Flow on data network and a positive semidefinite representable delay function

Jérôme Galtier * and Alexandre Laugier *
Jerome.Galtier@orange-ft.com
Alexandre.Laugier@orange-ft.com

January 6, 2007

Abstract

Data networks are subject to congestion, thereby the delay to go across the network may be large enough in order to dishearten customers to keep on using such a network. In this paper we address the problem of determining in a given network a routing which minimizes the delay or keeps it under a certain bound. This problem was already shown as $NP$ complete. Our main contribution is to study it in the special context of the positive semidefinite programming and we present a column generation approach to solve the underlying problem.

1 Introduction

The quality of service, QoS for short, offered by a telecommunication network can be expressed either in terms of delay or in terms of loss. The delay is the amount of time spent by an element, typically a packet, of the flow communication in order to go across a component of the network. The loss is the percentage of the number of packets committed to the network which are lost during the transmission. One can remark that if a packet is lost its transmission delay is infinite. We will study problems related to routing flows in a network with delay constraints. The transmission delay can be expressed on the transmission links of the network; then it is the time needed to go through a link. Or we may consider end-to-end delay; then it is the total amount of time needed to cross the network. The end-to-end delay may be the sum of the delays on the transmission links which form a route. In this case the criterion is said to
be additive. Notice it is not always the case for instance if a packet is split into smaller ones on certain links. However we will consider in this paper only additive criteria for QoS. We will focus on the remainder on the following problems dealing with the delay.

The first one concerns the minimization of the mean delay over the set of transmission links. This problem has been studied by P. Mahey and A. Ouorou: [15], [7] and [13]. They remark that the problem can be decomposed on each arc and employed a proximal decomposition method. However their methods suppose differentiability. Unfortunately the congestion functions have a vertical asymptote when the capacity is reached. This can be a cause of numerical instability. Therefore we investigate positive semidefinite methods.

The second concerns the problem of computing a minimum cost routing subjected to a maximum delay. The case where the delay is bounded on each edge was investigated by F. Boyer, [6]. The author used Benders decomposition in order to solve a minimum cost flow problem where each arc is subjected to have its related delay less or equal to a certain value. Further information about flows with convex cost can be found in the survey by A. Ouorou, P. Mahey and J.P. Vial, [16], with additional information in [12]. This problem may have alternative settings. On the one hand, the delay may be a function from the set of edges to \( \mathbb{N} \), i.e. a fixed number is associated to each edge. On the other hand, the delay on an edge may be a function of the value of the flow circulating on this edge and of the capacity of the edge. We only investigate the case where the capacity is fixed. Thus when the flow equals the capacity we have a vertical asymptote.

With respect to this second problem our aim is to determine a routing of the demand which minimizes the routing cost for a given linear function of the flow and such that the total delay spent in the network in order to go from source of the demand to its sink is bounded by a given value.

Now let us expose the models of the problem that we address in this article. The network is represented by a graph \( G = (V, E) \) where the set of vertices \( V \) is the set of switching points of the network and the set of edges \( E \) is the set of transmission links. The capacity of an edge is denoted by \( C_e \). Let \( d \) be the amount of demand which has to be carried from a source \( s \) to a sink \( t \) and \( \Pi \) be the set of all the paths linking \( s \) to \( t \), \( \phi_p \) is the part of the flow routed on the path \( p \in \Pi \). We denote by \( t_e \) the necessary delay to go through the edge \( e \). Our model requires that the total delay along any \( s-t \) path used in the solution, do not exceed a prescribed maximum allowed value \( \tau \). According with these notations the problems of interest can be formulated in the following way.

Now, we introduce the Kleinrock’s delay function which is handled in this article. According to Kleinrock, [11], the necessary delay to send a packet through the edge \( e \) is \( t_e = \frac{\phi_e}{C_e - \phi_e} \), where \( \phi_e = \sum_{p \ni e} \phi_p \). This function expresses the transmission delay of packets on transmission links if their inter-arrival times follow independent exponential laws and the service times observe an exponential law. These requirements are not so strong if we consider a backbone network where the number of connections is very large, and these connections can be considered independent.

First of all we express the problem which consists in computing the flow which minimizes the sum of the delay over the set of edges. This problem is as below:

\[
MnMdf = \begin{cases} 
\text{Min} & \sum_{e \in E} t_e \\
\text{s.t.} & \sum_{p \ni \Pi} \phi_p \geq d \\
& \phi_e = \sum_{p \ni e} \phi_p \leq C_e \ \forall e \in E \\
& \phi_p \geq 0 \ \forall p \in \Pi, \ t_e = \frac{\phi_e}{C_e - \phi_e} \ \forall e \in E 
\end{cases}
\]
Although there is no guarantee for the end-to-end delay, this model is interesting since it allows to find a routing such that the mean delay is minimum and this implies that the time spent in an individual edge is not too large. The two following are related with end-to-end QoS. Notice that in order to express this kind of QoS we adopt an arc-path formulation of the flow problem. According to the first of these two problems the delay over each edge $e$ is given by a fixed number $\tau_e$: this case corresponds to a network in which the transmission links have a low load. If $k_e$ is the cost for routing a unit of flow on edge $e$ this problem is as follows:

\[
\text{Mcfbd} = \begin{cases} 
\min & \sum_{e \in E} k_e \sum_{p \ni e} \phi_p \\
\text{s.t.} & \sum_{p \in \Pi} \phi_p \geq d \\
& \sum_{p \ni e} \phi_p \leq C_e \forall e \in E \\
& \sum_{e \ni p} t_e \leq \tau, \forall p \in \Pi \\
& \phi_p \geq 0 \forall p \in \Pi, t_e = \tau_e \forall e \in E 
\end{cases}
\]

Since the delay $t_e$ on edge $e$ is a fixed non negative integer, it is an input of the problem. Hence the set of paths $\Pi^{\leq \tau}$ satisfying the delay requirements can be computed a priori and all the other paths can be ignored. Then we may give a new formulation of the problem:

\[
\text{Mcfbd} = \begin{cases} 
\min & \sum_{e \in E} k_e \sum_{p \ni e} \phi_p \\
\text{s.t.} & \sum_{p \in \Pi^{\leq \tau}} \phi_p \geq d \\
& \sum_{p \ni e} \phi_p \leq C_e \forall e \in E \\
& \phi_p \geq 0 \forall p \in \Pi^{\leq \tau}, 
\end{cases}
\]

Remark that the set $\Pi^{\leq \tau}$ may be exponentially large in the size of the graph since it may contain all the paths that exist between the source and the sink of the demand. If the network is subject to congestion, we have to express the delay as a function of the flow circulating on the edges which have a fixed capacity. The problem which consists in computing a minimum cost flow with bounded delay constraints, the delay being given by the Kleinrock’s formula, can be written as shown below:

\[
\text{Mcfdk} = \begin{cases} 
\min & \sum_{e \in E} k_e \sum_{p \ni e} \phi_p \\
\text{s.t.} & \sum_{p \in \Pi} \phi_p \geq d \\
& \phi_e = \sum_{p \ni e} \phi_p \leq C_e \forall e \in E \\
& \sum_{e \ni p} t_e \leq \tau, \forall p \in \Pi \\
& \phi_p \geq 0 \forall p \in \Pi, t_e = \frac{\phi_e}{C_e-\phi_e} \forall e \in E 
\end{cases}
\]

Notice that for both problems $Mmdf$ and $Mcfdk$ the capacity constraint $\phi_e = \sum_{p \ni e} \phi_p \leq C_e$ are unnecessary since if they are violated the objective function of $Mmdf$ is unbounded and the
delay constraints of $Mcfdk$ are violated. In addition one can check that the problem $Mcfdk$ is over constrained with respect to the problem $Mmdf$ since paths are subjected to have a short delay even if they support no flow. If we want to deal with a not over constrained problem we have to introduce delay constraints with the following form: $\sum_{e \in p} t_e \leq \tau, \phi_p > 0$, but such a constraint is not well defined since it depends on the solution of the problem, we will mention a way that we want to investigate in order to solve such a problem at the end of this article.

The remainder of this article is organized as follows. In the next section we give a positive semidefinite formulation of the problems $Mmdf$ and $Mcfdk$ and discuss the complexity of the problems $Mmdf$ and $Mcfbd$. The section 3 is devoted to the problem $Mcfdk$. We will prove it is $NP$-hard after that we have shown that it satisfies some regularity condition known as Slater’s condition, this is done in order to use Lagrangian duality instead of extended duality as it is defined in [19] and [18] or by G. Pataki in [17] because these forms of the dual involve a too large number of variables in order to design an efficient algorithm. Then we establish optimality conditions. The last section is devoted to studies related to the dual problem of $Mcfdk$.

2 PSD formulations and complexity issues

The delay function we study is known as the Kleinrock’s delay function, see [11]. As previously seen, the transmission delay of a packet on arc $e$ is then given by: $t_e = \frac{\phi_e}{C_e - \phi_e}$. As mentioned by A. Nemirovski in [14] and by A. Ben-Tal and A. Nemirowski in [5] a convex function is representable via positive semidefinite matrices if its epigraph can be expressed with such matrices. As a definition the epigraph of the Kleinrock’s delay function is the set of points which satisfy $t_e \geq \frac{\phi_e}{C_e - \phi_e}$. Thereby:

$$
t_e \geq \frac{\phi_e}{C_e - \phi_e} \iff (t_e + 1)(C_e - \phi_e) - C_e \geq 0 \iff (t_e + 1) \frac{\sqrt{C_e}}{\sqrt{C_e}} (\phi_e) \geq 0$$

(1)

Remark that since the diagonal entries of a positive semidefinite matrix are nonnegative the positive semidefinite constraint implies that the flow circulating over the arc $e$ would be less or equal to the capacity of this arc, this yet shows that the capacity constraints are unnecessary. According to the representation of the delay function with a positive semidefinite matrix we can give the following formulation of the problem $Mmdf$:

$$Mmdf = \begin{cases} 
\text{Min} & \sum_{e \in E} t_e \\
\text{s.t.} & \sum_{p \in \Pi} \phi_p \geq d \\
& \left( t_e + 1 \frac{\sqrt{C_e}}{\sqrt{C_e}} \right) \geq 0 \forall e \in E \\
& \phi_p \geq 0 \forall p \in \Pi 
\end{cases}$$

Lemma 2.1 If there exists at least a path linking the the source and the sink of the flow with a non-zero remaining capacity, then for all positive $\epsilon$ the problem $Mmdf$ can be solved in polynomial time in $|V|, |E|$ and $\frac{1}{\epsilon}$ with an error no greater than $\epsilon$.

In the remainder of this article we refer to Slater’s condition as it is set in [10], see definition 2.3.1 on page 311 and theorem 2.3.2 on page 312.
Proof:
The number of variables handled by the problem \(Mmd\) equals the number of paths that exist between the source and the sink of the demand. The number of constraints is exactly \(|E| + 1\), thus the size of the problem grows exponentially with the size of the graph. Let us consider the dual of \(Mmd\) which is as follows:

\[
Mmd^* = \begin{dcases}
\text{Max } \alpha d - \sum_{e \in E} (x_e + C_c y_e + 2\sqrt{C_c} z_e) \\
\alpha - \sum_{e \in p} y_e \leq 0 \forall p \in \Pi \\
x_e \leq 1 \forall e \in E \\
\begin{pmatrix} x_e + 1 \\ z_e \\ y_e \end{pmatrix} \succeq 0 \forall e \in E
\end{dcases}
\]

As done in [21] we will show that both problems \(Mmd\) and its dual \(Mmd^*\) satisfy Slater’s condition and then reach the same common optimal value. It is easy to check that the incidence vectors of elementary paths (paths containing no loop) form an independent set of vectors, thus Slater’s condition would be satisfied if there exists a strictly feasible solution of \(Mmd\). By hypothesis we know that there exists at least a path with remaining capacity on all its arcs, then in order to get a strictly feasible solution it suffices to move from saturated paths a small amount of flow to the unfilled path. Thus we can say that the interior of the feasible region of \(Mmd\) is non-empty. It follows that \(Mmd\) and \(Mmd^*\) reach a common optimal value, the regions on which they are defined are bonded. Then the feasible region of \(Mmd\) is contained in a ball with radius \(R\) and contains a ball with radius \(r\). The dual problem \(Mmd^*\) contains an exponential large number of constraints related to the paths, but one can check that the separating problem over this class of inequalities can be solved by computing a shortest path, thus the separation problem can be solved in a polynomial time. Thereby this positive semidefinite program can be solved with an error no greater than \(\epsilon > 0\) in a polynomial time of the size of the program and \(\frac{1}{\epsilon}\), see for instance [2]. This shows that \(Mmd^*\) can be solved in polynomial time with an error no greater than \(\epsilon\) and the same arises for the problem \(Mmd\).

Before we leave the problem \(Mmd\) let us point out an interesting fact. Previously we said that methods using differentiability have huge numerical problems if the load of the network is close to 1. Remark that this is not the case when we use positive semidefinite programming. Note that the capacity condition, in our case are simply preserved because of the non negativity of the diagonal entries of the matrix, \(C_e - \phi_e \geq 0\).

Now let us set some complexity results about the problem \(Mcfbd\). As previously said the set \(\Pi \leq \tau\) may be exponentially large in the size of the graph, thus the complexity of the problem would depend on the complexity of the column generation problem.

**Lemma 2.2** If \(t_e = 1\ \forall e \in E\) (this means that all the arcs have the same crossing delay) the problem \(Mcfbd\) can be solved in a polynomial time of the size of the graph, else it is an \(\mathcal{NP}\)-hard problem.

Proof: As previously done consider the dual problem of \(Mcfbd\):
\[
Mcfd = \begin{cases}
\text{Min} & \sum_{e \in E} k_e \sum_{p \ni e} \phi_p \\
\text{s.t.} & \sum_{p \ni \Pi} \phi_p \geq d \\
& \sum_{e \in p} t_e \leq \tau, \forall p \in \Pi \\
& \left( \frac{t_e + 1}{\sqrt{C_e}} C_e - \phi_e \right) \geq 0 \forall e \in E
\end{cases}
\]

where \( \phi_e = \sum_{p \ni e} \phi_p \). As previously seen the capacity constraints are useless. One more difficulty arises than those seen with the problem \( Mcfbd \) due to the fact that the delay is a function of the current flow: it is that it is impossible to compute a priori a set like \( \Pi \leq \tau \). Moreover one can check that a path may have a non zero delay although it does not carry any flow, it suffices that one of its edge is shared by another path with a positive flow in order to have a positive delay on this edge, hence on all the path sharing this edge.

Since the number of paths between any two vertices of a graph is exponentially large the number of variables of the problem \( Mcfdk \) is also exponentially growing with the size of the graph. The numbers of constraints related to the delay on one edge is polynomial, as many as the number of edges, but the number of delay constraints may be exponentially large. Remark that when we generate a path we generate also the related delay constraint, therefore by the \( SEP = OPT \) theorem, \cite{9}, the complexity of \( Mcfbd \) would be of same order than those of the related column generation problem. We will discuss about it in the next section where we propose a column generation algorithm in order to solve this problem.

3 Optimality conditions for Mcfkd

The column generation implies we can express reduced cost, thereby we have to formulate the dual problem of \( Mcfdk \) and be sure that no duality gap arises. After writing \( Mcfdk \) in canonical form:

\[
Mcfbd^* = \begin{cases}
\text{Max} & ad - \sum_{e \in E} C_e \lambda_e \\
& \alpha - \sum_{e \in p} \lambda_e \leq \sum_{e \in p} k_e, \forall p \in \Pi \leq \tau \\
& 0 \leq \lambda_e \forall e \in E, 0 \leq \alpha
\end{cases}
\]
form we check that $Mcfdk$ satisfies Slater’s condition, see [20], which makes us sure that primal and dual have finite optima and reach the common optimal value, and then we give a formulation of its dual program $Mcfdk^*$. Let $A_d$ be the matrix for the demand constraint, $A_{tp}$ points out the matrix related to the delay constraint along path $p$ and $A_{Ce}$ denotes the matrix associated with capacity constraint on arc $e$. The reader will find a description of these matrices in the annex. Since the costs are non-negative we assume that no more than the demand would be routed. As detailed in the annex we introduce slack variables : $b_j$, $j \in \{1, \ldots, |\Pi|\}$ are related to delay constraints and $\eta^e_2$, $e \in E$ are related to capacity constraints. Then the problem $Mcfdk$ can be rewritten in the following way :

$$Mcfdk = \begin{cases} \min & C \cdot X \\ \text{s.t.} & A_d \cdot X = d \\ & A_{tp} \cdot X = \tau, \ \forall p \in \Pi \\ & A_{Ce} \cdot X = C_e \ \forall e \in E \\ & A_{\sqrt{C_e}} \cdot X = 2\sqrt{C_e} \ \forall e \in E \\ & X \succeq 0 \end{cases}$$

Now we check the fact that $Mcfdk$ and its dual $Mcfdk^*$, which is defined further, satisfy Slater’s condition.

**Lemma 3.1** If there exists a path $p$ which does not tight the delay constraint, then the problem $Mcfdk$ has a nonempty relative interior to the affine space $A_d \cdot X = d$.

**Proof:**

If there exits a path which does not fulfill the delay constraint and if all the others satisfy this constraint with equality, one can remove a fraction of flow from each of these paths and push it on the path which does not fulfill the delay constraint. Thus we may say that there exists a solution where all the slack variables corresponding to the delay constraints, denoted by $b_j$, $j \in \{1, \ldots, |\Pi|\}$ in the annex, are positive. Without loss of generality we may consider that none of the flows $\phi_p$ equals 0. If it is the case we may remove such a path from the set $\Pi$. As well we may suppose that the delay on each arc is positive. Moreover the remaining capacity on edge $e$, denoted by $\eta^e_2$ in the annex, cannot be equal to 0 otherwise the delay on arc $e$ is unbounded and the delay constraints related to each path which contains $e$ cannot be satisfied. Then we may say that if there exists a path which satisfy the delay constraint with strict inequality we may construct a matrix $X \succeq 0$ with all its diagonal entries positive therefore $X > 0$.

In order to define the dual program of $Mcfdk$ we associate a variable $\alpha$ to the demand constraint, a variable $\beta_p$ to each delay constraint along a path $p$ and a matrix $\begin{pmatrix} x_e & z_e \\ z_e & y_e \end{pmatrix}$ to the constraint related to arc $e$. Then the dual program can be formulated as below :

$$Mcfdk^* = \begin{cases} \max & ad - \tau \sum_{p \in \Pi} \beta_p - \sum_{e \in E} (x_e + C_e y_e + 2\sqrt{C_e} z_e) \\ \text{s.t.} & \alpha - \sum_{e \in p} y_e \leq \sum_{e \in p} k_e, \ \forall p \in \Pi \\ & -\sum_{p \ni e} \beta_p + x_e = 0, \ \forall e \in E \\ & \begin{pmatrix} x_e & z_e \\ z_e & y_e \end{pmatrix} \succeq 0, \\ & \alpha \geq 0, \ \beta_p \geq 0 \ \forall p \in \Pi \end{cases}$$

(2)
Corollary 3.1 The programsMcfdk andMcfdk* have a common bounded optimal value.

Proof:  
One can check easily that the matrices \(A_d, A_C, A_p\) and \(A_{\sqrt{C}}\) form an independent set of vectors in the space of the symmetric matrices. Then by lemma 3.1 we can say that Mcfdk and Mcfdk* satisfy the Slater’s condition, hence the corollary follows.

In this section we look for optimality conditions for the problem Mcfdk and conclude about its complexity. In order to establish optimality conditions we first show some property the optimal solutions of the dual Mcfdk*.

Lemma 3.2 Each optimal solution of the dual program Mcfdk* satisfies \(y_e = x_e(C_e - \Phi_e)^{1/2}\). Moreover if all the paths crossing arc \(e\) do not tight the delay constraint we have \(x_e = y_e = 0\).

Proof:  
We first observe looking Mcfdk* that for each \(e\), \(z_e\) satisfies \(z_e^2 \leq x_e y_e\). Moreover this is the only constraint applying to \(z_e\) therefore the objective sets \(z_e = \sqrt{x_e y_e}\). Applying complementary slackness we have

\[
(x_e + 1)^{1/2} C_e y_e = 0,
\]

which means

\[
x_e(t_e + 1) - \sqrt{C_e} x_e y_e + y_e(C_e - \Phi_e) = 0.
\]

But necessarily at least one optimal primal solution will tight the delay constraints and verifies \(t_e + 1 = C_e/(C_e - \Phi_e)\). Therefore

\[
x_e C_e/(C_e - \Phi_e) - \sqrt{C_e} x_e y_e + y_e(C_e - \Phi_e) = 0.
\]

We conclude by noticing that

\[
x_e C_e/(C_e - \Phi_e) - 2\sqrt{C_e} x_e y_e + y_e(C_e - \Phi_e) = \frac{(\sqrt{x_e C_e} - \sqrt{y_e(C_e - \Phi_e)})^2}{C_e - \Phi_e},
\]

which means \(y_e = x_e C_e/(C_e - \Phi_e)^2\).

Using complementary slackness we may say that \(\beta_p = 0\) if path \(p\) does not tight the delay constraint, thus \(\sum_{p \ni e} \beta_p = x_e = 0\).

Complementary slackness also leads to the following theorem characterizing optimal solutions of Mcfdk.

Theorem 3.1 Let \(\Pi_j\) be the set of paths which defines the problem Mcfdk after generating \(j\) variables of path \(\phi_p\). \(\phi^*_j\) denotes an optimal solution of the problem defined in this way. If all \(p \in \Pi - \Pi_j\) such that \(\sum_{e \in p} t_e < \tau\) satisfy \(\alpha \leq \sum_{e \in p} (y_e + k_e)\) then \(\phi^*_j\) is an optimal solution for Mcfdk.

Proof:  
If we have \(\alpha \leq \sum_{e \in p} (y_e + k_e)\) for all path \(p\) we may say that we reach an optimal dual solution, then by strong duality theorem the corresponding primal solution is optimal.
Corollary 3.2 The problem Mcfdk is NP-hard.

Proof:
By the previous theorem we may say that the column generation problem is equivalent to compute a shortest path according to costs \( k \) and such that the delay is bounded by \( \tau - \epsilon \) in a graph where the arcs are valued by the delays which are not equal (for instance when the flow is small enough the delay on arc \( e \) is \( \frac{1}{x_e} \)), this problem is known to be NP-complete, see [8, problem ND 30].

It follows from theorem 3.1 that the column generation problem reduces to a resource constrained shortest path which can be formulated as shown below.
Let \( \gamma_e \) be the reduced cost of arc \( e \) at the current step of the column generation algorithm, \( t_e \) denotes the amount of resource consumed on arc \( e \). Then the resource constrained shortest path problem is written in the following way if \( x_e \) equals 1 if arc \( e \) belongs to the path and 0 otherwise:

\[
\text{Rcsp} = \begin{cases} 
\min & \sum_{e \in E} \gamma_e x_e \\
\text{s.t.} & \sum_{e \in \delta(S)} x_e \geq 1, \forall S \subset V, s \in S, t \notin S \\
& \sum_{e \in E} t_e x_e \leq \tau \\
& x_e \in \{0, 1\}, \forall e \in E
\end{cases}
\]

J.E. Basley and N. Christofides designed an algorithm to solve this problem which based on Lagrangian relaxation where the delay constraints are dualized, see [4].

4 Karush Kuhn Tucker conditions and dual bounds

In this section we investigate Karush Kuhn Tucker optimality conditions in order to achieve new formulations of the dual problem and define some relaxations of it. We start from an alternative convex formulation of Problem Mcfdk*

\[
\text{Mcfdk2*} = \begin{cases} 
\min & -ad + \tau \sum_{p \in \Pi} \beta_p + \sum_{e \in E} (\sqrt{x_e} - \sqrt{C_e y_e})^2 \\
\text{s.t.} & \alpha - \sum_{e \in p} y_e - \sum_{e \in p} k_e \leq 0, \forall p \in \Pi \\
& -\sum_{p \ni e} \beta_p + x_e = 0, \forall e \in E \\
& \alpha \geq 0, \beta_p \geq 0 \forall p \in \Pi \\
& x_e \geq 0, y_e \geq 0 \forall e \in E
\end{cases}
\]

First let us recall the Karush-Kuhn-Tucker conditions. Consider a convex function \( f \) to be minimized over a convex set defined by constraints with form \( g_i \leq 0, \forall i \in I \), where \( g_i \) are convex function. Moreover suppose that both functions \( f \) and \( g_i \) are differentiable in a point \( x^* \). Then point \( x^* \) is an optimal solution of
The problem

\[
\begin{aligned}
\{ & \text{Min } f(x) \\
& \text{s.t. } g_i(x) \leq 0, \ \forall i \in I \\
\end{aligned}
\]

if and only if there exists \((u, u_0) \in \mathbb{R}^{n^+} \times \mathbb{R}^+\) such that \(u_0 \nabla f(x^*) + \sum_{i \in I} u_i \nabla g_i(x) = 0\) and \(u_i g_i(x) = 0 \ \forall i \in I\). Thus writing the Kuhn and Tucker conditions for the dual problem of \(Mcfbd\) and if \(g_p\) denotes a constraint related to the path \(p\) and \(g_e\) denotes a constraint related to edge \(e\) we obtain:

\[
\begin{aligned}
\frac{\partial f}{\partial p} &= -d \\
\frac{\partial f}{\partial x} &= 1 - \sqrt{\frac{C_{ye}}{2}} \\
\frac{\partial f}{\partial u} &= C_e(1 - \sqrt{\frac{C_{ye}}{2}})
\end{aligned}
\]

\[
\nabla g_e = \begin{cases} 
0 & \text{if } e \neq e_0 \\
1 & \text{if } e = e_0 \\
-1 & \text{if } p \supset p_0
\end{cases}
\]

\[
\nabla g_p = \begin{cases} 
1 & \text{if } e \in p \\
-1 & \text{if } e \notin p
\end{cases}
\]

Using these equations we can easily show that:

**Theorem 4.1** The problem \(Mcfdk\) as the same optimal value as:

\[
\begin{aligned}
\text{Max } & \alpha d - \sum_{p \in \Pi} \beta_p \left[ \frac{\tau + \sum_{e \in p} t_e^2}{C_e} \right] \\
\text{s.t. } & \alpha - \sum_{e \in P} k_e + \frac{(t_e + 1)^2}{C_e} \sum_{k \geq e} \beta_k \forall p \\
& \alpha \geq 0, \beta_p \geq 0 \ \forall p \in \Pi, \ t_e \geq 0 \ \forall e \in E
\end{aligned}
\]

(4)

Proof: Obviously if \(u = 0\) then \(u_i = 0 \ \forall i\) due to the Kuhn-Tucker conditions. Then we set \(t_{e_0} = u_{e_0}/u\) and \(\varphi_p = u_p/u\). The gradient condition shows that \(t_e = \sqrt{\frac{C_{ye}}{x_e}} - 1\) therefore \(x_e = \frac{C_{ye}}{(t_e + 1)^2}\) and \(y_e = \left(\sum_{p \ni e} \beta_p\right)\frac{(t_e + 1)^2}{C_e}\). The above problem is obtained by replacing \(x_e\) and \(y_e\) in \(Mcfdk2^*\), and therefore each solution given by \((x_e, e \in E, \ y_e, e \in E, \ \beta_p, p \in \Pi)\) induces a feasible solution in \((\tau, e \in E, \ \beta_p, p \in \Pi)\) in \(Mcfdk.tr\).

Conversely the solution of \(Mcfdk2^*\) can be derived from \(Mcfdk.tr\) by setting \(x_e = \frac{C_{ye}}{(t_e + 1)^2}\) and \(y_e = \left(\sum_{p \ni e} \beta_p\right)\frac{(t_e + 1)^2}{C_e}\). The reader can verify that the conditions of \(Mcfdk2^*\) are fulfilled.
Notice that any two cuts $\delta(S)$ and $\delta(S')$ define the same relaxation because $\sum_{p \in \Pi} \beta_p = \sum_{e \in \delta(S)} x_e = \sum_{e \in \delta(S')} x_e$.

5 Conclusion

Semi-definite programming may lead to an efficient way in order to solve flow problems with additional constraints representing some grade of service criteria since each of them has an asymptote and thus meet some computational difficulties when the flow is close to the capacity. The results that we obtain about flow problems can be extended to the case of the multicommodity flow problems, in particular for the problem $Mcfdk$. Indeed the positive semi-definite matrices used in order to express the delay have the same form but $\phi_e = \sum_{(st) \in D} \sum_{p \in \Pi_{st}} \sum_{e \in p} \phi_p$, where $D$ is the set of demands and $\Pi_{st}$ denotes the set of paths linking the source $s$ and the sink $t$. The optimality conditions of the theorem 3.1 have to be changed as follows. Let $\Pi_{st}^j$ be the set of paths linking $s$ and $t$ which defines the problem after $j$ iterations, if all $p \in \Pi_{st}^j - \Pi_{st}^{j-1}$ such that $\sum_{e \in p} t_e < \tau$ satisfy $\alpha_{st} \leq \sum_{e \in p} (y_e + k_e)$ and if this holds for any demand $(st) \in D$, then the solution obtained after $j$ iterations is optimal.

Now we want to investigate a non over constrained version of the problem $Mcfdk$, we mean the version mentioned previously in which only the paths carrying flow support a delay constraint. For this professor Gérard Cornuéjols suggested to us using disjunctive programming. Indeed, consider a path $p$ then we can consider on one hand a problem $Mcfdk_{p=0}$ defined in the hyperplane $\phi_p = 0$ and in which there is no delay constraint related to path $p$, and on the other hand a problem $Mcfdk_{p>0}$ defined in the half space $\phi_p > 0$ in which the path $p$ support the delay constraint. The optimal solution for this disjunction is reached on the convex hull of these two disjoint regions. This would refer to the work of E. Balas [3].

References


Here we give an explanation of the matrices used in order to express the problem Mcfdk in canonical form. In order to yield such an expression of the problem we need to replace inequalities by equalities, thus we will use slack variables when necessary. Thus the constraint \( \sum_{p \in \Pi} \phi_p \geq d \) becomes \( \sum_{p \in \Pi} \phi_p = d \) and the delay constraint along path \( p \) becomes \( \sum_{e \in p} t_e + b_p = t \), at last the capacity constraint may be written in the following way \( \eta_e^2 = C_e - \sum_{p \ni e} \phi_p \). This leads to write the matrix of the variables as below:

\[
X = \begin{pmatrix}
    b_1 & \cdots & b_{|\Pi|} \\
    \vdots & \ddots & \vdots \\
    \eta^1_{e_1} & \cdots & \eta^1_{e_{|E|}} \\
    \eta^2_{e_1} & \cdots & \eta^2_{e_{|E|}} \\
    \vdots & \ddots & \vdots \\
    \phi_1 & \cdots & \phi_{|\Pi|}
\end{pmatrix}
\]

Now we investigate the structure of each matrix representing the linear constraints. As previously seen \( A_d \) is the matrix useful to express demand constraint, it is in the following form:

\[
A_d = \begin{pmatrix}
    0 & \cdots & 0 & 0 & 0 \\
    \vdots & \ddots & \vdots & \vdots & \vdots \\
    0 & \cdots & 0 & 1 & \cdots \\
    0 & \cdots & 0 & 1 & \cdots \\
    1 & \cdots & 1 & \cdots & 1
\end{pmatrix}
\]

Notice that all the entries of this matrix are zero except those corresponding to the flow variables of the matrix \( X \) which equal 1, thus the inner product of the matrices \( X \) and \( A_d \) yields the demand constraint.

Remember that we called \( A_p \) the matrix related to the delay constraint along path \( p \), this one is as follows:
where $A_{e_i} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ if arc $e_i$ belongs to path $p$ and $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ otherwise. These matrices are corresponding to the blocks $t_{e_i} \begin{pmatrix} \eta_{e_1}^1 \\ \eta_{e_2}^1 \end{pmatrix}$ of the matrix $X$. All the others entries are zero excepted the one corresponding to the slack variable $b_p$ in $X$.

The matrix $A_{C_e}$ associated with the capacity constraint on the edge $e$ is organized as described below:

\[
A_{C_e} = \begin{pmatrix}
0 & & & & \\
& 0 & & & \\
& & 0 & & \\
& & & 0 & 0 \\
& & & & 0 1 \\
& & & & & \cdots \\
& & & & & & \delta_e^{p_1} \\
& & & & & & & \ddots \\
& & & & & & & & \cdots \\
& & & & & & & & & \delta_e^{p_{|P|}} \\
\end{pmatrix}
\]

where $\delta_e^p$ equals 1 if arc $e$ belongs to path $p$ and 0 otherwise. In order to express constraints related to variables $\eta_e^1$ we need the matrix $A_{\sqrt{C_e}}$ which has the following form:
Let $C$ be the matrix representing the objective function, it has the following structure:

$$
A_{\sqrt{\tau}} = \begin{pmatrix}
0 & & & & & \\
\vdots & \ddots & & & & \\
0 & & \ddots & & & \\
1 & 0 & & \ddots & & \\
\vdots & \ddots & & & \ddots & \\
0 & & & & & \ddots \\
\end{pmatrix}
$$

$$
C = \begin{pmatrix}
0 & & & & & \\
\vdots & \ddots & & & & \\
0 & & \ddots & & & \\
0 & & & \ddots & & \\
\vdots & \ddots & & & \ddots & \\
0 & & & & & \ddots \\
\end{pmatrix}
$$

where $c_{p_j} = \sum_{e \in P_j} k_e$. 