

# Cycle-Based Algorithms for Multicommodity Network Flow Problems with Separable Piecewise Convex Costs

**Mauricio C. de Souza**

*Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais,  
Rua Espirito Santo 35, cep: 30160 - 030, Belo Horizonte, MG, Brasil*

**Philippe Mahey**

*Laboratoire LIMOS, CNRS-UMR6158 and Université Blaise Pascal, Clermont-Ferrand, France*

**Bernard Gendron**

*Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128,  
Montréal, Québec, Canada*

**We present cycle-based algorithmic approaches to find local minima of a nonconvex and nonsmooth model for capacity expansion of a network supporting multicommodity flows. By exploiting complete optimality conditions for local minima, we give the convergence analysis of the negative-cost cycle canceling method. The cycle canceling method is embedded in a tabu search strategy to explore the solution space beyond the first local optimum. Reaching a local optimum, the idea is to accept a cost-increasing solution by pushing flow around a positive-cost cycle, and then to make use of the cycle cancelling method incorporating tabu search memory structures to find high quality local optima. Computational experiments on instances of the literature show that the tabu search algorithm can significantly improve feasible solutions obtained by the local optimization procedure, and it outperforms the capacity and flow assignment heuristic in terms of solution quality. © 2007 Wiley Periodicals, Inc. NETWORKS, Vol. 51(2), 133–141 2008**

**Keywords:** multicommodity flow problems; cycle canceling algorithms; tabu search.

## 1. INTRODUCTION

In this study, we consider a multicommodity network flow problem with piecewise convex arc cost functions, introduced by Luna and Mahey [15] to cope with capacity expansion problems in the optimal design of communications networks.

The problem is an extension of the classical flow assignment problem. Given a directed capacitated network with a traffic requirement matrix between some origin-destination pairs, the flow assignment problem consists of finding a feasible routing, which minimizes the average delay induced by the waiting queues at the switching nodes of the network. This delay is generally approximated by a separable smooth convex objective function (see [9] for example). The resulting convex cost multicommodity flow problem can be solved by several efficient algorithms [21]. In the problem we consider, we wish to decide which arc capacities should be expanded to further reduce the average delay and improve the so-called Quality of Service (QoS) of the resulting network. As expanding an arc induces a fixed cost, the problem results in finding a trade-off between investment and routing costs. Luna and Mahey [15] showed that the corresponding multicommodity flow problem can be modelled by using continuous piecewise convex cost functions on each arc, resulting in a nonconvex, nonsmooth, objective function. The problem can also be modelled as a mixed-integer nonlinear program [18], and can be seen as a particular case of the capacity and flow assignment problem (CFA), introduced by Gerla and Kleinrock [9]. To solve this problem, researchers have either tried to decouple the difficulty by alternately solving the capacity and the flow assignment problems (the so-called CA-FA procedure [6, 7, 10]), or used exact approaches like Benders decomposition [16].

In this study, we propose an efficient heuristic method based on the iterative solution of convex cost multicommodity flow subproblems, embedded in a tabu search framework. The method is based on the classical cycle canceling approach for network flows [14]. Ouorou and Mahey [20] derived

Received June 2005; accepted March 2007

Correspondence to: P. Mahey; e-mail: philippe.mahey@isima.fr

Contract grant sponsor: CNPQ grant; Contract grant number: 301603/2006-5

DOI 10.1002/net.20208

Published online 14 December 2007 in Wiley InterScience (www.interscience.wiley.com).

© 2007 Wiley Periodicals, Inc.

an efficient algorithm for smooth convex multicommodity flow problems by canceling minimal mean cycles. They have shown however that the cycle canceling method does not apply to nonsmooth convex cases. For the problem we consider, however, Mahey and Souza [19] proved that the absence of negative-cost cycles is necessary and sufficient for local optimality under the assumption that at breakpoints the marginal cost decreases, as observed when capacity expansion occurs. By exploiting this result, we give in the present work a convergence analysis of the negative-cost cycle canceling method. We also propose a tabu search strategy to explore the solution space beyond the first local optimum found by the negative-cost cycle canceling method. The idea is to accept a cost increasing solution pushing flow around a positive-cost cycle, using a tabu short-term memory in an attempt to find a better local optimum.

Tabu search methods have been proposed for another difficult, nonconvex, multicommodity network flow problem, the fixed-charge network design problem, in which the linear cost function on each arc is discontinuous, having a unique breakpoint at the origin. In Crainic et al. [4], simplex pivot-type moves and column generation are combined in a tabu search framework to explore the space of continuous flow variables, while evaluating the exact objective function. Ghamlouche et al. [11] proposed a different tabu search algorithm, where each new solution is obtained by pushing flows along cycles. As we are dealing with a nonlinear continuous cost function on each arc, the neighborhood definition and the strategy to guide the search differentiate our work from these other related contributions.

This study is organized as follows. In the next section, we present the problem and the negative-cost cycle free local optimality condition. Section 3 is dedicated to the negative-cost cycle canceling method and its convergence analysis. In Section 4, we propose a tabu search algorithm to explore the solution space beyond the first local optimum. We present computational experiments in Section 5. Concluding remarks and extensions are discussed in the last section.

## 2. PROBLEM DESCRIPTION

The underlying network topology is represented by a digraph  $\mathcal{G} = (V, A)$  with  $m$  nodes and  $n$  arcs. Traffic between any pair of nodes is treated as a separate commodity  $k$  with an origin-destination pair  $(s_k, t_k)$  and a demand requirement  $r_k \geq 0$ . We have then  $K$  flow vectors  $x_1, \dots, x_k, \dots, x_K$  of  $\mathbb{R}^n$  associated with the  $K$  commodities to be transported through  $\mathcal{G}$ . The total flow on arc  $j$  is denoted by  $x_j$ . The flow on arc  $j$  associated with commodity  $k$  is denoted  $x_{kj}$ . Let  $M$  be the  $m \times n$  node-arc incidence matrix of  $\mathcal{G}$ , and let  $b_k \in \mathbb{R}^m$  be the vector with all components 0 except  $b_k(s_k) = -b_k(t_k) = 1$ .

As we are concerned with capacity expansion, we assume each arc  $j \in A$  has an installed capacity  $c_{0j}$  and is expandable to a capacity  $c_{1j} > c_{0j}$  at a given fixed cost  $\pi_j$ . Luna

and Mahey [15] proposed the following continuous model, denoted in the sequel by (CCE):

$$\min \sum_{j=1}^n f_j(x_j) = \min \{ \Phi(c_{0j}, x_j), \Phi(c_{1j}, x_j) + \pi_j \} \quad (1)$$

$$\text{s.t. } \sum_{k=1}^K x_{kj} = x_j, \quad \forall j \in A, \quad (2)$$

$$Mx_k = r_k b_k, \quad k = 1, \dots, K, \quad (3)$$

$$x_j \leq c_{1j}, \quad \forall j \in A, \quad (4)$$

$$x_j, x_{kj} \geq 0, \quad \forall j \in A, k = 1, \dots, K. \quad (5)$$

In this formulation, the individual arc congestion functions  $\Phi : C \times \mathbb{R} \rightarrow \mathbb{R}$ , where  $C$  is a finite set of positive capacities, satisfy the following properties:

1.  $\Phi(c, \cdot)$  is strictly convex, monotone increasing on  $[0, c)$ ;
2.  $\Phi(c, \cdot)$  is continuously derivable on  $(0, c)$  and  $\partial \Phi(c_1, x)/\partial x < \partial \Phi(c_0, x)/\partial x$  for any  $0 < x < c_0 < c_1$ ;  $\partial \Phi^+(c_1, 0)/\partial x < \partial \Phi^+(c_0, 0)/\partial x$ , where  $\partial \Phi^+$  is the right partial derivative;
3.  $\Phi(c, 0) = 0$ ;
4.  $\Phi(c, x) \rightarrow +\infty$  if  $x \uparrow c$ .

The last property means that  $\Phi$  acts as a barrier that prevents the solution from approaching the capacity when minimizing (1). A well-known example of such a congestion function is Kleinrock's delay function  $\Phi(c, x) = \frac{x}{c-x}$ , which expresses the average delay of traffic  $x$  on an arc with capacity  $c$  assuming a Poisson process [3].

The objective function (1) of CCE is continuous but nonconvex and nonsmooth at the breakpoint  $\gamma_j c_{0j}$ ,  $0 < \gamma_j < 1$ . The investment fixed cost  $\pi_j$  and the threshold  $\gamma_j c_{0j}$  at which expansion occurs are related by  $\pi_j = \Phi(c_{0j}, \gamma_j c_{0j}) - \Phi(c_{1j}, \gamma_j c_{0j})$ . Constraints (2)–(5) define the classical node-arc formulation of multicommodity network flow problems [8, 21]. Figure 1 shows by a solid line the nonconvex resulting arc cost function of CCE. As can be observed in the figure,  $\gamma_j c_{0j}$  is the breakpoint beyond which it becomes advantageous to pay for expansion.

Let  $S$  be the set of feasible solutions defined by constraints (2)–(5). CCE is a piecewise convex multicommodity flow model since the objective function is convex on any subset of  $S$  where the flow on each arc  $j$  is restricted either to  $[0, \gamma_j c_{0j}]$  or to  $[\gamma_j c_{0j}, c_{1j}]$ . The consequence is that there exists at most one local optimum in any subset of  $S$  where the flow on each arc  $j$  is restricted either to  $[0, \gamma_j c_{0j}]$  or to  $[\gamma_j c_{0j}, c_{1j}]$ . This is the main motivation behind the development of the present work.

### 2.1. Neighborhood Definition

For any feasible solution  $x^*$  of CCE, we define:

$$A_0 = \{j \in A | x_j^* \in [0, \gamma_j c_{0j})\}$$

$$A_1 = \{j \in A | x_j^* \in (\gamma_j c_{0j}, c_{1j})\}$$

$$A_{01} = \{j \in A | x_j^* = \gamma_j c_{0j}\}.$$

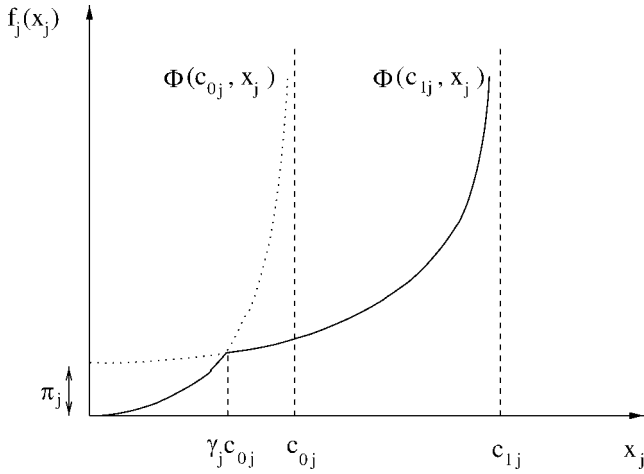


FIG. 1. The integrated function of congestion and expansion costs.

By letting  $g = |A_{01}|$ , we can define  $2^g$  subsets of the feasible set  $S$  by constraining  $x_j$  to  $[0, \gamma_j c_{0j}]$  (resp.  $x_j$  to  $[\gamma_j c_{0j}, c_{1j}]$ ) for  $j \in A_0$  (resp.  $j \in A_1$ ) and by alternately constraining  $x_j$  to  $[0, \gamma_j c_{0j}]$  or to  $[\gamma_j c_{0j}, c_{1j}]$  for  $j \in A_{01}$ . The union of these subsets defines the *neighborhood*  $N(x^*) \subseteq S$  of  $x^*$ . Note that  $x^*$  belongs to each of the  $2^g$  subsets, and that these subsets have disjoint interior points.

### 2.2. Local Optimality Conditions

Mahey and Souza [19] established that the absence of negative-cost cycles is a necessary and sufficient condition for local optimality of CCE. Let  $x$  be a feasible solution of CCE. Given a cycle  $\Theta$  and a sense of circulation, we will call  $\Theta$   $k$ -feasible if it presents a strictly positive residual, i.e., if we can augment the commodity flow  $x_{kj}$  on the forward arcs (denoted by  $\Theta^+$ ) and reduce it on the backward arcs (denoted by  $\Theta^-$ ). Its cost is given by

$$\lambda(x, \Theta) = \sum_{j \in \Theta^+} f_j^+(x_j) - \sum_{j \in \Theta^-} f_j^-(x_j)$$

where  $f_j^+(x_j)$  (resp.  $f_j^-(x_j)$ ) is the right (resp. left) partial derivative of the arc cost function  $f_j$ . Then we have the following result.

**Local optimality conditions [19]:** A feasible solution  $x^*$  is a local optimum of CCE if and only if, for all commodities  $k = 1, \dots, K$ , there does not exist any  $k$ -feasible cycle with negative cost.

We remark that this result cannot be extended to the convex nonsmooth case, as already observed in [20]. It works for the piecewise convex model CCE because, at the breakpoints,  $f_j^+(\gamma_j c_{0j}) < f_j^-(\gamma_j c_{0j})$ . This is the key characteristic distinguishing expansion problems, where the user is willing to pay a fixed expansion cost for a smaller marginal operational cost.

## 3. CYCLE CANCELING ALGORITHM

In this section, we describe the cycle canceling algorithm (CCA), which is derived from the above local optimality conditions. As in most cycle canceling algorithms, each iteration of CCA is divided in two steps: find a negative-cost cycle, and cancel the cycle by pushing flow on it. Two potential difficulties must be addressed:

- The multicommodity flow structure: This factor does not influence the procedure as each commodity is treated one at a time, i.e., the marginal costs are computed with respect to the total arc flow, but only the flow of one chosen commodity is updated.
- The nonlinear and nonsmooth cost function: Left and right derivatives of the arc cost function play the role of the arc costs, as we show how to compute an approximate local minimum of the one-dimensional cost function associated with the cycle canceling step.

### 3.1. Cycle-Finding Step

As the computation of the most negative-cost cycle is NP-hard, we follow an old idea of Weintraub [24] further improved by Barahona and Tardos [2], where a sequence of assignment subproblems is used to find the largest possible improvement in the objective function by canceling a family of node-disjoint cycles. Indeed, a feasible assignment of the network nodes to themselves corresponds to a set of node-disjoint cycles (eventually including self-loops) and computing the assignment with minimal cost can be done in polynomial time. Let  $A_k$  be the set of arcs that carry some positive flow of commodity  $k$ . To obtain the most negative set of disjoint  $k$ -feasible cycles, we can set the cost  $d_{iw}^k$  of assigning node  $i$  to node  $w$  to

$$d_{iw}^k = \begin{cases} f_j^+(x_j), & \text{if } j = (i, w) \in A_k, \\ -f_j^-(x_j), & \text{if } j = (w, i) \in A_k, \\ 0 & \text{if } i = w, \\ +\infty & \text{otherwise,} \end{cases} \quad (6)$$

where  $x$  is the current feasible multicommodity flow. Since the loops  $i \mapsto i$  do not correspond to cycles and have cost zero, either there exists an assignment with negative cost or there is no negative-cost  $k$ -feasible cycle. Observe too that the total cost of the  $k$ -feasible disjoint cycles is at least as negative as the cost of the most negative  $k$ -feasible cycle (the one which is not polynomially computable). But it is not the steepest descent direction as that one could involve several nondisjoint cycles. Finally, as the cycles obtained through the assignment subproblem are node-disjoint, we can cancel them sequentially in any order.

### 3.2. Cycle-Canceling Step

The  $k$ -canceling step augments (respectively, reduces) the  $k$ -th commodity flow on the forward (respectively, backward) arcs of the cycle  $\Theta$  so that the function  $\phi_\Theta(\alpha) = \sum_{j \in \Theta^+} f_j(x_j + \alpha) + \sum_{j \in \Theta^-} f_j(x_j - \alpha)$  is minimized. But the step size  $\alpha$  can

be smaller if some backward arc flow reaches its lower bound, i.e.,  $x_j^k - \alpha = 0$ . The line search will then be further restricted to keep the step within a subregion where the function  $\phi_\theta$  is convex, i.e., a subregion as defined in Section 2 where the flow on each arc in  $A_{01} \cap \Theta^-$  is restricted to  $[0, \gamma_j c_{0j}]$ , and the flow on each arc in  $A_{01} \cap \Theta^+$  is restricted to  $[\gamma_j c_{0j}, c_{1j}]$  (if there are other arcs in  $A_{01}$  which are not in the cycle, it is indifferent to restrict their flow either to  $[0, \gamma_j c_{0j}]$  or to  $[\gamma_j c_{0j}, c_{1j}]$ ). Let  $\bar{\alpha}$  be the largest allowed value for the step size, i.e.,  $\bar{\alpha} = \min_{j \in \Theta} \alpha_j$  where, for each  $j$ ,  $\alpha_j$  is computed as follows

$$\alpha_j = \begin{cases} \gamma_j c_{0j} - x_j, & \text{if } j \in \Theta^+ \text{ and } x_j < \gamma_j c_{0j} \\ \min \{x_j^k, x_j - \gamma_j c_{0j}\}, & \text{if } j \in \Theta^- \text{ and } x_j > \gamma_j c_{0j} \\ x_j^k, & \text{if } j \in \Theta^- \text{ and } x_j \leq \gamma_j c_{0j} \end{cases}$$

The canceling step can be computed by an efficient line search on the interval  $[0, \bar{\alpha}]$ , i.e., for instance, by a bisection strategy, testing the sign of  $\phi'_\theta(\alpha)$  at the midpoint of the search interval and reducing the size of the interval by a factor 2 until a given tolerance is attained [5]. Let  $\alpha^*$  be the optimal value of the step; then we can update the flow of the  $k$ -th commodity as follows:

$$x^k := x^k + \alpha^* \theta, \quad (7)$$

where  $\theta$  in the update formula (7) denotes the incidence vector of cycle  $\Theta$ .

We can observe that, in many cases, a larger step can be performed when one reaches the breakpoint value of some arc flow in the cycle. Indeed, suppose that  $x_j < \gamma_j c_{0j}$  and that the flow augments until  $x_j + \alpha = \gamma_j c_{0j}$  with  $\lambda(x + \alpha, \Theta)$  still negative. Then, as  $f_j^-(x_j + \alpha) > f_j^+(x_j + \alpha)$ , we can still augment the flow in the interval  $[\gamma_j c_{0j}, c_{1j}]$  corresponding to the adjacent subregion. The same situation is obtained for arcs where the breakpoint value is crossed from above ( $x_j > \gamma_j c_{0j}$  and  $j$  is a reverse arc in the cycle). As a consequence, the one-dimensional search on the negative cycle can be directly performed on the whole intervals  $[0, c_{1j}]$ , even if the function is nonconvex and nonsmooth.

### 3.3. Convergence Analysis

We now analyze the convergence of CCA using basic properties of descent algorithms. Observe that a canceling-step starts with a set  $\Omega$  of  $k$ -feasible disjoint cycles (for a fixed commodity  $k$ ) with  $|\Omega| = L$ . As the cycles are disjoint, the search direction is the indicator vector of the  $L$  disjoint cycles, i.e.,

$$[\theta_\Omega]_e = \begin{cases} +1, & \text{if } e \in \Theta_l^+, \text{ for some } l \\ -1, & \text{if } e \in \Theta_l^-, \text{ for some } l \\ 0, & \text{if } e \notin \Theta_l, \forall l \end{cases}$$

It is thus a feasible circulation for the  $k$ -th commodity. The single modification with respect to what was stated in the previous section is that we will use the same step size for all cycles in the bundle of disjoint cycles found by the cycle-finding procedure of Section 3.1. This is used here only for

the sake of simplification of the convergence analysis. As the cycles are disjoint, the canceling step will still be better if each cycle is canceled separately (thus with different optimal step sizes).

At a given feasible multicommodity flow  $x$ , let  $\phi(\alpha) = f(x + \alpha \theta_\Omega)$ , where  $\alpha$  is the step size applied to all cycles in the bundle  $\Omega$ , be the one-dimensional function associated with the search direction. Observe that  $\phi(\alpha) = \sum_{j \notin \Omega} f_j(x_j) + \sum_{l=1}^L \phi_{\theta_l}(\alpha)$ , using the individual cycle functions  $\phi_{\theta_l}$  defined in Section 3.2.

In the following lemma, the steepest descent direction will be denoted abusively by  $-\nabla f(x)$ . Observe that it is defined here with respect to a single commodity, all other commodity flows being kept fixed. Its cost is thus equal to  $-\|\nabla f(x)\|^2$ . We first establish that  $\theta_\Omega$  is a sufficient descent direction in the following sense:

**Lemma 1.** *The search direction associated with the optimal set of disjoint cycles is a sufficient descent direction in the sense that  $\phi'(0) \leq -\tau_0 \|\nabla f(x)\|^2$ , where  $0 < \tau_0 < 1$ .*

**Proof.** The steepest descent direction corresponds to the most negative cost of any circulation added to  $x$ . It is well known [1] that any circulation can be supported by at most  $m - n + 1$  feasible cycles. On the other hand, the cost of an optimal set of disjoint cycles is at least as negative as the cost of the most negative single cycle denoted by  $\bar{\Theta}$ . Then,  $\phi'(0) \leq \lambda(x, \bar{\Theta})$ . Completing the overestimation (for negative numbers), we have finally:

$$-\|\nabla f(x)\|^2 \leq \phi'(0) \leq \lambda(x, \bar{\Theta}) \leq \frac{-\|\nabla f(x)\|^2}{m - n + 1}$$

The result is thus proved by setting  $\tau_0 = \frac{1}{m - n + 1}$ . ■

**Definition 1.** *The step size  $\alpha$ , in other words the value of the circulation which is pushed on every cycle  $\Theta_l$ , is called admissible if it satisfies the following Armijo condition:*

$$f(x + \alpha \theta_\Omega) - f(x) \leq \tau_1 \alpha \phi'(0) \quad (8)$$

where  $\tau_1$  is a positive number less than  $1/2$ .

To understand how the individual bound constraints associated with the current subregion do not perturb the choice of an admissible step, we can observe that, when an arc flow reaches one of its bounds, the cycle is canceled and the function has strictly decreased (indeed, this point only concerns the case of a decreasing arc flow reaching the value zero, as we have seen that the breakpoint value can never be a stopping value). Thus, these cycle canceling steps do not affect the convergence of the Armijo iterations, as either there are infinitely many Armijo steps or there are no more negative-cost cycles. A practical way to estimate a good step size is to test condition (8) for  $\alpha = 1, 1/2, 1/4, \dots$  and take the largest value bounded by the intervals  $[0, c_{1j}]$ .

**Theorem 1.** *Suppose there exists a strictly feasible multicommodity solution to the problem with capacities  $c_{1j}$  for*

all  $j \in A$ . Then the sequence generated by algorithm CCA with feasible step sizes converges to a point which satisfies the local optimality conditions of CCE.

**Proof.** The objective function is continuously differentiable on the whole intervals. Then, as the direction is sufficiently decreasing by Lemma 1 and the Armijo condition is always satisfied when the step is not limited to the interval bounds, it is a well-known result (see for instance [5]) that the method will converge and each limit point is such that the gradient of  $f$  is zero or, equivalently, there are no negative-cost  $k$ -feasible cycles for all commodities. ■

Observe that, in the original paper by Weintraub [24], many assignment subproblems are solved at each step to approximate the most helpful cycle, in the sense of minimizing the decrease of the objective function *after* the flow update. This choice was exploited later by Barahona and Tardos [2] to obtain a polynomial algorithm in the linear case. Our choice is different as it relies on the idea of an approximation of the steepest-descent direction.

#### 4. TABU SEARCH METHOD

A local optimum of CCE can be obtained by applying CCA to a feasible solution. We propose a tabu search algorithm [12, 13] to explore the solution space beyond the first local optimum. For this purpose, we exploit the piecewise convex nature of the objective function. If we consider a local optimum  $x^*$  and its neighborhood  $N(x^*)$ , as defined in Section 2.1, we note that it is useless to perturb  $x^*$  towards some other feasible solution  $x'$  in the interior of  $N(x^*)$ , since CCA applied to  $x'$  would lead back to  $x^*$  in the descent step. Therefore, we need to drive the search not to any other feasible solution, but rather to a feasible solution in the neighborhood boundary of the current local optimum.

Each iteration of our tabu search algorithm consists of two distinct phases. In the first phase, the search is driven from a local optimum  $x^*$  to a feasible solution  $x'$  on the boundary of  $N(x^*)$ . To achieve that, we bring the flow on an arc  $j$  to its breakpoint,  $\gamma_j c_{0j}$ , by pushing flow around positive-cost cycles. During this phase, we accept a cost increasing solution. In the second phase, CCA is applied to  $x'$ . By using a short-term memory structure, we ensure that CCA, when applied to  $x'$ , leads the descent to a local optimum  $x^{*'}$  such that  $N(x^{*'})$  and  $N(x^*)$  have disjoint interior points. We also make use of a medium-term memory to guide the search in the first phase of an iteration. We now present how we perform the first phase of each iteration of the tabu search method, and then describe the memory structures used by the algorithm.

##### 4.1. Driving the Search to the Current Neighborhood Boundary

To begin each iteration, we first have to choose an arc  $j'$  such that  $x_j^* \neq \gamma_j c_{0j}$ . Then we must select a commodity  $k$  for which we modify the routing. Finally, we have to determine

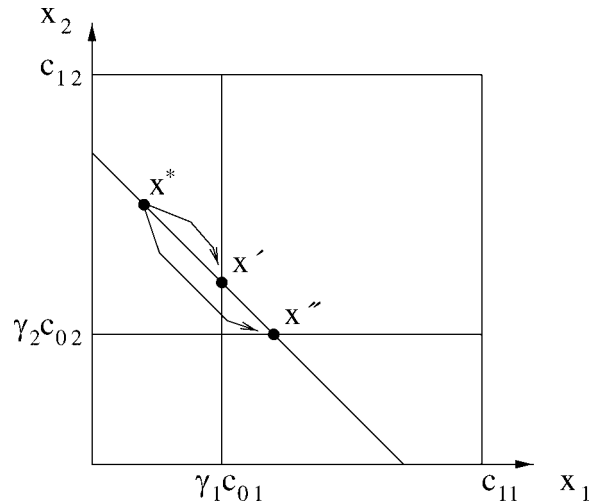


FIG. 2. Arc choices in order to intensify the search.

a  $k$ -feasible cycle along which we push flow until the value of  $x_j^*$  becomes  $\gamma_j c_{0j}$ .

The choice of arc  $j'$  reflects whether one intends to intensify or to diversify the search through the solution space. We are interested primarily in intensifying the search. If we denote by  $\text{int}N(x^*)$  the interior of  $N(x^*)$ , we choose arc  $j'$  to drive the search to a point  $x'$  in the boundary of  $N(x^*)$ , so that  $\text{int}N(x^*) \cap \text{int}N(x') \neq \emptyset$ . This implies that, by pushing flow along positive-cost cycles, we should not allow the flow on any arc to cross its breakpoint. Thus, we should maintain the condition that, for  $j = 1, \dots, n$ ,  $x_j' \leq \gamma_j c_{0j}$  (respectively,  $x_j' \geq \gamma_j c_{0j}$ ) if and only if  $x_j^* \leq \gamma_j c_{0j}$  (respectively,  $x_j^* \geq \gamma_j c_{0j}$ ).

Figure 2 shows a 2-dimensional example. Let us consider a graph with two nodes, an origin-destination pair of a commodity with demand  $r$ , and two different arcs, 1 and 2, both with the same tail and head. The feasible solution set is then  $S = \{(x_1, x_2) \in [0, c_{11}] \times [0, c_{12}] | x_1 + x_2 = r\}$ . Suppose that  $x^*$  is a local optimum of such an instance, with  $N(x^*) = C_1$  where  $C_1 = \{x \in S | x_1 \in [0, \gamma_1 c_{01}] \text{ and } x_2 \in [\gamma_2 c_{02}, c_{12}]\}$ . We intensify the search if we choose arc  $j' = 1$  to bring the flow to  $\gamma_1 c_{01}$ . This choice leads to  $x'$ , with  $N(x') = C_1 \cup C_2$  where  $C_2 = \{x \in S | x_1 \in [\gamma_1 c_{01}, c_{11}] \text{ and } x_2 \in [\gamma_2 c_{02}, c_{12}]\}$ , and indeed  $\text{int}N(x^*) \cap \text{int}N(x') \neq \emptyset$ . Otherwise, the choice of arc  $j' = 2$  leads to  $x''$ , with  $N(x'') = C_2 \cup C_3$  where  $C_3 = \{x \in S | x_1 \in [\gamma_1 c_{01}, c_{11}] \text{ and } x_2 \in [0, \gamma_2 c_{02}]\}$ , and we do not intensify the search since  $\text{int}N(x^*) \cap \text{int}N(x'') = \emptyset$  (note that the flow on arc  $j = 1$  has crossed its breakpoint).

The natural choice is to bring the flow on arc  $j^1 = \arg \min_{j \in A} \{|x_j^* - \gamma_j c_{0j}| | x_j^* \neq \gamma_j c_{0j}\}$  to  $\gamma_j c_{0j}$ . Since it is not known beforehand if actually there exists a feasible solution in which  $x_{j^1} = \gamma_{j^1} c_{0j^1}$ , we perform a trial-and-error procedure in an attempt to obtain such a feasible solution. The trial-and-error procedure described below is applied to the sequence of arcs such that  $x_j^* \neq \gamma_j c_{0j}$ , sorted in nondecreasing order of  $|x_j^* - \gamma_j c_{0j}|$ , until either the flow on an arc has been brought to its breakpoint, or the arc list has been traversed without success. The algorithm is stopped in the latter case. We describe this procedure in the following subsection.

**4.1.1. Trial-and-Error Procedure.** Suppose we are given a local optimum  $x^*$  and an arc  $j$  such that  $x_j^* > \gamma_j c_{0j}$ . We bring to  $\gamma_j c_{0j}$  the flow on  $j$  if we find both (i) a commodity  $k$  such that  $x_{kj}^* \geq |x_j^* - \gamma_j c_{0j}|$ , and (ii) a  $k$ -feasible cycle  $\Theta$  with  $j \in \Theta^-$  that supports pushing an amount of  $|x_j^* - \gamma_j c_{0j}|$  units of flow on it. This yields the following trial-and-error procedure, which iterates by pushing flow on  $k$ -feasible cycles in which  $j$  is backward until  $x_j^*$  decreases to the value  $\gamma_j c_{0j}$ . The case  $x_j^* < \gamma_j c_{0j}$  is handled in the same way, except that  $j$  must be a forward arc in  $\Theta$ .

1. Let  $K'$  be the set of candidate commodities. Set  $\sigma \leftarrow |x_j^* - \gamma_j c_{0j}|$ .
2. If  $K' = \emptyset$ , stop – there is no feasible solution with  $x_j = \gamma_j c_{0j}$ . Otherwise, choose  $k \in K'$  and set  $K' \leftarrow K' - \{k\}$ .
3. While  $\sigma > 0$  and there exists a  $k$ -feasible cycle  $\Theta$  in which  $j \in \Theta^-$  do:
  - Push along  $\Theta$  the largest amount of flow  $\alpha \in (0, \sigma]$  that  $\Theta$  supports.
  - Set  $\sigma \leftarrow \sigma - \alpha$ .
4. If  $\sigma > 0$ , return to Step 2. Otherwise, stop—a feasible solution with  $x_j = \gamma_j c_{0j}$  is obtained.

The set  $K'$ , in Step 1, contains the commodities for which a  $k$ -feasible cycle having  $j$  as one of its arcs might be found. In case  $x_j^* > \gamma_j c_{0j}$ , the procedure starts with  $K'$  being the set of commodities such that  $x_{kj}^* > 0$ . In Step 2, we choose a commodity  $k$  in nonincreasing order of  $x_{kj}^*$ , in an effort to perform a single iteration of the trial-and-error procedure. In case  $x_j^* < \gamma_j c_{0j}$ , the procedure starts with  $K'$  being the set of commodities such that  $x_{kj}^* < r_k$ . Note, however, that a commodity currently routed through arc  $j$  can become even more attractive for this commodity, since the marginal cost for augmenting flow on  $j$  decreases when reaching  $\gamma_j c_{0j}$ . It can as well become attractive to route through  $j$  a commodity that passes by an extremity of arc  $j$ . For these reasons, we choose first, in Step 2, commodities such that  $0 < x_{kj}^* \leq r_k - |x_j^* - \gamma_j c_{0j}|$  or  $x_{kj}^* > 0$  for an arc  $j'$  in the adjacency list of an extremity of  $j$ . Step 3 of the trial-and-error procedure needs a routine that either finds a  $k$ -feasible cycle containing arc  $j$ , or returns the information that such a cycle does not exist. We describe this procedure in the next subsection.

**4.1.2. Finding Positive-Cost  $k$ -Feasible Cycles.** Our procedure to find a positive-cost  $k$ -feasible cycle containing a chosen arc  $j$  is based on breadth-first search. We make use, for each node, of two predecessor indices,  $p_b$  and  $p_f$ , in order to trace respectively the backward and forward arcs of the  $k$ -feasible cycle. We use a boolean variable, *cycle\_found*, that indicates if there exists, or not, a  $k$ -feasible cycle. Let  $u$  and  $v$  designate respectively the tail and head of arc  $j$ . Consider for instance the case  $x_j^* > \gamma_j c_{0j}$  (the case  $x_j^* < \gamma_j c_{0j}$  is slightly different). Step 3 of the trial-and-error procedure calls the following routine:

1. Set  $p_b(i) \leftarrow 0$  and  $p_f(i) \leftarrow 0$  for all  $i \in V$ . Set  $q \leftarrow v$  and  $p_b(q) \leftarrow u$ .
2. While  $q \neq t_k$  do:

Scan the adjacency list  $Adj_q$  of node  $q$ .  
 Let  $l$  be the arc in  $Adj_q$  carrying the largest amount of flow of commodity  $k$ .  
 Let  $w$  designate the head of arc  $l$ . Set  $p_b(w) \leftarrow q$  and  $q \leftarrow w$ .

3. Perform a reverse breadth-first search from  $u$  considering only arcs carrying a strictly positive amount of flow of commodity  $k$ . Label the nodes visited ( $u$  included), and let  $L$  be the set of such nodes.
4. Set *cycle\_found*  $\leftarrow$  false.
5. While  $L \neq \emptyset$  and *cycle\_found* = false do:
  - Choose a node  $q \in L$  and set  $L \leftarrow L - \{q\}$ .
  - Perform a breadth-first search from  $q$  on arcs having a residual capacity ( $c_{ij} - x_j > \epsilon > 0$ ), and such that their heads are neither visited nor labeled.
  - Update  $p_f(w)$  for the nodes  $w$  being discovered.
  - If  $p_b(w) \neq 0$ , set *cycle\_found*  $\leftarrow$  true and stop the search - a  $k$ -feasible cycle with extremities  $q$  and  $w$  has been found.
6. If *cycle\_found* = true, compute the largest amount of flow  $\alpha \in (0, \sigma]$  that it supports by making use of the predecessor indices.

**4.1.3. Complexity Analysis.** We now discuss the worst-case complexity of the trial-and-error procedure. Every commodity  $k$  might be investigated in trying to bring the flow on a given arc  $j$  to  $\gamma_j c_{0j}$ . Given two flow vectors  $x'_k$  and  $x''_k$  of a commodity  $k$ , their difference,  $\Delta = x'_k - x''_k$ , can be expressed as the summation of at most  $(|A| - |V| + 1)$  cycles. The procedure to find a positive-cost  $k$ -feasible cycle has the same complexity as breadth-first search,  $O(|A|)$  [1]. Thus, the complexity of the whole procedure is  $O(|A|^3 K)$ .

#### 4.2. Memory Structures

We make use of two memory structures: a medium-term memory, which remains active during a fixed number of iterations, and a short-term memory, which acts within the local search phase of any single iteration.

The medium-term memory imposes restrictions on the selection of an arc to guide the search in the first phase of each iteration. Because it can happen that the local optimization only slightly deviates the flow on an arc  $j$  that has just been brought to  $\gamma_j c_{0j}$ , such an arc should be prevented from being chosen in the first phase of the subsequent iteration.

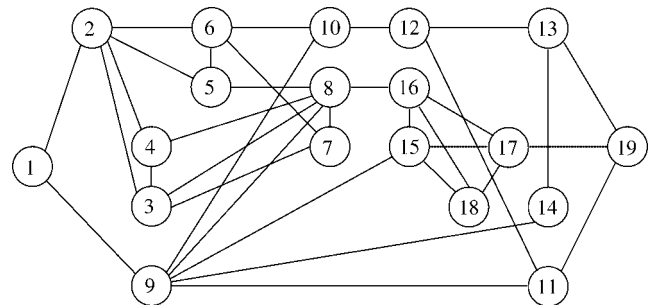


FIG. 3. Network used to generate the first group of test instances.

TABLE 1. Results for the homogeneous case.

$r$	$c_0$	$c_1$	IS	CA-FA	CCA	TB_CCA	Reductions (%)			LB
							r(IS)	r(CA-FA)	r(CCA)	
1	4	12	28.22	26.82	26.53	25.64	12.1	5.5	4.5	21.47
	5	15	21.84	21.27	21.27	20.23	9.5	6.1	6.1	17.08
	4	16	28.07	24.76	24.23	23.70	26.0	6.3	3.2	16.84
	5	20	21.25	19.58	19.40	19.21	15.2	2.7	1.4	13.45
2	4	12	53.95	49.95	49.33	48.49	12.6	3.4	2.0	43.59
	5	15	43.40	39.77	39.77	38.78	13.3	2.8	2.8	34.63
	4	16	49.17	44.93	44.50	40.91	24.5	11.9	10.6	33.77
	5	20	39.76	37.89	37.89	34.73	18.6	11.7	11.7	27.00

Otherwise, the search would be trapped between two bordering subsets. The medium-term memory has thus as attribute the arc  $j$  for which the flow has been brought to  $\gamma_j c_{0j}$ , which is kept tabu for *tabu tenure* iterations.

The short-term memory controls the local search in the second phase of an iteration, which consists of applying CCA from the solution  $x'$  obtained in the first phase. The short-term memory prevents the algorithm from going back to the recently visited local optimum  $x^*$ , avoiding the search being trapped in  $N(x^*)$ . Let  $j'$  be the arc for which flow has been brought to  $\gamma_{j'} c_{0j'}$  in the first phase. If, during this phase, we have increased (respectively, decreased) the value of  $x_{j'}$ , we classify as tabu the act of canceling a negative-cost cycle where  $j'$  is a backward (respectively, forward) arc with a current flow equal to  $\gamma_{j'} c_{0j'}$ . The short-term memory tabu status is kept active during the whole local optimization phase.

### 5. NUMERICAL EXPERIMENTS

We conducted numerical experiments on three different approaches to solve CCE: the classical capacity assignment-flow assignment (CA-FA) method [9], the cycle canceling (CCA) method described in Section 3, and the tabu search algorithm (TB\_CCA) proposed in Section 4.

The CA-FA algorithm alternates between a capacity assignment phase, with fixed routing, and a flow assignment phase, with fixed arc capacities, until no further improvements are possible. The capacity assignment phase is solved by inspection, while the routing phase is a convex

multicommodity network flow problem. To apply the CA-FA algorithm to CCE, we must decide which one of the two capacities,  $c_{0j}$  or  $c_{1j}$ , to assign whenever an arc  $j$  is at the breakpoint, i.e.,  $x_j = \gamma_j c_{0j}$ . Consequently, we have one of the two “active” congestion functions,  $\Phi(c_{0j}, x_j)$  or  $\Phi(c_{1j}, x_j)$ , on arc  $j$  in the subsequent routing phase. Note that CA-FA does not necessarily stop at a local minimum of CCE, since it does not take into account that  $\gamma_j c_{0j}$  is a point of discontinuity of the arc cost function derivative  $f'_j$ . As an example, assume we are given a feasible solution  $x^*$  in which an arc  $j$  is at the breakpoint. The two subsets composing the neighborhood  $N(x^*)$  are  $C_1$  and  $C_2$  associated, respectively, with the intervals  $[0, \gamma_j c_{0j}]$  and  $[\gamma_j c_{0j}, c_{1j}]$ . Suppose we decide, without loss of generality, to fix the capacity of arc  $j$  to its smallest value  $c_{0j}$ . If the routing does not change in the following flow assignment phase, then the algorithm stops. Since  $\Phi(c_{0j}, x_j)$  has been considered as the arc cost function on  $j$ , we can conclude that the local optimality conditions are satisfied for the subset  $C_1$ , but we cannot assess the optimality conditions for the subset  $C_2$ . Indeed, there might exist a negative-cost  $k$ -feasible cycle in which  $j$  is a forward arc, and  $x^*$  is not locally optimal for  $C_2$ .

We use the convex approximation proposed by Luna and Mahey [15] to generate lower bounds, as well as initial solutions for CA-FA and CCA. The proximal decomposition method described in Mahey et al. [17] is used to solve the convex multicommodity flow problems found in the initial convex approximation and in the routing phases of CA-FA. The solutions obtained by CCA are in turn initial solutions

TABLE 2. Results for the heterogeneous case.

$r$	$c_0$	$c_1$	IS	CA-FA	CCA	TB_CCA	Reductions (%)			LB
							r(IS)	r(CA-FA)	r(CCA)	
[0.5,2]	4	12	32.51	30.54	30.54	29.51	12.0	4.2	4.2	25.00
	5	15	25.57	24.74	24.74	23.30	11.4	7.3	7.3	19.91
	4	16	31.46	28.17	28.06	26.43	25.0	8.9	8.4	19.56
	5	20	25.42	23.98	23.68	21.92	22.4	13.2	11.3	15.63
[0.5,4]	4	12	51.45	48.21	48.21	47.04	10.2	2.7	2.7	43.26
	5	15	42.74	40.93	40.93	39.12	10.6	5.3	5.3	34.16
	4	16	44.79	43.23	42.57	40.63	12.5	7.8	5.8	33.26
	5	20	39.23	38.40	37.59	36.50	10.3	7.2	4.0	26.58

TABLE 3. Network characteristics of the second group of instances.

Instance	Topology	$ V $	$ E $	$K$
att	AT&T Worldnet backbone	90	274	272
fr250	Frame-relay	60	688	250
fr500	Frame-relay	60	906	500
hier50	2-level hierarchical	50	148	245

for TB\_CCA. The *tabu tenure* for the medium-term memory is an integer randomly generated in the interval  $[1, 5]$ . All the algorithms involved in our numerical experiments were implemented in C and performed on a PC Pentium III 700 MHz with 512 Mbytes of RAM memory.

Two groups of test instances were used. The first group is derived from a real network introduced by Mahey et al. [17], also used by Ouorou and Mahey [20]. Figure 3 shows this network, provided by the Centre National d'Études des Télécommunications, today France Télécom R&D. It has 19 nodes, 68 arcs (two arcs, one in each direction, for each pair of nodes directly connected), and 30 origin-destination pairs (each one treated as a different commodity). Installed capacities all have the same value  $c_0$ , and expansion to a capacity  $c_1$  is possible for every arc in the network. The parameter  $\gamma$  is set to 50% for every arc in the network, meaning that the expansion on an arc  $j$  occurs when the flow  $x_j$  reaches half of the installed capacity. The arc congestion costs are given by Kleinrock's function  $\Phi(c, x_j) = \frac{x_j}{c-x_j}$ , for a capacity value  $c$  and an arc  $j$ . To obtain different instances from this network, we vary the dimensional aspects, such as the ratio  $c_1/c_0$ , and the traffic load aspects, such as the heterogeneity of the demand requirements and its ratio over installed capacities.

Tables 1 and 2 display the results for the homogeneous and the heterogeneous demand requirements scenarios, respectively, for the network of Figure 3. We present problem characteristics in the first three columns (demand  $r$ , installed capacities  $c_0$ , and capacities for expansion  $c_1$ ). In Table 1, the traffic demands have the values indicated, while in Table 2, the traffic demand between each origin-destination pair is a value taken at random from the interval indicated. We then report in the fourth through seventh columns the objective function values of the following solutions: the initial solution IS obtained by the convex approximation [15], the solution obtained by CA-FA [9], the solution obtained by CCA, and the best solution found by TB\_CCA. In the next

three columns, we show the relative reductions provided by TB\_CCA with respect to the lower bound over the three other tested methods, i.e.,  $r(X) = \frac{X - \text{TB\_CCA}}{\text{LB}}$ %, where  $X$  is the objective function value of IS or of the solutions obtained by CA-FA or CCA, depending upon the case. In the last column, we report the lower bound LB given by the convex approximation [15].

In 9 out of 16 instances, solutions provided by CCA are better than the ones provided by CA-FA. It is important to point out that CCA performs particularly better than CA-FA when initial solutions have some arcs at the breakpoint. We relate this experimental observation to the fact that only CCA uses the information about the different right and left derivatives at the breakpoint. This happened in the four data configurations in Table 1 where  $c_0 = 4$ . In the solutions obtained by the CA-FA algorithm, we have one arc at the breakpoint in the following data configurations in Table 1:  $r = 1, c_0 = 4$ , with  $c_1 = 12$  and  $c_1 = 16$ . Consequently, we cannot assess the local optimality of the solutions obtained by the CA-FA in these two cases. For the solutions obtained by the CCA algorithm, we have one arc at the breakpoint in the following data configuration in Table 1:  $r = 1, c_0 = 4$  and  $c_1 = 12$ .

We gained significant improvements by applying TB\_CCA. In both the homogeneous and heterogeneous cases, average reductions from the local optima provided by CCA are 4.2% and 4.9%, respectively. Tables 1 and 2 clearly show that TB\_CCA outperforms CA-FA in terms of solution quality. Average reductions from the solution provided by CA-FA are 6.3% and 7.1% for the homogeneous and heterogeneous cases, respectively, and up to 13.2% in the latter case. This improvement in solution quality, however, requires an increased computational time. We performed TB\_CCA up to a maximum of 100 iterations without improvement in the best found solution. For these instances, algorithms CA-FA and CCA took less than 10s, while TB\_CCA required about 100s of CPU time.

The second group of data instances consists of four larger networks with different topologies. Table 3 summarizes the characteristics of the networks considered. These networks were used by Resende and Ribeiro [22] in the context of private virtual circuit routing. For these instances, we considered a traffic demand between every origin-destination pair equal to 1, a ratio  $c_1/c_0$  equal to 4, and the parameter  $\gamma$  set to 50%.

Table 4 displays the results for the second group of instances. The instances are identified in the first column. As

TABLE 4. Results for the second group of instances.

Instance	IS	CA-FA	CCA	TABU	Reductions (%)			LB
					r(IS)	r(CA-FA)	r(CCA)	
att	183.05	172.44	172.44	168.34	10.1	2.8	2.8	145.26
fr250	176.83	148.85	148.85	147.16	28.4	1.6	1.6	104.62
fr500	329.21	277.53	277.53	274.67	27.3	1.4	1.4	198.50
hier50	128.60	122.45	121.84	119.27	9.9	3.4	2.7	94.17



in Tables 1 and 2, the following columns display solution values for each algorithm, reductions provided by TB\_CCA, and lower bounds. Given the network sizes, we ran TB\_CCA up to a maximum of 10 iterations without improvement in the best found solution. On the instances fr250 and fr500 (which are the most time consuming), TB\_CCA required about one hour of CPU time. Even by performing only a maximum of 10 iterations without improvement, TB\_CCA obtained significant reductions for the four instances.

## 6. CONCLUDING REMARKS AND EXTENSIONS

We have presented and analyzed heuristic methods for a piecewise convex multicommodity flow problem. The methods are based on a cycle-canceling approach that subsumes the classical CA-FA method, in the sense that it can improve the best solution found by CA-FA which is not necessarily a local minimum. We have shown that the nonconvex nature of the arc cost functions does not alter the convergence properties of the cycle-canceling algorithm. We proposed a tabu search algorithm to explore the solution space beyond the first local optimum. The two-phase framework of each iteration leads to two kinds of cycling for which we used two distinct memory structures. A medium-term memory prevents the search from being blocked in between two subsets over each of which the objective function is convex. It restricts an arc to be chosen for guiding the search in the first phase for *tabu tenure* iterations. A short-term memory prevents the local optimization in the second phase from returning to the local optimum visited in the preceding iteration. Our main interest in conducting numerical experiments was to verify that the tabu search algorithm can significantly improve feasible solutions obtained by a single local optimization procedure. Some important extensions of this approach include the development of diversification strategies, and of heuristic rules to speed up the search, such as move estimations, candidate lists and neighborhood reduction (see for example Ribeiro and Souza [23]).

## Acknowledgments

The authors are grateful to Walid Benameur for his constructive comments and suggestions which helped us to improve the paper significantly.

## REFERENCES

- [1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network flows: Theory, algorithms and applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
- [2] F. Barahona and E. Tardos, Note on Weintraub's minimum-cost circulation algorithm, *SIAM J Computing* 18 (1989), 579–583.
- [3] D.P. Bertsekas and R.G. Gallager, *Data networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [4] T.G. Crainic, M. Gendreau, and J.M. Farvolden, A simplex-based tabu search method for capacitated network design, *INFORMS J Computing* 12 (2000), 223–236.
- [5] R. Fletcher, *Practical methods of optimization, Unconstrained optimization*, John Wiley & Sons, New York, 1987.
- [6] B. Gavish and K. Altinkemer, Backbone network design tools with economic tradeoffs, *ORSA J Computing* 2 (1990), 236–252.
- [7] B. Gavish and I. Neuman, A system for routing and capacity assignment in computer communications networks, *IEEE Trans Commun* 37 (1989), 360–366.
- [8] B. Gendron, T.G. Crainic, and A. Frangioni, “Multicommodity capacitated network design”, *Telecommunications network planning*, P. Soriano and B. Sansò (Editors), Kluwer, 1999, pp. 1–19.
- [9] M. Gerla and L. Kleinrock, On the topological design of distributed computer networks, *IEEE Trans Commun* 25 (1977), 48–60.
- [10] M. Gerla, J.A.S. Monteiro, and R. Pazos, Topology design and band with allocation in ATM nets, *IEEE J Selected Areas Commun* 7 (1989), 1253–1261.
- [11] I. Ghamlouche, T.G. Crainic, and M. Gendreau, Cycle-based neighborhoods for fixed-charge capacitated multicommodity network design, *Oper Res* 51 (2003), 655–667.
- [12] F. Glover, Tabu search – Part I, *ORSA J Computing* 1 (1989), 190–206.
- [13] F. Glover, Tabu search – Part II, *ORSA J Computing* 2 (1990), 4–32.
- [14] M. Klein, A primal method for minimal cost flows with applications to the assignment and transportation problems, *Manag Sci* 14 (1967), 205–220.
- [15] 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.
- [16] P. Mahey, A. Benchakroun, and F. Boyer, Capacity and flow assignment of data networks by generalized Benders decomposition, *J Global Optimization* 20 (2001), 173–193.
- [17] P. Mahey, A. Ouorou, L. LeBlanc, and J. Chifflet, A new proximal decomposition algorithm for routing in telecommunication networks, *Networks* 31 (1998), 227–238.
- [18] P. Mahey and M.C. de Souza, Solving multicommodity flow problems with separable piecewise convex costs, *Research Report LIMOS RR 04-09*, Université Blaise Pascal, France, June 2004. Available at <http://www.isima.fr/limos/publi/RR-04-09.pdf>.
- [19] P. Mahey and M.C. de Souza, Local optimality conditions for multicommodity flow problems with separable piecewise convex costs, *Oper Res Lett* 35 (2007), 221–226.
- [20] A. Ouorou and P. Mahey, A minimum mean cycle cancelling method for nonlinear multicommodity flow problems, *Eur J Oper Res* 121 (2000), 532–548.
- [21] A. Ouorou, P. Mahey, and J.P. Vial, A survey of algorithms for convex multicommodity flow problems, *Manag Sci* 46 (2000), 126–147.
- [22] M.G.C. Resende and C.C. Ribeiro, GRASP with path-relinking for private virtual circuit routing, *Networks* 41 (2003), 104–114.
- [23] C.C. Ribeiro and M.C. de Souza, Tabu search for the Steiner problem in graphs, *Networks* 36 (2000), 138–146.
- [24] A. Weintraub, A primal algorithm to solve network flow problems with convex costs, *Manag Sci* 21 (1974), 87–97.