***k*-Splittable Delay Constrained Routing Problem: A Branch-and-Price Approach**

Jérôme Truffot, Christophe Duhamel, and Philippe Mahey

LIMOS, UMR 6158-CNRS, Université Blaise-Pascal, BP 10125, Aubière 63173, France

Routing problems, which include a QoS-based path control, play a key role in broadband communication networks. We analyze here an algorithmic procedure based on branch-and-price algorithm and on the flow deviation method to solve a nonlinear *k*-splittable flow problem. The model can support end-to-end delay bounds on each path and we compare the behavior of the algorithm with and without these constraints. The trade-off between QoS guarantees and CPU time is clearly established and we show that minimizing the average delay on all arcs will yield solutions close to the optimal one at a significant computational saving. © 2009 Wiley Periodicals, Inc. NETWORKS, Vol. 00(00), 000–000 2009

Keywords: end-to-end delay; convex constraint; flow deviation; column generation; branch-and-price

1. INTRODUCTION

We consider here a new algorithmic approach for the *k*-splittable flow problem with end-to-end delay constraints, the (*k*-DCRP) problem, which aims at solving routing problems in modern broadband communications networks. In contrast with classical routing models, the (*k*-DCRP) model takes in consideration two characteristics, which are present in the general design of MPLS networks (see for instance [4]), namely: the need to limit the number of LSPs (Label-switched Paths) used to route the flows between all O-D pairs, and the introduction of end-to-end delay bounds for each commodity. Both these difficulties have been treated in the recent literature, but separately. The *k*-splittable flow problem has been first presented by Baier et al. [2] and then analyzed by Martens and Skutella [12]. The basic idea is to consider a compromise situation between the splittable and

unsplittable cases. Indeed, many different LSPs may be used for a given pair of origin-destination, but that number should remain small for technical reasons. In [4], the authors decide to minimize the number of LSPs, leading to a model which is close but different from the *k*-DCRP.

We consider here a *k*-splittable flow problem where the choice of at most *k* paths to route a given demand is determined to yield a minimal end-to-end delay. Even if the algorithm can be easily extended to multicommodity flow problems, we will limit the numerical study to single-commodity flows. To our knowledge, there is only one paper which considers an approximate solution to the case of multicommodity flows by separating the delay minimization from the control on the maximal number of paths; it is proposed by Duhamel and Mahey in [9], where they analyzed an adaptation of the Flow Deviation method for convex cost multicommodity flow including a control mechanism on the number of active paths.

The general *k*-splittable flow problem was shown to be NP-hard for all sufficiently small values of *k*¹. Martens and Skutella have defined the *k*-splittable flow as a single-commodity flow problem supported by a set of at most *k* distinct paths, but not necessarily disjoint. Besides the analysis of the main properties of *k*-splittable flows, they have proposed polynomial approximation schemes, but, except for the classical unsplittable flow problem which corresponds to the case *k* = 1 (see [1, 11] for example), no exact or even heuristic schemes have been tested until the proposition of a branch-and-price algorithm in [14] to solve the *k*-splittable maximum flow problem. An adaptation of the algorithm developed in [14] is used here as an internal subproblem to solve the *k*-DCRP.

Received November 2006; accepted July 2008

Correspondence to: C. Duhamel; e-mail: christophe.duhamel@isima.fr

Contract grant sponsor: Centre National de la Recherche Scientifique, CNRS (ACI-PRESTO Project)

DOI 10.1002/net.20311

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

© 2009 Wiley Periodicals, Inc.

¹ Indeed, if *k* is greater or equal than the number of paths given by the corresponding polynomial unconstrained flow problem, the problem is obviously polynomial.

As the end-to-end delay constraints are nonlinear and as they set a constraint on each active path, their introduction in the routing model turns it even more untractable. Practical simplifications are found in the literature since the pioneer work of Fratta et al. [10] by minimizing an average measure of the delay on all active arcs. This could lead to unrealistic solutions where too many paths (including those made of a concatenation of active subpaths) are penalized. As one does not know in advance which paths are going to carry some positive flow at the optimal solution, it is necessary to introduce boolean variables associated with the choice of each path in the nonlinear delay functions and this explains why few realistic models have been presented in the literature. The most notable work has been done by Ben-Ameur and Ouorou [5] where they define and model the Delay Constrained Routing Problem. They show that this problem is NP-hard and they propose several bounds by progressive convexifications of the delay constraints.

The main contribution is to show here how the modeling of k -splittable flows allows to take in consideration end-to-end delay measures on each active path used to route the flow. In comparison, classical routing models use an average delay measure induced by a convex approximation of the average congestion among all arcs carrying flow (see [10]). These models cannot give realistic guarantees for a QoS-based routing model and this was the main motivation for studying k -DCRP. However, quite surprisingly, the numerical comparisons between k -DCRP and the corresponding problem with an average delay measure (called below k -MADP) have shown the improvement of the solution relatively small, at a high computational cost.

In Section 2 a mathematical formulation for (k -DCRP) is proposed. Then, in Section 3 we describe how to solve (k -DCRP) by using a branch-and-price algorithm. The subproblem is solved by an approximation scheme similar to the idea proposed by Shahrokhi and Matula [13] and later developed by Bienstock [7]. The inner part of this approximation schema is a routing problem, solved by a Frank–Wolfe algorithm. In Section 4 numerical results are presented on small instances. The results will be compared with those provided by (k -MADP) before concluding.

2. MATHEMATICAL FORMULATION

Let $G = (N, A)$ be a digraph where N is the set of n nodes and A is the set of m arcs. Each arc $a \in A$ is given a capacity $u_a \geq 0$. We consider here a single commodity, i.e. an origin-destination pair (s, t) and a demand d to be sent from s to t on G . Let $\mathcal{P}$ be the set of elementary (s, t) -paths. The delay constrained routing problem (DCRP) consists in routing the requested demand while minimizing the maximal end-to-end delay. Observe that, in the design of MPLS networks with

QoS-based routing decisions, one wishes to guarantee an end-to-end-delay lower than a given threshold $\bar{\theta}$. Minimizing the maximal delay on all paths is a much more exigent objective as it will answer to a more precise question: what is the best delay that can be guaranteed for routing a given demand on a given capacitated network? In the k -DCRP model, the corresponding question will be: what is the best delay that can be guaranteed for routing a given demand on at most k routes?

For each arc $a \in A$, let $x_a \geq 0$ be the flow on it. The delay θ_p suffered by the data packets on a given active route p can be approximated by

$$\theta_p = \sum_{a \in p} \frac{\lambda}{u_a - x_a} \quad (1)$$

where λ is the average packet size (see [4, 10] for a description of the classical assumptions on the traffic to get that approximation, mostly referred as the Kleinrock's delay function).

Rather than choosing active paths from $\mathcal{P}$, one can describe each active path as a flow sub-problem by using arc-node formulation. Such an approach is well suited when considering a limit on the number of active paths in the routing solution [4]. This constraint is known to define the k -splittable flow [2]. Although these authors showed that it turns most network flow problems NP-hard, a few references were dedicated to exact algorithms. In [14], a branch-and-price method is used to solve the k -splittable maximum flow problem. Let H be the maximal number of active paths allowed to carry out the traffic. Thus, the k -splittable Delay Constrained Routing Problem (k -DCRP) consists in routing a delay constrained flow with at most H paths ($H = k$).

As there are H potentially active paths, each one is described as a flow subproblem. For each path number $h = 1 \dots H$, let $x_a^h \geq 0$ be the flow variable on the arc $a \in A$, let $y_a^h \in \{0, 1\}$ be the associated decision variable and let $d_h \geq 0$ be the associated amount of flow. Thus the decision variables y_a^h indicate the arcs a used by the active path indexed by h . This path carries a flow represented by the flow variables x_a^h . So this flow is constrained by the decision variables to use only the arcs for which $y_a^h = 1$.

Let $\delta^-(v)$ be the set of node v incoming arcs and let $\delta^+(v)$ be the set of node v outgoing arcs. For convenience, let x_a be the aggregated flow on node a , that is $x_a = \sum_{h=1}^H x_a^h$. Finally let θ be the maximal end-to-end delay over the routing. Then the arc-node model can be stated as follows:

$$\begin{aligned}
& \left\{ \begin{array}{l} \min \theta \\ \text{s.t.} \\ \sum_{a \in \delta^-(v)} x_a^h - \sum_{a \in \delta^+(v)} x_a^h = \begin{cases} -d_h & \text{if } v = s, \\ d_h & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases} \quad \forall h = 1 \dots H, \forall v \in N \quad (a) \\ \sum_{a \in \delta^-(v)} y_a^h - \sum_{a \in \delta^+(v)} y_a^h = \begin{cases} -1 & \text{if } v = s, \\ 1 & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases} \quad \forall h = 1 \dots H, \forall v \in N \quad (b) \\ x_a^h - u_a y_a^h \leq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (c) \\ x_a = \sum_{h=1}^H x_a^h \leq u_a \quad \forall a \in A \quad (d) \\ \sum_{h=1}^H d_h = d \quad (e) \\ \sum_{a \in \delta^-(v)} y_a^h \leq 1 \quad \forall v \in V \quad \forall h = 1 \dots H \quad (f) \\ \sum_{a \in A} \frac{\lambda y_a^h}{u_a - x_a} \leq \theta \quad \forall h = 1 \dots H \quad (g) \\ x_a^h \geq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (h) \\ y_a^h \in \{0, 1\} \quad \forall a \in A \quad \forall h = 1 \dots H \quad (i) \\ d_h \geq 0 \quad \forall h = 1 \dots H \quad (j) \end{array} \right. \quad (k - \text{DCRP}) \quad (2)
\end{aligned}$$

This model involves two flow conservation blocks (namely constraints 2(a) and 2(b)) for each path. The indicator variables y^h are needed to prevent multiple paths masquerading under the same x^h variables. In the same spirit, restrictions 2(f) are used to force each path h to be elementary that is, to prevent cycles to be connected to the path. In this manner, this constraints prevent multiple paths using the same y^h (see [14] for more details). 2(c) are the coupling constraints, 2(d) are the capacity constraints and 2(e) is the demand constraint. Finally, constraint 2(g) are the end-to-end delay constraints according to the formulation (1) for the delay.

The fact that we want to minimize the maximal delay on the active paths implies that all H chosen routes will effectively carry a positive fraction of the demand in the optimal solution. The introduction of path flow variables x_a^h is necessary because, modeling the flow constraint directly on the aggregate flow variables x_a , we may force the flow on a new path which is a combination of several active paths. This will thus violate the constraint on the maximal number of authorized paths.

This model is a mixed integer non linear model, as constraints 2(g) are nonlinear. The problem (k -DCRP) is NP-hard as a generalization of the (DCRP) problem presented by Ben-Ameur and Ouorou [5]. It is also a generalization of the k -splittable flow problem presented by Baier et al. [2]. Observe too that, as commented at the beginning of the section, in practice, one will stop the iterative process as soon as $\theta \leq \bar{\theta}$.

As mentioned previously, minimizing the average delay on a k -splittable flow (k -MADP) can provide a reasonable approximation even if some end-to-end delays constraints

might be violated. The model for (k -MADP) is as follows

$$(k - \text{MADP}) \left\{ \begin{array}{l} \min \psi(x) = \sum_{a \in A} \frac{x_a}{u_a - x_a} \\ \text{s.t.} \\ (a), (b), (c), (d), (e), (f), (h), (i), (j) \end{array} \right. \quad (3)$$

When comparing with (k -DCRP), nonlinear end-to-end delay constraints (g) are dropped in (k -MADP).

3. A BRANCH-AND-PRICE STRATEGY TO SOLVE K -DCRP

The model (k -DCRP) involves both nonlinear constraints and mixed integer variables. We propose in this section an adaptation of the Branch-and-Price method to cope with both difficulties. Branch-and-Price, as proposed early by Barnhart et al. [3] is a combination of Branch-and-Bound enumeration strategy with pricing (or Column Generation). To handle the k -splittability, a path generation scheme will be applied. It is based on a Dantzig-Wolfe reformulation of the model (k -DCRP). Therefore, each column corresponds to a (s, t) -path and branchings are used to project back towards integer solutions. Observe that the model for k -DCRP uses additional boolean variables y^h which are necessary to characterize that route h is indeed active in the current solution. The branching strategy will rely on our capacity to solve continuous relaxations of the model. We will show below how this can be done by solving a convex cost k -splittable flow problem, where the convex cost iteratively approximates the original maximum delay objective. That subproblem is solved

by a potential-reduction algorithm of the Frank-Wolfe's type which integrates the path generation (see Bienstock's book on potential-reduction algorithms [7]). Then, a post-optimization step is performed on each integer solution found during the branch-and-price. The algorithm will stop when no branching is found to improve the best upper bound given by these feasible integer solutions so that the optimal solution of (k -DCRP) is obtained. A description of the flowchart of the algorithm has been reported in Appendix A.

A branch and price scheme will be used to solve this problem. Thus, the integrity constraints on variables y_a^h are relaxed and the problem becomes a nonlinear problem.

3.1. Fractional Solution

We first simplify the expression of the end-to-end delay. Variables y_a^h are replaced by the ratio x_a^h/u_a in constraints 2(g). Indeed, $x_a^h > 0$ only if $y_a^h = 1$ and $x_a^h/u_a < 1$. We call (CR) the continuous relaxation of (k -DCRP) where constraints 2(g) are approximated by

$$\sum_{a \in A} \frac{\lambda x_a^h}{u_a(u_a - x_a)} \leq \theta \quad (4)$$

Thus, $[\theta_{\min}]_{(CR)} \leq [\theta_{\min}]_{(k-DCRP)}$.

$$(CR) \begin{cases} \min \theta \\ \text{s.t.} \\ 2(a), 2(b), 2(c), 2(d), 2(e), 2(f) \\ \sum_{a \in A} \frac{\lambda x_a^h}{u_a(u_a - x_a)} \leq \theta \quad \forall h = 1 \dots H \quad (g) \\ x_a^h \geq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (h) \\ y_a^h \geq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (i) \\ d_h \geq 0 \quad \forall h = 1 \dots H \quad (j) \end{cases} \quad (5)$$

This maximal end-to-end delay minimization problem (CR) corresponds to a min-max "packing" problem. We propose in the next section to solve this problem approximately by using an exponential potential function method [7, 15] based on the Shahrokhi and Matula algorithm [13] for the maximum concurrent flow problem. A similar approach was used in [9] where it was observed that the minimization of the congestion leads to an increase of the number of active paths.

3.2. A Potential Reduction Method to Solve (CR)

Let S be the set $\{x \in \mathbb{R}_+^{A \times \{1, \dots, H\}} \mid \sum_{h=1}^H x_a^h < u_a \forall a \in A\}$ of strictly feasible "pseudo-flows." Let $\theta^h \geq 0$ be the end-to-end delay function for the problem (CR) and $\theta \geq 0$ be the maximal end-to-end delay defined by

$$\theta_a^h(x) = \frac{\lambda x_a^h}{u_a(u_a - x_a)} \quad \forall x \in S \quad \forall a \in A \quad \forall h \in 1, \dots, H \quad (6)$$

$$\theta^h(x) = \sum_{a \in A} \theta_a^h(x) \quad \forall x \in S \quad \forall h \in 1, \dots, H \quad (7)$$

$$\theta(x) = \max_{h=1 \dots H} \theta^h(x) \quad \forall x \in S \quad (8)$$

The following proposition gives an important property of those functions and it will be used to ensure the correctness of our approach.

Proposition 1. *The function θ_a^h is convex on S for each a in A and for each $h = 1 \dots H$.*

See proof in Appendix B.

Then $\theta(x)$ is approximated by the exponential potential function $\phi(x, \mu)$ with $\mu > 0$.

$$\phi(x, \mu) = \mu \ln \sum_{h=1}^H \exp\left(\frac{\theta^h(x)}{\mu}\right) \quad (9)$$

This function has interesting properties: first it gives a good approximation of the function $\theta(x)$ in the sense that

$$\theta(x) \leq \phi(x, \mu) \leq \theta(x) + \mu \ln H, \quad \forall x \in S, \quad \forall \mu > 0 \quad (10)$$

Moreover, this function is convex on S as shown here.

Proposition 2. *The function $\phi(x, \mu)$ is convex on S for all $\mu > 0$.*

See proof in appendix C.

In addition, the calculation of the gradient of $\phi(x, \mu)$ gives

$$\begin{aligned} \frac{\partial \phi}{\partial x_a^h}(x, \mu) &= \frac{\lambda}{u_a(u_a - x_a)^2} \\ &\times \left(\sum_{i \neq h} v_i(x, \mu) x_a^i + v_h(x, \mu)(u_a - \bar{x}_a^h) \right) \\ &= \frac{\lambda(x_a^h + v_h(x, \mu)(u_a - x_a))}{u_a(u_a - x_a)^2} \\ &= \frac{\theta_a^h(x) + \frac{\lambda}{u_a} v_h(x, \mu)}{u_a - x_a} \end{aligned} \quad (11)$$

So the continuous relaxation (CR) is approximatively solved by minimizing the convex function $\phi(x, \mu)$ with the relaxed k -splittable flow constraints. This problem is expressed as a nonlinear model (CR2) where μ is a parameter.

$$(CR2) \begin{cases} \min \phi(x, \mu) \\ \text{s.t.} \\ 2(a), 2(b), 2(c), 2(d), 2(e), 2(f) \\ x_a^h \geq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (g) \\ y_a^h \geq 0 \quad \forall a \in A \quad \forall h = 1 \dots H \quad (h) \\ d_h \geq 0 \quad \forall h = 1 \dots H \quad (i) \end{cases} \quad (12)$$

The Frank-Wolfe method (indeed corresponding to the flow deviation method) is used to solve (CR2). The Frank-Wolfe algorithm is a linearization technique used to solve approximately convex minimization problems with linear inequality constraints. To resume the basic steps (see [6]), a linear program is solved at each iteration where the cost

is the first-order approximation of the nonlinear objective function estimated at the current point. The solution of the corresponding subproblem, in general a vertex of the polyhedron of the constraints, is used to perform a line search on the segment linking that optimal vertex and the current point. When applied to non linear flow problems, the LP subproblem yields a shortest path computation and the line search

corresponds to the deviation of a uniform quantity of flow from the old active paths to the new shortest path, simplifying dramatically the method in the form of the popular “Flow Deviation” method (see [8, 10]).

At each step i , let x^i be the current solution. The following model (DFSⁱ), associated with the linearization of the objective function ϕ , is the direction-finding subproblem.

$$(DFS^i) \left\{ \begin{array}{ll} \min \nabla \phi(x^i, \mu)^T x & \\ \text{s.t.} & \\ \sum_{a \in \delta^-(v)} x_a^h - \sum_{a \in \delta^+(v)} x_a^h = \begin{cases} -d_h & \text{if } v = s, \\ d_h & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases} & \forall h = 1 \dots H, \forall v \in N \quad (a) \\ \sum_{a \in \delta^-(v)} y_a^h - \sum_{a \in \delta^+(v)} y_a^h = \begin{cases} -1 & \text{if } v = s, \\ 1 & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases} & \forall h = 1 \dots H, \forall v \in N \quad (b) \\ x_a^h - u_a y_a^h \leq 0 & \forall a \in A \quad \forall h = 1 \dots H \quad (c) \\ \sum_{h=1}^H x_a^h \leq u_a & \forall a \in A \quad (d) \\ \sum_{h=1}^H d_h = d & \quad (e) \\ \sum_{a \in \delta^-(v)} y_a^h \leq 1 & \forall v \in V \quad \forall h = 1 \dots H \quad (f) \\ x_a^h \geq 0 & \forall a \in A \quad \forall h = 1 \dots H \quad (g) \\ y_a^h \geq 0 & \forall a \in A \quad \forall h = 1 \dots H \quad (h) \\ d_h \geq 0 & \forall h = 1 \dots H \quad (i) \end{array} \right. \quad (13)$$

A solution of (DFSⁱ) gives a direction along which the linear search finds a better solution. Thus, the following algorithm is used to find a lower bound to the continuous relaxation (CR2). Let μ and ϵ be two positive parameters, and let U be the maximal capacity over the arcs.

Algorithm 1 Continuous Relaxation

Let x^0 be a H -splittable flow routing demand.

repeat

$\bar{x}^i$ denote the solution of (DFSⁱ).

Set $x^{i+1} = (1 - \sigma^i)x^i + \sigma^i \bar{x}^i$, where $\sigma^i = \arg \min_{0 \leq \sigma \leq 1} \{\phi((1 - \sigma)x^i + \sigma \bar{x}^i, \mu)\}$.

Set $i \leftarrow i + 1$

until $\frac{\phi(x^i, \mu) - \phi(x^{i+1}, \mu)}{\epsilon^2 \phi(x^i, \mu)^2} <$

$\frac{32\lambda \left(\frac{UH^3}{\lambda^3} \phi^3(x^i, \mu) + m \right) + \frac{4\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x^i, \mu) + m \right)^2}$

Set the lower bound to $(1 - \epsilon)\phi(x^i, \mu) - \mu \ln H$

The theorem 1 gives us a proof of the approximation.

Theorem 1. Let $\phi^*(\mu)$ be an optimal solution of (CR2) and x^i the flow at the end of Algorithm 1. Then

$$(1 - \epsilon)\phi(x^i, \mu) \leq \phi^*(\mu) \quad (14)$$

See proof in Appendix D.

Observe that (DFSⁱ) is the continuous relaxation of a minimal linear cost k -splittable flow problem. We will solve it by a column generation algorithm.

3.3. Path-Generation

Let (DFS₂ⁱ) be the model obtained after proceeding to a Dantzig–Wolfe decomposition on (DFSⁱ). For each path $p \in \mathcal{P}$, let x_p^h be the flow contribution on path p for the path number h . Let δ_a^p be the indicator vector that identifies which arc $a \in A$ belongs to path p .

Then flow variables x_a^h of (DFSⁱ) are replaced by

$$x_a^h = \sum_{p \in \mathcal{P}} \delta_a^p x_p^h \quad \forall a \in A \quad \forall h = 1 \dots H \quad (15)$$

$$(DFS_2^i) \left\{ \begin{array}{ll} \min \sum_{h=1}^H \sum_{p \in \mathcal{P}} c_p^h x_p^h & \\ \text{s.t.} & \\ \sum_{h=1}^H \sum_{p \in \mathcal{P}} \delta_a^p x_p^h \leq u_a & \forall a \in A \quad (a) \\ \sum_{h=1}^H \sum_{p \in \mathcal{P}} x_p^h = d & \quad (b) \\ \sum_{p \in \mathcal{P}} x_p^h \leq 1 & \forall h = 1 \dots H \quad (c) \\ x_p^h \geq 0 & \forall p \in \mathcal{P} \quad \forall h = 1 \dots H \quad (d) \end{array} \right. \quad (16)$$

Algorithm 2 optimal path computation for (SP^h)

```
let  $A' = A$ 
let  $p^* = \emptyset, u^* = 0, w^* = \infty$ 
repeat
  compute a shortest path  $p$  from  $s$  to  $t$  on  $A'$  with  $c_a^h$  as arc lengths
  if  $p = \emptyset$  then
    break
  else
    let  $u$  be its capacity,  $w$  its cost
    if  $w < w^*$  then
       $p^* \leftarrow p$ 
       $u^* \leftarrow u$ 
       $w^* \leftarrow w$ 
    end if
     $A' \leftarrow A' \setminus \{a \in A' | u_a \leq u\}$ 
  end if
until  $A' = \emptyset$ 
return  $(p^*, u^*, w^*)$ 
```

For each path p , the capacity u_p and the cost c_p^h for path number h are given by

$$u_p = \min_{a \in p} u_a \quad (17)$$

$$c_p^h = \sum_{a \in p} \frac{\partial \phi}{\partial x_a^h}(x^i, \mu) \quad (18)$$

Constraints 13(d) give the capacity constraints 16(a). Constraints 13(a) imply $\sum_{p \in \mathcal{P}} x_p^h = d_h$ for all $h = 1 \dots H$. Then the constraint 13(e) lead to the demand constraint 16(b).

The constraints 16(c) correspond to the choice of at most one path p for each path index h (constraints 13(f)) as justified below.

Property 1. Variables y_a^h and constraints 13(b,c) can be dropped from (DFS_2^i) .

Proof. Let x be an feasible solution of (DFS_2^i) . Let $\bar{y}$ be defined by

$$\bar{y}_a^h = \frac{\sum_{p \in \mathcal{P}} \delta_a^p \frac{x_p^h}{u_p}}{\sum_{p \in \mathcal{P}} \frac{x_p^h}{u_p}} \quad \forall a \in A \quad \forall h = 1 \dots H \quad (19)$$

The definition of $\bar{y}$ ensures the respect of constraints 13(b). In the same way, constraints 16(c) and the definition of the capacity u_p imply (constraints 13(c))

$$u_a \bar{y}_a^h \geq \sum_{p \in \mathcal{P}} \delta_a^p \frac{u_a}{u_p} x_p^h \geq \sum_{p \in \mathcal{P}} \delta_a^p x_p^h = x_a^h \quad \forall a \in A \quad \forall h = 1 \dots H$$

The definition of the set $\mathcal{P}$ implies that paths p which cross a same node v are distinct and constraints 13(f) are satisfied:

$$\sum_{a \in \delta^-(v)} \bar{y}_a^h = \frac{\sum_{a \in \delta^-(v)} \sum_{p \in \mathcal{P}} \delta_a^p \frac{x_p^h}{u_p}}{\sum_{p \in \mathcal{P}} \delta_a^p \frac{x_p^h}{u_p}} \leq 1 \quad \forall v \in V \quad \forall h = 1 \dots H$$

So variables y_a^h and constraints 13(b,c) do not restrict the flow variables x_p^h anymore. ■

Then the column generation subproblem becomes an optimal path problem, i.e. minimization of the reduced cost $\bar{c}_p^h$

$$\bar{c}_p^h = c_p^h - \sum_{a \in p} \pi_a - \frac{\lambda_h}{u_p} - v = \sum_{a \in p} \left(\frac{\partial \phi}{\partial x_a^h}(x^i, \mu) - \pi_a \right) - \frac{\lambda_h}{u_p} - v \quad (20)$$

where $\pi_a \leq 0$ (resp. $\lambda_h \leq 0$ and $v \in \mathbb{R}$) are the dual variables corresponding to constraint 20(a) (resp. 20(c) and 20(b)). The computation of the minimum cost elementary path does not reduce to a simple shortest path problem, because the reduced cost involves a combination of two dual variables, π and λ (the cost associated to the demand constraints does not depend on the arcs of the path). Let $c_a^h = \partial \phi / \partial x_a^h(x^i, \mu) - \pi_a$ be the arc-component of the reduced cost. Given a path position h , the subproblem (SP^h) is as follows:

$$(SP^h): \min w = \sum_{p \in \mathcal{P}} \delta_a^p c_a^h - \lambda^h / u_p - v \quad (21)$$

Then the optimal path p^* is a combination between the shortest path and the highest capacity path (with c_a^h as arc lengths). Its computation can be done in the following way:

Property 2. Algorithm 2 computes an optimal path for the subproblem (SP^h) .

Proof. Let A' be a subset of A . Let p be the shortest path from s to t on A' with c_a^h as arc lengths. From the expression of the cost w , we obtain that, if p is not optimal on A' , the optimal path on A' has a larger capacity than the capacity of p . ■

Property 3. Algorithm 2 runs in polynomial time

Proof. There is no negative cost arc since $\pi_a \leq 0, \forall a \in A$. Thus, no negative cost cycle exists and any polynomial-time shortest path algorithm may be used. Next, at each iteration of the algorithm, at least one arc is removed from the set A' . Thus, there is at most m iterations, that is m computations of a shortest path. ■

Because the columns might be used anywhere in the branching tree, they are inserted into a global pool. The column generation will stop as soon as $w^* \geq 0$. Then, an optimal solution of (DFS^i) is easily computed from an optimal solution of (DFS_2^i) by Equation 15.

3.4. Recovering Integral Solutions

Algorithm 1 finds a lower bound for the continuous relaxation of $(k\text{-DCRP})$. To compute integer solutions, one can embed it into a branch and bound scheme.

FIG. 1. Divergence on aggregated flow x^h on node d .

Barnhart et al. [3] proposed an efficient branching for the routing problems. It is based on the concept of node of divergence over the aggregated flow $x_a^h = \sum_p \delta_a^p x_p^h$, $\forall a \in A$. A node of divergence is a node $d \in N$ such that the aggregated flow is coming from a single arc and going out on several arcs (see Fig. 1). Given a position $h \leq H$, a divergence occurs when the flow decision variables y_a^h are fractional. On Figure 1, those fractions are respectively $7/10$, $1/10$, and $2/10$. Thus, a fractional part of each path p is used.

Let δ_1 and δ_2 be a partition of $\delta^+(d)$ such that each set contains at least one arc carrying a positive amount of flow. Let $P_1 \subset \mathcal{P}$ and $P_2 \subset \mathcal{P}$ be the set of paths going through node d and using one arc of respectively δ_1 and δ_2 . Because the flow on x^h has to be integer (that is, has to be unsplittable), either it uses one arc in the set δ_1 or in the set δ_2 . Then the following branching is valid:

$$\left(\sum_{p \in P_1} y_p^h = 0 \right) \quad \text{vs.} \quad \left(\sum_{p \in P_2} y_p^h = 0 \right)$$

For a better efficiency, the branching should be applied on the first node of divergence and the sets P_1 and P_2 should be built such that each sum is the most fractional.

If such a node d does not exist, the aggregated flow x_a^h is carried by a single path and no branching is necessary.

3.5. Post-Optimization

The solution x_{CR} obtained in Section 3.4 is optimal with respect to the continuous objective function $\theta(x)$ defined in (CR). But we showed in Section 3.1 that the value of $\theta(x)$ for this solution is a lower bound for the optimal value θ^* of the k -DCRP model because of the relaxation of integrity constraints and the approximation of the decision variable y_a^h by the ratio x_a^h/u_a in constraint 2(g). However as the solution x_{CR} is a flow carried by H paths, it can be evaluated within the k -DCRP model. Let θ' be its corresponding value.

$$\theta(x_{\text{CR}}) \leq \theta^* \leq \theta' \quad (22)$$

Given the H specific routes computed in x_{CR} , it may be still possible to improve the value of θ by performing a post-optimization step. We have thus at most H paths p_h and we

compute the distribution d_h for all h which minimizes $\tilde{\theta}(d)$ defined by

$$\tilde{\theta}^h(d) = \sum_{a \in A} \frac{\lambda \delta_a^h}{u_a - x_a} = \sum_{a \in p_h} \frac{\lambda}{u_a - x_a} \quad (23)$$

$$\tilde{\theta}(d) = \max_{h \in \{1, \dots, H\}} \tilde{\theta}^h(d) \quad (24)$$

where $\delta_a^h = 1$ if $a \in p_h$ and 0 otherwise, and $x_a = \sum_{h=1}^H \delta_a^h d_h$.

The solution proposed by the branch and bound method is a feasible flow. The $\tilde{\theta}$ function ensures to always satisfy the capacity constraints. It only remains the demand constraints to be checked. The gradient of function $\tilde{\phi}$ is projected to find a feasible descent direction.

$$\tilde{\phi}(d) = \mu \ln \sum_{h=1}^H e^{\tilde{\theta}^h(d)/\mu} \quad (25)$$

The gradient of $\tilde{\phi}$ needs the definition of the partial derivatives of $\tilde{\theta}^h$ and of the expression of $\tilde{v}^h$

$$\frac{\partial \tilde{\theta}^h}{\partial d_i}(d) = \sum_{a \in p_h \cap p_i} \frac{\lambda}{(u_a - x_a)^2} \quad (26)$$

$$\tilde{v}^h(d) = \frac{e^{\tilde{\theta}^h(d)/\mu}}{\sum_{i=1}^H e^{\tilde{\theta}^i(d)/\mu}} \quad (27)$$

Then

$$\frac{\partial \tilde{\phi}}{\partial d_i}(d) = \sum_{h=1}^H \tilde{v}^h(d) \sum_{a \in p_h \cap p_i} \frac{\lambda}{(u_a - x_a)^2} \quad (28)$$

The descent direction $\Delta(d)$ is given by

$$\Delta_h(d) = \frac{1}{H} \sum_{i=1}^H \frac{\partial \tilde{\phi}}{\partial d_i}(d) - \frac{\partial \tilde{\phi}}{\partial d_h}(d) \quad (29)$$

In this direction, $\tilde{\theta}$ is minimized using its directional derivative

$$\tilde{\theta}'(d, \Delta) = \max_{h \in I(d)} [\nabla \tilde{\theta}^h(d)]^T \Delta \quad (30)$$

where $I(d) = \{h | \tilde{\theta}^h(d) = \tilde{\theta}(d)\}$.

3.6. Initial Integral Solution

To reduce the size of the branching tree in the branch-and-price algorithm, a primal solution is first computed at the root node. We can note that any k -splittable solution is feasible with respect to (k -DCRP). Thus, it is sufficient to use the solution of the k -splittable minimum cost flow problem. However the capacities are reduced to $4/5$ of the initial capacities to prevent this solution from being too close to saturation and to prevent θ from being too high. The cost of each arc is set to

FIG. 2. Transit grid with 12 nodes and 38 arcs.

λ/u_a^2 which corresponds to the gradient of $\lambda/(u_a - x_a)$ (see (1)) at the origin.

4. NUMERICAL RESULTS

Two sets of instances have been used to measure the efficiency of our approach. The first one is a set of optical telecommunication networks ranging from 10 nodes to 14 nodes and with low connectivity.

A second set of instances has been generated using the Transit Grid generator developed by G. Waissi². The topology of those instances (see Fig. 2) has been designed to look close to the transportation networks. In our experiments, grids with 10 nodes have been used. The capacities have been randomly chosen in the interval $[1; 1000]$. The demand is computed the following way: let H be the upper bound of the number of paths. A k -splittable maximum flow is computed [14]. Let F^* be the value of this optimal solution. Then the demand is set to $3F^*/4$ and thus depends on H .

The experiments were done on a Pentium 4 computer with 4 Gb memory running linux 2.6.6. The time limit has been set to 3 h for each run. The program has been coded in C and compiled with gcc 3.3.2 with the -O2 flag activated.

(k -MADP) is solved by a similar approach. We can show that the property 1 can be applied to the (k -MADP). Then the continuous relaxation looks like (CR2) with a different objective (ψ). To compare a solution of (k -MADP) with a solution of (k -DCRP), we will provide the corresponding value of the highest end-to-end delay. Obviously the flow distribution on the paths of the optimal solution is optimal for function ϕ . To have a fair comparison of the performance of each model, the post-optimization step described in Section 3.4 is applied to this solution. Then we will provide the new value of the highest end-to-end delay.

Table 1 displays the results for the telecommunication networks whereas Table 2 keeps track of the 10 nodes transit grids. For each instance, experiments are run for increasing

values of H (and therefore a varying demand as previously explained). The results are reported for both (k -DCRP) (columns “End-To-End Delay”) and (k -MADP) (columns “Minimum Average Delay”). For each method, the optimal value (column “ z^* ”) and the CPU time in seconds (column “CPU”) are reported. For the (k -MADP), the column “delay” corresponds to the evaluation of the optimal solution with respect to the objective function of (k -DCRP) (maximal delay along an active path). The column “PO” shows the same value after a post-optimization step.

From Table 1, it can be seen for (k -DCRP) that the CPU time grows quickly as H increases. For small values of H , (k -MADP) produces good approximations of the optimal solution of (k -DCRP), once the post-optimization procedure (PO) is applied. However, as H increases the gap between (k -MADP) and (k -DCRP) increases. Another good feature from (k -MADP) is the low CPU time requirement when comparing with (k -DCRP). On the “abilene” instance it is reduced by two orders of magnitude. On the “zang” instance an approximation for the higher H value ($H = 5$) can be computed.

Table 2 display the results on the transit grid instances. Those instances are harder to solve for several reasons: first, their topology is denser than telecommunication networks. Second, the capacities are highly variable. On the instances, it can be seen that (k -MADP) provides a good approximation within a reasonable amount of time. (k -DCRP) is able to solve instances with H up to 3. As the CPU times grow quickly with H , $H = 4$ is currently the limit for (k -DCRP).

5. CONCLUSION

We have proposed an arc-node flow model for the nonlinear delay constraint k -splittable flow problem (k -DCRP) where each potential path is modelled by a separate flow. This allowed us to use a branch-and-price strategy after performing a Dantzig–Wolfe reformulation to allow column generation scheme. The flow deviation method has been applied to solve the relaxed subproblem which based on potential function. Finally, a post-optimization step is done on each integer solution to get the optimal solution.

The choice of the exponential potential function ϕ leads to rather good numerical performances as long as H remains low (namely $H \leq 3$) and the approach works only for a small number of active paths. The approximation ψ , based on the Kleinrock’s average delay function, has been proposed in the (k -MADP) model. It yields nice solutions within a reasonable amount of time, confirming former theoretical studies by Bienstock [7].

This new approach looks promising for hard routing problems like (k -DCRP) combining both nonlinearity and decision variables. We believe finally that the use of (k -MADP) as warm start for (k -DCRP) could increase the performance of our approach.

²<http://www.informatik.uni-trier.de/~naeher/Professur/research/generators/maxflow/tg/index.html>

APPENDIX A. THE FLOWCHART OF THE ALGORITHM

A branch-and-price approach for the k -DCRP.

APPENDIX B. CONVEXITY OF THE FUNCTION θ_A^H

This function θ_a^h depends only on variables associated with the arc a . Then the partial derivatives with respect to the variables associated with other arcs are equal to zero. Thus, we study only the partial derivatives with respect to variables x_a^i for $i = 1, \dots, H$. For convenience, let $\bar{x}_a^h$ be the complement of x_a^h in x_a

$$\bar{x}_a^h = \sum_{i \neq h} x_a^i = x_a - x_a^h \quad (31)$$

The partial derivatives calculation gives

$$\frac{\partial \theta_a^h}{\partial x_a^i}(x) = \begin{cases} \frac{\lambda(u_a - \bar{x}_a^h)}{u_a(u_a - x_a)^2} & \text{if } i = h. \\ \frac{\lambda x_a^h}{u_a(u_a - x_a)^2} & \text{otherwise} \end{cases} \quad (32)$$

Then the second-order partial derivatives are

$$\frac{\partial^2 \theta_a^h}{\partial x_a^i \partial x_a^j}(x) = \begin{cases} \frac{2\lambda x_a^h}{u_a(u_a - x_a)^3} & \text{if } i \neq h \text{ and } j \neq h, \\ \frac{2\lambda(u_a - \bar{x}_a^h)}{u_a(u_a - x_a)^3} & \text{if } i = h \text{ and } j = h, \\ \frac{\lambda(u_a - \bar{x}_a^h + x_a^h)}{u_a(u_a - x_a)^3} & \text{otherwise.} \end{cases} \quad (33)$$

TABLE 1. Telecommunication networks with low connectivity.

Instance		Minimum average delay				End-to-end delay	
Graph	H	z^*	CPU (s)	Delay	PO	z^*	CPU (s)
Abilene 11–14	1	15.00	0.04	2.00	2.00	2.00	0.06
	2	32.91	0.05	2.30	2.20	2.20	0.17
	3	16.44	0.34	1.43	1.41	1.38	2.57
	4	16.44	1.50	1.43	1.41	1.38	10.22
	5	16.44	1.96	1.43	1.41	1.38	31.26
	6	16.44	2.84	1.43	1.41	1.38	249.09
Cost 11–25	1	5.10	0.02	0.11	0.11	0.11	0.01
	2	9.02	0.39	0.22	0.17	0.17	5.19
	3	12.28	4.73	0.30	0.21	0.21	42.53
	4	16.14	18.77	0.41	0.27	0.27	290.12
	5	7.97	474.28	0.26	0.21	0.16	4926.12
Nsf 14–21	1	5.10	0.01	0.17	0.17	0.17	0.15
	2	7.56	0.33	0.17	0.17	0.17	1.88
	3	12.59	0.80	0.28	0.23	0.23	8.98
	4	15.25	9.05	0.30	0.28	0.26	66.81
	5	8.76	32.79	0.22	0.19	0.18	229.90
	6	8.74	134.20	0.22	0.18	0.18	895.62
Zang 10–15	1	7.63	0.04	0.35	0.35	0.35	0.16
	2	11.76	0.68	0.33	0.32	0.32	6.84
	3	21.66	5.51	0.46	0.43	0.43	44.81
	4	10.49	49.99	0.30	0.28	0.27	1060.92
	5	10.46	538.17	0.30	0.28	≤ 0.27	—
	6	≤ 10.46	—	0.30	0.28	≤ 0.29	—

Let $H_a^h(x)$ be the Hessian matrix of the function θ_a^h . Then,
 $\forall x \in S, \forall a \in A, \forall h = 1, \dots, H$

$$\begin{aligned}
x^T H_a^h(x) x &= \sum_{i=1}^H \sum_{j=1}^H x_a^i \frac{\partial^2 \theta_a^h}{\partial x_a^i \partial x_a^j}(x) x_a^j \\
&= \frac{\lambda}{u_a(u_a - x_a)^3} \left(2 \sum_{i \neq h} \sum_{j \neq h} x_a^h x_a^i x_a^j \right. \\
&\quad \left. + 2 \sum_{i \neq h} (u_a - \bar{x}_a^h + x_a^h) x_a^h x_a^i + 2(u_a - \bar{x}_a^h) (x_a^h)^2 \right) \\
&= \frac{2\lambda x_a^h}{u_a(u_a - x_a)^3} ((\bar{x}_a^h)^2 \\
&\quad + (u_a - \bar{x}_a^h + x_a^h) \bar{x}_a^h + (u_a - \bar{x}_a^h) x_a^h) \\
&= \frac{2\lambda x_a^h}{u_a(u_a - x_a)^3} (u_a (\bar{x}_a^h + x_a^h)) \\
&= \frac{2\lambda x_a^h x_a}{(u_a - x_a)^3} \geq 0
\end{aligned} \tag{34}$$

Thus $x^T H_a^h x \geq 0, \forall x \in S, \forall a \in A, \forall h = 1 \dots H$

APPENDIX C. CONVEXITY OF THE FUNCTION ϕ

The proof is adapted from [15]. The calculation of partial derivatives and second-order partial derivatives gives

$$\nabla_x \phi(x, \mu) = \sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \tag{35}$$

$$\begin{aligned}
\nabla_x^2 \phi(x, \mu) &= \sum_{h=1}^H \left(v_h(x, \mu) \nabla^2 \theta^h(x) + \frac{1}{\mu} v_h(x, \mu) \nabla \theta^h(x) \nabla \theta^h(x)^T \right) \\
&\quad - \frac{1}{\mu} \left(\sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \right) \left(\sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \right)^T
\end{aligned} \tag{36}$$

where

$$v_h(x, \mu) = \frac{\exp(\theta^h(x)/\mu)}{\sum_{h=1}^H \exp(\theta^h(x)/\mu)} \in (0, 1), \sum_{h=1}^H v_h(x, \mu) = 1. \tag{37}$$

Let us note

$$A = [\nabla \theta^1(x), \dots, \nabla \theta^H(x)],$$

$$\Lambda = \text{diag}(v_h(x, \mu)),$$

$$\Upsilon = \begin{bmatrix} v_1(x, \mu) \\ \vdots \\ v_P(x, \mu) \end{bmatrix} [v_1(x, \mu), \dots, v_P(x, \mu)],$$

$$D = \Lambda - \Upsilon$$

TABLE 2. 10 nodes transit grids.

Instance		Minimum average delay				End-to-end delay	
Graph	H	z^*	CPU (s)	Delay	PO	z^*	CPU (s)
tg10-3	1	2.92	0.06	2.06	2.06	2.06	0.40
12-40	2	5.84	4.42	2.23	2.22	2.22	24.84
	3	8.09	76.88	2.45	2.33	2.33	1173.87
	4	10.26	1871.23	2.57	2.47	≤ 2.46	—
	5	≤ 11.15	—	2.72	2.59	≤ 2.58	—
tg10-4	1	9.14	0.18	2.99	2.99	2.99	3.35
12-40	2	14.82	9.81	2.85	2.77	2.77	47.22
	3	18.72	220.06	3.71	3.10	3.06	2195.04
	4	≤ 19.94	—	4.06	3.44	≤ 3.22	—
tg10-5	1	12.74	0.05	2.18	2.18	2.18	0.59
12-38	2	12.42	9.71	1.89	1.81	1.81	85.07
	3	15.94	667.70	3.35	2.31	2.21	9592.66
	4	≤ 17.55	—	3.20	3.06	≤ 2.59	—
tg10-6	1	7.14	0.04	1.54	1.54	1.54	0.72
12-40	2	14.16	2.20	2.22	1.93	1.93	15.03
	3	17.94	72.28	2.36	2.31	2.10	526.02
	4	21.86	3953.02	3.22	2.41	≤ 2.16	—
	5	≤ 12.19	—	2.24	1.87	≤ 1.64	—
tg10-7	1	6.79	0.03	1.86	1.86	1.86	1.36
12-40	2	13.49	8.50	3.05	2.56	2.55	58.27
	3	18.56	419.56	3.31	2.87	2.86	6233.33
	4	≤ 20.05	—	3.61	3.32	≤ 3.08	—
tg10-10	1	10.68	0.06	2.87	2.87	2.87	1.54
12-40	2	18.38	4.74	4.60	3.71	3.70	47.60
	3	19.78	403.24	4.08	3.81	3.50	4592.37
	4	≤ 22.65	—	5.63	3.86	≤ 4.12	—

Then we have

$$\begin{aligned}
& \frac{1}{\mu} \left(\sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \nabla \theta^h(x)^T \right. \\
& \quad \left. - \left(\sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \right) \left(\sum_{h=1}^H v_h(x, \mu) \nabla \theta^h(x) \right)^T \right) \\
&= \frac{1}{\mu} (A \Lambda A^T - A \Upsilon A^T) = \frac{1}{\mu} A D A^T \quad (38)
\end{aligned}$$

By Gershgorin theorem, the eigenvalues of D are located in the union U_H of H circles.

$$U_H = \bigcup_{h=1}^H \{z \in \mathbb{C} : |z - d_{hh}| \leq R_h(D)\}$$

where

$$\begin{aligned}
d_{hh} &= v_h(x, \mu) - v_h^2(x, \mu) \quad \forall h = 1, \dots, H \\
R_h(D) &= \sum_{i \neq h} v_h(x, \mu) v_i(x, \mu) \quad \forall h = 1, \dots, H
\end{aligned}$$

Then

$$\begin{aligned}
R_h(D) &= \sum_{i=1}^H v_h(x, \mu) v_i(x, \mu) - v_h^2(x, \mu) \\
&= v_h(x, \mu) \sum_{i=1}^H v_i(x, \mu) - v_h^2(x, \mu) \\
&= v_h(x, \mu) - v_h^2(x, \mu) \\
&= d_{hh}
\end{aligned}$$

Thus, each eigenvalue of $\Lambda - \Upsilon$ is non-negative and the matrix $\frac{1}{\mu} A D A^T$ is positive semidefinite.

On the other hand, we have

$$\begin{aligned}
& x^T \left(\sum_{h=1}^H v_h(x, \mu) \nabla^2 \theta^h(x) \right) x \\
&= \sum_{h=1}^H v_h(x, \mu) \sum_{a \in A} x^T \nabla^2 \theta_a^h(x) x \\
&= \sum_{h=1}^H v_h(x, \mu) \sum_{a \in A} \frac{2\lambda x_a^h x_a}{(u_a - x_a)^3}
\end{aligned}$$

$$= 2\lambda \sum_{a \in A} \left(\frac{x_a}{(u_a - x_a)^3} \sum_{h=1}^H v_h(x, \mu) x_a^h \right)$$

Let $x \in S$ be chosen such that

$$x^T \left(\sum_{h=1}^H v_h(x, \mu) \nabla^2 \theta^h(x) \right) x = 0$$

Then

$$\frac{x_a}{(u_a - x_a)^3} \sum_{h=1}^H v_h(x, \mu) x_a^h = 0 \quad \forall a \in A \quad \forall x \in S$$

However as $v_h(x, \mu) > 0$ for all h , we deduce that $x = 0$.

Thus $\sum_{h=1}^H v_h(x, \mu) \nabla^2 \theta^h(x)$ is positive definite and so is $\nabla_x^2 \phi(x, \mu)$.

APPENDIX D. PROOF OF THEOREM 1

The proof is adapted from [8].

Consider an input vector x^i . For convenience let

$$g(\sigma) = \phi((1 - \sigma)x^i + \sigma \bar{x}^i, \mu) = \phi(x^i + \sigma(\bar{x}^i - x^i), \mu) \quad (39)$$

we obtain

$$g(\sigma) - g(0) \leq (\phi^*(\mu) - \phi(x^i, \mu))\sigma + \frac{1}{2}g''(\alpha)\sigma^2 \quad \text{where } 0 < \alpha < \sigma \quad (40)$$

To bound the quadratic term in (40), let us use the simplification $x_\alpha = (1 - \alpha)x^i + \alpha \bar{x}^i$ and $d = \bar{x}^i - x^i$. As $v_h(x_\alpha, \mu) < 1 \quad \forall h \in \{1, \dots, H\}$, we obtain from (36) and (38)

$$g''(\alpha) \leq \sum_{h=1}^H d^T \nabla^2 \theta^h(x_\alpha) d + \frac{1}{\mu} d^T (ADA^T) d \quad (41)$$

The definition of the second-order derivatives in (33) implies

$$\begin{aligned} & \sum_{h=1}^H d^T \nabla^2 \theta^h(x_\alpha) d \\ &= \sum_{a \in A} \frac{\lambda}{u_a(u_a - x_{\alpha,a})^3} \sum_{h=1}^H \left(2 \sum_{i \neq h} \sum_{j \neq h} d_a^i x_{\alpha,a}^h d_a^j \right. \\ & \quad \left. + 2 \sum_{i \neq h} (u_a - \bar{x}_{\alpha,a}^h + x_{\alpha,a}^h) d_a^i d_a^h + 2(u_a - \bar{x}_{\alpha,a}^h) (d_a^h)^2 \right) \\ &= \sum_{a \in A} \frac{2\lambda}{u_a(u_a - x_{\alpha,a})^3} \sum_{h=1}^H \left((\bar{d}_a^h)^2 x_{\alpha,a}^h \right. \\ & \quad \left. + (u_a - \bar{x}_{\alpha,a}^h + x_{\alpha,a}^h) \bar{d}_a^h d_a^h + (u_a - \bar{x}_{\alpha,a}^h) (d_a^h)^2 \right) \end{aligned}$$

$$\begin{aligned} &= \sum_{a \in A} \frac{2\lambda}{u_a(u_a - x_{\alpha,a})^3} \sum_{h=1}^H ((d_a^h)^2 x_{\alpha,a}^h + (u_a - x_{\alpha,a}) d_a^h d_a^h) \\ &= \sum_{a \in A} \frac{2\lambda d_a^2}{(u_a - x_{\alpha,a})^3} \end{aligned} \quad (42)$$

Observe that

$$|d_a| = |\bar{x}_a^i - x_a^i| \leq u_a \leq U \quad \forall a \in A \quad (43)$$

$$\frac{1}{(1-x)^3} \leq 8 \left(\frac{x}{1-x} \right)^3 + 8 \quad \forall 0 \leq x \leq 1 \quad (44)$$

Without loss of generality, we suppose that $u_a \geq 1$, for all $a \in A$. Then

$$\begin{aligned} \sum_{h=1}^H d^T \nabla^2 \theta^h(x_\alpha) d &\leq 16\lambda \left(U^2 \sum_{a \in A} \left(\frac{\sum_{h=1}^H x_{\alpha,a}^h}{u_a(u_a - x_{\alpha,a})} \right)^3 + m \right) \\ &\leq 16\lambda \left(U^2 \left(\sum_{h=1}^H \sum_{a \in A} \frac{x_{\alpha,a}^h}{u_a(u_a - x_{\alpha,a})} \right)^3 + m \right) \\ &\leq 16\lambda \left(U^2 \left(\sum_{h=1}^H \frac{\theta^h(x_\alpha)}{\lambda} \right)^3 + m \right) \\ &\leq 16\lambda \left(U^2 \left(\frac{H\theta(x_\alpha)}{\lambda} \right)^3 + m \right) \\ &\leq 16\lambda \left(\frac{U^2 H^3}{\lambda^3} \phi^3(x_\alpha, \mu) + m \right) \end{aligned} \quad (45)$$

On the other hand, the previous spectral analysis of the symmetric positive semidefinite matrix D proves that its largest eigenvalue γ is bounded above

$$\gamma = \max_{h \in \{1, \dots, H\}} 2v_h(x, \mu)(1 - v_h(x, \mu)) \leq \frac{1}{2} \quad (46)$$

Thus

$$\begin{aligned} \frac{1}{\mu} d^T ADA^T d &\leq \frac{1}{2\mu} \|A^T d\|^2 \\ &\leq \frac{1}{2\mu} \sum_{h=1}^H (\nabla \theta^h(x_\alpha)^T d)^2 \\ &\leq \frac{1}{2\mu} \left(\sum_{h=1}^H \sum_{a \in A} \frac{\lambda}{u_a(u_a - x_{\alpha,a})^2} \right. \\ & \quad \left. \times \left(x_{\alpha,a}^h \sum_{i \neq h} d_a^i + (u_a - \bar{x}_{\alpha,a}^h) d_a^h \right) \right)^2 \\ &\leq \frac{1}{2\mu} \left(\sum_{a \in A} \sum_{h=1}^H \frac{\lambda}{u_a(u_a - x_{\alpha,a})^2} (x_{\alpha,a}^h d_a + (u_a - x_{\alpha,a}) d_a^h) \right)^2 \\ &\leq \frac{1}{2\mu} \left(\sum_{a \in A} \frac{\lambda d_a}{(u_a - x_{\alpha,a})^2} \right)^2 \end{aligned} \quad (47)$$

Observe that

$$\frac{1}{(1-x)^2} \leq 2 \left(\frac{x}{1-x} \right)^2 + 2 \quad \forall 0 \leq x \leq 1 \quad (48)$$

Then

$$\begin{aligned} \frac{1}{\mu} d^T ADA^T d &\leq \frac{2\lambda^2}{\mu} \left(U \sum_{a \in A} \left(\frac{x_{\alpha,a}}{(u_a - x_{\alpha,a})} \right)^2 + m \right)^2 \\ &\leq \frac{2\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x_\alpha, \mu) + m \right)^2 \end{aligned} \quad (49)$$

If σ is chosen such that

$$\phi(x^i + \sigma(\bar{x}^i - x^i), \mu) \leq \phi(x^i, \mu) \quad (50)$$

then (45), (49) and the convexity of ϕ implies

$$\begin{aligned} g''(\alpha) &\leq 16\lambda \left(\frac{U^2 H^3}{\lambda^3} \phi^3(x^i, \mu) + m \right) \\ &\quad + \frac{2\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x^i, \mu) + m \right)^2 \end{aligned} \quad (51)$$

Substituting in (40), we obtain

$$\begin{aligned} \phi(x^i + \sigma(\bar{x}^i - x^i), \mu) - \phi(x^i, \mu) &\leq (\phi^*(\mu) - \phi(x^i, \mu))\sigma \\ &\quad + \frac{1}{2} \left(16\lambda \left(\frac{U^2 H^3}{\lambda^3} \phi^3(x^i, \mu) + m \right) \right. \\ &\quad \left. + \frac{2\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x^i, \mu) + m \right)^2 \right) \sigma^2 \end{aligned} \quad (52)$$

Like explained in [8], after a calculation to minimize the quadratic right-hand side, we get

$$\begin{aligned} \phi(x^{i+1}, \mu) - \phi(x^i, \mu) \\ \leq - \frac{(\phi^*(\mu) - \phi(x^i, \mu))^2}{32\lambda \left(\frac{U^2 H^3}{\lambda^3} \phi^3(x^i, \mu) + m \right) + \frac{4\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x^i, \mu) + m \right)^2} \end{aligned} \quad (53)$$

If $\phi^*(\mu) < (1 - \epsilon)\phi(x^i, \mu)$, we therefore obtain

$$\begin{aligned} \phi(x^{i+1}, \mu) - \phi(x^i, \mu) \\ \leq - \frac{\epsilon^2 \phi^2(x^i, \mu)}{32\lambda \left(\frac{U^2 H^3}{\lambda^3} \phi^3(x^i, \mu) + m \right) + \frac{4\lambda^2}{\mu} \left(\frac{UH^2}{\lambda^2} \phi^2(x^i, \mu) + m \right)^2} \end{aligned} \quad (54)$$

and the approximation is proved.

REFERENCES

- [1] A. Atamtürk and D. Rajan, On splittable and unsplittable capacitated network design arc-set polyhedra, *Math Program* 92 (2002), 315–333.
- [2] G. Baier, E. Köhler, and M. Skutella, On the k-splittable flow problem, *Proceedings of the 10th Annual European Symposium on Algorithms*, Springer-Verlag, 2002, pp. 101–103.
- [3] C. Barnhart, C.A. Hane, and P.H. Vance, Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems, *Oper Res* 48 (2000), 318–326.
- [4] S. Beker, D. Kofman, and N. Puech, Off-Line MPLS layout design and reconfiguration: Reducing complexity under dynamic traffic conditions, *INOC Conference*, Evry (France), 2003, pp. 61–66.
- [5] W. Ben-Ameur and A. Ouorou, Mathematical models of the delay constrained routing problem, *Algorithmic Oper Res* 1 (2006), 94–103.
- [6] D.P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, Boston, 1999.
- [7] D. Bienstock, *Potential function methods for approximately solving linear programming problems, theory and practice*, Kluwer Academic Publishers, Boston, 2002.
- [8] D. Bienstock and O. Raskina, Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem, *Math Program* 91 (2002), 479–492.
- [9] C. Duhamel and P. Mahey, Multicommodity flow problems with a bounded number of paths: A flow deviation approach, *Networks* 49 (2007), 80–89.
- [10] L. Fratta, M. Gerla, and L. Kleinrock, The flow deviation method: An approach to store-and-forward communication network design, *Networks* 3 (1973), 97–133.
- [11] J.M. Kleinberg, *Approximation algorithms for disjoint paths problems*, PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1996.
- [12] M. Martens and M. Skutella, Flows on few paths: Algorithms and lower bounds, *Networks* 48 (2006), 68–76.
- [13] F. Shahrokhi and D.W. Matula, The maximum concurrent flow problem, *J ACM* 37 (1990), 318–334.
- [14] J. Truffot and C. Duhamel, A branch and price algorithm for the k-splittable maximum flow problem, *LIMOS*, Aubière, France, 2006.
- [15] S. Xu, Smoothing method for minimax problems, *Comput Optim Appl* 20 (2001), 267–279.