



HAL
open science

A minimum mean cycle cancelling method for nonlinear multicommodity flow problems

Adam Ouorou, Philippe Mahey

► **To cite this version:**

Adam Ouorou, Philippe Mahey. A minimum mean cycle cancelling method for nonlinear multicommodity flow problems. *European Journal of Operational Research*, 2000, 121 (3), pp.532-548. hal-01644719

HAL Id: hal-01644719

<https://hal.science/hal-01644719>

Submitted on 22 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Minimum Mean Cycle Cancelling Method for Nonlinear Multicommodity Flow Problems

A. Ouorou P. Mahey

Laboratoire d'Informatique, de Modélisation et
d'Optimisation des Systèmes, Université Blaise Pascal,
ISIMA, BP 125, 63173 Aubière cedex, France.

Abstract

We propose a new method based on minimum mean cycles cancelling for multicommodity flow problems with separable convex cost ruling out saturated capacities. This method is inspired by the cycles cancelling method first worked out by Goldberg and Tarjan for minimum cost circulations [9]. Convergence of the method is formally proved and a variant with a more flexible selection of cycles is proposed. Also, we report some computational experience on the message routing problem in telecommunication networks using actual and randomly generated networks.

Key words: Network programming, multicommodity flows, cycles cancelling methods, separable convex programming.

1 Introduction

Multicommodity Flow problems are among the most difficult and important problems in Network Optimization. Most textbooks on network flows consider separately the convex minimum cost flow problem and the multicommodity flow problem [1, 10], but scarcely the nonlinear multicommodity flow problem. The first nonlinear multicommodity flow models have been extensively studied in connection with the design and analysis of communications systems and traffic assignment. For example, in a packet-switched communication network, messages packets generated at sources connected to the network, have to be routed over the network links to their destinations. The routing problem consists of determining an assignment of routes in order to minimize some cost function. The most common performance measure used in the literature is the average message delay and the problem then turns out to be a multicommodity flow problems with a nonlinear separable convex cost [2]. Much of the motivation for this work comes from this message routing problem. However the range

of applications of nonlinear multicommodity models also includes transportation networks, energy transport systems, water distribution networks and many other areas.

In practice, the large scale of multicommodity flow problems makes a straightforward application of mathematical programming tools computationally cumbersome. On the other hand, the rich block-angular structure of the models makes attractive the design of specially tailored algorithms such as decomposition algorithms. Early approaches were specializations of general nonlinear programming [15, 6, 2, 7]. Recently emphasis has been placed on solution techniques such as interior point [22], analytic center cutting plane [8] and proximal algorithms [5, 16]. See [18] for a survey of convex multicommodity flow problems. In this work, we focus on the convex multicommodity flow problem and we point out that the proposed algorithms extend the mean cycle cancelling algorithm for convex cost flows but do not apply to linear multicommodity flow problems. Our aim in this paper is to show that the idea of minimum mean cycle cancelling originally worked out by Goldberg and Tarjan [9] and extended later by Karzanov and McCormick [13] for separable convex optimization in unimodular linear spaces, can be exploited to give a new method for multicommodity flow problems with separable convex costs. This method is a block-coordinate class method which makes descent steps that involve altering of one commodity and the vector of total flows around a cycle. We identify such a cycle using minimum mean directed cycles algorithms in an auxiliary graph related to that commodity.

The rest of the paper is organized as follows. In the next section, we formulate the nonlinear multicommodity flow problem. In section 3, we give an optimality criterion for the problem based on the *absolute mean value* of a commodity. From this optimality criterion, we derive a new method, cancelling *minimum mean feasible* cycles for each commodity, and prove its convergence in section 4. In section 5, we give a variant of the method which alternates between primal and dual steps: a step improving the primal variables and another one for the dual variables. In section 6, we present some numerical experiments obtained on the message routing problem with actual and randomly generated networks so as to demonstrate the effectiveness of the proposed methods.

2 The Nonlinear Multicommodity Flow Problem

Let $G = (V, E)$ be a directed network with m nodes and n arcs and K flow vectors x_1, \dots, x_K of \mathbb{R}^n associated with K commodities to be transported through G . The flow of commodity k on arc j is denoted x_{kj} . Let M be the $m \times n$ node-arc incidence matrix of G . We suppose that each commodity has a single origin destination pair (s_k, t_k) and that we are given the demand requirement $r_k \geq 0$ between s_k and t_k . Let $b_k \in \mathbb{R}^m$ be the vector with all

components 0 except $b_{ks_k} = -b_{kt_k} = 1$. For each $k = 1, \dots, K$, we define the set Ω_k of vectors $x_k \geq 0$ which satisfy the network constraints $Mx_k = r_k b_k$:

$$\Omega_k = \{x_k \in \mathbb{R}^n : Mx_k = r_k b_k, x_k \geq 0\}.$$

We define a *multicommodity flow* as a set of vectors $x = (\xi, x_1, x_2, \dots, x_K)$ with $x_k \in \Omega_k$ and $\xi = \sum_k x_k$ (in words, ξ is the vector of total flows).

To each arc j , we associate a real c_j representing its capacity, and a nonlinear cost function f_j in order to take into account congestion effects in the network G . We assume that for each j , f_j is a proper convex continuously differentiable function which is everywhere finite value and $f_j(\xi_j)$ increases radically as ξ_j exceeds the capacity c_j . This corresponds to the existence of barriers that prevent the solution from approaching the capacity when f_j is minimized. These assumptions are justified for almost all functions one encounters in practice for data networks. For instance, in data network routing problem, the cost function is often based on a queueing model of average delay and f_j has typically the form:

$$f_j(\xi_j) = \frac{\xi_j}{c_j - \xi_j}, \quad \xi_j < c_j. \quad (1)$$

Our hypotheses on the f_j functions exclude the linear capacitated case. In that situation, one way to use the idea of cancelling negative cycle, is to introduce a barrier function to force an interior point convergence toward the optimal solution. Let d_j be the linear cost on arc j . Then we can use:

$$f_j(\xi_j) = d_j \xi_j - \eta \log(c_j - \xi_j), \quad \xi_j < c_j,$$

with $\eta > 0$ sufficiently large. These developments are exposed in [19]. Here we shall be concerned with the minimum cost multicommodity flow problem:

$$(P) \quad \begin{cases} \min & f(x) = \sum_{j=1}^n f_j(\xi_j) \\ \text{s.t.} & \xi_j = \sum_{k=1}^K x_{kj}, \quad \xi_j < c_j, \quad j = 1, \dots, n, \\ & Mx_k = r_k b_k, \quad x_k \geq 0, \quad k = 1, \dots, K. \end{cases}$$

The assumptions on f_j force the solution of problem (P) to meet capacity constraints and the problem then turns out to be formally uncapacitated. We suppose that there is a feasible solution in the interior of the effective domain of f , $\text{dom}(f) = \{x : f(x) < +\infty\}$. Hence the set of Kuhn-Tucker vectors is not empty [21].

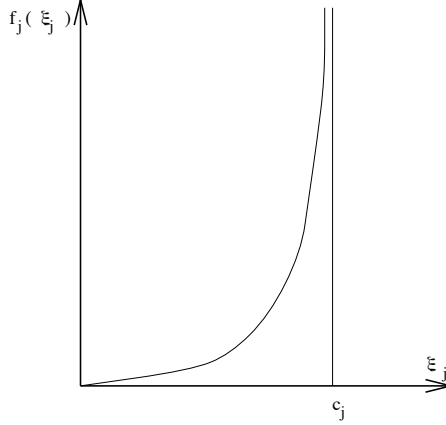


Figure 1: Example of Cost Function f_j .

In the sequel, we denote by \mathcal{X} the following set:

$$\mathcal{X} = \left\{ (\xi, x_1, \dots, x_K) : x_k \in \Omega_k \forall k, \xi = \sum_k x_k \right\}.$$

For a multicommodity $x = (\xi, x_1, \dots, x_K)$, we define for each commodity k , the sets

$$E_k(x) = \{j \in E : x_{kj} > 0\} \quad \text{and} \quad \bar{E}_k(x) = E \setminus E_k(x) = \{j \in E : x_{kj} = 0\}.$$

3 Optimality Criterion

In this section, we introduce a characterization of optimality for problem (P). We first state some basic definitions and notations which will be used in the paper.

A *chain* in G is a sequence of nodes and arcs of G , $i_1, j_1, i_2, j_2, \dots, i_{p-1}, j_{p-1}, i_p$ such that for $l = 1, \dots, p-1$ either $j_l = (i_l, i_{l+1})$ or $j_l = (i_{l+1}, i_l)$. A *cycle* is a chain $i_1, j_1, i_2, j_2, \dots, i_{p-1}, j_{p-1}, i_p$ without any repetition of nodes together with the arc (i_1, i_p) or (i_p, i_1) . In the sequel, we will refer to a cycle as a set of arcs without any explicit mention of the nodes. Given a cycle γ , a *forward* arc $j = (i_1, i_2)$ is an arc such that the cycle visits node i_1 before node i_2 ; it is a *backward* arc otherwise. We denote by γ^+ the set of forward arcs of γ and γ^- the set of its backward arcs. To any cycle γ , we associate the vector $\Gamma(\gamma)$ with components

$$\Gamma_j(\gamma) = \begin{cases} 1 & \text{if } j \in \gamma^+, \\ -1 & \text{if } j \in \gamma^-, \\ 0 & \text{otherwise.} \end{cases}$$

Let $x = (\xi, x_1, \dots, x_K) \in \mathcal{X}$. For a single commodity flow $x_k \in \Omega_k$, a cycle γ is *k-feasible* if for any $j \in \gamma^-$, $x_{kj} > 0$ (i.e. $\gamma^- \subseteq E_k(x)$). We denote by $\Lambda_k(x)$ the set of *k-feasible* cycles. Obviously, if $\gamma \in \Lambda_k(x)$, $x_k + \alpha\Gamma(\gamma) \in \Omega_k$ for all $\alpha \in [0, \min_{j \in E_k(x)} x_{kj}]$.

Let f'_j be the first derivative of f_j and $|\gamma|$ the number of arcs in cycle γ . For a multi-commodity flow $x = (\xi, x_1, \dots, x_K) \in \mathcal{X}$, we define $c(x, \gamma)$ and $\bar{c}(x, \gamma)$ to be respectively the *cost* and the *mean cost* of γ with respect to x :

$$c(x, \gamma) = \sum_{j \in \gamma^+} f'_j(\xi_j) - \sum_{j \in \gamma^-} f'_j(\xi_j), \quad \bar{c}(x, \gamma) = c(x, \gamma)/|\gamma|.$$

A cycle γ is said to be *negative* if $c(x, \gamma) < 0$.

Our characterization of the optimal solutions of problem (P) depends on the *k-absolute mean* for each commodity k , which is defined to be

$$\lambda_k(x) = \max \left(0, - \min_{\gamma \in \Lambda_k(x)} \bar{c}(x, \gamma) \right).$$

Note that $\lambda_k(x) \geq 0$ by definition. A *k-feasible* cycle γ is said to be a *k-minimum mean cycle* if $\bar{c}(x, \gamma) = \min_{\hat{\gamma} \in \Lambda_k(x)} \bar{c}(x, \hat{\gamma})$. To find a *k-minimum mean cycle*, we define an auxiliary digraph \bar{G}_k with arcs cost with respect to x as follows: \bar{G}_k contains all arcs in E and the reverses of only arcs in $E_k(x)$. Each arc $j \in E$ has the cost $f'_j(\xi_j)$, and the reverse arc \bar{j} of $j \in E_k(x)$ has the cost $-f'_j(\xi_j)$. There exists many polynomial algorithms for computing minimum mean directed cycles (see for instance [11, 12, 23]). A directed cycle $\bar{\gamma}$ in \bar{G}_k corresponds to a *k-feasible* cycle γ in G as follows: if $j \in \bar{\gamma}$, then $j \in \gamma^+$; if $\bar{j} \in \bar{\gamma}$ then $j \in \gamma^-$.

In our study, we will use the following result on *k-absolute means*.

Proposition 1 *Let $x = (\xi, x_1, \dots, x_K) \in \mathcal{X}$. For each commodity k , λ is an upper bound of $\lambda_k(x)$ if and only if there exists a vector $p_k \in \mathbb{R}^m$ such that if we assign to each node i of G the components p_{ki} and define the vector $z_k \in \mathbb{R}^n$ with components $z_{kj} = p_{ki} - p_{ks}$, $j = (i, s)$, we have*

$$\begin{cases} f'_j(\xi_j) - \lambda \leq z_{kj} \leq f'_j(\xi_j) + \lambda \text{ for all } j \in E_k(x), \\ f'_j(\xi_j) - \lambda - \alpha_{kj} \leq z_{kj} \leq f'_j(\xi_j) + \lambda, \alpha_{kj} \geq 0, \text{ for all } j \in \bar{E}_k(x). \end{cases} \quad (2)$$

*If $\lambda = \lambda_k(x) > 0$ and $\gamma \in \Lambda_k(x)$ is a *k-minimum mean cycle*, then*

$$f'_j(\xi_j) - z_{kj} = -\lambda \text{ for all } j \in \gamma^+ \text{ and } z_{kj} - f'_j(\xi_j) = -\lambda \text{ for all } j \in \gamma^-.$$

Proof. We follow McCormick [17] for a similar statement in [13]. Consider a multicommodity flow $x = (\xi, x_1, \dots, x_K) \in \mathcal{X}$. A k -minimum mean cycle can be seen as a solution of the linear program

$$\begin{aligned} \min \quad & (f'x^+ - f'x^-)/(ex^+ + ex^-) \\ \text{s.t.} \quad & M(x^+ - x^-) = 0, \\ & x_j^- \leq 0, \quad j \in \overline{E}_k(x), \\ & x^+, x^- \geq 0. \end{aligned} \tag{3}$$

where $e \in \mathbb{R}^n$ with $e_j = 1$ for all j and $f' \in \mathbb{R}^n$ is the vector of coordinates $f'_j(\xi_j)$.

Note that if (x^+, x^-) is optimal for this problem, so is $\alpha(x^+, x^-)$ for any $\alpha > 0$. Then we can require the denominator to be equal to one, so an equivalent problem is

$$\begin{aligned} \min \quad & f'x^+ - f'x^- \\ \text{s.t.} \quad & M(x^+ - x^-) = 0, \\ & ex^+ + ex^- = 1, \\ & x_j^- \leq 0, \quad j \in \overline{E}_k(x), \\ & x^+, x^- \geq 0. \end{aligned} \tag{4}$$

The dual of (4) is

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & p_{ki} - p_{ks} + \delta \leq f'_j(\xi_j), \quad j = 1, \dots, n, \\ & -(p_{ki} - p_{ks}) + \delta \leq -f'_j(\xi_j), \quad j \in E_k(x), \\ & -(p_{ki} - p_{ks}) + \alpha_{kj} + \delta \leq -f'_j(\xi_j), \quad j \in \overline{E}_k(x), \\ & p_k \in \mathbb{R}^m, \quad \delta \in \mathbb{R}, \quad \alpha_{kj} \leq 0. \end{aligned} \tag{5}$$

We let z_k be the vector of coordinates $z_{kj} = p_{ki} - p_{ks}$ for each arc $j = (i, s)$. Then the dual problem can be rewritten as

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & \delta \leq f'_j(\xi_j) - z_{kj}, \quad j = 1, \dots, n, \\ & \delta \leq z_{kj} - f'_j(\xi_j), \quad j \in E_k(x), \\ & \delta \leq z_{kj} - \alpha_{kj} - f'_j(\xi_j), \quad j \in \overline{E}_k(x), \\ & z_k \in \mathbb{R}^n, \quad \delta \in \mathbb{R}, \quad \alpha_{kj} \leq 0. \end{aligned} \tag{6}$$

This problem can be interpreted as searching the largest least negative value of δ that satisfies (2). Hence it is easy to derive the results in the proposition. \square

The vectors $p_k \in \mathbb{R}^m$ and $z_k \in \mathbb{R}^n$ associated with the multicommodity flow $x \in \mathcal{X}$ in proposition 1, can be seen as respectively the *price* and *tension* vectors associated to commodity k , and $f'_j(\xi_j) - z_{kj}$ as the *reduced cost* of arc j with respect to commodity k .

We now state optimality conditions for characterizing when a given feasible solution for (P) is optimal:

Theorem 1 *A feasible multicommodity flow x is an optimal solution for problem (P) if and only if $\lambda_k(x) = 0$ for all k .*

Proof. We first show that if x is optimal for (P) then $\lambda_k(x) = 0$ for each k .

We argue by contradiction. Suppose there exists a commodity k such that $\lambda_k(x) > 0$. Let γ be a k -feasible cycle such that $\bar{c}(x, \gamma) = -\lambda_k(x)$. From the continuity of f'_j , there exists an $\alpha > 0$ such that

$$f'_j(\xi_j + \alpha) - f'_j(\xi_j) < \lambda_k(x) \text{ for } j \in \gamma^+ \text{ and } f'_j(\xi_j) - f'_j(\xi_j - \alpha) < \lambda_k(x) \text{ for } j \in \gamma^-.$$

Define $\varepsilon = \min(\alpha, \min_{j \in \gamma^-} x_{kj})$; then $\varepsilon > 0$, and as f'_j is nondecreasing by convexity, we have

$$f'_j(\xi_j + \varepsilon) - f'_j(\xi_j) < \lambda_k(x) \text{ if } j \in \gamma^+ \text{ and } f'_j(\xi_j) - f'_j(\xi_j - \varepsilon) < \lambda_k(x) \text{ if } j \in \gamma^-.$$

Let $y = (\zeta, y_1, \dots, y_K)$ be the multicommodity flow obtained from x by the following k -cancelling cycle step.

$$\begin{cases} \zeta = \xi + \varepsilon \Gamma(\gamma), \\ y_k = x_k + \varepsilon \Gamma(\gamma), \\ y_p = x_p \text{ for all } p \neq k. \end{cases} \quad (7)$$

The k -cancelling step increases the total flows and the flows of the commodity k on forward arcs in the k -feasible cycle γ and decreases the total flows and the flows of the commodity k on backward arcs in the cycle. It maintains the feasibility of the new commodity y (i.e $y \in \mathcal{X}$). By convexity of the functions f_j , we have

$$\begin{cases} f_j(\xi_j + \varepsilon) - f_j(\xi_j) \leq \varepsilon f'_j(\xi_j + \varepsilon) = \varepsilon f'_j(\xi_j) + \varepsilon(f'_j(\xi_j + \varepsilon) - f'_j(\xi_j)) & \text{if } j \in \gamma^+, \\ f_j(\xi_j - \varepsilon) - f_j(\xi_j) \leq -\varepsilon f'_j(\xi_j - \varepsilon) = -\varepsilon f'_j(\xi_j) + \varepsilon(f'_j(\xi_j) - f'_j(\xi_j - \varepsilon)) & \text{if } j \in \gamma^-. \end{cases}$$

Hence

$$f(y) - f(x) = \sum_j [f_j(\zeta_j) - f_j(\xi_j)] < \varepsilon c(x, \gamma) + \varepsilon |\gamma| \lambda_k(x) = 0.$$

Now let $\bar{x} \in \mathcal{X}$ such that $\lambda_k(\bar{x}) = 0$ for all $k = 1, \dots, K$ and $\bar{z}_k \in \mathbb{R}^n$, $k = 1, \dots, K$ the tension vectors associated with \bar{x} in proposition 1. Then (2) yields

$$\bar{z}_{kj} = f'_j(\bar{\xi}_j) \text{ for all } j \in E_k(\bar{x}), \quad \bar{z}_{kj} \leq f'_j(\bar{\xi}_j) \text{ for all } j \in \bar{E}_k(\bar{x}) \quad (8)$$

Consider now the vector $\bar{z}_0 \in \mathbb{R}^n$ which components are defined by $\bar{z}_{0j} = f'_j(\bar{\xi}_j)$ if $\bar{\xi}_j > 0$ and $\bar{z}_{0j} = \max_k \bar{z}_{kj}$ if $\bar{\xi}_j = 0$. Then $(\bar{x}, \bar{z}_0, \bar{z}_k, k = 1, \dots, K)$ verify the following Kuhn-Tucker

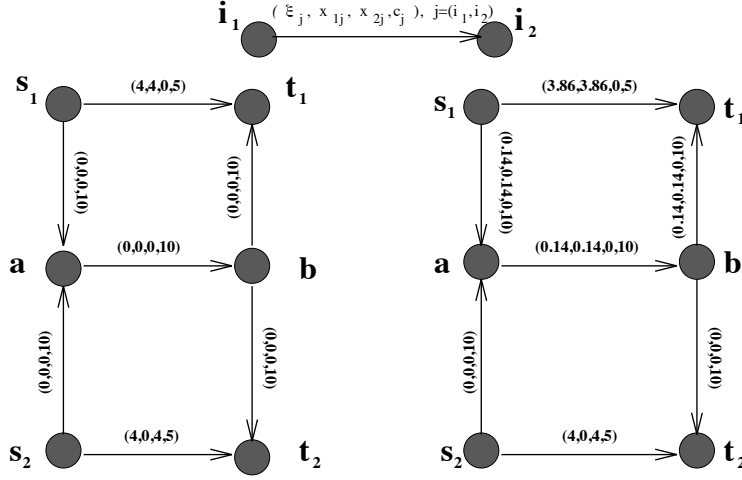


Figure 2: Illustration of k -cancelling step.

conditions for problem (P).

$$\begin{cases} \bar{z}_{0j} = f'_j(\bar{\xi}_j), & \text{if } \bar{\xi}_j > 0, \\ \bar{z}_{0j} \leq f'_j(\bar{\xi}_j), & \text{if } \bar{\xi}_j = 0, \\ \bar{z}_{0j} = \bar{z}_{kj}, & \text{if } \bar{x}_{kj} > 0 \quad j = 1, \dots, n, \quad k = 1, \dots, K \\ \bar{z}_{0j} \geq \bar{z}_{kj} & \text{if } \bar{x}_{kj} = 0 \quad j = 1, \dots, n, \quad k = 1, \dots, K \end{cases}$$

implying the optimality of \bar{x} .

(We use the notations z_0 for the dual vector associated with the constraints $\xi = \sum_{k=1}^K x_k$ while the potential vectors p_k which define the z_k 's in (2), are associated with the constraints $Mx_k = r_k b_{k\cdot}$) \square

Example of k -cancelling step

To illustrate the k -cancelling step, we consider the multicommodity flow problem of figure 2 with two commodities: the source-destination pairs are (s_1, t_1) and (s_2, t_2) and the requirements $r_1 = r_2 = 4$. We suppose that the functions f_j are given by (1). The numbers on arcs correspond to $(\xi_j, x_{1j}, x_{2j}, c_j)$.

Obviously the first multicommodity flow shown in figure 2 with cost $f_1 = 8.0$ is not optimal ($\lambda_1(x) = \lambda_2(x) = 1.17$). The k -cancelling step with $k = 1$, $\varepsilon = 0.14$ and the k -minimum mean cycle

$$(\gamma^+ = \{(s_1, a), (a, b), (b, t_1)\}, \gamma^- = \{(s_1, t_1)\})$$

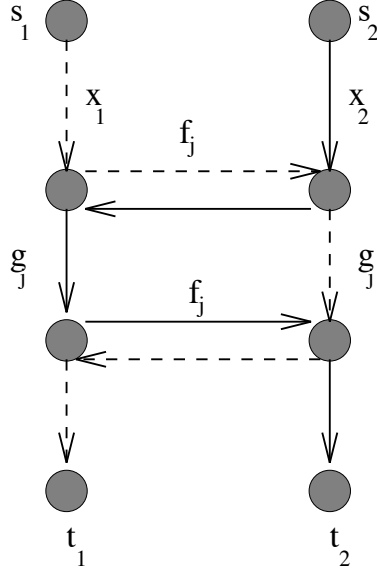


Figure 3: Necessity of differentiability of cost functions.

leads to a better multicommodity flow with cost $f_2 = 7.42$.

Remark 1 *Theorem 1 expresses that the set of optimal solutions for problem (P) is*

$$\mathcal{S} = \{x \in \mathcal{X} : \lambda_k(x) = 0 \text{ for all } k\}$$

An alternative statement is that $x \in \mathcal{X}$ is optimal for (P) if and only if there is no negative k -feasible cycle for any commodity k .

Discussion

In the single commodity case [13], the differentiability of the cost function is not needed and the cost of a cycle were defined as [13]:

$$c(x, \gamma) = \sum_{j \in \gamma^+} f_j^+(\xi_j) - \sum_{j \in \gamma^-} f_j^-(\xi_j) \quad (9)$$

where f_j^+ and f_j^- are respectively the right and the left derivative of f_j .

It is instructive to point out why the differentiability of the cost functions is needed in the multicommodity flow context. First, let's illustrate the necessity of this assumption with the following example with no capacities (see figure 3).

In the example, for horizontal links the cost functions are $f_j(\xi_j) = \xi_j$, while for the vertical links, the cost functions are given by

$$g_j(\xi_j) = \begin{cases} 0 & \text{if } \xi_j \leq 1, \\ 2(\xi_j - 1) & \text{if } \xi_j \geq 1. \end{cases}$$

There are two commodities sharing the network: commodity k ($= 1, 2$) send one unit of flow from s_k to t_k . The optimal multicommodity flow sends both units on the vertical links. Consider now the multicommodity flow $x = (x_1, x_2)$ that sends both units across the horizontal links (x_1 uses the links $-->$ and x_2 the links $--->$. See figure 3). We have $\lambda_1(x) = \lambda_2(x) = 0$ while x is not optimal. That is $\lambda_k(x) = 0$ for all k does not imply the optimality in this case.

The reason relies on the following observation. Suppose the functions f_j are not differentiable. Then, the cost of a cycle is defined by (9), and one may easily show that (2) in this case, is replaced by

$$\begin{cases} f_j^-(\xi_j) - \lambda \leq z_{kj} \leq f_j^+(\xi_j) + \lambda \text{ for all } j \in E_k(x), \\ f_j^-(\xi_j) - \lambda - \alpha_{kj} \leq z_{kj} \leq f_j^+(\xi_j) + \lambda, \alpha_{kj} \geq 0, \text{ for all } j \in \overline{E}_k(x). \end{cases} \quad (10)$$

Hence, for $\hat{x} \in \mathcal{X}$, $\lambda_k(\hat{x}) = 0$ for all k yields with \hat{z}_k , $k = 1 \dots, K$ the corresponding vectors in (10)

$$\hat{z}_{kj} \in \partial f_j(\hat{\xi}_j) \text{ for all } j \in E_k(\hat{x}), \quad \hat{z}_{kj} \leq f_j^+(\hat{\xi}_j) \text{ for all } j \in \overline{E}_k(\hat{x}), \quad (11)$$

which does not imply the Kuhn-Tucker conditions of problem (P) in the non differentiable case:

$$\begin{cases} z_j^* \in \partial f_j(\xi_j^*) & \text{if } \xi_j^* > 0, \\ z_j^* \leq f_j^+(\xi_j^*) & \text{if } \xi_j^* = 0, \\ z_j^* = z_{kj}^* & \text{if } x_{kj}^* > 0, \\ z_j^* \geq z_{kj}^* & \text{if } x_{kj}^* = 0. \end{cases} \quad (12)$$

Indeed, a consequence of (12) is that if two commodities p and k share a link j , their tensions on that arc must be equal. One cannot derive this fact from (11), as for an arc j such that $\hat{x}_{pj} > 0$ and $\hat{x}_{kj} > 0$, \hat{z}_{kj} is not necessary equal to \hat{z}_{pj} . On the other hand, (12) implies (11). The only way to ensure that (11) implies (12) whenever $K > 1$, is to require $\partial f_j(\xi_j)$ to be reduced to a singleton i.e. f_j to be differentiable.

4 The Cancelling Cycles Method

The optimality criterion of the previous section suggests the following algorithmic approach for solving problem (P), which we call the *cancelling cycles method*. We begin with a feasible multicommodity flow $x^0 \in \mathcal{X}$. At each iteration, a commodity k with positive absolute mean is chosen and a k -minimum mean cycle is found with the aim to improve the cost

function and decrease the k -absolute mean. The algorithm terminates when no negative k -feasible cycle can be found for any k . Theorem 1 implies that when the algorithm terminates, it has found an optimal solution for problem (P). The algorithm is sketched below:

Cancelling Cycles Method (CCM)

1. Find an initial feasible solution x^0 in \mathcal{X} . Set $t = 0$.
2. If $\lambda_k(x^t) = 0$, for all k , stop : x^t is optimal.
3. Choose a commodity k with $\lambda_k(x^t) > 0$. Find a k -feasible cycle $\gamma_k \in \Lambda_k(x^t)$ such that $\bar{c}(x^t, \gamma_k) = -\lambda_k(x^t)$ (as described in section 3). Compute

$$\varepsilon_j \text{ such that } \begin{cases} f'_j(\xi_j^t + \varepsilon_j) = f'_j(\xi_j^t) + \lambda_k(x^t) & \text{if } j \in \gamma_k^+ \\ f'_j(\xi_j^t - \varepsilon_j) = f'_j(\xi_j^t) - \lambda_k(x^t) & \text{if } j \in \gamma_k^- \end{cases} \quad (13)$$

Set $\varepsilon = \min[\min_{j \in \gamma_k^+} \varepsilon_j, \min_{j \in \gamma_k^-} x_{kj}^t]$ and perform a k -cancelling cycle step to obtain x^{t+1} as indicated in equations (7).

4. Let $t := t + 1$ and repeat.

A black box is supposed to be available to compute ε_j in (13); in most applications, it is easily computed. The current solution x^t is improved by the k -cancelling step as shown in the first part of proof of theorem 1. Observe that CCM is a primal method as it works on the original problem (P) by searching through the feasible set \mathcal{X} for the optimal solution. Each iterate is feasible and as it is shown in proposition 2 below, the objective function monotonically decreases in the procedure. The algorithm possesses some common features with coordinate descent methods in the sense that at each iteration, a descent step with respect to one commodity is performed in order to compute a new better multicommodity flow.

There is a number of ways to choose the commodity to be changed at a given iteration. We make the assumption that *commodities are examined in a cyclic fashion*, that is commodity x_1 is examined first, then x_2 and so forth through x_K . Then one repeats the process starting with x_1 again. We also assume that every commodity k can be chosen an *infinite* number of times.

In order to obtain our convergence result, we must show that the algorithm produces a sequence $x^t \in \mathcal{X}$ such that every limit point \bar{x} belongs to \mathcal{S} (see remark 1). We first show some preliminary results.

Let k^t and γ_{k^t} be respectively the commodity chosen at iteration t and the corresponding cancelled cycle. For $j \in \gamma_{k^t}^+$, we have $\xi_j^t \leq \xi_j^{t+1} \leq \xi_j^t + \varepsilon_j$ and $f_j'(\xi_j^t + \varepsilon_j) = f_j'(\xi_j^t) + \lambda_{k^t}(x^t)$ by (13). As f_j' is nondecreasing from convexity of f_j we have

$$f_j'(\xi_j^t) \leq f_j'(\xi_j^{t+1}) \leq f_j'(\xi_j^t) + \lambda_{k^t}(x^t) \quad \text{if } j \in \gamma_{k^t}^+ \quad (14)$$

Similarly, one can show that

$$f_j'(\xi_j^t) \geq f_j'(\xi_j^{t+1}) \geq f_j'(\xi_j^t) - \lambda_{k^t}(x^t) \quad \text{if } j \in \gamma_{k^t}^- \quad (15)$$

We can then prove that the k^t -absolute mean decreases after the k^t -cancelling step:

Proposition 2 *After a k^t -cancelling step, we have $f(x^{t+1}) \leq f(x^t)$ and $\lambda_{k^t}(x^{t+1}) \leq \lambda_{k^t}(x^t)$.*

Proof. Since $\xi_j^{t+1} = \xi_j^t$ outside the cycle γ_{k^t} , we have

$$f(x^{t+1}) - f(x^t) = \sum_{j \in \gamma_{k^t}^+} [f_j(\xi_j^{t+1}) - f_j(\xi_j^t)] + \sum_{j \in \gamma_{k^t}^-} [f_j(\xi_j^{t+1}) - f_j(\xi_j^t)]$$

By convexity of each f_j , we have

$$f_j(\xi_j^{t+1}) - f_j(\xi_j^t) \leq \varepsilon f_j'(\xi_j^{t+1}) \leq \varepsilon [f_j'(\xi_j^t) + \lambda_{k^t}(x^t)] \quad \text{for } j \in \gamma_{k^t}^+ \text{ [cf. (14)]}$$

$$f_j(\xi_j^{t+1}) - f_j(\xi_j^t) \leq -\varepsilon f_j'(\xi_j^{t+1}) \leq -\varepsilon [f_j'(\xi_j^t) - \lambda_{k^t}(x^t)] \quad \text{for } j \in \gamma_{k^t}^- \text{ [cf. (15)]}$$

Hence

$$f(x^{t+1}) - f(x^t) \leq \varepsilon \left[\sum_{j \in \gamma_{k^t}^+} f_j'(\xi_j^t) - \sum_{j \in \gamma_{k^t}^-} f_j'(\xi_j^t) \right] + \varepsilon \lambda_{k^t}(x^t) |\gamma_{k^t}| = \varepsilon [c(x^t, \gamma_{k^t}) + \lambda_{k^t}(x^t) |\gamma_{k^t}|] = 0$$

To show that the k^t -absolute mean does not increase, we show that (2) holds for $f_j'(\xi_j^{t+1})$ and z_{k^t} the tension vector associated with x^t in (2). Since x^t and x^{t+1} coincide outside the cycle γ_{k^t} , it suffices to show this property only for $j \in \gamma_{k^t}$.

- $j \in \gamma_{k^t}^+$. We have $f_j'(\xi_j^{t+1}) \geq f_j'(\xi_j^t)$ and $f_j'(\xi_j^t) = z_{k^t j} - \lambda_{k^t}(x^t)$. Thus

$$f_j'(\xi_j^{t+1}) - z_{k^t j} \geq -\lambda_{k^t}(x^t)$$

Also, $f_j'(\xi_j^{t+1}) \leq f_j'(\xi_j^t) + \lambda_{k^t}(x^t)$ by 14. As $f_j'(\xi_j^t) = z_{k^t j} - \lambda_{k^t}(x^t)$ by proposition 1, we have

$$-f_j'(\xi_j^{t+1}) \geq -z_{k^t j} + \lambda_{k^t}(x^t) - \lambda_{k^t}(x^t) \geq -z_{k^t j} - \lambda_{k^t}(x^t) \quad \text{i.e. } z_{k^t j} - f_j'(\xi_j^{t+1}) \geq -\lambda_{k^t}(x^t)$$

- $j \in \gamma_{kt}^-$. We have $f'_j(\xi_j^{t+1}) \leq f'_j(\xi_j^t)$ (first part of (15)), and $-f'_j(\xi_j^{t+1}) \geq -f'_j(\xi_j^t) = -z_{ktj} - \lambda_{kt}(x^t)$ (proposition 1), that is

$$z_{ktj} - f'_j(\xi_j^{t+1}) \geq -\lambda_{kt}(x^t).$$

Again from (15), $f'_j(\xi_j^{t+1}) \geq f'_j(\xi_j^t) - \lambda_{kt}(x^t)$ and from proposition 1, $f'_j(\xi_j^t) = z_{ktj} + \lambda_{kt}(x^t)$. So

$$f'_j(\xi_j^{t+1}) \geq z_{ktj} - \lambda_{kt}(x^t) + \lambda_{kt}(x^t) \geq z_{ktj} - \lambda_{kt}(x^t) \text{ i.e. } f'_j(\xi_j^{t+1}) - z_{ktj} \geq -\lambda_{kt}(x^t)$$

Now, for any cycle $\gamma' \in \Lambda_{kt}(x^{t+1})$, we have

$$c(x^{t+1}, \gamma') = \sum_{j \in \gamma'^+} f'_j(\xi_j^{t+1}) - \sum_{j \in \gamma'^-} f'_j(\xi_j^{t+1}) \geq \sum_{j \in \gamma'^+} z_{ktj} - \sum_{j \in \gamma'^-} z_{ktj} - \lambda_{kt}(x^t) |\gamma'|$$

As by construction (see proposition 1),

$$\sum_{j \in \gamma'^+} z_{ktj} - \sum_{j \in \gamma'^-} z_{ktj} = 0$$

we have $c(x^{t+1}, \gamma') \geq -\lambda_{kt}(x^t) |\gamma'|$ which imply $\lambda_{kt}(x^{t+1}) \leq \lambda_{kt}(x^t)$ □

Note that because the commodities share the same network and as the k^t -cancelling step changes the total flows carried by arcs in the corresponding cycle, this change reflects itself in a change of absolute means of all commodities. An absolute mean of a commodity $p \neq k^t$ may decrease or increase after this k^t -cancelling step, even if $\Lambda_p(x^{t+1}) = \Lambda_p(x^t)$ for $p \neq k^t$.

We are now ready to show our main result.

Theorem 2 *Let $\{x^t\}$ be a sequence of iterates generated by the algorithm CCM. Then, any limit point \bar{x} of $\{x^t\}$ is an optimal solution to (P).*

Proof. Clearly, the sequence $\{x^t\}$ of multicommodity flows generated by CCM is bounded. Let \bar{x} be any limit point of $\{x^t\}$ and let $\{x^t\}_{t \in \mathcal{T}}$ be a subsequence of $\{x^t\}$ converging to \bar{x} . As \mathcal{X} is closed, $\bar{x} \in \mathcal{X}$, and we have $\lim_{t \in \mathcal{T}} f(x^t) = f(\bar{x})$ by continuity of f . If we can show that $\lambda_k(\bar{x}) = 0$ for all k , theorem 1 implies that $\{x^t\}$ converges to an optimal solution of problem (P).

Recall that each commodity is examined cyclically (every K iterations). Then, we can consider a subsequence \mathcal{T}_k associated with a certain k and assume that the k -minimum mean cycle, say γ_k , is the same for $t \in \mathcal{T}_k$ (indeed, each commodity is treated an infinite number of times and the number of cycles is finite). From proposition 2, the k -absolute mean is

reduced an infinite number of times (but may increase between these steps) and is always ≥ 0 . By possibly restricting to a subsequence of \mathcal{T}_k , we may assume that $\{\lambda_k(x^t)\}$ is a decreasing sequence and bounded below by 0 (for all k). Suppose now that $\lambda_k(x^t) \geq \alpha_k > 0$ for each $t > \bar{t}$, $t \in \mathcal{T}_k$. We then have for $t > \bar{t}$ (using the convexity of f_j),

$$f(x^{t+1}) - f(x^t) \leq \varepsilon_t \left[\sum_{j \in \gamma_k^+} f'_j(\xi_j^{t+1}) - \sum_{j \in \gamma_k^-} f'_j(\xi_j^{t+1}) \right] = -\lambda_k(x^{t+1}) \varepsilon_t |\gamma_k|.$$

(The equality comes from the fact that the k -minimum mean cycle is the same for $t > \bar{t}$.) Since $|\gamma_k| \geq 2$,

$$f(x^{t+1}) - f(x^t) \leq -2\alpha_k \varepsilon_t < 0$$

For $t > \bar{t}$ and $t \notin \mathcal{T}_k$, we have $f(x^{t+1}) - f(x^t) \leq 0$ (proposition 2). These relations imply that $f(x^t) \rightarrow -\infty$, contradicting the hypothesis that f is bounded from below on \mathcal{X} (the optimal value is assumed to be finite; see section 2). Hence, $\lambda_k(x^t) \rightarrow 0$ as $t \rightarrow \infty$ for all k .

Fix any k and let γ be a k -feasible cycle at \bar{x} . We have for each $t \in \mathcal{T}_k$ (see proposition 1 and also the proof of the second part of proposition 2)

$$\sum_{j \in \gamma^+} f'_j(\xi_j^{t+1}) - \sum_{j \in \gamma^-} f'_j(\xi_j^{t+1}) \geq \left[\sum_{j \in \gamma^+} z_{kj}^t - \sum_{j \in \gamma^-} z_{kj}^t \right] - \lambda_k(x^t) |\gamma| = -\lambda_k(x^t) |\gamma|$$

Taking the limit as $t \rightarrow \infty$, we have

$$\lim_{t \rightarrow \infty} \left[\sum_{j \in \gamma^+} f'_j(\xi_j^{t+1}) - \sum_{j \in \gamma^-} f'_j(\xi_j^{t+1}) \right] \geq 0$$

And since f'_j is continuous by assumption,

$$\sum_{j \in \gamma^+} f'_j(\bar{\xi}_j) - \sum_{j \in \gamma^-} f'_j(\bar{\xi}_j) \geq 0$$

or equivalently, $\lambda_k(\bar{x}) = 0$, which proves that \bar{x} is an optimal solution by theorem 1. \square

All operations involved in CCM iterations are just addition, subtraction, comparison of rationals, and division of a rational by an integer not exceeding the number of arcs n while computing $\lambda_k(x)$ for a given commodity k (see [11] for example). Hence, one can ask the question of whether there is a polynomial bound in the size of input for all computations in CCM. Monotonic decrease of all absolute means at each iteration is not observed as in [13]. Hence it is not a trivial task to quantify the decrease obtained after some given number of iterations and bound the complexity of the algorithm.

5 A Cancel and Tighten Method (CTM)

We propose in this section a variant of the minimum mean cycle cancelling method which is based on two steps: a cancel one which intends to improve the primal variables by a more flexible selection of cycles for cancelling and another step for improving the dual variables. The idea of the Fast Cancel and Tighten algorithm of [13] (see also [9]) is extended here to derive this method for nonlinear multicommodity flow problems of section 2. Most of the definitions and proofs are inspired from [13].

Consider a multicommodity flow $x = (\xi, x_1, \dots, x_K) \in \mathcal{X}$ and let $z_k \in \mathbb{R}^n$ be any tension vector. For each commodity k , we define the k -distance between x and z_k as

$$\lambda_k(x, z_k) = \max_{j=1}^n \lambda_{kj}(x, z_k),$$

where

$$\lambda_{kj}(x, z_k) = \begin{cases} z_{kj} - f'_j(\xi_j) & \text{if } z_{kj} > f'_j(\xi_j), \\ f'_j(\xi_j) - z_{kj} & \text{if } z_{kj} < f'_j(\xi_j) \text{ and } j \in E_k(x), \\ 0 & \text{if } z_{kj} < f'_j(\xi_j) \text{ and } j \in \overline{E}_k(x), \\ 0 & \text{if } f'_j(\xi_j) = z_{kj} \text{ and } j \in E_k(x), \\ 0 & \text{if } f'_j(\xi_j) \geq z_{kj} \text{ and } j \in \overline{E}_k(x). \end{cases}$$

By definition $\lambda_k(x, z_k) \geq 0$ for all k . We set $\bar{\lambda}_k = \lambda_k(x, z_k)$. An arc j is k -positively far from z_k if $f'_j(\xi) - z_{kj} < -\bar{\lambda}_k/2$, and k -negatively far if $z_{kj} - f'_j(\xi) < -\bar{\lambda}_k/2$. Otherwise, the arc is said to be k -positively close (respectively k -negatively close) to z_k . Note that an arc cannot be k -positively and k -negatively far at the same time. We denote $\mathcal{P}_k(x, z_k)$ (respectively $\mathcal{N}_k(x, z_k)$) the set of k -positively far arcs (respectively k -negatively far arcs) from z_k . A cycle γ is k -far from z_k if $\gamma^+ \subseteq \mathcal{P}_k(x, z_k)$ and $\gamma^- \subseteq \mathcal{N}_k(x, z_k) \cap E_k(x)$. We have

$$f'_j(\xi_j) - z_{kj} \geq -\bar{\lambda}_k \text{ for all } j \text{ and } z_{kj} - f'_j(\xi_j) \geq -\bar{\lambda}_k \text{ for all } j \in E_k(x) \quad (16)$$

Hence by proposition 1, $\bar{\lambda}_k \geq \lambda_k(x)$. Note that, if $\lambda_k = 0$ for all k , then x is optimal for (P) by theorem 1.

The cancel step of CTM consists of cancelling all k -far cycles for each k as in CCM¹. For that purpose, we build for each commodity k , an auxiliary graph \overline{G}_k containing arcs in $\mathcal{P}_k(x, z_k)$ and reverse arcs in $\mathcal{N}_k(x, z_k) \cap E_k(x)$. The graph \overline{G}_k is acyclic if and only if it

¹Note that a k -far cycle is a negative k -feasible cycle if $\lambda_k > 0$

possesses a topological ordering of its nodes [1]. Algorithms for detecting directed cycle or producing a topological ordering of nodes in a linear time, can be found in the literature (see [1] for example). If a topological ordering exists for \overline{G}_k , there is no k -far cycle. Otherwise there is a directed cycle in \overline{G}_k which corresponds to a k -far cycle in G .

The computation of ε_j is somewhat different here :

$$\varepsilon_j \text{ is such that } \begin{cases} f'_j(\xi_j + \varepsilon_j) = z_{kj} & \text{if } j \in \gamma_k^+, \\ f'_j(\xi_j - \varepsilon_j) = z_{kj} & \text{if } j \in \gamma_k^-. \end{cases} \quad (17)$$

The dual step is based on the following proposition (recall that m denotes the number of nodes in the graph G)

Proposition 3 *Let $h_k \in \mathbb{R}^n$ be a vector with integral components that satisfies*

$$|h_{kj}| \leq m - 1 \quad \text{for all } j \quad (18)$$

$$h_{kj} \geq 1 \quad \text{for all } j \in \mathcal{P}_k \quad (19)$$

$$h_{kj} \leq -1 \quad \text{for all } j \in \mathcal{N}_k \quad (20)$$

Then replacing z_k by $z'_k = z_k - (\bar{\lambda}_k/2m)h_k$ reduces $\bar{\lambda}_k$ to $\bar{\lambda}'_k = (1 - 1/2m)\bar{\lambda}_k$ and (16) holds for z'_k and $\bar{\lambda}'_k$

Proof. The proof is similar to the proof of proposition 4.2 in [13]. We analyse each case for an arc j . Suppose j is k -positively close to z_k . Then

$$\begin{aligned} f'_j(\xi_j) - z'_{kj} &= f'_j(\xi_j) - z_{kj} + (\bar{\lambda}_k/2m)h_{kj} \\ &\geq f'_j(\xi_j) - z_{kj} - (m-1)\bar{\lambda}_k/2m \quad \text{by (18)} \\ &= f'_j(\xi_j) - z_{kj} - \bar{\lambda}_k/2 + \bar{\lambda}_k/2m \end{aligned} \quad (21)$$

As $f'_j(\xi) - z_{kj} \geq -\bar{\lambda}_k/2$ for an arc k -positively close to z_k , we have $f'_j(\xi_j) - z'_{kj} \geq -\bar{\lambda}'_k$. We may assume $z_{kj} - f'_j(\xi_j) \geq 0$ else we would deal this in k -negatively close or far from z_k case. Hence,

$$\begin{aligned} z'_{kj} - f'_j(\xi_j) &= (z_{kj} - (\bar{\lambda}_k/2m)h_{kj}) - f'_j(\xi_j) \\ &\geq z_{kj} - f'_j(\xi_j) - (m-1)\bar{\lambda}_k/2m \quad \text{by (18)} \\ &= -(m-1)\bar{\lambda}_k/2m > -\bar{\lambda}'_k \end{aligned}$$

For an arc k -negatively close to z_k , (21) holds and we may assume that $f'_j(\xi_j) - z_{kj} \geq 0$ else we deal this in k -positively close or far from z_k case. Hence

$$f'_j(\xi_j) - z'_{kj} \geq -(m-1)\bar{\lambda}_k/2m > \bar{\lambda}'_k$$

Now, as $z_{kj} - f'_j(\xi_j) \geq -\bar{\lambda}_k/2$, we have

$$\begin{aligned} z'_{kj} - f'_j(\xi_j) &= (z_{kj} - (\bar{\lambda}_k/2m)h_{kj}) - f'_j(\xi_j) \\ &\geq z_{kj} - f'_j(\xi_j) - (m-1)\bar{\lambda}_k/2m \text{ by (18)} \\ &\geq -\bar{\lambda}_k/2 - (m-1)\bar{\lambda}_k/2m = -\bar{\lambda}'_k \end{aligned}$$

Consider an arc j k -positively far from z_k

$$\begin{aligned} f'_j(\xi_j) - z'_{kj} &= f'_j(\xi_j) - z_{kj} + (\bar{\lambda}_k/2m)h_{kj} \\ &\geq f'_j(\xi_j) - z_{kj} + \bar{\lambda}_k/2m \text{ by (19)} \\ &\geq -\bar{\lambda}_k(1 - /2m) \text{ by (16)} \end{aligned}$$

Also, we may assume that $z_{kj} - f'_j(\xi_j) \geq 0$, else we deal this in k -negatively close case.

$$\begin{aligned} z'_{kj} - f'_j(\xi_j) &= (z_{kj} - (\bar{\lambda}_k/2m)h_{kj}) - f'_j(\xi_j) \\ &\geq z_{kj} - f'_j(\xi_j) - (m-1)\bar{\lambda}_k/2m \text{ by (18)} \\ &\geq -(m-1)\bar{\lambda}_k/2m > -\bar{\lambda}'_k \end{aligned}$$

The proof for an arc j k -negatively far from z_k is similar. □

To find the vector h_k , we adapt the trick used in [9]: When all k -far cycles have been cancelled, \bar{G}_k is acyclic, so a topological ordering \mathcal{O} of nodes exists. The label differences $h_{kj} = \mathcal{O}_i - \mathcal{O}_s$, $j = (i, s)$ provide a vector h_k that satisfies conditions of proposition 3. The convergence of CTM is proved as in theorem 2.

6 Numerical Experiments

In this section, we report some computational results with the algorithms developed in the previous sections.

One of the most important performance measure of a data network is the average delay required to transmit a packet from origin to destination. We thus consider *routing problems* in data networks where the delay function associated with a link j is typically of the form [14],[2]:

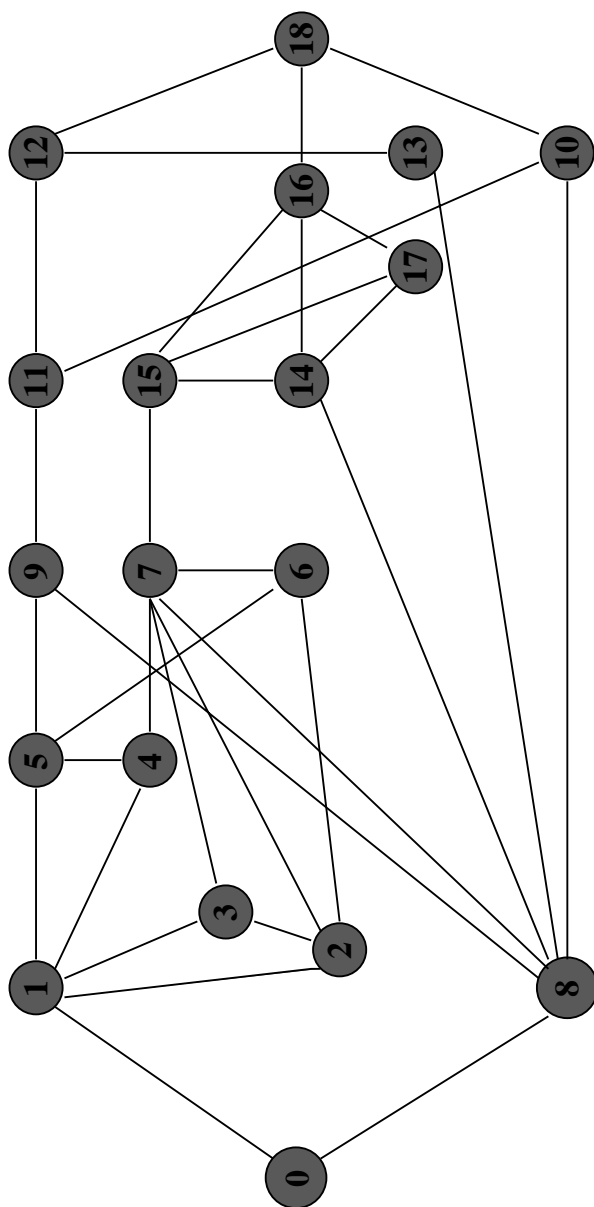
$$f_j(\xi_j) = \frac{\xi_j}{c_j - \xi_j}, \quad \xi_j < c_j. \quad (22)$$

The network shown in figure 4 is an actual network with 19 nodes and 68 arcs: for each pair of nodes, there are two arcs, one in each direction. It is issued from a real world traffic situation which was given by the Centre National d'Etudes des Télécommunications with 30 commodities.

The first derivative function of (22) is given by

$$f'_j(\xi_j) = \frac{c_j}{(c_j - \xi_j)^2}, \quad \xi_j < c_j.$$

Figure 4: The 19 nodes-68 arcs Network.



In the present application, it is easy to compute ε_j in (13) and (17). We used Karp’s algorithm [11] mainly because of it is very simple to implement, even if its average behaviour (known to be close to its worst case bound) is likely to be worse than more recent methods like [12, 23].

Following some preliminary experiments, we found that it is beneficial to perform a line search between ξ^t and $\xi^t + \varepsilon\Gamma(\gamma)$ (indeed $\varepsilon\Gamma(\gamma)$ is a search direction). After a cancelling step, the cycle may stay negative (see example on figure 2). One may look for a step length that maintains feasibility and make the cost of the cycle nonnegative. A way to achieve that is to repeat the process above (with the new multicommodity flow) until the cost of the cycle becomes nonnegative. With this trick, we found that the number of iterations and the running time are reduced significantly.

In the routing problem, the paths flows are not unique although the total link flows are unique because of the strict convexity of the cost function (22) with respect to ξ_j . If individual routes used by each commodity are needed, a flow decomposition algorithm [1] can be used to recover them.

Implementation of our algorithms with the above basic ideas have been written in the C programming language. The algorithms were compiled with the compiler `xlc` with the default code optimization “-O”. All testing was performed on an IBM RISC/System 6000 machine.

We recall that m , n and K denote respectively the numbers of nodes, arcs and commodities.

6.1 Comparison of CCM and CTM

First we compare the two proposed algorithms using the following termination criterion for both of the algorithms CCM and CTM

$$\max_{k=1}^K \lambda_k(x^t) \leq \mu \quad (23)$$

where μ is a positive tolerance parameter. For the results of table 1, the tolerance μ is set to 10^{-3} , then to 10^{-4} . The computational results on table 1 are obtained using the network shown in figure 4. We vary the number of commodities to have the six test problems on table 1.

An *iteration count for a cyclic examination of commodities* (i.e in the order of their numbering) so that the number of iterations in the tables means the number of times we cycle through all commodities. The column headed “Path Dispersion” on the tables gives the maximum number of paths used by a commodity to route its corresponding demand

Table 1: CCM and CTM

$\mu = 10^{-3}$					
K		Cost	CPU Time (in sec.)	# iterations	Path dispersion
5	CCM	2.812025	0.01	1	2
	CTM	2.812048	0.14	5	2
10	CCM	5.017776	0.01	1	2
	CTM	5.017816	0.27	5	2
15	CCM	8.704390	0.03	2	2
	CTM	8.704455	0.46	4	2
20	CCM	12.747241	0.06	5	3
	CTM	12.747290	0.95	8	3
25	CCM	18.436107	0.23	14	4
	CTM	18.436083	2.16	27	4
30	CCM	23.493523	0.51	29	3
	CTM	23.493511	3.77	72	4
$\mu = 10^{-4}$					
K		Cost	CPU Time (in sec.)	# iterations	Path dispersion
5	CCM	2.812025	0.01	1	2
	CTM	2.812026	0.22	8	2
10	CCM	5.017776	0.01	1	2
	CTM	5.017776	0.48	8	2
15	CCM	8.704385	0.04	3	2
	CTM	8.704385	0.88	9	2
20	CCM	12.747178	0.1	8	3
	CTM	12.747178	1.94	15	3
25	CCM	18.435911	0.43	27	4
	CTM	18.435913	5.0	40	4
30	CCM	23.493456	0.72	37	3
	CTM	23.493459	10.47	90	3

while the column “Cost” corresponds to the objective function values computed for the solutions given by the algorithms. The time units reported on all the table are seconds.

As CTM uses a more flexible selection of negative cycles, it is expected to be faster than CCM as in [13] but computational experiments has shown that it is more sensitive to the interaction of commodities as one can observe on tables 1. A modification of the current multicommodity by the way of a given commodity k has an influence on p -distance of the other commodities. Hence even if some p -distance has been cancelled at an iteration, it may increase and the commodity p must be treated again.

As CCM seems to perform better, we consider it for the following experiments.

6.2 Effect of Problem Size and Accuracy parameter

We first examine in this section the performance of CCM for various problem sizes. The three factors $m =$ the number of nodes, $n =$ the number of arcs and $K =$ the number of commodities determine the problem size. To this aim, we use a network generator to have some randomly test generated problems which are listed in increasing order of the number of arcs in table 2. The network generator has been used in [8] and is described in that paper. The results of these runs with $\mu = 10^{-4}$ are reported in table 2. We collect in this table the number of iterations, the CPU time in seconds (to reach the 10^{-4} accuracy) and the maximum number of paths used by a commodity.

Table 2: CCM on Randomly Generated Test Problems with $\mu = 10^{-4}$

Problem	m	n	K	# iterations	CPU Time	Path Dispersion
1	60	280	100	22	10.67	4
2	60	280	500	11	24.7	3
3	60	280	2000	4	40.87	3
4	100	300	120	5	6.63	3
5	100	600	200	22	61.84	4
6	200	800	200	17	175.44	5
7	200	800	500	17	417.83	4
8	200	1000	3000	10	1862.38	3
9	300	1200	1000	16	2263.78	4
10	300	1600	1000	17	2965.43	4
11	300	2000	1000	31	6645.32	8

Recall that the number of iterations is the time we cycle through the commodities. For example for problems 1 and 2 which have the same numbers of nodes and arcs, the k -cancelling step has been made at most $2200 = 22 \times 100$ for the former and at most $5500 = 11 \times 500$ for the latter. As indicated by the CPU times, the routing problem becomes difficult as its size increases. Table 2 shows that the CPU time increases as the number of arcs increases (see problems 9, 10 and 11) and the same observation holds for the number of commodities (problem 1, 2 and 3). On the other hand, the number of nodes does not seem to affect the CPU time (see problems 1 and 5). The time per iteration is of course dominated by the minimum mean cycles computations which depend more on the number of arcs and commodities.

The analysis of these computational results shows promise for CCM to become a competitive algorithm for large-scale routing problems. For example, problem 8 involves $3000 \times 200 + 1000 = 601000$ constraints and about $3000 \times 1000 = 3000000$ variables.

We next examine the effect on efficiency of a tighter accuracy requirement in the final stopping criterion. We use the same basic topology of figure 4 with 30 commodities and the randomly generated problem 1 and 2 (see table 2), and $\mu = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$. In table 3 we report the numbers of iterations and the CPU time required to reach the accuracy μ . These results show that tightening the convergence tolerance lead to much higher time. Note that practitioners are normally satisfied with “near” optimal solutions.

6.3 Comparison with the Flow Deviation Method (FD)

To examine CCM further, we conduct a limited experimentation to compare it with the Flow Deviation method (FD)² [6, 15] which is known as one of the most effective methods for routing problems in data networks. The Flow Deviation method is a special case of the so-called Frank-Wolfe method for solving nonlinear optimization problems with linear constraints. A single iteration of the Flow Deviation method consists of solving shortest path problems between each origin-destination pair (s_k, t_k) to find a search direction and performing a line search procedure. It is then comparable to a single iteration of CCM. We use the same initialization procedure for both algorithms. We have chosen to stop the two algorithms on their proper optimality test. For CCM, as in all the results above, we use the stopping test (23). In the Flow Deviation algorithm, a lower bound on the optimal value is computed at each iteration, and the procedure is stopped when the current objective function value $f(x^t)$ is within some $\mu_{FD} > 0$ of the best lower bound LB found at any iteration:

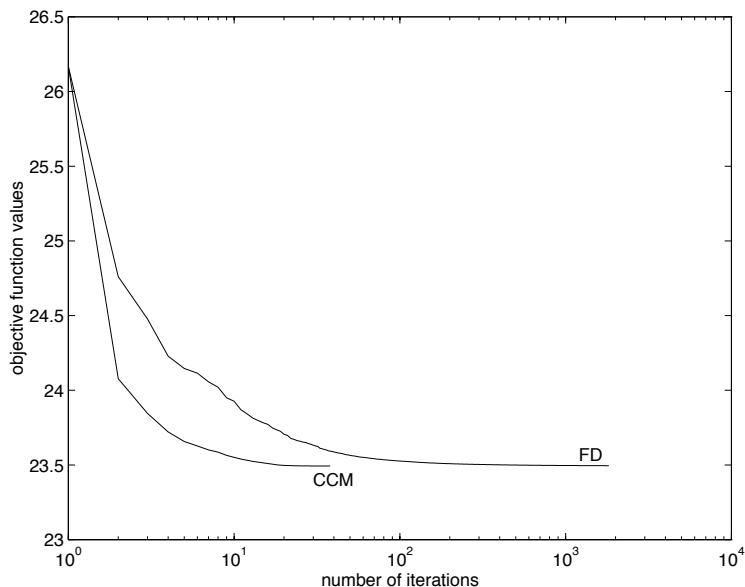
$$\frac{f(x^t) - LB}{LB} \leq \mu_{FD}. \quad (24)$$

²The code of FD was originally written in FORTRAN.

We use the same test problems as in section 6.1 on network of figure 4 in the experiments of this section. The convergence tolerance μ_{FD} is set to 10^{-2} for FD and we use $\mu = 10^{-4}$ for CCM. We have found that with these choices, the objective function values given by the two algorithms differs only slightly.

One can observe from the results on table 4, that the convergence of CCM is quite rapid. We cannot however draw some definitive conclusions from these limited experiments. The results are encouraging for CCM to investigate a more complete comparison with other efficient algorithms for nonlinear multicommodity flow problems of section 2. Figure 5 shows the monotonic decrease of the objective function, as shown in proposition 2. This behaviour is also observed with the Flow deviation algorithm on the same figure. It is obtained using the network of figure 4 with $K = 30$ and semilogarithmic scales from MATLAB (the `semilogx` command). In table 5, we show successive objective function values for the two algorithms. Once the two algorithms come near a solution, they tend to slow down as shown on figure 5 and table 5. This behaviour suggests that both algorithms may be used as part of hybrid solution procedure to initialize some methods using more complex search directions. For the routing problems in data networks that motivates this study, an adequate approximation to an optimal solution is enough in practice.

Figure 5: Objective function values vs number of iterations.



Recall that, in the routing problem, the paths flows are not unique although the total link flows are unique: two different set of paths might yield to the same total flow. FD is

likely to use a large number of paths (the reason is explained in [16] and confirmed by some numerical experiments in [18]). We observe on table 4 that CCM is likely to involve a small number of paths between a source-destination pair in general, increasing the reliability of the system.

CCM needs to store all individual flow vectors; in our code, we use a $K \times n$ array to store them. The memory storage requirement of CCM is then substantial than for FD if the optimal total flow is the only quantity of interest. In this case, FD requires a small amount of storage and thus may be preferable over CCM for very large network problems. When detailed information about the optimal path flows is required, FD needs to store the shortest paths generated at each iteration, and then its storage requirement depends on the number of iterations.

Table 3: Behaviour of CCM w.r.t. μ

μ	# iterations	CPU Time
10^{-1}	1	0.03
10^{-2}	17	0.28
10^{-3}	29	0.51
10^{-4}	37	0.72
10^{-5}	49	0.96
10^{-6}	74	1.34

network of figure 4 with $K = 30$

μ	# iterations	CPU Time
10^{-1}	1	0.83
10^{-2}	9	4.24
10^{-3}	18	8.21
10^{-4}	22	10.67
10^{-5}	29	13.88
10^{-6}	42	19.69

randomly generated problem 1

μ	# iterations	CPU Time
10^{-1}	1	2.11
10^{-2}	3	8.11
10^{-3}	8	18.4
10^{-4}	11	24.7
10^{-5}	12	26.94
10^{-6}	17	37.31

randomly generated problem 2

Table 4: CCM and FD

K		Cost	CPU Time (in sec.)	# iterations	Path dispersion
5	CCM	2.812025	0.01	1	2
	FD	2.812142	0.53	589	2
10	CCM	5.017776	0.01	1	2
	FD	5.018010	0.41	465	2
15	CCM	8.704385	0.04	3	2
	FD	8.705045	0.14	155	2
20	CCM	12.747178	0.1	8	3
	FD	12.747975	0.48	525	4
25	CCM	18.435911	0.43	27	4
	FD	18.437096	1.05	1129	5
30	CCM	23.493456	0.72	37	3
	FD	23.495117	1.72	1822	9

Table 5: Successive iterations of CCM and FD

CCM		FD	
# iterations	Cost	# iterations	Cost
0	26.166667	0	26.166667
10	23.539396	72	23.539316
20	23.496063	1155	23.496065
25	23.493951	1800	23.495135
30	23.493482	1815	23.495123
31	23.493471	1816	23.495122
32	23.493465	1817	23.495121
33	23.493461	1818	23.495120
34	23.493459	1819	23.495120
35	23.4934567	1820	23.4951184
36	23.4934563	1821	23.4951175
37	23.4934561	1822	23.4951166

7 Conclusion

We have presented a new method for nonlinear multicommodity flow problems with convex costs. Our computational experiences quite limited indicate that the proposed method is effective and competitive with one of the best known methods for routing problem in data networks. For our numerical experimentation, due to its simplicity, we use Karp algorithm [11] to find minimum mean cycles. It is well known that the best case of this algorithm is the same as its worst case. Therefore, we may expect the proposed method to be able to solve larger problems using minimum mean cycles algorithms with a better average case behaviour than Karp's one, such as algorithms proposed in [12, 23]. The question of whether there is a polynomial bound (as in [13] for single commodity) in the size of input for all computations is left for further research.

Acknowledgments. We are indebted to Prof. S.T. McCormick for many helpful suggestions and support, and to Prof Larry J. LeBlanc for kindly providing the Flow Deviation code. Prof. J.-Ph. Vial and Dr R. Sarkissian have also gave to the authors access to their generator of test problems. Partial support from the Centre National d'Etudes des Télécommunications (France Telecom), is gratefully acknowledged by the first author.

References

- [1] R.V. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows : Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [2] D.P. Bertsekas and R.G. Gallager , *Data Networks*, Prentice-Hall, 1987.
- [3] D.P. Bertsekas, R. Gendron, W.K. Tsai, "Implementation of an optimal multicommodity network flow algorithm based on gradient projection and a path flow formulation", *Report LIDS-P-1364. Cambridge, MA: MIT Laboratory for Information and Decision Systems*, 1984.
- [4] D.P. Bertsekas, P. Hosein and P. Tseng, "Relaxation methods for network flow problems with convex arc costs", *SIAM J. Control and Opt.* Vol 25, no 5, (1987), pp. 1219-1243.
- [5] J.E. Eckstein, M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers", *Large Scale Optimization : State of the Art Kluwer Academic Publishers B.V.*, 1993, pp. 119-138.
- [6] L. Fratta, M. Gerla and L. Kleinrock, "The flow deviation method : an approach to store-and-forward communication network design", *Networks*, 3(1973), pp. 97-133.
- [7] M. Fukushima, "A nonsmooth optimization approach to nonlinear multicommodity network flow problems", *J. Oper. Res. Soc. of Japan* 27, 2, pp. 151-177, 1984.

- [8] J.-L. Goffin, J. Gondzio, R. Sarkissian, J.P. Vial, "Solving nonlinear multicommodity flow problems by the analytic center cutting plane method", *Math. Prog.* 76(1997) pp. 131-154.
- [9] A.V. Goldberg, R.E. Tarjan, "Finding minimum-cost circulation by cancelling negative cycles", *JACM* 36, no 4, 1989, pp. 873-886.
- [10] M. Gondran, M. Minoux, *Graphes et Algorithmes*, Eyrolles, 1979.
- [11] R.M. Karp, "A characterization of the minimum cycle mean in a digraph", *Discrete Mathematics*, 23, 1978, pp. 309-311.
- [12] R.M. Karp, J.B. Orlin, "Parametric shortest path algorithms with an application to cyclic staffing", *Discrete Applied Mathematics*, 3, 1981, pp. 37-45.
- [13] A.V. Karzanov and S.T. McCormick, "Polynomial methods for separable convex optimization in unimodular linear spaces with applications", *SIAM J. on Computing*, 26, pp. 1245-1275, 1997.
- [14] L. Kleinrock, *Communications, Nets, Stochastic Message Flow and Delay*, Dover, 1972.
- [15] L. LeBlanc, "Mathematical programming algorithms for large scale network equilibrium and network design problems", *Ph.D. Dissertation*, IE/MS Dept, Northwestern University, Evanston IL, 1973.
- [16] P. Mahey, A. Ouorou, L. LeBlanc, J. Chifflet, "A new proximal decomposition algorithm for routing in telecommunication networks", accepted to appear in *Networks*.
- [17] S.T. McCormick "Alternating proof of Lemma 2.2", Working paper September 1994, *Faculty of Commerce and Business Administration, University of British Columbia, Vancouver*.
- [18] A. Ouorou, P. Mahey, J.-Ph. Vial, "A survey of algorithms for convex multicommodity flow problems", *Technical Report*, Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes, Université Blaise Pascal, 1997.
- [19] A. Ouorou, P. Mahey, "Solving linear multicommodity network flow problems with an interior minimum mean cycle cancelling method", *Technical Report*, Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes, Université Blaise Pascal, 1998.
- [20] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*, Topics in Transportation, VSP, Utrecht, The Netherlands, 1994.
- [21] R. T. Rockafellar, *Convex Analysis*, Princeton Univ. Press, 1970.
- [22] G.L. Schultz, R.R. Meyer, "An interior point method for block angular optimization", *SIAM J. Optimization*, Vol. 1, no 4, 1991, pp. 583-602.
- [23] Young N.E., R.E. Tarjan, J.B. Orlin, "Faster parametric shortest path and minimum balance algorithms", *Networks*, 21, pp. 205-221, 1991.