i(2)} \\ . \\ . \\ . \\ \overline{x}_{\alpha_i(\delta(i))} \end{pmatrix} \tag{2}$$

and $e^T \overline{y}^i = 1$ (e denotes the vector of all ones of appropriate dimension, here $P(i)$). The linear system (2) can be more conviently written as $A^i \overline{y}^i - B^i \overline{x} = 0$ where B^i is the $(\delta(i) \times n)$ $0-1$ matrix constructed as follows: for $t \in [1, \delta(i)]$ its only nonzero coefficients in row t has value 1 and belongs to column $\alpha_i(t)$.

Now, all the construction above may be carried out for every node $i = 1, ..., m$ of the network in turn. It follows that, if $\overline{x}$ is any solution to (P_1), then there exists $0-1$ vectors $\overline{y}^i$ $(i = 1, ..., m)$ satisfying the following set of constraints

$$(\mathbf{II}) \begin{cases} \overline{x} \in \mathcal{X} \text{ and} \\ A^i \overline{y}^i - B^i \overline{x} = 0 \\ e^T \overline{y}^i = 1 & (\forall i = 1, ..., m) \\ \overline{y}^i \in \{0, 1\}^{P(i)} \end{cases} .$$

Note that the requirement $\overline{x} \in V_1 \times V_2 \times ... \times V_n$ would be redundant in (**II**).

Consider then the following mixed integer linear programming problem

$$(P_2) \quad \begin{cases} \min \frac{1}{2} \sum_{i=1}^{m} \sum_{p=1}^{P(i)} \gamma_p^i y_p^i \\ A^i y^i - B^i x = 0 & (*) \\ e^T y^i = 1, & (**) \quad \forall i = 1, \dots, m \\ y^i \in \{0, 1\}^{P(i)}, \\ x \in \mathcal{X}. \end{cases}$$

PROPOSITION 2. *([9]) (P_1) and (P_2) are equivalent in the following sense:*

(i) If $\overline{x}$ is any solution to (P_1) and $\overline{y}^i$ $(i = 1, ..., m)$ the associated vectors satisfying (**II**) *then $\overline{x}$ and $\overline{y}^i$ $(i = 1, ..., m)$ form a solution to (P_2).*

(ii) If $\overline{x}$ and $\overline{y}^i$ $(i = 1, ..., m)$ form a solution to (P_2) then $\overline{x}$ is a solution to (P_1)

(iii) (P_1) and (P_2) have the same optimal value.

The number of y variables in (P_2) depends on the number of nodes ($m \geqslant 50$ is typically in problems of practical size), the node degrees (typically an average of 5) and the number of discontinuity points on each edge cost function (typically an average of 5). Therefore, an estimate of the typical number of columns in (P_2) is $m \times 5^5$. On the other hand the total number of constraints (*)–(**) is limited to $m + 2n$.

Another aspect of the problem is that, if we want an accurate linear description of the feasible multicommodity flow polyhedron $\mathcal{X}$, a sufficient number of linear constraints of the x variables will have to be explicitly brought into the model. Therefore, for examples of practical sizes ($m \geqslant 50$ nodes, $n \geqslant 80$ edges, say) we can expect ($\mathbf{P}_2$) to be a large scale mixed integer linear program for which there is no hope of getting guaranteed exact optimal solutions with the best currently available techniques, as claimed Gabrel and Minoux in [8, 9].

Our d.c. approach is directly applied to Problem (P_0) via an approximation of the individual link cost functions $f_u(x_u)$ by differences of convex polyhedral functions. The resulting polyhedral d.c. program will be solved by DCA (Section 4). In the next section we shall present the general framework of d.c. programming and DCA.

3. D.C. Programming and DCA

Here we summarize the material needed for an easy understanding of d.c. programming and DCA which will be used to solve the minimum cost multicommodity flow problem (P_0). Our working space is $\mathbf{X} = \mathbb{R}^n$ equipped with the canonical inner product $\langle \cdot, \cdot \rangle$ and the corresponding Euclidean norm $\| \cdot \|$, thus the dual space $\mathbf{Y}$ of $\mathbf{X}$ can be identified with $\mathbf{X}$ itself. We follow [39] for definitions of usual tools of modern convex analysis where functions could take the infinite values $\pm\infty$. A function $\theta : \mathbf{X} \to \mathbb{R} \cup \{\pm\infty\}$ is said to be proper if it takes nowhere the value $-\infty$ and is not identically equal to $+\infty$. The set of all lower semicontinuous proper convex functions on $\mathbf{X}$ is denoted $\Gamma_0(\mathbf{X})$. For $g \in \Gamma_0(\mathbf{X})$, the *conjugate function* g^* of g is a function belonging to $\Gamma_0(\mathbf{Y})$ and defined by

$$g^*(y) = \sup\{\langle x, y \rangle - g(x) : x \in \mathbf{X}\}$$

and we have $g^{**} = g$.

Let $g \in \Gamma_0(\mathbf{X})$ and let $x^0 \in \operatorname{dom} g$ and $\epsilon > 0$, then $\partial_\epsilon g(x^0)$ stands for the $\epsilon-$ *subdifferential* of g at x^0 and is given by

$$\partial_\epsilon g(x^0) = \{y^0 \in \mathbf{Y} : g(x) \geqslant g(x^0) + \langle x - x^0, y^0 \rangle - \epsilon, \forall x \in \mathbf{X}\}$$

while $\partial g(x^0)$ corresponding to $\epsilon = 0$, stands for the *usual (or exact) subdifferential* of g at x^0. Recall that

$$y^0 \in \partial g(x^0) \Longleftrightarrow x^0 \in \partial g^*(y^0) \Longleftrightarrow \langle x^0, y^0 \rangle = g(x^0) + g^*(y^0).$$

One says that g is *subdifferentiable* at x^0 if $\partial g(x^0)$ is nonempty.

Also, the indicator function χ_C of a closed convex set is defined by $\chi_C(x) = 0$ if $x \in C$, $+\infty$ otherwise.

A function $\theta \in \Gamma_0(\mathbf{X})$ is said to be *polyhedral convex* if [39]

$$\theta(x) = \max\{\langle a^i, x\rangle - \alpha_i : i = 1, \dots, m\} + \chi_S(x), \quad \forall x \in \mathbf{X},$$

where $a^i \in \mathbf{Y}$, $\alpha_i \in \mathbb{R}$ for $i = 1, \dots, m$ and S is a nonempty polyhedral convex set in $\mathbf{X}$. Recall that [39] the conjugate of a polyhedral convex function is polyhedral convexe and the sum of polyhedral convex functions is polyhedral convex too.

Let $\rho \geqslant 0$ and C be a convex subset of $\mathbf{X}$. One says that a function θ: $C \longrightarrow \mathbb{R} \cup \{+\infty\}$ is ρ-convex if

$$\theta[\lambda x + (1-\lambda)x'] \leqslant \lambda\theta(x) + (1-\lambda)\theta(x') \\ - \frac{\lambda(1-\lambda)}{2}\rho\|x - x'\|^2, \forall\lambda \in]0,1[, \forall x, x' \in C.$$

It amounts to saying that $\theta - (\rho/2)\|\cdot\|^2$ is convex on C. The *modulus of strong convexity* of θ on C, denoted by $\rho(\theta, C)$ or $\rho(\theta)$ if $C = \mathbf{X}$, is given by:

$$\rho(\theta, C) = \sup\{\rho \geqslant 0 : \theta - (\rho/2)\|\cdot\|^2 \text{ is convex on } C\}. \tag{3}$$

Clearly, θ is convex on C if and only if $\rho(\theta, C) = 0$. One says that θ is *strongly convex* on C if $\rho(\theta, C) > 0$.

A general *d.c. program* is of the following form with $g, h \in \Gamma_0(\mathbf{X})$

$$(\mathrm{P}_{dc}) \qquad \begin{cases} \alpha = \inf \ f(x) := g(x) - h(x) \\ s.t. \qquad x \in \mathbf{X}, \end{cases}$$

where we adopt the convention $+\infty - (+\infty) = +\infty$ to avoid ambiguity. One says that $g - h$ is a *d.c. decomposition* (or d.c. representation) of f, and g, h are its *convex d.c. components*. If g and h are finite on $\mathbf{X}$, then $f = g - h$ is said to be finite d.c. function on $\mathbf{X}$.

Note that the finiteness of α merely implies that

$$\operatorname{dom} g \subset \operatorname{dom} h \quad \text{and} \quad \operatorname{dom} h^* \subset \operatorname{dom} g^*. \tag{4}$$

Such inclusions will be assumed throughout the paper.

A point x^* is said to be *a local minimizer* of $g - h$ if $g(x^*) - h(x^*)$ is finite (i.e., $x^* \in \operatorname{dom} g \cap \operatorname{dom} h$) and there exists a neighbourhood U of x^* such that

$$g(x^*) - h(x^*) \leqslant g(x) - h(x), \ \ \forall x \in U. \tag{5}$$

Under the convention $+\infty - (+\infty) = +\infty$, the property (5) is equivalent to $g(x^*) - h(x^*) \leqslant g(x) - h(x), \quad \forall x \in U \cap \operatorname{dom} g$.

A point x^* is said to be *a critical point* of $g - h$ if $\partial g(x^*) \cap \partial h(x^*) \neq \emptyset$.

The d.c. duality, (due to Toland [42], who generalized in a very elegant and natural way the early works of Pham Dinh T. on convex maximization programming), associates the d.c. program (P_{dc}) with the following one called its dual d.c. program

$$(D_{dc}) \qquad \begin{cases} \alpha = \inf \ h^*(y) - g^*(y) \\ s.t. \qquad y \in \mathbf{Y} \end{cases}$$

with the help of the functional conjugate notion and states relationships between them. More precisely, as a sort of getting to the root of convex functions (namely a convex function $\theta \in \Gamma_0(\mathbf{X})$ is characterized as *the pointwise supremum of a collection of affine minorizations*, in particular there holds the following expression

$$\theta(x) = \sup\{\langle x, y\rangle - \theta^*(y) : y \in \mathbf{Y}\}, \ \forall x \in \mathbf{X} \tag{6}$$

that will appear in the concept of our DCA again), the d.c. duality is built by replacing, in problem (P_{dc}), the function h with its corresponding expression of (6).

If at least one of convex d.c. components is polyhedral convex, then Problem (P_{dc}) is called a *polyhedral d.c. program.* In this case the dual program (D_{dc}) is also a polyhedral d.c. program.. The special class of polyhedral d.c. programs, which plays a key role in nonconvex optimization, possesses worthy properties, from both theoretical and computational viewpoints, *as necessary and sufficient local optimality conditions, and finite convergence for DCA* (see, e.g., [23, 37, 38]).

Thanks to a *symmetry* in the d.c. duality (the bidual d.c. program is exactly the primal one) and the *d.c. duality transportation of global minimizers* ([23, 37, 38]), solving a d.c. program implies solving the dual one and *vice versa*. It may be useful if one of them is easier to solve than the other. The equality of the optimal value in the primal and dual programs can be easily translated (with the help of ϵ-*subdifferential* of the d.c. components) in global optimality conditions, namely x^* is a global solution to (P_{dc}) if and only if

$$\partial_\epsilon h(x^*) \subset \partial_\epsilon g(x^*) \ \forall \epsilon \geqslant 0.$$

Unfortunately as we should expect, these conditions are rather difficult to use for devising solution methods to d.c. programs.

Local d.c. optimality conditions constitute (with the d.c. duality) the basis of the DCA. In general, it is not easy to state them as in global d.c. optimality and there have been found very few properties which are useful in practice [23, 37, 38].

REMARK 1. Problem (P_{dc}) is a false d.c. program if the function $f = g - h$ is actually convex on $\mathbf{X}$. For example the problem of minimizing a convex function f on $\mathbf{X}$ can be (equivalently) casted in the d.c. framework as that of minimizing a d.c. function $g - h$, where $g = f + \theta$, $h = \theta$ and θ is a finite convex function on $\mathbf{X}$. In such case it is proved that the subdifferential inclusion $\partial h(x^*) \subset \partial g(x^*)$ is equivalent to $0 \in \partial f(x^*)$, i.e. x^* is a solution to the problem being considered. We indicate in [23, 37, 38] other ways of generating equivalent d.c. programs by using regularization techniques. These features proper to the d.c. framework are crucial in the use of the DCA for solving nonconvex problems (or false d.c. problems). There are as many DCA as there are d.c. decompositions.

The DCA for general d.c. programs.

The DCA consists in the construction of the two sequences $\{x^k\} \subset \mathbf{X}$ and $\{y^k\} \subset \mathbf{Y}$ (of attempting candidates for being primal and dual solutions respectively) that we improve at each iteration (the sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing) in an appropriate way such that their respective limits x^∞ and y^∞satisfy the local optimality condition

$$\partial h(x^\infty) \subset \partial g(x^\infty) \text{ and } \partial g^*(y^\infty) \subset \partial h^*(y^\infty),\ i.e.\ (x^\infty, y^\infty) \in \mathcal{P}_l \times \mathcal{D}_l,$$

or are critical points of $g - h$ and $h^* - g^*$ respectively.

These sequences are generated as follows: x^{k+1} (resp. y^k) is a solution to the *convex program* (P_k) (resp. (D_k)) defined by

$$(\mathrm{P}_k) \qquad \begin{cases} \inf & \{g(x) - [h(x^k) + \langle x - x^k, y^k\rangle]\} \\ s.t. & x \in \mathbf{X}, \end{cases}$$

$$(\mathrm{D}_k) \qquad \begin{cases} \inf & \{h^*(y) - [g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle]\} \\ s.t. & y \in \mathbf{Y}. \end{cases}$$

In view of the relation: (P_k) (resp. (D_k)) is obtained from (P_{dc}) (resp. (D_{dc})) by replacing h (resp. g^*) with its affine minorization defined by $y^k \in \partial h(x^k)$ (resp. $x^k \in \partial g^*(y^{k-1})$), the DCA yields the next scheme:

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k). \tag{7}$$

It corresponds actually to the *simplified* DCA (which will be shortly called DCA through the paper for simplicity) where x^{k+1} (resp. y^k) is arbitrarily chosen in $\partial g^*(y^k)$ (resp. $\partial h(x^k)$. In the *complete* form of DCA, we impose the following natural choice

$$x^{k+1} \in \arg\min\{g(x) - h(x) : x \in \partial g^*(y^k)\} \tag{8}$$

and

$$y^k \in \arg\min\{h^*(y) - g^*(y) : y \in \partial h(x^k)\} \tag{9}$$

Problems (8) and (9) are equivalent to convex maximization problems (10) and (11) respectively

$$x^{k+1} \in \arg\min\{\langle x, y^k\rangle - h(x) : x \in \partial g^*(y^k)\} \tag{10}$$

$$y^k \in \arg\min\{\langle x^k, y\rangle - g^*(y) : y \in \partial h(x^k)\}. \tag{11}$$

The complete DCA ensures that $(x^\infty, y^\infty) \in \mathcal{P}_l \times \mathcal{D}_l$. It can be viewed as a sort of decomposition approach of the primal and dual problems (P_{dc}), (D_{dc}). From a practical point of view, although Problems (8) and (9) are simpler than (P_{dc}), (D_{dc}) (we work in $\partial h(x^{k+1})$ and $\partial g^*(y^k)$ with convex maximization problems), they remain nonconvex programs and thus are still hard to solve. In practice, except the cases where the convex maximization problems (10) and (11) are easy to treat, one generally uses the simplified DCA to solve d.c. programs.

The DCA was introduced as an extension of the subgradient algorithms (for convex maximization programming) to d.c. programming by Pham Dinh Tao in 1986. But this field has been really developed from 1994 only with joint works by Le Thi Hoai An and Pham Dinh Tao (see [23], [37], [38] and the references therein) for solving nonsmooth nonconvex optimization problems. To our knowledge, DCA is actually one of a few algorithms (in the convex analysis approach to d.c. programming) which allows to solve large-scale d.c. programs.

It had been proved in Pham Dinh Tao and Le Thi Hoai An [23, 37, 38] that, for the simplified DCA, we have

(i) The sequences $\{g(x^k)-h(x^k)\}$ and $\{h^*(y^k)-g^*(y^k)\}$ are decreasing and
- $g(x^{k+1})-h(x^{k+1}) = g(x^k)-h(x^k)$ if and only if $y^k \in \partial g(x^k)\cap\partial h(x^k)$, $y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1})$ and $[\rho(g) + \rho(h)]\|x^{k+1} - x^k\| = 0$.
- $h^*(y^{k+1})-g^*(y^{k+1}) = h^*(y^k)-g^*(y^k)$ if and only if $x^{k+1} \in \partial g^*(y^k)\cap \partial h^*(y^k)$, $x^{k+1} \in \partial g^*(y^{k+1}) \cap \partial h^*(y^{k+1})$ and $[\rho(g^*) + \rho(h^*)]\|y^{k+1} - y^k\| = 0$.

In such a case DCA terminates at the k^{th} iteration.

(ii) If $\rho(g) + \rho(h) > 0$ (resp. $\rho(g^*) + \rho(h^*) > 0$), then the series $\{\|x^{k+1} - x^k\|^2\}$ (resp. $\{\|y^{k+1} - y^k\|^2\}$) converges.

(iii) If the optimal value α of problem (P_{dc}) is finite and the sequences $\{x^k\}$ and $\{y^k\}$ are bounded then every limit point x^∞ (resp. y^∞) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).

(iv) DCA has a linear convergence for general d.c. programs.

(v) In polyhedral d.c. programs, the sequences DCA $\{x^k\}$ and $\{y^k\}$ contain finitely many elements and DCA has a finite convergence.

We have the same results for the complete DCA, except that in (i) (resp. (iii)) we must add the following property:

$\partial h(x^k) \subset \partial g(x^k)$ and $\partial g^*(y^k) \subset \partial h^*(y^k)$ (resp. $\partial h(x^\infty) \subset \partial g(x^\infty)$ and $\partial g^*(y^\infty) \subset \partial h^*(y^\infty)$).

The above description of DCA does not really reveal the main features of this approach which could partly explain the qualities *(running time, robustness, stability, rate of convergence and globality of sought solutions)* of DCA from the computational viewpoint. For a deeper insight into DCA the reader is referred to [26].

REMARK 2. So there are as many DCA as there are d.c. decompositions and it is of particular interest to study various equivalent d.c. forms for the primal and dual d.c. problems. It is worth mentioning, for instance, that by using conjointly suitable d.c. decompositions of convex functions and proximal regularization techniques we can obtain the proximal point algorithm and the Goldstein-Levitin-Polyak subgradient method (in convex programming) as special cases of DCA.

The choice of the d.c. decomposition of the objective function in a d.c. program and the initial point for DCA are open questions to be studied for the specific structure of the problem being considered. In practice, for solving a given d.c. program, we try to choose g and h such that sequences $\{x^k\}$ and $\{y^k\}$ can be easily calculated, i.e. either they are in explicit form or their computations are inexpensive.

We shall apply *all these d.c. enhancement features* to solve multicommodity network optimization with step increasing cost functions (P_0) which are formulated as d.c. programs.

4. Approximate polyhedral d. c. programs to (P_0) and solutions by DCA

Since the individual link cost functions $f_u(x_u)$ are discontinuous on $[0, \beta_u]$, the objective function $f(x)$ is not a d.c. function on $\mathbf{X} = \mathbb{R}^n$. There are many ways to approximate the function f by d.c. functions on $\mathbf{X}$ to build approximate d.c. programs (see [25] and Nguyen's PhD Thesis [30] where our d.c. models have been used). According to the special structure of $f_u(x_u)$ and DCA's good behaviour in polyhedral d.c. programs [23, 26, 37, 38]), we have chosen *polyhedral d.c. functions* to approximate f.

4.1. APPROXIMATE POLYHEDRAL D.C. FUNCTIONS TO THE STEP INCREASING COST FUNCTIONS f_u

Beside the discontinuity points $0 = v_u^0 < v_u^1 < \ldots < v_u^{q(u)} = \beta_u$ we introduce the following ones $\widetilde{v}_u^i$, $i = 0, \ldots, q(u) - 1$ such that

$$v_u^0 < \widetilde{v}_u^0 < \widetilde{v}_u^1 < v_u^1 < \ldots < \widetilde{v}_u^{q(u)-1} < v_u^{q(u)-1} < v_u^{q(u)}$$

and $\max\{\widetilde{v}_u^0 - v_u^0, v_u^i - \widetilde{v}_u^i : i = 1, \ldots, q(u) - 1\} \leqslant \eta$, a positive number sufficiently small, and the associated (finite) polyhedral convex functions on $\mathbb{R}$, $L_u^i(x_u), i = 0, \ldots, q(u)$:

- L_u^0 is constantly equal to γ_u^0 on $\left]-\infty, v_u^0\right]$, affine on $\left[\gamma_u^0, +\infty\right[$ such that $L_u^0(v_u^0) = \gamma_u^0$ and $L_u^0(\widetilde{v}_u^0) = \gamma_u^1$ (with slope c_u^0)
- L_u^i is constantly equal to γ_u^i on $\left]-\infty, \widetilde{v}_u^i\right]$, affine on $\left[\widetilde{v}_u^i, +\infty\right[$such that $L_u^i(\widetilde{v}_u^i) = \gamma_u^i$ and $L_u^i(v_u^i) = \gamma_u^{i+1}$ (with slope c_u^i) for $i = 1, .., q(u) - 1$,
- $L_u^{q(u)}(x_u) = \gamma_u^{q(u)}$ for $u \in \mathbb{R}$.

Consider now the piecewise linear function $F_u(x_u)$, for $u \in U$, defined on $\mathbb{R}$ by

$$F_u(x_u) = \begin{cases} L_u^0(x_u) & \text{if } x_u \leqslant \widetilde{v}_u^0 \\ L_u^1(x_u) & \text{if } \widetilde{v}_u^0 \leqslant x_u \leqslant v_u^1 \\ L_u^i(x_u) & \text{if } v_u^{i-1} \leqslant x_u \leqslant v_u^i,\ i = 2, ..., q(u) - 1 \\ L_u^{q(u)}(x_u) & \text{if } v_u^{q(u)-1} \leqslant x_u \end{cases}$$

which will be used to approximate the function $f_u(x_u)$.

Being the pointwise infimum of the finite collection of polyhedral convex functions L_u^i, $i = 1, ..., q(u)$,

$$F_u = \min\{L_u^i : i = 1, ..., q(u)\}$$

the function F_u is a polyhedral d.c. function with the following d.c. decomposition

$$F_u = G_u - H_u$$

where G_u and H_u are (finite) polyhedral convex functions on $\mathbb{R}$ defined by

$$G_u := \sum_{q=0}^{q(u)} L_u^i, \qquad H_u := \max_{i=0,...,q(u)} \left(\sum_{j=0, j\neq i}^{q(u)} L_u^i \right). \tag{12}$$

In order to simplify computations in DCA for solving the approximate polyhedral d.c. program ($\mathbf{P}_{dc}$) we will express the functions G_u and H_u as pointwise supremum of a finite collection of affine functions. A simple calculation gives the following results: $G(u)(x_u) =$

$$\begin{cases} \sum_{i=0}^{q(u)} \gamma_u^i & \text{if } x_u \leqslant v_u^0 \\ c_u^0 x_u + \sum_{i=0}^{q(u)} \gamma_u^i & \text{if } v_u^0 \leqslant x_u \leqslant \widetilde{v}_u^1 \\ \left(\sum_{j=0}^{i-1} c_u^j \right) x_u + \left[\sum_{i=0}^{q(u)} \gamma_u^i - \sum_{j=1}^{i-1} c_u^j \widetilde{v}_u^j \right] & \text{if } \widetilde{v}_u^{i-1} \leqslant x_u \leqslant \widetilde{v}_u^i, \\ & i = 2, ..., q(u) - 1 \\ \left(\sum_{j=0}^{q(u)-1} c_u^j \right) x_u + \left[\sum_{i=0}^{q(u)} \gamma_u^i - \sum_{j=1}^{q(u)-1} c_u^j \widetilde{v}_u^j \right] & \text{if } x_u \geqslant \widetilde{v}_u^{q(u)-1} \end{cases},$$

and $H_u(x_u) =$

$$\begin{cases} \sum_{i=0}^{q(u)} \gamma_u^i & \text{if } x_u \leqslant \widetilde{v}_u^0 \\ c_u^0 x_u + \sum_{i=0}^{q(u)} \gamma_u^i - c_u^0 \widetilde{v}_u^0 & \text{if } \widetilde{v}_u^0 \leqslant x_u \leqslant v_u^1 \\ \left(\sum_{j=0}^{i-1} c_u^j\right) x_u + \left[\sum_{i=0}^{q(u)} \gamma_u^i - c_u^0 \widetilde{v}_u^0 - \sum_{j=1}^{i-1} c_u^j v_u^j\right] & \text{if } v_u^{i-1} \leqslant x_u \leqslant v_u^i, \\ & i = 2, \ldots, q(u) - 1 \\ \left(\sum_{j=0}^{q(u)-1} c_u^j\right) x_u + \left[\sum_{i=0}^{q(u)} \gamma_u^i - c_u^0 \widetilde{v}_u^0 - \sum_{j=1}^{q(u)-1} c_u^j v_u^j\right] & \text{if } x_u \geqslant v_u^{q(u)-1},\ i = q(u) \end{cases}.$$

It implies that

$$G_u(x_u) = \max\{a_u^i x_u + b_u^i : i = 0, \ldots, q(u)\}\ \forall x_u \in \mathbb{R} \tag{13}$$

where

$$a_u^i = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{j=0}^{i-1} c_u^j & \text{if } i = 1 \ldots, q(u) \end{cases}$$

and

$$b_u^i = \begin{cases} \sum_{i=0}^{q(u)} \gamma_u^i & \text{if } i = 0, 1 \\ \sum_{i=0}^{q(u)} \gamma_u^i - \sum_{j=1}^{i-1} c_u^j \widetilde{v}_u^j & \text{if } i = 2, \ldots, q(u) \end{cases}.$$

Likewise

$$H_u(x_u) = \max\{\widetilde{a}_u^i x_u + \widetilde{b}_u^i : i = 0, \ldots, q(u)\}\ \forall x_u \in \mathbb{R} \tag{14}$$

where

$$\widetilde{a}_u^i = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{j=0}^{i-1} c_u^j & \text{if } i = 1 \ldots, q(u) \end{cases} = a_u^i$$

and

$$\widetilde{b}_u^i = \begin{cases} \sum_{i=0}^{q(u)} \gamma_u^i & \text{if } i = 0 \\ \sum_{i=0}^{q(u)} \gamma_u^i - \sum_{j=1}^{i-1} c_u^j \widetilde{v}_u^j & \text{if } i = 1, ..., q(u). \end{cases}$$

4.2. APPROXIMATE POLYHEDRAL D.C. FUNCTION TO THE OBJECTIVE FUNCTION $f(x)$ OF (P_0)

The approximate polyhedral d.c. functions $F_u(x_u)$ to the link cost functions $f_u(x_u)$ lead naturally to the following approximate function $F(x)$ to the objective function $f(x)$ of (P_0)

$$F(x) = \sum_{u \in U} F_u(x_u), \ \forall x = (x_u) \in \mathbb{R}^n.$$

In other words

$$F = G - H$$

where the functions G and H are (finite) polyhedral convex on $\mathbb{R}^n$

$$G(x) := \sum_{u \in U} G_u(x_u), H(x) := \sum_{u \in U} H_u(x_u), \ \forall x = (x_u) \in \mathbb{R}^n.$$

It follows that F is a polyhedral d.c. function and the resulting approximate polyhedral d.c. program to the multicommodity network optimization problem with step increasing cost functions (P_0) then is of the form

$$\min\{F(x) = G(x) - H(x) \quad : \quad x \in \mathcal{X} \cap \mathcal{D}\} \quad (\mathbf{P}_{dc}).$$

4.3. DCA FOR SOLVING THE APPROXIMATE POLYHEDRAL D.C. PROGRAM (P_{dc})

According to Section 3 and identifying the set of edges U with $\{1, ..., n\}$ the DCA applied to (P_{dc}) consists of constructing two sequences $\{x^k\}$ and $\{y^k\}$ such that

$$\text{Given } x^0 \in \mathbb{R}^n, \ x^k \longrightarrow y^k \in \partial H(x^k) \longrightarrow x^{k+1} \in \partial(G + \chi_{\mathcal{X} \cap \mathcal{D}})^*(y^k)$$

($\chi_{\mathcal{X} \cap \mathcal{D}}$ denotes, as indicated in Section 3, the indicator function of the closed convex set $\mathcal{X} \cap \mathcal{D}$ in $\mathbb{R}^n$).

4.3.1. *Computing* $\partial H(x)$

Since

$$H(x) := \sum_{u=1}^{n} H_u(x_u),\ \forall x = (x_u) \in \mathbb{R}^n,$$

we have [18, 19]

$$\partial H(x) = \prod_{u=1}^{n} \partial H_u(x_u),\ \forall x = (x_u) \in \mathbb{R}^n. \tag{15}$$

Instead of (12) the computation of $\partial_u H_u(x_u)$ is simpler using the expression of the finite polyhedral convex function $H_u(x_u)$ as the pointwise supremum of the finite collection of affine functions (14). Indeed we then have [18, 19]

$$\partial H_u(x_u) = co\{\widetilde{a}_u^i : i, ..., q(u), \widetilde{a}_u^i x_u + \widetilde{b}_u^i = H_u(x_u)\} \tag{16}$$

(co stands for the *convex hull*).

4.3.2. *Computing* $\partial(G + \chi_{\mathcal{X}\cap\mathcal{D}})^*(y^k)$

Unlike the computation of $\partial H(x)$ that is explicit, the computation of a subgradient of $\partial(G + \chi_{\mathcal{X}\cap\mathcal{D}})^*(y^k)$ amounts to minimizing the polyhedral convex function $G(x) - \langle x, y\rangle$ over the compact convex $\mathcal{X} \cap \mathcal{D}$

$$\min\{G(x) - \langle x, y^k\rangle : x \in \mathcal{X} \cap \mathcal{D}\} \quad (CP)^{k+1}.$$

More precisely $\partial(G + \chi_{\mathcal{X}\cap\mathcal{D}})^*(y^k)$ is exactly the solution set of this problem.

Using the expression of the polyhedral convex function $G(x)$ as the pointwise supremum of the finite collection of affine functions (13), this problem becomes

$$\min\left\{\sum_{u=1}^{n} \max_{i=0,...,q(u)}\{(a_u^i - y_u^k)x_u + b_u^i\} : x \in \mathcal{X} \cap \mathcal{D}\right\}.$$

It is equivalent to the linear program

$$\min\left\{\begin{array}{ll} \sum_{u=1}^{n} t_u : & (a_u^i - y_u^k)x_u + b_u^i \leqslant t_u, i = 0, ..., q(u), u = 1, ..., n, \\ & x \in \mathcal{X} \cap \mathcal{D},\ t \in \mathbb{R}^n \end{array}\right\} (LP)^{k+1}$$

in the following sense:

(i) If x^* is an optimal solution to $(CP)^{k+1}$ then (x^*, t^*), with $t_u^* = \max_{i=0,...,q(u)} \{(a_u^i - y_u^k)x_u + b_u^i\}$ for $u = 1, ..., n$, is an optimal solution to $(LP)^{k+1}$.

(ii) If (x^*, t^*) is an optimal solution to $(LP)^{k+1}$ then $t_u^* = \max_{i=0,...,q(u)}\{(a_u^i - y_u^k)x_u + b_u^i\}$ for $u = 1, ..., n$ and x^* is an optimal solution to $(CP)^{k+1}$.

Problem $(LP)^{k+1}$ can be written as

$$\begin{cases} \min\limits_{(t,x)} \langle \mathbb{1}, t \rangle \\ \left[\mathbf{X}^k \ \ \mathbf{T}^k \right] \begin{pmatrix} x \\ t \end{pmatrix} \leqslant -Ib \quad (*) \qquad (\mathbf{LP})^{k+1} \\ x \in \mathcal{X} \cap \mathcal{D} \\ t \in \mathbb{R}^n \end{cases}$$

where

$$\left[\overbrace{\begin{matrix} X_1^{q(1)} & & & \\ & .. & & \\ & & X_u^{q(u)} & \\ & & & .. \\ & & & & X_n^{q(n)} \end{matrix}}^{\mathbf{X}^k} \ \ \overbrace{\begin{matrix} -\mathbb{1}_1^{q(1)} & & & \\ & .. & & \\ & & -\mathbb{1}_u^{q(u)} & \\ & & & .. \\ & & & & -\mathbb{1}_n^{q(n)} \end{matrix}}^{\mathbf{T}^k} \right] \begin{pmatrix} x_1 \\ : \\ x_u \\ : \\ x_n \\ t_1 \\ : \\ t_u \\ : \\ t_n \end{pmatrix}$$

$$\leqslant \overbrace{\begin{pmatrix} -Ib_1 \\ .. \\ -Ib_u \\ .. \\ -Ib_n \end{pmatrix}}^{-Ib},$$

$$X_u^{q(u)} = (a_u^1 - y_u^k, .., a_u^{q(u)} - y_u^k)^T, \quad \mathbb{1}_u^{q(u)} = (1, .., 1)^T, \quad Ib_u = (b_u^1, .., b_u^{q(u)})^T.$$

Finally, using the matrix formulation (1) of the set of all feasible multicommodity flows $\mathcal{X}$, the set of linear constraints in $(\mathbf{LP})^{k+1}$ has the block diagonal structure with $[mK + n + \sum_{u=1}^{n} q(u)]$ rows and $[2n(K+1)]$ columns

$$\begin{bmatrix} \mathbf{A} & & & & & & \\ & .. & & & & & \\ & & \mathbf{A} & & & & \\ & & & .. & & & \\ & & & & \mathbf{A} & & \\ \mathbf{II} & .. & \mathbf{II} & .. & \mathbf{II} & -\mathbf{I}_n & \\ & & & & & \mathbf{X}^k & \mathbf{T}^k \end{bmatrix} \begin{pmatrix} \xi^1 \\ . \\ . \\ \xi^i \\ . \\ . \\ \xi^K \\ x \\ t \end{pmatrix} \begin{matrix} = \\ .. \\ = \\ .. \\ = \\ = \\ \leqslant \end{matrix} \begin{pmatrix} d_1\delta^1 \\ .. \\ d_i\delta^i \\ .. \\ d_K\delta^K \\ 0 \\ -Ib \end{pmatrix}. \tag{17}$$

4.4. DESCRIPTION OF THE DCA FOR SOLVING MULTICOMMODITY NETWORK OPTIMIZATION PROBLEMS WITH STEP INCREASING COST FUNCTIONS (P_0)

We are now in a position to summarize the DCA for solving (P_0). It is the DCA applied to $(\mathbf{P}_{dc})$ (Subsection 4.2) and can be described as follows:

DCA for solving ($\mathbf{P}_0$):

1. *Initialization*: Let $x^0 \in \mathbb{R}^n$ be given and $k \geqslant 0$.
2. *Compute* y^k by using (15), (16) and solve $(\mathrm{LP})^{k+1}$ to compute x^{k+1}.
3. *Stopping rule*: If $f(x^k) - f(x^{k+1}) \leqslant \varepsilon$, terminate. Otherwise increase k by 1 and return to 1.

4.5. RELAXED LINEAR PROBLEMS AND LOWER BOUNDS FOR (P_0)

Consider the following convex optimization problem

$$\min\{(f + \chi_{\mathcal{X}\cap\mathcal{D}})^{**}(x) : x \in \mathbb{R}^n\}(P_0)^{**}$$

that is obtained by replacing the objective function $(f + \chi_{\mathcal{X}\cap\mathcal{D}})$ of (P_0) with its closed convex hull $(f + \chi_{\mathcal{X}\cap\mathcal{D}})^{**}$, (i.e. the pointwise supremum of all affine functions majorized by f) [39, 18]. It is known Problems (P_0) and $(P_0)^{**}$ have the same optimal value and the solution set of (P_0) is contained in the solution set of $(P_0)^{**}$ [18]. But the function $(f + \chi_{\mathcal{X}\cap\mathcal{D}})^{**}$ is impractical and we should be content with the closed convex hull of $(f + \chi_{\mathcal{D}})$ and the resulting convex optimization problem

$$\min\{(f + \chi_{\mathcal{D}})^{**}(x) : x \in \mathcal{X}\} \quad (\mathbf{CP}).$$

The following relations between (P_0) and ($\mathbf{CP}$) are straighforward

(i) $(f + \chi_{\mathcal{D}})^{**}(x) \leqslant (f + \chi_{\mathcal{X}\cap\mathcal{D}})^{**}(x) \leqslant (f + \chi_{\mathcal{X}\cap\mathcal{D}})(x)$ for all $x \in \mathbb{R}^n$,

(ii) the optimal value of ($\mathbf{CP}$) is a lower bound for ($\mathbf{P}_0$),

(iii) if $(f + \chi_{\mathcal{D}})^{**}(x^*) = f(x^*)$ where x^* is an optimal solution to ($\mathbf{CP}$), then ($\mathbf{P}_0$) and ($\mathbf{CP}$) have the same optimal value and $\overline{x}$ solves (P_0) if and only if $\overline{x}$ solves ($\mathbf{CP}$) and $(f + \chi_{\mathcal{D}})^{**}(\overline{x}) = f(\overline{x})$.

The tractability of $(f + \chi_{\mathcal{D}})^{**}$ comes from the separability of both the bound constraints $\mathcal{D} = \Pi_{u\in U}[0, \beta_u]$ and the total cost function f

$$f(x) = \sum_{u\in U} f_u(x_u).$$

Indeed we have

$$(f + \chi_{\mathcal{D}})(x) = \sum_{u\in U}(f_u + \chi_{[0,\beta_u]})(x_u)$$

and then the closed convex hull of $(f + \chi_{\mathcal{D}})$ can be expressed as

$$(f + \chi_{\mathcal{D}})^{**}(x) = \sum_{u\in U}(f_u + \chi_{[0,\beta_u]})^{**}(x_u).$$

The next result shows that $(f + \chi_{\mathcal{D}})$ is piecewie linear on $\mathcal{D}$.

PROPOSITION 3. *For each $u \in U$, the functions $(f_u + \chi_{[0,\beta_u]})^*$ and $(f_u + \chi_{[0,\beta_u]})^{**}$ are piecewise linear on $\mathbb{R}$ and on $[0, \beta_u]$, respectively, and we have*

(i) $(f_u + \chi_{[0,\beta_u]})^*(y_u) = \max\{s_u^k y_u - \eta_u^k : k = 0, ..., 2q(u)\}$ *for* $y_u \in \mathbb{R}$*, where*

$$s_u^k = \begin{cases} v_u^k & \text{if } k = 0, ..., q(u) - 1 \\ v_u^{k-q(u)+1} & \text{if } k = q(u), ..., 2q(u), \end{cases}$$

$$\eta_u^k = \begin{cases} \gamma_u^k & \text{if } k = 0, ..., q(u) - 1 \\ \gamma_u^{k-q(u)+1} & \text{if } k = q(u), ..., 2q(u). \end{cases}$$

(ii) $(f_u + \chi_{[0,\beta_u]})^{**}(x_u) = \min\{\sum_{k=0}^{2q(u)} \lambda_u^k \eta_u^k : x_u = \sum_{k=0}^{2q(u)} \lambda_u^k s_u^k,\ \lambda_u^k \geqslant 0,\ k = 0, ..., 2q(u),$ *and* $\sum_{k=0}^{2q(u)} \lambda_u^k = 1\}$, *for* $x_u \in$ *dom* $(f_u + \chi_{[0,\beta_u]})^{**} = [0, \beta_u]$.

Proof. (i) is easily proved and (ii) is a simple consequence of (i) [18, 39].

The convexified problem

$$(\mathbf{CP}) \qquad \begin{cases} \min f + \chi_{\mathcal{D}})^{**}(x) = \sum_{u \in U} (f_u + \chi_{[0,\beta_u]})^{**}(x_u) \\ x \in \mathcal{X} \end{cases}$$

can then be equivalently converted in to the following linear program

$$\min \sum_{u \in U} \tau_u \tag{18}$$

$$\sum_{k=0}^{2q(u)} \lambda_u^k \eta_u^k \leqslant \tau_u, \text{ for } u \in U$$

$$\sum_{k=0}^{2q(u)} \lambda_u^k s_u^k \in \mathcal{X},$$

$$\lambda_u^k \geqslant 0 \text{ for } k = 0, ..., 2q(u) \text{ and } \sum_{k=0}^{2q(u)} \lambda_u^k = 1$$

whose optimal value is a lower bound for the multicommodity network optimization problem (P_0).

Finally the linearized problems ($\mathbf{LP}_0$) obtained from ($\mathbf{P}_0$) by replacing each link cost function f_u with simple affine minorization l_u of f_u on $[0, \beta_u]$.

$$(\mathbf{LP}_0) \qquad \begin{cases} \min l(x) = \sum_{u \in U} l_u(x_u) \\ x \in \mathcal{X} \cap \mathcal{D} \end{cases}$$

could also be used to provide lower bounds to Problem of (P_0). Since $l_u \leqslant (f_u + \chi_{[0,\beta_u]})^{**}$ on $[0, \beta_u]$,the optimal value of Problem ($\mathbf{CP}$) is greater than optimal values of ($\mathbf{LP}_0$) but the last problem is simpler to solve.

5. Preliminary computational experiments and conclusions

The DCA described in Section 4 has been implemented and test on a series of tests problems, typical of telecommunication networks, and derived from [8, 9]). These authors have designed a generator of instances of the min cost multicommodity flow problem with step increasing link cost functions resembling to those encountered in real telecommunication network design applications.

5.1. DATA

We have considered two cases of a network $G = (V, U)$ where $V = \{s_1, ..., s_m\}$ and $U = \{u_1, ..., u_n\}$ with $m = 16$, $n = 25$ in the first case and $m = 42$ and $n = 66$ in the second one. The multicommodity flow circulating on these network is composed of K simultaneous simple flows.

In our test problems, some of the basic characteristics of real problems have been produced, in particular

- the number $q(u)$ of discontinuity points is a random integer in $[13, 15]$ (resp. $[3, 4]$) for the first case(resp. the second case);
- for each $u \in U$, β_u is, on average, equal to twice or three times a feasible flow $\overline{x}_u$;
- the $q(u)$ discontinuity points are randomly chosen in the range $[0, \beta_u]$;
- compute the cost γ_u^i associated with each discontinuity point v_u^i $(i = 0, ..., q(u))$ according to the formula

$$\gamma_u^{i+1} = f_u(v_u^{i+1}) = \max\{a(v_u^{i+1})^\tau d_u,\ f_u(v_u^i) + C\},\ i = 0, ..., q(u) - 1,$$

where d_u is the distance (in kilometer) of link u, τ is a coefficient chosen in the range $[0.7, 1]$, a is a random number chosen in the range $[E, F]$. The positive numbers C, E and F will be precised in each test problem. In case 1, the distances (in kilometers) between these nodes and the multicommodity flow requirements between each node pair are taken from [8, 9].

In all our test problems the points $\widetilde{v}_u^i$, $i = 0, ..., q(u) - 1$ (Subsection 4.1) are defined as follows: for $u \in U$

$$\widetilde{v}_u^i = v_u^i - \frac{v_u^i - v_u^{i-1}}{10}, i = 1, ..., q(u) - 1$$

and

$$\widetilde{v}_u^0 = v_u^0 + \min\{\frac{v_u^i - v_u^{i-1}}{10} : i = 1, ..., q(u) - 1\}.$$

The first case $(m = 16, n = 25)$ consists of three series of test problems corresponding to different values of τ, C, E and F. Here K=120.

1. Test problems **1** *('regular profile')* : $[E, F] = [0.9, 1]$, $C = 4$ and $\tau = 0.9$ (case 1.1), 1 (case 1.2). The step increasing link cost functions have regular height steps.

2. Test problems **2** *('irregular profile')* **:** $[E, F] = [0.9, 1.1]$, $C = 2$ and $\tau = 0.6$ (case 2.1) , 0.7 (case **2**.2), 0.8 (case 2.3), 0.9 (case 2.4), 1 (case 2.5). The step increasing link cost functions have very irregular height steps.

3. Test problems **3** *('concave profile')* **:** $[E, F] = [1, 1.1]$, $C = 2$ and $\tau = 0.6$ (case 3.1), 0.7 (case 3.2), 0.8 (3.3), 0.9 (case 3.4), 1 (case 3.5). The step increasing link cost functions may have a 'concave profile' that represents typically economies of scales. Such cost functions make the problem strongly combinatorial.

The second case ($m = 45$, $n = 62$) deals with $K = 113, 226, 452$.

5.2. COMPUTATIONAL RESULTS

In our first series of tests problems we have used, instead of the convexified problem (18), the linearized problems ($\mathbf{LP}_0$).

To solve linear programs $(\mathbf{LP})^{k+1}$ (computing the sequence $\{y^k\}$ generated by DCA, see 4.3.2) and the linearized problem ($\mathbf{LP}_0$) we have used CPLEX library subroutines.

For each example solved, Tables 1 and 2 show:

m: number of nodes in the network, n: number of links;

NV: number of x variables in $(\mathbf{LP})^k$;

NC: number of constraints in $(\mathbf{LP})^k$;

ND: average of number of discontinuity points of functions $f_u(x_u)$, $\forall u \in U$;

LB: lower bound obtained by solving the linearized problem ($\mathbf{LP}_0$), i.e., $l(\overline{x}) + L$ where $\overline{x}$ is the optimal solution to ($\mathbf{LP}_0$) and $L = \sum_{u_i n U} L_u$ is an additional value to compensate for a bad under-approximation of l_u, $u \in U$ in a neighborhood of $(\overline{x}_u)$;

P: number of iterations performed by DCA at the convergence;

$r = \frac{f(x^P) - LB}{f(x^P)}(\%)$.

We have started DCA with different choices of initial points x^0:

(i) $x^0 = 0$, in this case x^0 may not be feasible for ($\mathbf{P}_0$) and $F(x^0)$ could be less than $F(x^P)$. This fact does not evidently contradict the decade of the sequence $\{F(x^P)\}$. On the other hand x^k must be feasible for $k \geqslant 1$ and $F(x^1) > F(x^2) > \ldots > F(x^P)$.

(ii) x^0 is a feasible solution, here the optimal solution to the linearized problem ($\mathbf{LP}_0$).

(iii) $x^0 = (x_u^0)_{u \in U}$, where x_u^0 is randomly chosen in $[0, \beta_u]$ for each $u \in U$.

(iv) $x^0 = (x_u^0)_{u \in U}$, where $x_u^0 = \beta_u$ for each $u \in U$.

Note that in the last two cases x^0 may be infeasible.

Since the choice (i) is always better in all our test problems (P and $f(x^P)$ are the smallest compared with the other choices of x^0), we will present only computational results with this choice.

Table 1. The first case: $m = 16$, $n = 25$; $(13 \leqslant ND \leqslant 15)$

Test problems	$NV \times NC$	P	LB	$f(x^P)$	r
$K = 120$					
1.1	2281 × 6050	3	3089	3305	6.53
1.2	2281 × 6050	3	3089	3338	7.45
2.1	2283 × 6050	4	792	949	16.5
2.2	2283 × 6050	5	1239	1403	11.6
2.3	2283 × 6050	5	2060	2369	13
2.4	2283 × 6050	4	3344	3762	11
2.5	2283 × 6050	3	5417	5906	8.2
3.1	2283 × 6050	4	820	949	13.6
3.2	2283 × 6050	5	1294	1548	16.4
3.3	2283 × 6050	5	2112	2480	14.8
3.4	2283 × 6050	6	3400	3928	13.4
3.5	2283 × 6050	4	5797	6321	8.2

The quality of this choice can be explained by the fact: the link cost functions $f_u(x_u)$ being increasing, $x^0 = 0$ should be the unique solution to (P_0) if it was a feasible multicommodity flow.

We have not given here the running time of DCA. It corresponds exactly to that of solving P linear programs $(\mathbf{LP})^k : k = 1, .., P$. An efficient solution of these problems must take into account the following properties:

(i) the network block diagonal structure (17) of $(\mathbf{LP})^k$ allows exploiting sparsity,

(ii) post-optimization techniques may be used since Problems $(\mathbf{LP})^k$, $k = 1, \ldots, P$ differ only from the constraints $(_*)$.

5.3. CONCLUSIONS

We have presented a new approach based on d.c. programming and DCA to solve the multicommodity network optimization with step increasing cost functions. Preliminary computational experiments (Tables 1 and 2) are presented on a series of test problems with structures similar to those encountered in telecommunication networks. They show that the d.c. approach and DCA provide feasible multicommodity flows x^* such that $r := f(x^*) - LB/f(x^*)$ lies in the range [4.2 %, 16.5 %] with an average of 11.5 %. It is clear that r may be better estimated using better lower bounds LB. Meanwhile, to the best of our knowledge, that is the first time so good upper bound have been obtained. These results have been more or less expected since the d.c. approach have been successfully applied to many and various classes of nonconvex nondifferentiable optimization problems, especially for large scale settings where found local solutions are quite often global

Table 2. The second case: $m = 42$, $n = 66$; $(3 \leqslant ND \leqslant 4)$.

$K = 113$			
Profile	$NV \times NC$	P	r
'regular'	5047 × 15049	5	4.21
'irregular'	5047 × 15049	5	13
'concave'	5047 × 15049	5	15
$K = 226$			
Profile	$NV \times NC$	P	r
'regular'	9855 × 29964	5	5.86
'irregular'	9855 × 29964	6	13.5
'concave'	9855 × 29964	5	15.5
$K = 452$			
Profile	$NV \times NC$	P	r
'regular'	19347 × 79796	5	5.25
'irregular'	19347 × 79796	5	14
'concave'	19347 × 79796	6	16

ones. It is worth noting that DCA is quite simple and inexpensive, in particular for approximate polyhedral d.c. programs (P_{dc}) to (P_0) it has finite convergence: in these series of test problems the number of iterations performed by DCA at the convergence P varies between 3 and 6 with an average of 4.6. Our d.c. approach and DCA may then be applied to large scale multicommodity network optimization problems.

To confirm globality of solutions computed by DCA or to improve them in the negation we can combine DCA with branch and bound techniques. Our branching procedure consists simply of subdivisions of the box $\mathcal{D} = \Pi_{u \in U}[0, \beta_u]$ in smaller ones $\mathcal{D}_k$ while the bounding procedure solves the linear programs $(\mathbf{CP})_k$ obtained from $(\mathbf{CP})$ in replacing $\mathcal{D}$ with $\mathcal{D}_k$. The best current feasible multicommodity flow w generated by branch and bound techniques will be the new starting point for DCA if $f(w) < f(x^*)$ (escaping procedure) and so on. Such a combination may considerably reduce the differences of current upper bounds and lower bounds to $(\mathbf{P}_0)$ and may so allow globally solving large scale problems.

On the other hand we believe that our d.c. approach and DCA are efficient in the solution of the (CA), (FA) and (CFA) problems beside usual relaxation techniques (applied to large scale mixed integer programming problems like (P_2)) which are in general very expensive. These issues are currently under research.

References

1. K.T. Ahuja, T. Magnanti and J. Orlin. *Networks Flows: Theory, Algorithms and Applications.* Prentice Hall, 1993.
2. A.A. Assad. Multicommodity network flows-A survey. *Networks* 8: 37–91, 1978.
3. A. Balakrisnan and S.C. Graves. A composite algorithm for a concave-cost network flow problem. *Networks* 19: 175–202, 1989.
4. D.P. Bertsekas and R.G. Gallager. *Data Netwoks.* Prentice Hall, 1987.
5. F. Boyer. *Conception et Routage des Réseaux de Télécommunications.* Thèse de Doctorat de l'Université Blaise Pascal, Clermont -Ferrand, 1997.
6. V.J.M. Ferreira Filho and R.D. Galvao. A Survey of Computer Network Design Problems. *Investigacion Operativa* 4: 183–211, 1994.
7. L. Fratta, M. Gerla and L. Kleinrock. The flow deviation method: an approach to store-and-forward communications network design. *Networks* 3: 97–133, 1973.
8. V. Gabrel and M. Minoux. Large scale LP relaxations for minimum cost multicommodity flow problems with step increasing cost functions and computational results. *Technical Report, Laboratoire MASI, Univ. Paris 6,* 1996.
9. V. Gabrel and M. Minoux. LP relaxations better than convexification for multicommodity network optimization problems with step increasing cost functions. *Acta Mathematica Vietnamica* 22: 128–145, 1997.
10. B. Gavish. Augmented Lagrangian based bounds for centralized network design. *IEEE Transactions on Communications* COM-33, 1247–1257, 1985.
11. B. Gavish and K. Altinkemer. Backbone network design tools with economic tradeoffs *ORSA J. on Computing* 2/3: 236–252.
12. B. Gavish and G.W. Graves. System for routing and capacity assignment in computer communication network. *IEEE Transactions on Communications* COM-37: 360–366, 1989.
13. M. Gerla and L.Kleinrock. On the tological design of distributed computer networks. *IEEE Transactions on Communications* COM-25: 48–60, 1977.
14. M. Gerla *The Design of Store-and-Forward Networks for Computer Communications*. PhD Thesis, UCLA, 1973.
15. M. Gerla and L. Keinrock. On the topological design of distributed computer networks. *IEEE Transactions on Communications* COM-25: 48–60, 1977.
16. M. Gerla, R. Monteiro and R. Pazos, Topology design and bandwith allocation in ATM nets. *IEEE transactions on selected Area in Communications* SAC-7, 1989.
17. M. Gondran and M. Minoux. *Graphes et Algorithmes* Eyrolles, Paris 2nd edition 1995.
18. J.B. Hiriart Urruty and C. Lemarechal *Convex Analysis and Minimization Algorithms*. Springer, Berlin, 1993.
19. P.J. Laurent. *Approximation et Optimisation*. Hermann, Paris, 1972.
20. R. Horst, P.M. Pardalos and V.T. Nguyen. *Introduction to Global Optimization*, Kluwer, Dordrecht, 1995.
21. R. Horst and V.T. Nguyen. D.C. Programming: Overview. *Journal of Optimization Theory and Applications*, 103: 1–43, 1999.
22. J.L. Kennington. A survey of linear cost multicommodity network flows. *Operations Research* 26: 209–236, 1978.
23. H.A. Le Thi. *Contribution à l'optimisation non convexe et l'optimisation globale: Théorie, Algorithmes et Applications* . Habilitation à Diriger des Recherches, Université de Rouen, 1997.
24. H.A. Le Thi and T. Pham Dinh. Solving a class of linearly constrained indefinite quadratic problems by D.c. algorithms. *Journal of Global Optimization* 11: 253–285, 1997.
25. H.A. Le Thi and T. Pham Dinh. D.c. models of real world nonconvex optimization problems. *Technical Report,* LMI, INSA-Rouen, 1998.

26. H.A. Le Thi and T. Pham Dinh. D.c. programming approach for large-scale molecular optimization via the general distance geometry problem, in *Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches,* C.A. Floudas and P.M. Pardalos (Eds.), pp. 301–339, Kluwer Academic Publishers, Dordrecht, 2000.
27. H.A. Le Thi and T. Pham Dinh, A continuous approach for large-scale linearly constrained quadratic zero-one programming. *Optimization*, 45(3) 1–28, 2001.
28. J. Mac Gregor Smith and P. Winter (eds). Topological NetWork Design. *Annals of Operations Research* 33, 1991.
29. P. Mahey and T. Pham Dinh. Proximal decomposition on the graph of a maximal monotone operator. *SIAM Journal on Optimization,* 5: 454–468, 1995.
30. T.Q. Nguyen. *Une approche D.c. en Optimisation dans les Réseaux. Algorithmes, Codes et Simulations Numériques.* PhD Thesis, Université de Rouen, 1999.
31. P. Mahey, A. Ourou, L. Leblanc and J. Chifflet. A new proximal decomposition algorithm for routing in telecommunications networks. *Networks* 31: 227–338, 1998.
32. P. Mahey and H.P.L. Luna. Bounds for Global Optimization of Capacity Expansion and Flow Assignment Problems. *to appear in Operations Research Letters.*
33. P. Mahey and H.P.L. Luna. Separable convexification techniques for capacity and flow assignment problems. *To appear.*
34. V.T. Nguyen. D.C. Programming. *in Encyclopedia of Optimization* Kluwer Academic Publishers, Dordrecht.
35. M. Minoux. Network synthesis and Optimum Network design problems: Models, Solution Methods and Applications. *Networks* 19: 313-360, 1989.
36. A. Ouorou. *Décomposition proximale des problèmes de multiflots à critère convexe-Applications aux problèmes de routage dans les réseaux de communications*. Thèse de Doctorat, Université de Clermont-Ferrand, 1995.
37. T. Pham Dinh and H.A. Le Thi Convex analysis approach to d.c. programming: Theory, Algorithms and Applications (dedicated to Professor Hoang Tuy on the occasion of his 70th birthday). *Acta Mathematica Vietnamica.* 22(1): 289–355.
38. T. Pham Dinh and H.A. Le Thi. D.c. optimization algorithms for trust region problem. *SIAM J. Optimization*, 8(2): 476–505.
39. R.T. Rockafellar. *Convex Analysis*. Princeton University, Princeton 1970.
40. R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control and Optimization*, 14(5): 877–898.
41. B. Sanso, F. Soumis and M. Gendreau. On the evaluation of telecommunication networks reliability using routing models. *IEEE Transactions on Communications* COM-3, 1494-1501.
42. J.F. Toland. On subdifferential calculus and duality in nonconvex optimization. *Bull. Soc. Math. France Mémoire* 60: 173–180.