A Tutorial on Max-Min Fairness and its Applications to Routing, Load-Balancing and Network Design

Dritan Nace*
Laboratoire Heudiasyc UMR CNRS 6599
Université de Technologie de Compiègne,
60205 Compiègne Cedex, France
nace@utc.fr
and
Michal Pioro†
Institute of Telecommunications
Warsaw University of Technology, Poland,
Department of Communication Systems
Lund University, Sweden
mpp@tele.pw.edu.pl

May 6, 2006

Abstract

This tutorial is devoted to the notion of Max-Min Fairness (MMF), associated optimization problems, and their applications to multi-commodity flow networks. We first introduce a theoretical background for the MMF problem and discuss its relation to lexicographic optimization. We next present resolution algorithms for the MMF optimization, and then give some applications to telecommunication networks, more particularly to routing, load-balancing, and network design.

1 Introduction

In this tutorial on Max-Min Fairness (MMF) we focus on applications MMF to multi-commodity flow networks. Generally speaking, MMF is applicable in numerous areas where it is desirable to achieve an equitable distribution of certain resources shared by competing demands and is therefore closely related to max-min or min-max optimization problems, widely studied in literature. With respect to applications of MMF to telecommunication networks, a lot of

*Supported by France Telecom R&D.

†Supported by Polish State Committee for Scientific Research (grant no. 3 T11D 001 27 "Design Methods for NGI Core Networks"), and Swedish Research Council (grant no. 621-2003-2767 "Design of Modern Telecommunication Networks—Multicommodity Flow Optimization Approach").

work related to rate adaptation and congestion control in TCP (*Transmission Control Protocol*) networks (see [6, 10, 12, 24, 31, 33] etc.), has already been done. Still, it is our conviction that not enough has been done to understand the relations of MMF with routing and network design. Hence, we first introduce the theoretical background of the MMF notion and its relations with lexicographic optimization, and then, after a brief description of the state of art on MMF, we present in a greater detail its applications to routing, load-balancing, and network design in telecommunication networks.

1.1 A MMF routing problem

Probably the best known problem in communication network traffic engineering related to the notion of max-min fairness is the routing problem studied by, among others, Bertsekas and Gallager in Section 6.4.2 of their book "Data Networks" [4]. The problem, referred to as MMF/SRP (MMF Simple Routing Problem) in the sequel, deals with fair bandwidth allocation among connections in a packet network with given link capacities.

The problem is as follows. Consider a network composed of V nodes, E links with given capacities c_e $(e = 1, 2, ..., E)$, and a set of D connections (demands). Each connection d $(d = 1, 2, ..., D)$ is assigned a predefined path P_d between its end nodes; δ_{ed} is a binary coefficient indicating whether link e belongs to path P_d $(\delta_{ed} = 1)$ or not $(\delta_{ed} = 0)$ (i.e., whether or not $e \in P_d$). Now let x_d denote the bandwidth allocated to path P_d and $\boldsymbol{x} = (x_1, x_2, ..., x_D) \in \mathbb{R}^D$ be the corresponding allocation vector. We are interested in finding a feasible allocation vector $\boldsymbol{x}$ (the set of all feasible allocation vectors will be denoted by X) which is *fair*.

Certainly, vector $\boldsymbol{x}$ is feasible if $\boldsymbol{x} \geq \boldsymbol{0}$ and

$$\sum_{d=1}^{D} \delta_{ed} x_d \leq c_e, \quad e = 1, 2, ..., E \tag{1}$$

(constraint (1) is called the capacity constraint and assures that for any link e its load does not exceed its capacity). Yet, it is not at all obvious, how to define a fair flow allocation vector $\boldsymbol{x}$. We think, however, that the reader will have no doubts that the solution $\boldsymbol{x}^0$ given by the following algorithm will be fair according to the common sense.

Algorithm 1 (MMF/SRP)

Input: Number of links E and connections D, coincidence coefficients $(\delta_{ed} : e = 1, 2, ..., E,\ d = 1, 2, ..., D)$.
Output: Fair solution $\boldsymbol{x}^0 \in X$.

- **Step 1:** Set $\boldsymbol{x}^0 = \boldsymbol{0}$ and $k = 0$ (k is the iteration counter).
- **Step 2:** Put $k := k + 1$. Set $\tau = \min \{ \ c_e / \sum_{d=1}^{D} \delta_{ed} : \ e = 1, 2, ..., E \ \}$ and put $c_e := c_e - \tau \times (\sum_{d=1}^{D} \delta_{ed})$ for $e = 1, 2, ..., E$. Put $x_d^0 := x_d^0 + \tau$ for $d = 1, 2, ..., D$. Remove all saturated links (i.e., all links with $c_e = 0$). Together with each removed link e, remove all connections d for which their paths P_d use the removed link (i.e., all d with $\delta_{ed} = 1$). Denote the set of the removed connections by $\mathcal{D}_k$.

- **Step 3:** If there are no connections left then stop, otherwise go to Step 2.

The above algorithm starts from the zero allocation and uniformly increases the individual allocations until one (or more) links gets saturated. Then the connections that cannot be improved are removed from the network, link capacities modified (decreased), and the process continues for the remaining connections. Thus, after the first execution of Step 2, the current vector $\boldsymbol{x}^0$ allocates simultaneously as much bandwidth as possible to *all* connections. Then, the connections for which it is not possible to further increase the allocated bandwidth are removed, and the process continued because in general it is still possible to increase the bandwidth for a subset of connections (for those which do not use saturated links). In the final solution the set $\mathcal{D}_1$ of the most "handicapped" connections (i.e., connections which can get the least bandwidth) will be assigned the maximum they can get, then the set $\mathcal{D}_2$ of the next most handicapped connections will get the maximum, and so on. Note that the value of counter k returns the number of times Step 2 is executed, and that there are exactly k distinct values in the final vector $\boldsymbol{x}^0$.

In fact, Bertsekas and Gallager introduce MMF/SRP in a quite different way. They define an allocation vector $\boldsymbol{x}^0 \in X$ to be max-min fair if it is not possible to increase the allocated bandwidth x_d^0 of any connection d (connections are called sessions in [4]) only at the expense of connections whose allocated bandwidths are greater than x_d^0, i.e., such an increase is possible only if some connections with the allocated bandwidth less or equal to x_d^0 are decreased. More precisely, an allocation vector $\boldsymbol{x}^0$ is said to be max-min fair in X if $\boldsymbol{x}^0 \in X$ and it fulfills the following property.

Property 1. *For any allocation vector $\boldsymbol{x} \in X$ and for any connection d such that $x_d > x_d^0$ there exists a connection d' such that $x_{d'} < x_{d'}^0 \leq x_d^0$.*

MMF/SRP has several important properties which are directly implied by the construction of $\boldsymbol{x}^0$ in Algorithm 1. Below we list three of them.

- The optimal solution $\boldsymbol{x}^0$ is unique.
- Property 1.
- For each connection d there exists a saturated (bottleneck) link e on path P_d (i.e., $e \in P_d$ and $c_e = \sum_{d=1}^{D} \delta_{ed} x_d^0$) such that x_d^0 is as at least as large as the bandwidth allocated to any other connection using link e. Note that this means that the number of times Step 2 is executed is at most E, i.e., $k \leq E$.

In the sequel we will introduce a general MMF optimization problem which will generalize MMF/SRP, and present algorithms for its various versions, including convex and non-convex formulations. We will give in Chapter 3 an example illustrating how the such algorithms, including MMF/SRP, are applied to routing in networks. We will also present more network applications and some numerical results.

1.2 General formulations of the MMF optimization problem

There are several ways for formally introducing the idea of max-min fairness. Here we choose to do this through the use of the notion of lexicographical order of ordered outcome vectors, i.e., the way followed in [28, 40, 42] and others.

Consider a vector of real-valued functions defined on an arbitrary set $X \subseteq \mathbb{R}^n$ of real n-vectors $\boldsymbol{x} = (x_1, x_2, ..., x_n)$:

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})) \text{ where } f_j : X \to R, \ j = 1, 2, ..., m. \tag{2}$$

Vector $\boldsymbol{f}$ will be referred to as a vector of outcomes, objectives, or criteria.

Definition 1. *Vector $\boldsymbol{y} \in \mathbb{R}^m$ is called lexicographically greater than vector $\boldsymbol{z} \in \mathbb{R}^m$, $\boldsymbol{y} \succ \boldsymbol{z}$, if there exists $j \in \{1, ..., m\}$ such that $y_i = z_i$, for all $i \in \{1, ..., j-1\}$ and $y_j > z_j$. If $\boldsymbol{y} \succ \boldsymbol{z}$ or $\boldsymbol{y} = \boldsymbol{z}$ then we write $\boldsymbol{y} \succeq \boldsymbol{z}$.*

Definition 2. *The lexicographical maximization problem (LXM/P) for given X and $\boldsymbol{f}$ is denoted by*

$$\boldsymbol{lexmax}_{\,\boldsymbol{x} \in X} \ \boldsymbol{f}(\boldsymbol{x}), \tag{3}$$

and consists in finding a vector $\boldsymbol{x}^0$ for which $\boldsymbol{f}(\boldsymbol{x}^0)$ is lexicographically maximal over X. This means that for every vector $\boldsymbol{x} \in X$, $\boldsymbol{f}(\boldsymbol{x}^0)$ is lexicographically greater than or equal to $\boldsymbol{f}(\boldsymbol{x})$: $\forall \ \boldsymbol{x} \in X, \ \boldsymbol{f}(\boldsymbol{x}^0) \succeq \boldsymbol{f}(\boldsymbol{x})$.

A natural way of solving LXM/P is to first maximize $f_1(\boldsymbol{x})$ over X, then, denoting the maximum value of $f_1(\boldsymbol{x})$ with f_1^0, maximizing $f_2(\boldsymbol{x})$ for all vectors $\boldsymbol{x} \in X$ such that $f_1(\boldsymbol{x}) \geq f_1^0$, and so on. The resolution algorithm for LXM/P is as follows.

Algorithm 2 (General algorithm for LXM/P)

Input: Optimization space X and criteria $\boldsymbol{f}$.
Output: Solution $\boldsymbol{x}^0 \in X$ and the optimal (lexicographically maximal) criterion vector $\boldsymbol{f}^0 = (f_1^0, f_2^0, ..., f_m^0)$.

- **Step 1:** Set $j = 1$ and $X_1 = X$.
- **Step 2:** Solve the following single-objective problem

$$max_{\ \boldsymbol{x} \in X_j} \ f_j(\boldsymbol{x}) \tag{4}$$

  and denote an optimal solution and the optimal solution value by $\boldsymbol{x}^0 \in X_j$ and f_j^0, respectively. If $j = m$ then stop: $(\boldsymbol{x}^0, (f_1^0, f_2^0, ..., f_m^0))$ is an optimal solution to LXM/P.
- **Step 3:** Set $X_{j+1} = X_j \cap \{\boldsymbol{x} : f_j(\boldsymbol{x}) \geq f_j^0\}$ and $j = j + 1$. Go to Step 2.

In LXM/P, maximization of the first outcome f_1 is the most important, maximization of the second outcome f_2 is the next important, and so on. Consequently, the optimization scheme given in the above algorithm is quite simple.

For the MMF optimization problems, i.e., for the optimization problems considered in this tutorial, the resolution algorithms are more complicated because we do not assume any priority of the objectives. We rather treat them equally and try to first maximize the minimum outcome (or all the minimum outcomes, if there are more than one), whichever it is. Hence, to first step in the MMF optimization is to solve the problem:

$$max_{\ (\boldsymbol{x}, \tau) \in X'} \ \tau, \tag{5}$$

where $X' = \{(\boldsymbol{x}, \tau) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in X,\ f_j(\boldsymbol{x}) \geq \tau,\ j = 1, 2, ..., m\}$. (The reader should recognize the difference between problem (5) and the first problem encountered by Algorithm 2.)

After solving problem (5) there may be room for further increasing some of the outcomes (but not all). How to do this in general is, however, not obvious and the notion of max-min fairness optimization has to be formally introduced for this purpose. Let $\langle \boldsymbol{y} \rangle = (\langle y \rangle_1, \langle y \rangle_2, ..., \langle y \rangle_m)$ denote a version of vector $\boldsymbol{y} = (y_1, y_2, ..., y_m) \in \mathbb{R}^m$ ordered in the non-decreasing order (i.e., for some permutation φ on the set $\{1, 2, ..., m\}$ it holds that $\langle y \rangle_j = y_{\varphi(j)}$ for $j = 1, 2, ..., m$ and $\langle y \rangle_1 \leq \langle y \rangle_2 \leq ... \leq \langle y \rangle_m$).

Definition 3. *The max-min fairness optimization problem (MMF/OP) for given X and $\boldsymbol{f}$ is as follows*

$$\textbf{\textit{lexmax}}_{\,\boldsymbol{x} \in X}\ \langle \boldsymbol{f}(\boldsymbol{x}) \rangle. \tag{6}$$

Hence, MMF/OP consists in lexicographical maximization of the ***sorted outcome vector*** $\boldsymbol{f}(\boldsymbol{x})$ *over X. Any optimal solution vector $\boldsymbol{x}^0 \in X$ of (6) is called max-min fair on set X with respect to criteria $\boldsymbol{f}$.*

At this stage it should be obvious to the reader that the optimal objective τ^0 of problem (5) yields the smallest entry (outcome) of the optimal solution to MMF/OP (6), as it has been already suggested.

Let $\phi : \mathbb{R} \to \mathbb{R}$ be an increasing function and let $f'_j = \phi \circ f_j, j = 1, 2, ..., m$. Then the following proposition holds.

Proposition 1. *Some $\boldsymbol{x}^0 \in X$ is MMF on set X with respect to criteria $\boldsymbol{f}$ if, and only if, $\boldsymbol{x}^0$ is MMF on X with respect to $\boldsymbol{f}'$.*

The above result will be useful when dealing with non-linear link load functions in the network load balancing problem defined in Chapter 2.

Let us now turn back to problem MMF/SRP considered in the previous section. Using the general notion of MMF/OP (3), problem MMF/SRP takes the following form:

$$\textbf{\textit{lexmax}}_{\,\boldsymbol{x} \in X}\ \langle (x_1, x_2, ..., x_D) \rangle. \tag{7}$$

It can be easily shown (see Section 8.1.2 in [42]) that Algorithm 1 does indeed solve problem (7). It turns out, however, that general resolution algorithms for MMF/OP are not so simple as Algorithm 1, as we will soon learn. In fact, the presentation of such algorithms and their applications to a class of routing problems in communication networks is the main purpose of this tutorial.

1.3 An alternative definition of MMF

The notion of MMF given in Definition 3 is general, i.e., applicable to all sets X and criteria $\boldsymbol{f}$. Some authors, however, adopt a less general (and in our opinion less intuitive) definition of MMF related to Property 1 ([4, 35, 46]).

Definition 4. *A vector $\boldsymbol{y}^0$ is max-min fair on set $Q \subseteq \mathbb{R}^m$ if, and only if, $\forall\, \boldsymbol{y} \in Q,\ \exists\, k \in \{1, ..., m\},\ (y_k > y_k^0) \Rightarrow (\exists\, j \in \{1, ..., m\},\ y_j < y_j^0 \leq y_k^0)$.*

It can be shown (see Section 2.1 in [46]) that if an MMF vector (in the sense of Definition 4) exists then it is unique.

The main disadvantage of Definition 4 is that an MMF vector does not necessarily exist for a general set Q. We will see such examples in the balance of this tutorial.

On the other hand, for the important case of convex sets Q, an MMF vector in the sense of Definition 4 always exists. In fact, in this case definitions 3 and 4 are equivalent. Moreover, the two definitions are always equivalent when the solution of problem (6) is unique.

An observant reader has probably already noticed that we formally cannot directly relate definitions 3 and 4 since they deal with different objects (Definition 3 considers the criterion function $\boldsymbol{f}$ while Definition 4 deals with a vector of variables $\boldsymbol{y}$ instead). To make the two definitions comparable we define a set $Q(X, \boldsymbol{f}) \subseteq R_m$ of vectors $\boldsymbol{y} = (y_1, y_2, ..., y_m)$:

$$Q(X, \boldsymbol{f}) = \{\boldsymbol{y} : \ \boldsymbol{y} \leq \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x} \in X \ \} \tag{8}$$

(where $\boldsymbol{y} \leq \boldsymbol{f}(\boldsymbol{x})$ denotes that $y_j \leq f_j(\boldsymbol{x})$ for $j = 1, 2, ..., m$), and reformulate problem (6) as follows:

$$lexmax_{\ \boldsymbol{y} \in Q(X, \boldsymbol{f})} \ \langle \boldsymbol{y} \rangle. \tag{9}$$

It should be clear to the reader that problems (6) and (9) are equivalent.

We note that set $Q(X, \boldsymbol{f})$ is convex whenever set X is convex and functions $f_j, j = 1, 2, ..., m$ are concave. It can be proved that in such a case the two definitions are equivalent, i.e., there exists a vector $\boldsymbol{x}^0 \in X$ and a vector $\boldsymbol{y}^0 \in Q(X, \boldsymbol{f})$ such that $\boldsymbol{x}^0$ is MMF with respect to criteria $\boldsymbol{f}$ on X according to Definition 3, and $\boldsymbol{y}^0$ is MMF on $Q(X, \boldsymbol{f})$ according to Definition 4, and that $\boldsymbol{f}(\boldsymbol{x}^0) = \boldsymbol{y}^0$. Moreover, vector $\boldsymbol{y}^0$ is unique. We shall consider the case of convex $Q(X, \boldsymbol{f})$ in the next section.

To further illustrate the relation between the definitions of MMF consider the routing problem MMF/SRP (considered in Section 1.1) for a simple network composed of just one link ($E = 1$) with capacity $c_1 = 1$, and two distinct demands ($D = 2$, $d = 1, 2$) between the end nodes of the link. The solution of MMF/SRP is clearly $x_1^0 = x_2^0 = 0.5$, and the resulting vector $\boldsymbol{x}^0$ is MMF according to both definitions. Still, if we assume integrality of the demand flows (i.e., that the flows must be integers) we arrive at two different MMF vectors according to Definition 3: $\boldsymbol{x}' = (1, 0)$ and $\boldsymbol{x}'' = (0, 1)$. None of these vectors is MMF according to Definition 4. (In the integral case the solution space is not convex and the MMF vector according to Definition 4 does not exist at all.)

Finally, we notice that the MMF vector $\boldsymbol{x}^0$ on set X according to Definition 3 is called *leximin maximal* on X in [46].

1.4 Convex MMF optimization problems

Throughout this subsection we shall assume that the set X is convex and that all the criteria functions $f_j(\boldsymbol{x}), j = 1, 2, ...m$, are concave. With these assumptions the problem MMF/OP will be called convex MMF/OP (MMF/CXOP in short). As we shall see the assumed convexity/concavity will ensure that all the single-objective optimization sub-problems encountered in the sequel of this section are convex. The convexity/concavity assumption is quite strong; in particular it implies the following proposition.

Proposition 2. *Suppose X is convex and $\boldsymbol{f}$ are concave, and let $\boldsymbol{x}'$ and $\boldsymbol{x}''$ be two different optimal solutions of MMF/OP (6). Then*

$$\boldsymbol{f}(\boldsymbol{x}') = \boldsymbol{f}(\boldsymbol{x}''). \tag{10}$$

Proposition 2 states that for the convex case not only $\langle \boldsymbol{f}(\boldsymbol{x}') \rangle = \langle \boldsymbol{f}(\boldsymbol{x}'') \rangle$, which is obvious, but also that the MMF solution is unique in the criterion space. The proposition also implies that the solution $\boldsymbol{y}^0$ of the modified problem (9) is unique, and hence that definitions 3 and 4 are equivalent in the convex case. In the sequel we will denote the unique MMF vector by $\boldsymbol{y}^0 = (y_1^0, y_2^0, ..., y_m^0)$ and its sorted, non-decreasing version by $\boldsymbol{Y}^0 = (Y_1^0, Y_2^0, ..., Y_m^0)$.

Now we shall present a general algorithm for convex MMF/OP. Suppose B is a subset of the index set $M = \{1, 2, ..., m\}$, $B \subseteq M$, and let $\boldsymbol{t}^B = (t_j : j \in B)$ be a $|B|$-element vector. Also, let B' denote the set complementary to B : $B' = M \setminus B$. For given B and $\boldsymbol{t}^B$ we define the following (convex!) mathematical programming problem in variables $\boldsymbol{x}$ and τ:

$\mathcal{P}(B, \boldsymbol{t}^B)$:

$$\begin{aligned}
&\mathit{maximize} && \tau && && \text{(11a)}\\
&\mathit{subject\ to} && f_j(\boldsymbol{x}) \geq \tau && j \in B' && \text{(11b)}\\
& && f_j(\boldsymbol{x}) \geq t_j^B && j \in B && \text{(11c)}\\
& && \boldsymbol{x} \in X. && && \text{(11d)}
\end{aligned}$$

It is clear that the solution τ^0 of problem $\mathcal{P}(\emptyset, \emptyset)$ defined by (11) for empty set B and empty sequence $\boldsymbol{t}^B$ will yield the smallest value of $\boldsymbol{Y}^0$, i.e., the value Y_1^0 (and possibly some other consecutive entries of $\boldsymbol{Y}^0$). This observation suggests the following algorithm for solving the convex version of problem MMF/OP (6).

Algorithm 3 (General algorithm for MMF/CXOP)

Input: Convex 0ptimization space X and concave criteria $\boldsymbol{f}$.
Output: MMF solution $\boldsymbol{x}^0 \in X$ and the (optimal) non-sorted MMF criterion vector $\boldsymbol{t}^B$.

- **Step 1:** Set $B = \emptyset$ (empty set) and $\boldsymbol{t}^B = \emptyset$ (empty sequence).
- **Step 2:** If $B = M$ then **stop** ($\langle \boldsymbol{t}^B \rangle$ is the optimal solution of problem MMF/OP, i.e., $\langle \boldsymbol{t}^B \rangle = \boldsymbol{Y}^0$). Else, solve $\mathcal{P}(B, \boldsymbol{t}^B)$ and denote the resulting optimal solution by $(\boldsymbol{x}^0, \tau^0)$.
- **Step 3:** For each index $k \in B'$ such that $f_k(\boldsymbol{x}^0) = \tau^0$ solve the following test problem $\mathcal{T}(B, \boldsymbol{t}^B, \tau^0, k)$:

$$\begin{aligned}
&\mathit{maximize} && f_k(\boldsymbol{x}) && && \text{(12a)}\\
&\mathit{subject\ to} && f_j(\boldsymbol{x}) \geq \tau^0 && j \in B' \setminus \{k\} && \text{(12b)}\\
& && f_j(\boldsymbol{x}) \geq t_j^B && j \in B && \text{(12c)}\\
& && \boldsymbol{x} \in X. && && \text{(12d)}
\end{aligned}$$

If for optimal $\boldsymbol{x}'$, while solving test $\mathcal{T}(B, \boldsymbol{t}^B, \tau^0, k)$ we have $f_k(\boldsymbol{x}') = \tau^0$ (i.e., when criterion $f_k(\boldsymbol{x})$ cannot be further increased), then we put $B := B \cup \{k\}$ and $t_k^B := \tau^0$.

- **Step 3:** Go to Step 2.

It can happen that as a result of solving the test in Step 3 for some index $k \in B'$, it will turn out that $f_l(\boldsymbol{x}') > \tau^0$ for some other, not yet tested, index $l \in B'$ $(l \neq k)$. In such an (advantageous) case, the objective function with index l does not have to be tested, as its value can be further increased without disturbing the maximal values $\boldsymbol{t}^B$.

Observe that set B is the current set of blocking indices, i.e., the indices j for which the value $f_j(\boldsymbol{x}^0)$ is equal to t_j^B in every optimal solution of problem MMF/OP. Note also that although the tests in Step 3 are performe separately for individual indices $k \in B'$, the values of objective functions f_k for the indices $k \in B'$ (where set B' results from Step 2) can be increased *simultaneously* above the value of τ^0 in the next execution of Step 2. This follows from convexity of the set defined by constraints (11b-d): if $f_j(\boldsymbol{x}^k) = a^k > \tau^0$ and $\boldsymbol{x}^k$ satisfies (11b-d), then a convex combination of the vectors $\boldsymbol{x}^k$, $\boldsymbol{x} = \sum_{k \in B'} \alpha^j \boldsymbol{x}^k$ $(\sum_{k \in B'} \alpha^k = 1,\ \alpha^k > 0,\ k \in B')$ also satisfies (11b-d), and $f_k(\boldsymbol{x}) > \tau^0$ for all $k \in B'$.

It is natural to ask what is the relation of the general algorithm given above and Algorithm 1 for problem MMF/SRP in Section 1.1. Certainly, MMF/SRP is a convex MMF problem so Algorithm 3 applies to it. It should be clear to the reader that the value of τ computed (in a direct way) in Step 2 of Algorithm 1 is just equal to τ^0, and hence it is an optimal solution of a consecutive problem $\mathcal{P}(B, \boldsymbol{t}^B)$ in Step 2 of Algorithm 3. On the other hand, the general tests $\mathcal{T}(B, \boldsymbol{t}^B, \tau^0, k)$ are not used in Algorithm 1 since in the case of MMF/SRP they become very simple: we test the saturation of links instead.

A slightly different version of Algorithm 3 is given on page 8 in [40] (see also Remark 8.2 in [42]). In this version the test of Step 3 is performed only until the first non-blocking $k \in B'$ is detected—then the algorithm goes immediately back to Step 2. The main idea behind such a modification is that in this way the number of tests run in Step 3 of Algorithm 3 can in many cases be decreased, leading to an overall improved efficiency of the MMF solution algorithm (provided that the complexity of problems (11) and (12) is similar).

1.5 Linear MMF optimization problems

Whether modified or not, Algorithm 3 presented in the previous section can be time consuming due to an excessive number of instances of the problems (11) and (12) that have to be solved in the iteration process. On the other hand, we have already seen in Algorithm 1 that in a particular case of the routing problem MMF/SRP, the tests (12) may become very easy to perform. Below we shall show that in general the tests $\mathcal{T}(B, \boldsymbol{t}^B, \tau^0, k)$ can be made much more efficient provided that optimal dual variables for problems $\mathcal{P}(B, \boldsymbol{t}^B)$ can be effectively computed.

Although the following results derived for dual variables hold for general convex MMF problems (i.e., for MMF/CXOP), the most important application of their use is LP since only for LP the optimal dual variables are readily obtained (e.g., by using the simplex method).

Suppose $\boldsymbol{\lambda} = (\lambda_j : \ j \in B')$ denotes the vector of dual variables (multipliers) associated with constraints (11b). This leads to the following Lagrangean function for problem $\mathcal{P}(B, \boldsymbol{t}^B)$:

$$L(\boldsymbol{x}, \tau; \boldsymbol{\lambda}) = -\tau + \sum\nolimits_{j \in B'} \lambda_j(\tau - f_j(\boldsymbol{x})) = (\sum\nolimits_{j \in B'} \lambda_j - 1)\tau - \sum\nolimits_{j \in B'} \lambda_j f_j(\boldsymbol{x}). \tag{13}$$

The domain of Lagrangian (13) is defined by

$$\boldsymbol{x} \in Z \text{ (where } Z \text{ is determined by constraints (12c} - \text{d))} \tag{14a}$$

$$-\infty < \tau < +\infty \tag{14b}$$

$$\boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{14c}$$

Hence, the dual function is formally defined as

$$W(\boldsymbol{\lambda}) = \min\nolimits_{\tau, \boldsymbol{x} \in Z} L(\boldsymbol{x}, \tau; \boldsymbol{\lambda}), \ \boldsymbol{\lambda} \geq \boldsymbol{0} \tag{15}$$

and the dual problem reads:

$$\textit{maximize } W(\boldsymbol{\lambda}) \ \textit{ over } \ \boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{16}$$

The following proposition can be proved [42].

Proposition 3. *Let $\boldsymbol{\lambda}^0$ be the vector of optimal dual variables solving the dual problem (16). Then*

$$\sum\nolimits_{j \in B'} \lambda_j^0 = 1 \tag{17}$$

and if $\lambda_j^0 > 0$ for some $j \in B'$, then $f_j(\boldsymbol{x})$ cannot be improved, i.e., $f_j(\boldsymbol{x}^0) = \tau^0$ for every optimal primal solution $(\boldsymbol{x}^0, \tau^0)$ of (11).

Note that in general the inverse of the second part of Proposition 3 does not hold: $\lambda_j^0 = 0$ does not necessarily imply that $f_j(\boldsymbol{x})$ can be improved (for an example see [41, 42]).

In fact, it can be proved [42, Chpt. 13] that the inverse implication holds if and only if set B is regular (set B is called *regular* if for any non-empty proper subset G of B, in the modified formulation $\mathcal{P}(B \backslash G, \boldsymbol{t}^{B \backslash G})$ the value of $f_k(\boldsymbol{x})$ can be improved for at least one of the indices $k \in B \backslash G$). Whether or not the consecutive sets B are regular, The following algorithm solves the convex problem MMF/CXOP.

Algorithm 4 (Algorithm for MMF/CXOP based on dual variables)

Input: Convex 0ptimization space X and linear criteria $\boldsymbol{f}$.
Output: MMF solution $\boldsymbol{x}^0 \in X$ and the (optimal) non-sorted MMF criterion vector $\boldsymbol{t}^B$.

- **Step 1:** Set $B = \emptyset$ (empty set) and $\boldsymbol{t}^B = \emptyset$ (empty sequence).
- **Step 2:** If $B = M$ then **stop** ($\langle \boldsymbol{t}^B \rangle$ is the optimal solution of problem MMF/OP, i.e., $\langle \boldsymbol{t}^B \rangle = \boldsymbol{Y}^0$). Else, solve $\mathcal{P}(B, \boldsymbol{t}^B)$ and denote the resulting optimal solution by $(\boldsymbol{x}^0, \tau^0; \boldsymbol{\lambda}^0)$.
- **Step 3:** Set $B := B \cup \{j \in B' : \ \lambda_j^0 > 0\}$. Go to Step 2.

Observe that if for some $j \in B'$ with $\lambda_j^0 = 0$, $f_j(\boldsymbol{x})$ cannot be further improved, then in Step 2 the value of τ^0 will not be improved; still at least one such index j will be detected (due to property (17)) and included into set B in the next execution of Step 3.

Let us now focus on the linear case, that is when criteria $f_j(x)$ are linear ($j = 1, 2, ..., m$), and when the set X is described by a system of linear equations/inequalities. When using Simplex for solving LP problems, the optimal dual variables used in Algorithm 4 can be obtained directly from the simplex tableau. (in fact the Simplex tableau was a basis of early implementations of the MMF solution for LP problems [7, 8, 26]).

Still, as remarked above, we cannot be sure that all blocking/non-blocking indices $j \in B'$ are detected in Step 3 of Algorithm 4. We can handle this issue either by using the methods computing strictly the complementary optimal solution (i.e., when there is only one zero value for each complementary pair of slack variables) or by resolving an auxiliary LP problem as shown in the balance of this section (for details see in [34]). Let us recall that the complementary slackness property, one of the most fundamental property of LP theory, states the conditions relating a pair of complementary optimal solutions, that is, primal and dual solution. Let consider some constraint i with respect to the primal problem and a given solution; let s_i and λ_i denote respectively the corresponding slack and dual variables. The complementary slackness theorem states that some solution is optimal if and only if at least one of variables s_i, λ_i is null. Some solution is called strictly complementary optimal if only one of these variables take zero while the other is strictly positive. To see how this can be achieved we first formulate still another (a more general) algorithm for MMF/CXOP.

Algorithm 5 (Algorithm for MMF/CXOP)

Input: Convex optimization space X and linear criteria $\boldsymbol{f}$.
Output: MMF solution $\boldsymbol{x}^0 \in X$ and the (optimal) non-sorted MMF criterion vector $\boldsymbol{t}^B$.

- **Step 1:** Set $B = \emptyset$ (empty set) and $\boldsymbol{t}^B = \emptyset$ (empty sequence).
- **Step 2:** If $B = M$ then **stop** ($\langle \boldsymbol{t}^B \rangle$ is the optimal solution of problem MMF/OP, i.e., $\langle \boldsymbol{t}^B \rangle = \boldsymbol{Y}^0$). Else, solve $\mathcal{P}(B, \boldsymbol{t}^B)$ and denote the resulting optimal solution by $(\boldsymbol{x}^0, \tau^0)$.
- **Step 3:** Let $Z(B, \boldsymbol{t}^B)$ be the set of all optimal solutions of $\mathcal{P}(B, \boldsymbol{t}^B)$. Put $B := B \cup \{j \in B' : \ f_j(\boldsymbol{x}) = \tau^0 \ \text{ for all } \boldsymbol{x} \in Z(B, \boldsymbol{t}^B)\}$. Go to Step 2.

Finding directly all binding constraints as required in Step 3 of Algorithm 5 could be obtained by LP solvers that compute strictly complementary optimal solutions, as an IPM (Interior Point Method) methods based on the central trajectory approach (see [18]). However, it is possible to achieve strictly complementary solutions by any LP solver using the method given first in [19] and summarized below for the first execution of Step 2 in Algorithm 5.

Let τ^0 be the optimal solution value of problem $\mathcal{P}(B, \boldsymbol{t}^B)$ with B empty. We need to find the binding constraints (i.e., the constraints that are tight for all optimal solutions) among the constraints (11b). This can be accomplished by solving an auxiliary LP problem similar to $\mathcal{T}(B, \boldsymbol{t}^B, \tau^0, k)$. We introduce a new decision variable $\gamma \geq 1$ that multiplies each constraint, and replace $\gamma\boldsymbol{x}$ by $\boldsymbol{y}$ (this can be done when criteria $\boldsymbol{f}$ are linear). Next, we add slack variables s_j that are bounded from above by 1. The resulting problem is as follows:

$$\begin{aligned} &\textit{maximize} && \textstyle\sum_{i=1}^m s_i && \text{(18a)} \\ &\textit{subject to} && f_i(\boldsymbol{y}) - \gamma\tau^0 - s_j \geq 0 \quad j = 1, 2, ..., m && \text{(18b)} \\ & && \gamma \geq 1, \boldsymbol{y} \geq \boldsymbol{0}, 0 \leq s_j \leq 1, \ j = 1, 2, ..., m. && \text{(18c)} \end{aligned}$$

If constraint j is not necessarily tight for the initial problem $\mathcal{P}(B, \boldsymbol{t}^B), (B = \emptyset)$, then we can choose a value of γ large enough in the transformed problem so that $s_j = 1$. So, constraint j for (11b) is not binding if, and only if, $s_j = 1$ in an optimal solution for LP (18a) - (18b). In total, at most $2m - 1$ linear programs are solved throughout this algorithm (at most m linear programs if using an IPM method computing strictly complementary solutions). Finally, the following result holds (see [34] for a detailed proof):

Theorem 1. *The vector $\boldsymbol{t}^B$ obtained at the end of Algorithm 5 is max-min fair and it is obtained in polynomial time.*

Similar results and approaches can be developed for the min-max fair problem. Generally speaking, the latter problem arises when one wants to minimize lexicographically the sorted outcome vector (see [34] for further details). This is called in [46] leximax minimal.

2 Max-min fairness, related works

Historically, among the first related works on lexicographical ordering, we find these concerned with game theory, see for instance [44, 27] and more recently [25]. Particularly worthy of mention is Schmeidler [44], who introduced the notion of lexicographic order when defining the nucleolus of a characteristic function game, as we recall it now. A characteristic function game, as defined in [44, 27], consists of a set $N = \{1, 2, ..., n\}$ of players and a characteristic function $v : 2^n \to \mathbb{R}_+$ that associates a value $v(S) \geq 0$ to every subset (coalition) $S \subseteq N$. The problem is to find a fair allocation of the total gain $v(N)$ among all players $i \in N$. A payoff vector $\boldsymbol{x} \in \mathbb{R}^n$ is defined such that $\boldsymbol{x}_i \geq 0$. For each set $S \subseteq N$, we let $\boldsymbol{x}(S)$ denote $\sum_{i \in S} \boldsymbol{x}_i$ and $\sum_{i \in N} \boldsymbol{x}_i = v(N)$. Last, for any payoff vector $\boldsymbol{x}$, let γ be the vector whose components take on values $v(S) - \boldsymbol{x}(S)$, for all $S \subseteq N$. The nucleolus is then defined as the vector x whose corresponding γ is min-max fair. Obviously, γ can be computed as the min-max fair criterion to the following system, where f_S is given by $v(S) - \sum_{i \in S} x_i$ for all $S \subseteq N, \boldsymbol{x} \geq 0$:

$$f_S \leq \gamma_S \qquad \forall S \subset N \tag{19a}$$

$$f_N = v(N) \tag{19b}$$

In [44], the author has shown the uniqueness and the existence of such an allocation. Clearly, this problem can be resolved by the algorithms given in Chapter 1 applied to min-max fairness instead of max-min fairness. Notice that counterpart of Agorithm 5 for the leximax minimization, that is min-max fairness, problem is given in [34] and an example of min-max fairness will be given when dealing with load balancing in networks. Finally, a similar method to above is also presented in [25].
Another "historical" work is presented in [32] where the author has studied the fair maximum flow problem for single-source multi-sinks networks. This problem can be stated as follows: in a given capacitated network, find the maximal flow from a given source to a set of sinks, such that the amount of flow supplied to sinks is as fair as possible. He has shown that the well-known labeling method for searching augmenting paths proposed for the traditional maximum flow problem is not applicable when, instead of maximizing the flow globally, one needs to distribute it fairly. However, the author has put forward an elegant method for computing maximum fair single-source flow. He showed that the value of the maximum fair single-source flow is equal to this of the maximum flow.

¿From the applications of MMF to networks point of view, two related issues are essentially concerned with. The first and the most widely studied is rate allocation and congestion control for TCP/IP networks. The second issue concerns the routing, load-balancing and network design and it is also the main scope of this paper.

Max-min fairness applied in rate allocation and congestion control for TCP/IP networks. Some questions relating to fairness, given extensive coverage in the literature ([24, 12, 10, 31, 6, 14, 15, 23, 48, 33] etc.) have arisen with world-wide Internet deployment. Although these works are out of the

scope of this tutorial, we think that recalling briefly some of main directions of work on this area should be helpful for the reader to establish the connection with fairness applications to routing and network design. At the origin we find [23], where the authors have studied and proposed congestion control schemes in order to prevent the network from entering congestive collapse[1]. Key issues are the extent to which congestion control mechanisms influence resource sharing among the competing sessions, the fairness of this resource sharing, and the desirability of integrating fairness as a design objective. Let us also recall that the two main pillars of the Internet are the *routing layer* (IP) and the *transport layer* (TCP and UDP). An important result (see [12]) concerning TCP behavior is that the AIMD (*Additive Increase, Multiplicative Decrease*) algorithm[2] (on which TCP is based) converges to an efficient and fair equilibrium point corresponding to *max-min fair* allocation in a network composed of a single bottleneck link with n users sharing it. However, we may note that this result does not hold for networks with multiple bottlenecks. In [24] it is shown that the rate control based on AIMD, achieves proportional fairness. The reality is more complex [48] and this result is not applicable to the Reno version of TCP (used is today's networks) which is not strictly speaking proportionally fair. The Vegas variant of TCP, however, is proportionally fair, and the rate allotted to connections is inversely proportional to the RTT value (*Round Trip Time*). Hence, assuming max-min fair resource allocation obviously does not allow an accurate presentation of the network behavior, but it yields an acceptable degree of approximation which can lead to very useful conclusions for different application settings. We also note that it is by no means a simple matter to model mathematically the exact allocation of bandwidth to competing connections which is entailed by the networks control mechanism (AIMD). It could be therefore interesting to use approximated formulations which allow computationally-tractable models to be defined.
In résumé, too much work is done on the AIMD window-based mechanisms and it is now well understood in terms of fairness, stability, oscillations and other properties. A lot of efforts and work is also done to apply these mechanisms for real-time streaming applications. A relevant example is the attempt to introduce the protocol TCP-friendly, which intends to impose a certain congestion control for non-TCP applications ([14, 15]). A finer approach than AIMD relies on modeling of TCP throughput to directly adjust source's sending rate as a function of packet loss rate and round-trip time. Lastly, there is also a lot of work in studying the MMF on multicast applications for both uni-rate and multi-rate session cases. Finally, let us notice that studying the fairness from control congestion point of view point of view, is quite interesting in establishing some immediate connection between the transport layer and the network layer.

Max-min fairness and multi-commodity flow applications. Max-min fairness applied to routing has been subject of several works this last decade [16, 20, 21, 28, 9, 11, 30, 35, 38, 42, 43], etc. Among them, some work has been

[1]Congestive collapse is the name given to the state that a packet switched computer network can get into when congestion in the network is so bad that almost no useful communication is happening.

[2]Based on the principle of AIMD, a TCP connection probes for extra bandwidth by increasing its congestion window linearly with time, and on detecting congestion, reducing its window multiplicatively by a factor of two.

devoted to the static routing (connections and corresponding routing paths are given) case, where source rates are subject to change (see for instance [4, 16]). In [4] the author presents the *progressive filling* algorithm for achieving a max-min fair distribution of resources to flows for the fixed single path routing case. The principle of the progressive filling algorithm is already used in Algorithm 1. In [16] the authors propose a LP model for the static case and in addition the extension of their algorithm with a heuristic for computing variants of fair routing. Other works [9, 11, 30] examine and compare performances of representative routing algorithms. In the latter studies the authors have tried to find an algorithm with fair properties and/or maximizing the overall throughput. They are all on-line routing algorithms: that is, for each new connection, the source computes in real-time the appropriate path according to local or collected information. In [35, 36], the problem of max-min fair bandwidth-sharing among TCP/IP connections when routing is not fixed has been considered from off-line point of view. This work has led to some complex LP models that loses the linearity when considering fairness levels. The proposed algorithm can be seen as an extension of the Algorithm 1 (MMF/SRP) given in Chapter 1 except that the routing is not fixed and at each iteration a new routing is computed while the previously saturated links and the corresponding fair sharing remain fixed until the end of the algorithm. However it is shown that one can get around this difficulty (that is non-linearity) by fixing the corresponding bandwidth sharing value and updating it continuously and not assuring the polynomiality. Finally, two other approaches, in the spirit of Algorithms 3, 4 and 5, enriched by theoretical analysis have been given in [41, 42, 38, 37]. ¿From an OR theory point of view, the max-min fair routing problem is equivalent to the max-min fair multi-commodity flow problem, which can also be seen as an extension of the concurrent flow problem (see for instance [45]). Recall that the concurrent flow problem [45] is defined as a multicommodity flow problem in a capacitated network $N = (V, A, C)$ and a set of commodities D. For each commodity d, the goal is to send T_d units from the source node o^d to its destination node t^d. If there is no feasible solution, then the objective is to maximize the satisfaction ratio for each commodity, that is, find the largest α such that for each commodity we can send simultaneously αT_d units. In the lexicographically maximum concurrent flow problem, we look for a flow vector that is leximin maximal. This models the situation in which one wants to maximize the worst satisfaction ratio; then one wants to maximize the second worst satisfaction ratio, and so on. Several variants of the fair (multi)flow problem have been presented and analyzed in [1, 4, 32, 37], etc. One can distinguish here two different cases: splittable and unsplittable fair flows with respect to the number of paths used to transfer the flow (each flow supplied to a sink is transported through a single (resp. multiple) path for the unsplittable (resp. splittable) case). Conversely, the fair single-source unsplittable flow problem is shown to be NP-complete: special cases of the latter problem include several fundamental load-balancing problems, [28]. In this latter work, the authors propose an interesting approximated algorithm for the fair unsplittable single-source flow problem. However, in many application contexts (e.g. transport), problems are generally not limited to single-source networks: several commodities are required to share the same underlying network, which is the context that we focus here. Fairness also finds applications in issues relating to load balancing. Some relevant work on balanced networks is presented in [21] where the authors propose an approach for lexicographically-optimal balanced networks. The problem they address is allocating bandwidth between two endpoints of a backbone network so that the network is equitably loaded, and therefore it is limited to the case of single-commodity network. In contrast, our concern here is a more general case, namely the multi-commodity network, and the models we present can be extended to load-balancing problems as we will show later in this chapter. In [43] the authors have presented an application of the MMF on network design as it will be described later in this paper.

3 Max-min fair routing

Telecommunication context. One of the most relevant applications of MMF to telecommunication networks is max-min fair routing. Generally speaking, the routing problem for telecommunication operators consists of two consecutive tasks: a good estimation of traffic demand, and the subsequent determination of the appropriate routes, or the routing of these multiple commodities in the underlying network. The latter problem, for a range of application settings, is modeled as a multi-commodity flow problem in a capacitated network with respect to a traffic demand matrix. Forecasts of traffic demand (summarized in the matrix) are generally expressed as amounts of traffic (*e.g. in Mbs/sec*) to be transported, or number of expected connections to be done between pairs of nodes. In practice, routing has been seen as a component of other more complex and general problems related to the design and/or survivability of telecommunication networks. Let remind that there are two main (opposed) ways to route traffic in a network. At the least constrained end of the spectrum, routing is said splittable (or bifurcated) if any traffic demand is allowed to be split across multiple paths. The converse of splittable routing is unsplittable (or non-bifurcated, single-path) routing, where each demand have to be routed along the same single path. Unsplittable-demand routing gives rise to NP-complete problems. From an operational point of view, although unsplittable-demand routing is more or less a requirement in IP networks, network managers can use MPLS technology to enable the splitting of traffic across several paths, and so splittable demand routing might be an acceptable objective. Anther point pleading in favor of splittable routing is the reduced number of routes (generally one or two per demand), given from a splittable routing solution obtained through multi-commodity flow models. From a mathematical point of view, the case of splittable demand routing can be viewed as a relaxation (some integrality constraints are relaxed) of the unsplittable case, thus rendering the problem computationally tractable [36]. The routing problem that we study here, namely max-min fair routing, is intended to achieve the fairest distribution of resources via routing, that is, each demand has to be served fairly, as well as network resources will allow. This is especially suitable for elastic traffic flows, which account for a major part of Internet traffic. In practice, the traffic demands change faster than the networks and the network managers cannot afford updating their routing schemes according to changes in the traffic distribution. Therefore, one approach frequently used by for network managers seeking to manage resources wisely while confronted with dynamic real-time traffic conditions is to design the best possible (optimized) static network conditions for a set of estimated average or worst-case traffic demands. Fortunately, an interesting property of MMF routing is its robustness; we will show that the obtained MMF routing solution is, at a certain extent, robust to demand traffic changes. Last, we notice that the problem of max-min fair routing finds application as a tool for IP network planners wishing to evaluate the impact of a certain degree of fairness on the resource consumption. In this case, the max-min fairness could be viewed as an approximation of the real behavior of TCP. We strongly believe that the degree of approximation is still acceptable to derive useful conclusions in this particular context. On the other hand, it might very soon happen that network planners start thinking on imposing a certain level of fairness into their network, using, for instance, MPLS based mechanisms.

3.1 Mathematical modeling and resolution approach

Let us define formally in the following a multi-commodity flow, a demand satisfaction vector and the max-min fair routing problem.

Following the notation given before, let a capacitated network be given by the triplet $N = (V, A, C)$. Let D be the set of commodities d labeled in $\{1, 2, ..., |D|\}$. Each commodity has a source node o^d and a sink node p^d, and a target value T_d. A flow f^d (of value $b \geq 0$), in network N with respect to commodity d is a function from A to $\mathbb{R}^+$ such that:

$$\forall (i,j) \in A \qquad f_{ij}^d \leq C_{ij} \tag{20a}$$

$$\forall i \in V \quad \sum_{j:(i,j)\in A} f_{ij}^d - \sum_{j:(j,i)\in A} f_{ji}^d = \left\{ \begin{array}{cc} b, & i = o^d \\ -b, & i = p^d \\ = 0, & otherwise \end{array} \right\} \tag{20b}$$

where f_{ij}^d give the arc-flow values. Each flow satisfying constraints (20a) and (20b) is called feasible. A multi-commodity flow f is composed of feasible flows f^d with respect to commodities $d \in D$ such that: $\forall (i,j) \in A, \sum_{d\in D} f_{ij}^d \leq C_{ij}$. This multi-commodity flow is called feasible. The theorem of decomposition [1] suggests that a flow can be decomposed into a set of tuples (path, carried value) and a multi-commodity flow into a set of flows, whose aggregation satisfies capacity constraints. This leads to an alternative mathematical formulation of multi-commodity flow problems, that is, arc-path formulation. Let recall that in the arc-path formulation, each path in the network is associated to a variable. This yields an important number of variables. However, such models can be solved very efficiently in practice through column generation[13]. This is particularly true for not meshed networks where the number of paths is limited. From telecommunication point of view, such models are far more convent since it allows to restrict the length of routing paths (see for instance [5]).

The connection that one can do between a multicommodity flow and a routing is straightforward. It is easy to see that some feasible multi-commodity flow, which has a value not less than T_d for each commodity d, yields a feasible routing with respect to demands requiring routing T_d units of traffic. Let us consider now some feasible multi-commodity flow and for each $d \in D$ replace b with $t_d T_d$ in (20b). The demand satisfaction (ratio) vector and the MMF routing problem can then be defined as follows:

Definition 5. *Given a routing f, the vector $t = \{t_d, d \in D\}$ whose components give the satisfaction ratio of traffic demands is called the demand satisfaction vector of routing f.*

Obviously, the demand satisfaction vector is defined on $X \subseteq \mathbb{R}_+^{|D|}$. The MMF routing problem can then be defined as computing some routing so that the corresponding demand satisfaction vector is MMF on X.

3.1.1 Solution method

In the light of the methods presented in Chapter 1, the MMF routing problem can be summarized as computing iteratively the components of the demand satisfaction vector, which are the decision variables of our problem. Intuitively, the main idea behind our algorithm is that the first lowest value has to be maximized before the second lowest value is maximized, and inductively, the maximization for a component is carried out after components whose values are less good than the given value have been maximized. A demand whose satisfaction (ratio) value has already been maximized, is called saturated. The algorithm terminates when all demands are saturated. Algorithms 3, 4, 5 can all be applied to the MMF routing problem. Let see how Algorithm 5 can be applied to the MMF routing problem.

Henceforward an arc will be referred as saturated when there is no more capacity left on it, that is, all its capacity has already been assigned to flows.
- Let D_k denote the set of commodities saturated simultaneously at the same value during step k, and B (resp. B') the set of indices of saturated (resp. non saturated) demands given by $\bigcup_{1\leq p<k} D_p$ (resp. $D \setminus \bigcup_{1\leq p<k} D_p$).

The algorithm can then be stated as follows:

Algorithm MMF routing

Input: A capacitated network $N = (V, A, C)$ and a set of commodities $d \in D$.

Output: The max-min fair routing and the associated demand satisfaction vector.

- Set $k = 1; B = \phi$ - While $B \neq D$ do:
- Solve the (LP) routing problem $\mathcal{P}(B, \boldsymbol{t}^B)$, compute τ; /* see below */
- Identify all demands saturated at this step, (set D_k), put $B := B \cup \{D_k\}$, $t^{D_k} := \tau$ and $k = k + 1$.

where the LP formulation of problem $\mathcal{P}(B, \boldsymbol{t}^B)$ and a similar LP model for testing whether some demand is saturated in the sense of max-min fairness are detailed below.

Formulating and solving the $\mathcal{P}(B, \boldsymbol{t}^B)$ *routing problem:*
The $\mathcal{P}(B, \boldsymbol{t}^B)$ problem can be modeled using a classic arc-node LP formulation as follows:

$$
\begin{array}{llll}
Max & \tau & & \\
(\omega_{i,j}) & \sum_{d\in D} f_{ij}^d \leq C_{ij}, & \forall (i,j) \in A, & \text{(21a)} \\
(\pi_{i,d}) & \sum_{j:(i,j)\in A} f_{ij}^d - \sum_{j:(j,i)\in A} f_{ji}^d = 0, & \forall d \in D, i \in V \setminus \{o^d, p^d\} & \text{(21b)} \\
(\pi_{i,d}) & \sum_{j:(i,j)\in A} f_{ij}^d - \sum_{j:(j,i)\in A} f_{ji}^d = \left\{ \begin{array}{l} t_d T_d, i = o^d \\ -t_d T_d, i = p^d \end{array} \right. & \forall i \in V, \forall d \in B & \text{(21c)} \\
(\pi_{o^d,d}) & \sum_{j:(j,i)\in A} f_{ji}^d - \sum_{j:(i,j)\in A} f_{ij}^d + t_d T_d \leq 0, & \forall d \in B', i = o^d & \text{(21d)} \\
(\pi_{p^d,d}) & \sum_{j:(i,j)\in A} f_{ij}^d - \sum_{j:(j,i)\in A} f_{ji}^d + t_d T_d \leq 0, & \forall d \in B', i = p^d & \text{(21e)} \\
 & f_{ij}^d \geq 0 & \forall (i,j) \in A, \forall d \in D & \text{(21f)}
\end{array}
$$

where constraints (21a) are capacity constraints, constraints (21b) are mass balance constraints and (21c, 21d, 21e) are traffic constraints. Finally, constraints (21f) indicate that arc-flow values must be non-negative while in parentheses we give the respective dual variables. At the end of the solving procedure we obtain a multi-commodity flow f, τ and the dual values associated with constraints. In practice, several commodity flows can be simultaneously saturated at the end of the k^{th} step. Assuming that we are using an IPM method to solve the above problem, the set D_k gives the set of indices of constraints (21d) which dual values are strictly positive.

3.2 Routing and resource sharing

In real TCP/IP networks the traffic demands can be seen as a number of connections to be done. This leads also to another notion closely related to max-min fairness. Assuming that access rates are unlimited, it is easy to see that each connection is necessarily saturated in one of its links, called a bottleneck link. Its (end-to-end) rate is then equal to the minimum of the shares offered by each link in the path. It can also be expressed as the share offered by the bottleneck link to connections saturated in it, which we term the *max-min fair share*. For a better explication, consider the following simple example of a linear network consisting of 3 routers A, B, C and 2 links AB, BC with capacities set respectively to 2MB/sec and 3MB/sec. Assume there are 3 connections: $A \rightarrow B, A \rightarrow C$ and $B \rightarrow C$, each of them routed through the unique path in the network.

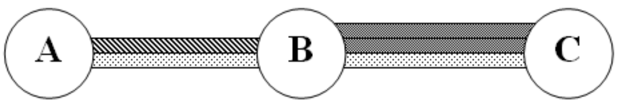

Figure 1: Example of max-min fair resource sharing between connections.

The connection $A \rightarrow B$ is clearly saturated by link AB (there is no other choice); similarly, the connection $B \rightarrow C$ is saturated by link BC. We see intuitively that if we start to increase equally the amount of bandwidth allocated to each one of the three connections, the first link to become saturated will be AB, with a resulting allocation of 1 unit for each connection. Going one step further, the bandwidth of the connection $B \rightarrow C$ can still be increased by one unit before the second link BC also becomes saturated. The final max-min fair rate of connections is $[1, 1, 2]$ while the max-min fair share of links is $[1, 2]$. The above procedure can be generalized in any network where the routing is already known.

Let us look at an example [36] for explaining the relation between MMF routing and resource sharing. Suppose that we have a network with 4 routers : A, B, C, D, connected by 4 links AB, AC, BD, CD with capacities fixed respectively at 1MB/sec, 3MB/sec, 1MB/sec and 2MB/sec. The connections to be routed are $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $B \rightarrow D$ and $C \rightarrow D$.

Let us first see how the Algorithm 1 handles this example (i.e., when the routing is fixed as shown in Table 1), (see Figure 3.2). We remark that at the end of the first step, connections $A \rightarrow B$, $A \rightarrow D$, $B \rightarrow D$ are occupied with a rate of 0.5MB/sec. Consequently, links AB and BD become saturated. During the next step, the connection $C \rightarrow D$ is saturated with a rate of 2MB/sec. At

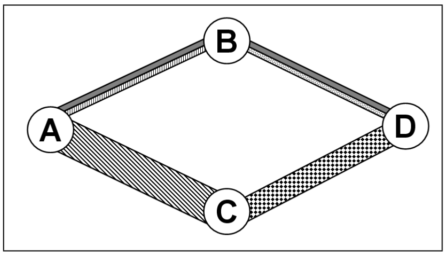

Figure 2: The resulting resource sharing for a fixed routing (MMF/SRP).

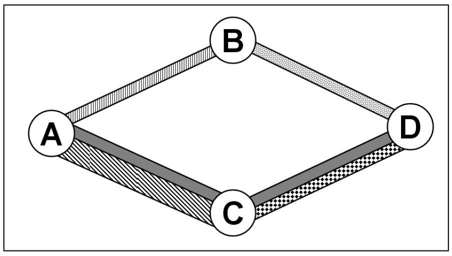

Figure 3: The resulting resource sharing for MMF routing.

the final step, $A \rightarrow C$ is saturated with a rate of 3MB/sec. The total throughput is 6.5 MB/sec.

In Figure 3.2 we show how the routing and the ressource sharing when the MMF routing algorithm is applied. The total throughput is 6MB/sec, less than the first case but the sharing/distribution is substantially fairer.

We notice that the MMF routing solution allows a fair sharing of resources both with respect to the max-min fair sharing links and end-to-end connection rates. Thus, the connection $A \rightarrow D$ uses the path $A - C - D$, which is composed of the two least loaded links that are AC and CD.

3.3 Some theoretical properties

In this section we study the optimality and polynomiality of our approach, and we highlight certain properties of MMF routing and MMF demand satisfaction vector. As the MMF Algorithm is a direct application of the Algorithm 5, then

	Given Routing		MMF Routing.	
Connection	Path	Flow	Path	Flow
A → B	A-B	0.5	A-B	1
A → C	A-C	3	A-C	2
A → D	A-B-D	0.5	A-C-D	1
B → D	B-D	0.5	B-D	1
C → D	C-D	2	C-D	1
Total		6.5		6

Table 1: Comparison of resource sharing for three routing strategies

the theoretical results (MMF and polynomiality) stated in [34] holds also for the MMF routing Algorithm, and we have the following result:

Theorem 2. *The multi-commodity routing obtained at the end of the MMF routing algorithm is max-min fair and it can be obtained in polynomial time.*

We can go further and state that any MMF routing solution yields not only max-min fair allocation of end-to-end rates of connections, but also ensures that the share's link vector is also max-min fair [36]:

Theorem 3. *The MMF routing solution has the following properties:*
a) there is no other routing that would allow the fair sharing value of some saturated link to be increased at the expense of other links offering more.
b) there is no other routing that would allow the rate allocation of saturated connections to be increased at the expense of connections better served.

In the sequel we will exhibit some other properties related to max-min fair routing. First, let us note with S_k, the set of saturated links at the end of the step k. Let us consider the set of multi-commodity flows satisfying all constraints associated with the problem $\mathcal{P}(B, \boldsymbol{t}^B)$, (step k). Obviously, for each new optimal solution there is at least one newly-saturated demands or arc. Let us denote as D_k (resp. S_k) these sets, and D_k^* (resp. S_k^*) the intersection of all sets of newly saturated demands (arcs) with respect to all possible routings obtained as solution of problem $\mathcal{P}(B, \boldsymbol{t}^B)$. Then, we can state the following properties:

Property 2. *All sets D_k^* and S_k^* are not empty.*

Let us begin by considering the set D_k^* and suppose by absurdity that it is empty. Indeed, without loss of generality, let us suppose that there exist two routings Φ_1 and Φ_2 satisfying all constraints with respect to $\mathcal{P}(B, \boldsymbol{t}^B)$ (step k), and having distinct sets of saturated flows at the same value τ. These routings are such that $Q_1 \subset B'$ (resp. $Q_2 \subset B'$) represents the set of newly saturated commodity flows for Φ_1 (resp. Φ_2) and $Q_1 \cap Q_2 = \emptyset$. Let $\Phi = \frac{(\Phi_1 + \Phi_2)}{2}$. It is obvious that the routing Φ satisfies capacity and mass-balance constraints associated with the problem $\mathcal{P}(B, \boldsymbol{t}^B)$. Furthermore, it is not saturated in any demand contained in B'. In these conditions, we increase at the same rate the flow value to each f_{ij}^d with respect to non-saturated demands ($d \in B'$) until the network becomes saturated. Obviously, τ will be increased and will have a value strictly superior to that obtained by Φ_1 and Φ_2, which contradicts the fact that Φ_1 and Φ_2 are obtained as solutions of problem $\mathcal{P}(B, \boldsymbol{t}^B)$ during the k^{th} step. We have consequently proved the first part of the above property. The second part can be proved in a similar way. It will be noted that each routing solution of problem $\mathcal{P}(B, \boldsymbol{t}^B)$ necessarily saturates some arcs separating the extremities of the saturated demands in the graph. Using the same logic, that is, assuming the existence of two routings Φ_1 and Φ_2 satisfying all constraints with respect to $\mathcal{P}(B, \boldsymbol{t}^B)$, and having distinct sets of saturated arcs at the end of step k, it is clear that $\Phi = \frac{(\Phi_1 + \Phi_2)}{2}$ does not saturate any of them. We can therefore state that the intersection of all these sets of arcs is necessarily not empty. □

We state without proof in the following some simple properties of routing solutions obtained at the end of a given step of the algorithm.

Property 3. *All routing obtained at the end of the MMF Algorithm saturate demands in a unique order*[3].

Property 4. *The MMF routing Algorithm terminates in at most* $Min\{|D|, |A|\}$ *steps.*

The last result can be improved when dealing with undirected networks.

Proposition 4. *All first-level max-min fair routing in a capacitated k-arc connected undirected network saturate in at least k arcs.*

Notice first that each first-level max-min fair routing is necessarily saturated in at least one commodity d. Furthermore, the terminal nodes o_d and p_d cannot be connected through paths containing only non-saturated arcs because the flow value of the demand d cannot be increased. Consequently, all disjoint paths (at least k) between o_d and p_d are saturated. Which is to say that at least k arcs are saturated at the end of the first step[4]. $\square$

In the same way we note that the terminal nodes of saturated commodities are placed in two distinct/complementary sets linked by saturated arcs, thus defining a cut. At the end of the first step, let N_1 denote the subnetwork obtained from the initial network N when removing the saturated arcs. Similarly, at the end of the k^{th} step, let N_k denote the subnetwork obtained when removing from N_{k-1} the newly-saturated arcs. At the end of the k^{th} step, the terminal nodes of newly saturated demands are necessarily disconnected in the subnetwork N_k and the number of components is strictly superior to this in N_{k-1}. This is because these new demands were not saturated before executing the most recent step, and consequently the respective terminal nodes were in the same component. This component cannot therefore remain connected in the new subnetwork N_k, and consequently the number of components becomes strictly superior to that obtained at the end of the preceding step. So after each step, the number of components of the remained subnetwork is incremented by at least one. Since at the end of the first step the number of components is at least 2, it follows that after $k-1$ further steps the number of components becomes greater or equal to $k+1$. Combining this with the fact that a connected network cannot be decomposed into more than $|V|$ components, we see that there cannot be more than $|V|-1$ decompositions of the network or steps of the algorithm, and have thus proved the following result:

Proposition 5. *The max-min fair flow vector in undirected networks has at most* $Min\{|D|, |A|, |V|-1\}$ *distinct values.*

3.4 Robustness of MMF multi-commodity flows

First, it is worth noting that the max-min fair routing computing approach is able to cope with a homogeneous traffic increase/decrease across all demands and is particularly suitable for elastic traffic. More generally, the MMF solution provides robust routing schemes even for the general case when a set of traffic demands represented in a matrix have to be routed through a given network. It is well known that traffic prevision is a difficult task and errors are common

[3]The demands contained in the same D_k are interchangeable in the saturating order.

[4]A large number of demands could be thus saturated during the first step.

in real-life situations. The max-min fair routing also provides a robust routing scheme. Generally speaking it is obvious that different routing schemes do not face in the same way the overload traffic situations. With respect to our MMF routing solution, we can show that variations in traffic within certain bounds are completely absorbed (we assume that routing is feasible). As an illustration, let us suppose that the traffic for all demands $d \in D$ are increased within a given ratio r_d with respect to the traffic prevision (T_d). We are able to state that as long as $r_d \leq \lambda[d] - 1$, for all $d \in D$, the computed routing will remain feasible and all demands will be entirely routed. It is then easy to prove, following Proposition 1, that the overload ratio vector given by $\{\lambda[d] - 1, \forall d \in D\}$ is also MMF.

Property 5. *The overload ratio vector associated with a max-min fair routing solution is max-min fair.*

In other words, any MMF routing solution guarantees for each demand a certain ratio of overload such that it is not possible to do better without decreasing the guaranteed overload ratio of other traffic demands with lower values.

4 Lexicographically minimum load networks

In this chapter, we consider the problem of load-balancing (as referred in [21]) in a given network. Thus, we aim to distribute the load fairly among the network links while satisfying a given set of traffic constraints. More precisely, we aim not only to minimize the maximal load among links, but also to minimize lexicographically the sorted (non-increasing) load values through network links in order to obtain what is called a *lexicographically minimum load network* [34]. In contrast to the max-min fair multi-commodity flow problem, this one is equivalent to the problem of loading fairly the network in the sense min-max. This problem arises in telecommunication networks when the operator needs to define routing with respect to a given traffic demand matrix such that the network load is fairly distributed among the network links. We consider the multi-commodity network and we focus on the splittable case. Some relevant work on balanced networks is presented in [21] where the authors propose a strongly polynomial approach for lexicographically-optimal balanced networks. The problem they address is allocating bandwidth between two endpoints of a backbone network so that the network is equitably loaded, and therefore it is limited to the case of single-commodity network. How to generalize this result to multi-commodity networks is already shown in [34], but we consider here a large range of link load functions, especially non-linear, frequently used in telecommunication area, and show that each of them can be brought to the linear case (see also [39]). Let us begin by giving some definitions and notation.

4.1 Link load functions

A load function $Y(f)$ (linear or not) defined as $Y : A \to \mathbb{R}^+$ gives the load associated with each arc. An arc load function is generally closely related to the arc occupation ratio and thus to the flow traversing it. It can typically be expressed linearly as $\frac{f_a}{C_a}$, where $f_a = \sum f_{ij}^d : (ij) = a$ and C_a is the capacity of link a. Another currently used criterion (to maximize) is measuring the residual capacity $C_a - f_a$. The global network resources optimization can also involve non-linear link criterions, as for instance $(\alpha - 1)^{-1}(1 - f_a/C_a)^{1-\alpha}$ or $(\alpha - 1)^{-1}(C_a - f_a)^{1-\alpha}$ $\alpha \in R^+ \setminus \{1\}$; (see e.g. [3, 29]). We show that, in any of these cases, it is possible to use one of the two previous linear examples to obtain optimal solutions.

4.1.1 Link load vector and computing approach framework

Similarly to flow vector, one can define the link load vector with respect to a multi-commodity flow and a load function.

Definition 6. *Given a network (V, A, C), a multi-commodity flow f and a load function Y, the vector $\gamma = \{\gamma_a, a \in A\}$ whose components give the load value associated with each arc (i.e., $\gamma_a = Y(a)$) is called the link load vector of multi-commodity flow f.*

A link load vector is called *feasible* if there exists a feasible multi-commodity flow with the same load vector. Let F denote the set of all the feasible multi-commodity flows in network N with respect to a set of commodities D. We use the phrase *lexicographic minimum load network problem* to refer to the problem of computing a feasible multi-commodity flow such that its link load vector is min-max fair (that is leximax minimal) in the set of feasible load vectors with respect to F. Now, the lexicographic minimum load network problem can be stated as: *given a capacitated network* $N = (V, A, C)$ *and a set of commodities* D *with traffic values* $\{T_d\}_{d \in D}$*, find the min-max fair link load vector* γ *and a corresponding feasible multi-commodity flow.* Computing min-max fair vectors can be achieved by similar algorithms to these given in Chapter 1, except that instated of maximizing, one need to minimize iteratively the components of the load vector sorted through non-decreasing values. This problem is already analyzed in [34] where a similar approach to Algorithm 5 is given and some connection between the two algorithms is established.

The algorithm can then be stated as follows:

Algorithm Min-Max Fair loaded network

Input: A capacitated network $N = (V, A, C)$ and a set of commodities $d \in D$.

Output The min-max fair link load vector and the associated multi-commodity flow.

- Set $k = 1; B = \phi$ - While $B \neq A$ do:
- Solve the (LP) problem $\mathcal{P}(B, \boldsymbol{t}^B)$, compute τ; /* see below */
- Identify the set of all links minimally loaded at this step, (called A_k) put $B := B \cup \{A_k\}$, $t^{A_k} := \tau$ and $k = k + 1$.

where the LP formulation of problem $\mathcal{P}(B, \boldsymbol{t}^B)$ and a similar LP model for testing whether some demand is saturated in the sense of max-min fairness are detailed below.

Formulating and solving the $\mathcal{P}(B, \boldsymbol{t}^B)$ *problem:*
The $\mathcal{P}(B, \boldsymbol{t}^B)$ problem can be modeled using a classic arc-node LP formulation as follows:

A link whose load has already been minimized, is called *minimally loaded.* The algorithm terminates when all links are minimally loaded. During each step we resolve a LP problem similar to P_k described for the Algorithm MMF routing. Let consider the most common load function, which gives the relative load $\frac{f_a}{C_a}$ that we need to minimize. The corresponding problem $\mathcal{P}(B, \boldsymbol{t}^B)$ can be then stated as:

$$Minimize \quad \tau$$

$$\sum_{j:(i,j)\in A} f_{ij}^d - \sum_{j:(j,i)\in A} f_{ji}^d = \begin{cases} T_d, i = s^d \\ -T_d, i = t^d \\ 0, otherwise \end{cases} \qquad \forall i \in V, \forall d \in D \qquad (22a)$$

$$\sum_{d\in D} f_{ij}^d \leq C_{ij} \qquad \forall (i,j) \in A \qquad (22b)$$

$$\sum_{d\in D} f_{ij}^d \leq \alpha C_{ij} \qquad \forall (i,j) \in B' \qquad (22c)$$

$$\sum_{d\in D} f_{ij}^d = \gamma[i,j] C_{ij} \qquad \forall (i,j) \in B \qquad (22d)$$

$$f_{ij}^d \geq 0 \quad \forall (i,j) \in A, \forall d \in D, \qquad (22e)$$

where constraints (22a) are flow conservation constraints, (22b) are capacity constraints and (22c, 22d) are load constraints. Finally, constraints (22e) indicate that arc-flow values must be non-negative. Similarly to notation used for the previous algorithms, B denotes the set of links loaded through the first $k-1$ iterations of the algorithm and B' its complementary subset with respect to A. To finish with the iteration, we need to identify all links loaded to τ by searching for the binding constraints (22c). Then, we set $\gamma[i,j] = \tau$ for all lastly loaded links. One can follow a reasoning similar to this used for the Algorithm MMF routing and deduce that there is at least one such minimally loaded link at the end of each iteration. Furthermore, one can show that there are at most $|v|-1$ distinct load values in the obtained load vector.

4.1.2 Non-linear link load functions

In practice there are several load functions employed by network managers, most of them non-linear. A wellknown load function, called Kleinrock function, is given by $\frac{f_a}{(C_a - f_a)}$ and it takes value in $[0, +\infty)$. It can be directly proved that any multi-commodity flow achieving min-max fairness for the relative load function achieves also min-max fair load for the Kleinrock function (see [34]).

Corollary 1. *Any multi-commodity flow that achieves lexicographic minimum load for the linear load function* $\frac{f_a}{C_a}$ *also achieves lexicographic minimum load for the Kleinrock load function* $\frac{f_a}{(C_a - f_a)}$*, and vice-versa.*

Maybe we can give the framework of the proof...

Instead of considering separately the different link load functions we will propose a more general way to deal with the most frequently used link load functions. Let have a look at the following general link load function: $(\alpha - 1)^{-1}(1 - f_a/C_a)^{1-\alpha}$. With respect to Proposition 1, one can prove that the result also holds when replacing "max-min fair" with "min-max fair". Then, we obtain the following corollary which is easy to verify since the above function can be expressed as increasing function of $\frac{f_a}{C_a}$.

Corollary 2. *Let* $\alpha \in R^+ \setminus \{1\}$*. Any multi-commodity flow that achieves lexicographic minimum load for the linear load function* $\frac{f_a}{C_a}$ *also achieves lexicographic minimum load for the generalized link load function* $(\alpha - 1)^{-1}(1 - f_a/C_a)^{1-\alpha}$*, and vice-versa.*

The second general link load function frequently used is $(\alpha - 1)^{-1}(C_a - f_a)^{1-\alpha}$ $\alpha \in R^+ \setminus \{1\}$. It can easily be seen that the above function is a decreasing function of $(C_a - f_a)$. $(C_a - f_a)$ expresses the residual capacity of the link and it can be seen as an "unusual" load vector, which needs to be maximized lexicographically. Then, the approach is quite similar and the problem $\mathcal{P}(B, \boldsymbol{t}^B)$ can be formulated similarly to above except changing the objective function to maximize and replacing load constraints (22c, 22d) by (23a, 23b):

$$C_{ij} - \sum_{d \in D} f_{ij}^d \geq \alpha \quad \forall (i,j) \in B' \tag{23a}$$

$$C_{ij} - \sum_{d \in D} f_{ij}^d = \gamma[i,j] \quad \forall (i,j) \in B \tag{23b}$$

Now we can use a similar trick to this for Corollary 1 based on the following result:

Proposition 6. *Let ϕ be a decreasing function in R. Some x achieves a max-min fair (resp. min-max fair) vector γ with respect to system (1), if and only if x achieves a min-max fair (resp. max-min fair) vector for the corresponding system composed of functions $\{\phi \circ f_i\}$ given by $\phi(\gamma)$.*

Then, we obtain the following corollary which solves the case of the second general link load function.

Corollary 3. *Let $\alpha \in R^+ \setminus \{1\}$. Any multi-commodity flow whose load vector is max-min fair for the linear load function $C_a - f_a$ achieves lexicographic minimum load for the generalized link load function $(\alpha-1)^{-1}(C_a - f_a)^{1-\alpha}$, and vice-versa.*

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, New Jersey, 1993.

[2] W. Ben-Ameur and E. Gourdin. Internet routing and related topology issues. In *Proc. of DRCN 2000*, 2000.

[3] W. Ben Ameur, N. Michel, E. Gourdin et B. Liau. Routing strategies for IP networks. *Telektronikk*, 2/3, pp 145-158, 2001.

[4] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Engelwood Cliffs, N.J., 1992.

[5] D. Bertsimas and J. N. Tsitsiklis *Introduction to Linear Optimization.* Athena Scientific, 1997.

[6] T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In *Proc. of ACM SIGMETRICS 2001*, 2001.

[7] F.A. Behringer, A simplex based algorithm for the lexicographically extended linear maxmin problem, European Journal of Operational Research 7 (1981) 274-283.

[8] F.A. Behringer, Linear multiobjective maxmin optimization and some Pareto and lexmaxmin extensions, OR Spektrum 8 (1986) 25-32.

[9] S. Boulahia-Oueslati. *Qualité de service et routage des flots élastiques dans un réseau multiservice.* PhD thesis, ENST, Paris, France, November 2000.

[10] A. Charny, D. D. Clark and R. Jain. Congestion control with explicit rate indication. In *Proc. of IEEE International Conference on Communications*, 1995.

[11] S. Chen and K. Nahrstedt. Maxmin fair routing in connection-oriented networks. Tech. report, Dept. Comp. Science, UIUC, USA, 1998.

[12] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[13] G. Desaulniers, J. Desrosiers, M. M. Solomon (eds.) *Column Generation.* Springer-Verlag, N.Y, June 2005.

[14] S. Floyd, K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Trans. On Networking, 7(4), August 1999.*

[15] S. Floyd. Congestion control principle. *IETF draft ¡draft-floyd-cong-02.txt¿, March 2000.*

[16] G. Fodor, G. Malicsko, M. Pioro and T. Szymanski. Path Optimization for Elastic Traffic under Fairness Constraints. In *Proc. of ITC 2001.*

[17] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, N.J., 1962.

[18] R. M. Freund and S. Mizuno. Interior Point Methods: Current Status and Future Directions. In *High Performance Optimization*, H. Frenk et al. (eds.), Kluwer Academic Publishers, pp. 441-466, 2000.

[19] R. M. Freund, R. Roundy, and M.J. Todd. Identifying the Set of Always-Active Constraints in a System of Linear Inequalities by a Single Linear Program. *Working paper 1674-85*, Massachusetts Institute of Technology (MIT), Sloan School of Management, 1985.

[20] J. Galtier. Semi-definite programming as a simple extension to linear programming: convex optimization with queueing, equity and other telecom functionals. In *Proc. of AlgoTel 2001.* INRIA, 2001.

[21] L. Georgiadis, P. Georgatsos, K. Floros, S. Sartzetakis Lexicographically Optimal Balanced Networks. In *Proc. Infocom'01*, pp. 689-698, 2001.

[22] A. Goel, A. Meyerson, and S. Plotkin. Combining fairness with throughput: Online routing with multiple objectives. In *Proc. 32nd ACM STOC*, 2000.

[23] V. Jacobson, M. J. Karels Congestion Avoidance and Control (1988) In *ACM Computer Communication Review; Proceedings of the Sigcomm '88 Symposium* in Stanford, CA, 1988.

[24] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, vol. 49, pp. 237-252, 1998.

[25] W. Kern, D. Paulusma. Matching Games: the Least Core and the Nucleolus. *Mathematics of Operations Research* Vol. 28, No. 2, pp. 294308, May 2003.

[26] R.S. Klein, H. Luss, D.R. Smith, A lexicographic minimax algorithm for multiperiod resource allocation, Mathematical Programming 55 (1992) 213–234.

[27] E. Kohlberg. On the Nucleolus of a Characteristic Function Game. *SIAM Journal of Applied Mathematics*, Vol. 20, No. 1, 1971.

[28] J. Kleinberg, Y. Rabani, and E. Tardos. Fairness in routing and load balancing. In *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, 1999.

[29] L. Kleinrock. *Communications Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964.

[30] Q. Ma. *Quality-of-Service Routing in Integrated Service Networks.* PhD thesis, Carnegie Mellon Univ., Pittsburgh, USA, Jan. 1998.

[31] L. Massoulié and J.W. Roberts. Bandwidth sharing: objectives and algorithms. In *Proc. of IEEE INFOCOM'99*, 1999.

[32] N. Megiddo. Optimal flows in networks with sources and sinks. *Mathematical Programming*, 7, 1974.

[33] J. Y. Le Boudec Rate adaptation, congestion control and fairness: a tutorial. *http://ica1www.epfl.ch/PS_files/LEB3132.pdf*, Fevrier 2005.

[34] D. Nace, J. B. Orlin. Lexicographically Minimum and Maximum Load Linear Programming Problems, to appear in Operations Research.

[35] D. Nace. A linear programming based approach for computing optimal splittable fair routing. In *Proc. of The Seventh IEEE Symposium on Computers and Communications 2002, ISCC'2002. pp. 468-474*, July 2002, Taormine, Italy.

[36] D. Nace, L.N. Doan, Eric Gourdin and Bernard Liau. Computing optimal max-min fair resource allocation for elastic flows *to apear in dec. 2006, IEEE Transactions on Networking*

[37] D. Nace, L.N. Doan. A polynomial approach to the fair multi-flow problem. *Rapport Interne, Heudiasyc, UTC, November 2002.*

[38] D. Nace, N-L. Doan, Some Results on Max-min Fair Routing, INOC 2003, Evry, 27-29 October 2003. *In Proc. of INOC 2003, Evry, October 2003.*

[39] D. Nace, O. Klopfenstein On the Lexicographically Minimum Loaded Networks, *In Proc. of INOC 2005, mars 2005, Lisbone Portugal.*

[40] W. Ogryczak, M. Pioro and A. Tomaszewski, Telecommunications Network Design and Max-Min Optimization Problem. *Journal of Telecommunications and Information Thechnology* 3/2005, pp. 1-14.

[41] M. Pioro, P. Nilsson, E. Kubilinskas and G. Fodor. On Efficient Max-min Fair Routing ALgorithms. *In Proc. of ISCC 2003, Antalya, Turkey, July 2003.*

[42] M. Pioro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*, Morgan Kaufmann Publishers (2004).

[43] P. Nilsson, M. Pioro, Z. Dziong Link Protection within an Existing Backbone Network *In Proc. of INOC 2003, Evry, October 2003*, pp. 435-441.

[44] D. Schmeidler. The Nucleolus of a Characteristic Function Game. *SIAM Journal of Applied Mathematics*, Vol. 17, 1163-1170, 1969.

[45] F. Shahrokhi and D.W. Matula. The Maximum Concurrent Flow Problem. *in Journal of the ACM*, Vol. 37, pp. 318-334, 1990.

[46] B. Radunovic, J.-Y. Le Boudec. A Unified Framework for Max-Min and Min-Max Fairness with Applications. *Proceedings of 40th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, October 2002.

[47] J. Ros and W. K. Tsai, A Theory of Maxmin Rate Allocation with Minimal Rate Guarantee. *Technical Report, http://www.eng.uci.edu/ netrol, June 2000.*

[48] M. Vojvonovic, J.-Y. Le Boudec and C. Boutremans. Global Fairness of additive-increase and multiplicative-decrease with heterogeneous round trip-times. In *Proc. of IEEE INFOCOM'00*, 2000.