QoS-aware Multicommodity Flows and Transportation Planning ⋆

George Tsaggouris[1,2], Christos Zaroliagis[1,2]

[1] Computer Technology Institute, N. Kazantzaki Str,
Patras University Campus, 26500 Patras, Greece
[2] Department of Computer Engineering and Informatics,
University of Patras, 26500 Patras, Greece
{tsaggour,zaro}@ceid.upatras.gr

Abstract. We consider the *QoS-aware Multicommodity Flow* problem, a natural generalization of the weighted multicommodity flow problem where the demands and commodity values are elastic to the Quality-of-Service characteristics of the underlying network. The problem is fundamental in transportation planning and also has important applications beyond the transportation domain. We provide a FPTAS for the QoS-aware Multicommodity Flow problem by building upon a Lagrangian relaxation method and a recent FPTAS for the non-additive shortest path problem.

1 Introduction

Consider a capacitated directed network $G = (V, E)$ in which we wish to route k commodities to meet certain initial demands. Each commodity i is associated with a specific origin-destination pair (s_i, t_i), a demand d_i, and a value v_i representing the *profit* of routing one unit of flow from that commodity. Also, for each commodity i, a weight $wt_i : E \rightarrow I\!R_0^+$ is defined that quantifies the provided *quality of service (QoS)* when this commodity is routed along an edge e or a path p, where $wt_i(p) = \sum_{e \in p} wt_i(e)$. Smaller weight means better QoS. When a commodity is not routed along its shortest w.r.t. wt_i (optimal w.r.t. the QoS) path due to capacity restrictions, then (i) a portion of its demand d_i drops (the worse the QoS of the path, the larger the portion of d_i that is lost), and (ii) its value v_i is reduced (the worse the QoS, the larger the reduction). In other words, demands and values are *elastic* to the provided QoS. The objective is to compute the maximum weighted multicommodity flow (sum over all commodities and over all paths of the flow routed from every commodity on each path multiplied by the commodity's value) subject to the QoS-elastic demands and values. We call the above the *QoS-aware Multicommodity Flow* (MCF) problem.

The QoS-aware MCF problem is a natural generalization of the weighted MCF problem that (is motivated by and) plays a key role in transportation

⋆ This work was partially supported by the FET Unit of EC (IST priority – 6th FP), under contracts no. FP6-021235-2 (ARRIVAL) and no. IST-2002-001907 (DELIS).

planning: one of the prime issues that planners of transport operators in public transportation networks have to deal with concerns the routing of various commodities (customers with common origin-destination pairs) to meet certain demands [9,13,14]. A customer, when provided with a non-optimal path (route) due to unavailable capacity, s/he will most likely switch to another operator or even other means of transport and the probability in doing so increases as the QoS drops (actually, as a result of statistical measurements over several years, major European railway companies know quite accurately the percentage of customers they lose in such cases as a function of the path's QoS [9,14]). To minimize the loss of customers, the value charged for the requested service is usually reduced to make the alternative (worse in QoS) path, offered for that service, attractive.

Consequently, transportation planners are confronted with the following network and line planning issues:

– Which is the maximum profit obtained with the current capacity policy that incurs certain QoS-elastic demands and values?
– How much will this profit improve if the capacity is increased?
– Which is the necessary capacity to achieve a profit above a certain threshold?

A fast algorithm for the QoS-aware MCF problem would allow transportation planners to address effectively such network and line planning issues by identifying capacity bottlenecks and proceed accordingly.

It is worth mentioning that the QoS-aware MCF problem is also fundamental in applications beyond the transportation domain. For instance, in networking (e.g., multimedia) applications over the Internet [8], or in information dissemination over various communication networks [3]. In such a setting, a "server" (owned by some service provider) sends information to "clients", who retrieve answers to queries they have posed regarding various types of information. Common queries are typically grouped together. Answering a query incurs a cost and a data acquisition time that depends on the communication capacity. When a "client" is provided with an non-optimal service (e.g., long data acquisition time due to capacity restrictions), s/he will most likely switch to another provider. On the other hand, the provider may reduce the cost of such a service in order to minimize the loss.

A related problem, called *max-flow with QoS guarantee* (QoS max-flow), has been considered in [2]. The problem asks for computing the maximum (unweighted) MCF routed along a set of paths whose cost does not exceed a specific bound, and has been shown to be NP-hard in [2]. In the same paper [2], a pseudopolynomial time approximation scheme for QoS max-flow is given. It can be easily seen that QoS max-flow is a special case of the QoS-aware MCF problem (Section 5).

In this paper, we show that the QoS-aware MCF problem can be formulated (in a non-straightforward manner) as a fractional packing LP, and provide a FPTAS for its approximate solution. Our algorithm builds upon the Garg & Könemann Lagrangian relaxation method for fractional packing LPs [5], combined with the phases technique by Fleischer [4]. A crucial step of the method is to construct an oracle that identifies the most violated constraint of the dual LP. While in the classical weighted MCF problem the construction of the oracle is harmless (reduces to the standard, single objective shortest path problem), this is *not* the case with the QoS-aware MCF problem. The construction turns out to be non-trivial, since it reduces to a multiobjective (actually non-additive) shortest path problem due to the QoS-elastic demands and values. Building upon a recent FPTAS for non-additive shortest paths [12], we are able to construct the required oracle and hence provide a FPTAS for the QoS-aware MCF problem. Our approach gives also a FPTAS for the QoS max-flow problem, thus improving upon the result in [2].

The rest of the paper is organized as follows. We start (Section 2) with some necessary preliminaries, a formal definition of the problem and its LP formulation. We then proceed (Section 3) with a review of the GK method [5] upon which our algorithm builds. Subsequently, we give the details of our FPTAS (Section 4), and present extensions of our results to constrained versions of the QoS-aware problem (Section 5). We conclude in Section 6.

2 Preliminaries and Problem Formulation

2.1 Problem Definition and LP Formulation

We are given a digraph $G = (V, E)$, along with a capacity function $u : E \rightarrow I\!R_0^+$ on its edges. We are also given a set of k commodities. A commodity i, $1 \leq i \leq k$, is a tuple $(s_i, t_i, d_i, wt_i(\cdot), \mathrm{f_i}(\cdot), \mathrm{v_i}(\cdot))$, whose attributes are defined as follows. Attributes $s_i \in V$ and $t_i \in V$ are the source and the target nodes, respectively, while $d_i \in I\!R_0^+$ is the demand of the commodity. The weight function $wt_i : E \rightarrow I\!R_0^+$ quantifies the *quality of service (QoS)* for commodity i (smaller weight means better QoS). For any s_i-t_i path p, $wt_i(p) := \sum_{e \in p} wt_i(e)$ and let $\delta_i(s_i, t_i)$ be the length of the shortest path from s_i to t_i w.r.t. the weight function $wt_i(\cdot)$. The non-decreasing function $\mathrm{f_i} : [1, \infty) \rightarrow [0, 1]$ is the *elasticity function* of i that determines the portion $\mathrm{f_i}(x)$ of the commodity's demand d_i that is lost if the provided path is x times worse than the shortest path w.r.t. $wt_i(\cdot)$; that is, if a units of d_i were supposed to be sent in case the provided path was shortest (optimal), then only $(1 - \mathrm{f_i}(x))a$ units will be shipped through the actually provided (non-optimal) path, while $\mathrm{f_i}(x)a$ units will be lost. Commodity i is also associated with a non-increasing *profit function* $\mathrm{v_i} : [1, \infty) \rightarrow I\!R_0^+$ that gives the profit $\mathrm{v_i}(x)$ from shipping one unit of flow of commodity i through a path that is x times worse than the shortest path w.r.t. $wt_i(\cdot)$. The objective is to maximize the total profit, i.e., the sum over all commodities and over all paths of the flow routed from every commodity on each path multiplied by the commodity's profit, subject to the capacity and demand constraints, and w.r.t. the QoS-elasticity of demands and profits. We call the above the *QoS-aware Multicommodity Flow* (MCF) problem.

Let $P_i = \{p : p \text{ is an } s_i\text{-}t_i \text{ path}\}$ be the set of *candidate paths* along which flow from commodity i can be sent. Consider such a particular path $p \in P_i$ and let $X_i(p) \in I\!R_0^+$ denote the flow of commodity i routed along p. The definition of the elasticity function implies that for each unit of flow of commodity i routed along p, there are $\frac{1}{1-\mathrm{f_i}(x)}$ units *consumed* from the demand of the commodity. Thus, we define a *consumption function* $\mathrm{h_i} : [1,\infty) \rightarrow [1,\infty)$ with $\mathrm{h_i}(x) = \frac{1}{1-\mathrm{f_i}(x)}$. Since $\mathrm{f_i}$ is non-decreasing, $\mathrm{h_i}$ is also non-decreasing. Accordingly, we define the *consumption* $h_i(p) \geq 1$ of a path p as the amount of demand consumed for each unit of flow routed along p:

$$h_i(p) = \mathrm{h_i}\left(\frac{wt_i(p)}{\delta_i(s_i,t_i)}\right).$$

Similarly, we define the *value* $v_i(p)$ of a path p as the profit from routing one unit of flow of commodity i through p:

$$v_i(p) = \mathrm{v_i}\left(\frac{wt_i(p)}{\delta_i(s_i,t_i)}\right).$$

Using the above definitions, the QoS-aware MCF problem can be described by the following linear program (LP).

$$\max \sum_{i=1}^{k} \sum_{p \in P_i} v_i(p) X_i(p) \tag{1}$$

$$s.t. \sum_{i=1}^{k} \sum_{e \in p, p \in P_i} X_i(p) \leq u(e), \forall e \in E \tag{2}$$

$$\sum_{p \in P_i} X_i(p) h_i(p) \leq d_i, \forall i = 1 \dots k \tag{3}$$

$$X_i(p) \geq 0, \forall i = 1 \dots k, \forall p \in P_i$$

2.2 Non-Additive Shortest Paths

In this section, we introduce the non-additive shortest path (NASP) problem that will be used as a subroutine in our FPTAS for the QoS-aware MCF problem.

In NASP, we are given a digraph $G = (V, E)$ and a d-dimensional function vector $\mathbf{c} : E \rightarrow [IR^+]^d$ associating each edge e with a vector of attributes $\mathbf{c}(e)$ and a path p with a vector of attributes $\mathbf{c}(p) = \sum_{e \in p} \mathbf{c}(e)$. We are also given a d-attribute non-decreasing and *non-linear* utility function $\mathcal{U} : [IR^+]^d \rightarrow IR$. The objective is to find a path p^*, from a specific source node s to a destination t, that minimizes the objective function, i.e., $p^* = \mathrm{argmin}_{p \in P(s,t)} \mathcal{U}(\mathbf{c}(p))$. (It is easy to see that in the case where $\mathcal{U}$ is linear, NASP reduces to the classical single-objective shortest path problem.) For the general case of non-linear $\mathcal{U}$, it is not difficult to see that NASP is NP-hard.

The first FPTAS for NASP were independently presented in [11] (for any $d > 1$ and polynomially bounded utility function) and in [1] (for $d = 2$ and quasi-polynomially bounded utility function), with the former having a better time complexity. Recently, an improved FPTAS for NASP was given [12] that holds for *any* $d > 1$ and a larger than quasi-polynomially bounded family of utility functions. The new result improves considerably upon those in [1,11] w.r.t. time (dependence on $1/\varepsilon$), number of objectives, and class of utility functions.

The following lemma is an immediate consequence of [12, Theorem 4].

Lemma 1. *Let the utility function of NASP be of the form* $\mathcal{U}([x_1, x_2]^T) = x_1\mathcal{U}_1(x_2) + \mathcal{U}_2(x_2)$*, where* $\mathcal{U}_1, \mathcal{U}_2$ *are any non-negative and non-decreasing functions. Then, for any* $\varepsilon > 0$*, there is an algorithm that computes an* $(1+\varepsilon)$*-approximation to the optimum of NASP in time* $O(n^2 m \frac{\log(nC_1)}{\varepsilon})$*, where* $C_1 = \frac{\max_{e\in E} c_1(e)}{\min_{e\in E} c_1(e)}$.

3 Review of the GK Method

The linear program for the QoS-aware MCF problem (given in Section 2.1) is a (pure) fractional packing LP, i.e., a linear program of the form $\max\{c^T x | Ax \leq b, x \geq 0\}$, where $A_{M\times N}$, $b_{M\times 1}$ and $c_{N\times 1}$ have positive entries. By scaling we also assume that $A(i,j) \leq b(i)$, $\forall i, j$. The dual of that problem is $\min\{b^T y | A^T y \geq c, y \geq 0\}$. In [5], Garg and Könemann present a remarkably elegant and simple FPTAS for solving fractional packing LPs. Their algorithm maintains a primal and a dual solution. At each step they identify the most violated constraint in the dual and increase the corresponding primal variable, and the dual variables. The most violated constraint is identified by using an exact oracle.

The algorithm works as follows. Let the *length* of a column j with respect to the dual variables y be $\text{length}_y(j) = \sum_i \frac{A(i,j)}{c(j)} y(i)$. Let $a(y)$ denote the length of the minimum-length column, i.e., $a(y) = \min_j \text{length}_y(j)$. Let also $D(y) = b^T y$ be the dual objective value with respect to y. Then, the dual problem is equivalent to finding an assignment y that minimizes $\frac{D(y)}{a(y)}$. The procedure is iterative. Let y_{k-1} be the dual variables and f_{k-1} be the value of the primal solution at the beginning of the k-th iteration. The initial values of the dual variables are $y_0(i) = \delta/b(i)$, where δ is a constant to be chosen later, and the primal variables are initially zero. In the k-th iteration, a call to an oracle is made that returns the minimum length column q of A, i.e., $\text{length}_{y_{k-1}}(q) = \alpha(y_{k-1})$. Let now $p = \text{argmin}_i \frac{b(i)}{A(i,q)}$ be the "minimum capacity" row. In this iteration, we increase the primal variable $x(q)$ by $\frac{b(p)}{A(p,q)}$, thus the primal objective becomes $f_k = f_{k-1} + c(q)\frac{b(p)}{A(p,q)}$. The dual variables are updated as

$$y_k(i) = y_{k-1}(i)\left(1 + \varepsilon\frac{b(p)/A(p,q)}{b(i)/A(i,q)}\right),$$

where $\varepsilon > 0$ is a constant depending on the desired approximation ratio. For brevity we denote $a(y_k)$ and $D(y_k)$ by $a(k)$ and $D(k)$, respectively. The procedure stops at the first iteration t such that $D(t) \geq 1$. The final primal solution constructed may not be feasible since some of the packing constraints may be violated. However, scaling the final value of the primal variables by $\log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}$ gives a feasible solution (see Lemma 6 in the Appendix).

The above algorithm can be straightforwardly extended to work with an approximate oracle[1]. Simply, in the k-th iteration we call an oracle that returns an $(1+w)$-approximation of the minimum length column of A. If q is the column returned by the oracle, then we have that $\text{length}_{y_{k-1}}(q) \leq (1+w)a(y_{k-1})$. By working similarly to [5] and choosing $\delta = (1+\varepsilon)((1+\varepsilon)M)^{-1/\varepsilon}$, we can show the following theorem (whose proof is in the Appendix for the sake of completeness).

Theorem 1. *There is an algorithm that computes an $(1-\varepsilon)^{-2}(1+w)$-approximation to the packing LP after at most $M\lceil\log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}\rceil = M\lceil\frac{1}{\varepsilon}(1+\log_{1+\varepsilon} M)\rceil$ iterations, where M is the number of rows.*

4 The FPTAS for QoS-aware Multicommodity Flows

In this section, we describe the (non-straightforward) details of solving the QoS-aware MCF problem by building upon the GK method. We start by obtaining the dual of the LP formulation of the QoS-aware MCF problem. We introduce for each edge e a dual variable $l(e)$ that corresponds to the capacity constraint (2) on e, and for each commodity i we introduce a dual variable ϕ_i that corresponds to the demand constraint (3) on i. The dual LP becomes

$$\min \ D = \sum_{e\in E} l(e)u(e) + \sum_{i=1}^{k} \phi_i d_i \tag{4}$$

$$s.t. \ \ l(p) + \phi_i h_i(p) \geq v_i(p), \forall i = 1\dots k, \forall p \in P_i \tag{5}$$

$$l(p) \geq 0, \forall i = 1\dots k, \forall p \in P_i \quad \phi_i \geq 0, \forall i = 1\dots k$$

where $l(p) := \sum_{e\in p} l(e)$.

To apply the GK method, it must hold $u(e) \geq 1$, $\forall e \in E$, and $d_i \geq h_i(p)$, $\forall 1 \leq i \leq k$, $p \in P_i$. To ensure this, we scale the capacities and demands by $\min\{\min_{e\in E} u(e), \min_{1\leq i\leq k}\frac{d_i}{h_i^{max}}\}$, where $h_i^{max} = h_i(\frac{(n-1)\max_{e\in E} wt_i(e)}{\delta_i(s_i,t_i)})$ is an upper bound on the maximum possible value of $h_i(\cdot)$.

Given an assignment (l,ϕ) for the dual variables, the length of a dual constraint is defined as $\text{length}_{(l,\phi)}(i,p) = \frac{l(p)+\phi_i h_i(p)}{v_i(p)}$ and the length of the most violated constraint is denoted by $a(l,\phi) = \min_{1\leq i\leq k}\min_{p\in P_i}\text{length}_{(l,\phi)}(i,p)$. The algorithm maintains a dual variable $l(e)$ for each edge e, initially equal to $\frac{\delta}{u(e)}$, and a dual variable ϕ_i for each commodity i, initially equal to $\frac{\delta}{d_i}$, where $\delta = (1+\varepsilon)((1+\varepsilon)(m+k))^{-\frac{1}{\varepsilon}}$.

The algorithm proceeds in iterations. Initially all flows are zero. In each iteration, it makes a call to an oracle that returns a commodity i' and a path

[1] Such an extension of the GK approach to work with approximate oracles was known before [6], and its combination with the phases technique of Fleischer [4] for solving packing problems has been first observed by Young [15] for solving the more general case of mixed packing LPs.

$p \in P_{i'}$ that approximately minimizes $\text{length}_{(l,\phi)}(i,q)$ over all $1 \leq i \leq k$ and $q \in P_i$; i.e., we have $\text{length}_{(l,\phi)}(i',p) \leq (1+\varepsilon)a(l,\phi)$. It then augments $\Delta = \min\{\frac{d'_i}{h'_i(p)}, \min_{e\in p} u(e)\}$ units of flow from commodity i' through p and updates the corresponding dual variables by setting $l(e) = l(e)(1+\varepsilon\frac{\Delta}{u(e)})$, $\forall e \in p$, and $\phi_{i'} = \phi_{i'}(1+\varepsilon\frac{\Delta h_{i'}(p)}{d_{i'}})$. The algorithm terminates at the first iteration for which $D = \sum_{e\in E} l(e)u(e) + \sum_{i=1}^k \phi_i d_i > 1$, and scales the final flow by $\log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}$.

We now turn to the most crucial step of the algorithm: to build a suitable approximate oracle to identify the most violated constraint (5) of the dual. Our task is to approximately minimize, overall $1 \leq i \leq k$ and $q \in P_i$, the function

$$\frac{l(q)+\phi_i h_i(q)}{v_i(q)} = \frac{l(q)+\phi_i \cdot \mathrm{h}_i\left(\frac{wt_i(q)}{\delta_i(s_i,t_i)}\right)}{\mathrm{v}_i\left(\frac{wt_i(q)}{\delta_i(s_i,t_i)}\right)}.$$

Note that for a fixed i, this requires the solution of a NASP instance with objective function $\mathcal{U}([x_1,x_2]^T) = \frac{x_1+\phi_i\mathrm{h}_i\left(\frac{x_2}{\delta_i(s_i,t_i)}\right)}{\mathrm{v}_i\left(\frac{x_2}{\delta_i(s_i,t_i)}\right)}$ and cost vector $\mathbf{c} = [l, wt_i]^T$. Note also that the above function is of the form required by Lemma 1 with $\mathcal{U}_1(x) = \frac{1}{\mathrm{v}_i\left(\frac{x}{\delta_i(s_i,t_i)}\right)}$ and $\mathcal{U}_2(x) = \phi_i \cdot \frac{\mathrm{h}_i\left(\frac{x}{\delta_i(s_i,t_i)}\right)}{\mathrm{v}_i\left(\frac{x}{\delta_i(s_i,t_i)}\right)}$. Consequently, we can apply Lemma 1 for any fixed i and make use of a non-additive shortest path routine $\bar{p} = \text{NASP}(G,s_i,t_i,l,wt_i,\varepsilon)$ that returns an s_i-t_i path $\bar{p}$ that approximately (within $(1+\varepsilon)$) minimizes the above function, overall $q \in P_i$, in time $O(n^2m\frac{\log(nL)}{\varepsilon})$, where $L = \frac{\max_{e\in E} l(e)}{\min_{e\in E} l(e)}$.

To efficiently implement the oracle, we do not call the NASP routine for every value of i. Instead, the oracle proceeds in phases (like in [4]), maintaining a lower bound estimation $\overline{a}$ of $a(l,\phi)$, initially equal to $\overline{a} = \frac{1}{1+\varepsilon}\min_{1\leq i\leq k}\{\frac{l(p_i)+\phi_i h_i(p_i)}{v_i(p_i)} | p_i = \text{NASP}(G,s_i,t_i,l,wt_i,\varepsilon)\}$. In each phase, the oracle examines the commodities one by one by performing NASP computations. For each commodity i the oracle returns a path $p = \text{NASP}(G,s_i,t_i,l,wt_i,\varepsilon)\}$ for which $\frac{l(p)+\phi_i h_i(p)}{v_i(p)} \leq \overline{a}(1+\varepsilon)^2$. As long as such a path can be found, the oracle sticks to commodity i. Otherwise, it continues with commodity $i+1$. After all k commodities are considered in a phase, we know that $a(l,\phi) \geq (1+\varepsilon)\overline{a}$ and proceed to the next phase by setting $\overline{a} = (1+\varepsilon)\overline{a}$. The pseudocodes of our algorithm and the oracle are given in Fig. 1.

To discuss correctness and time bounds, we start with the following lemma that establishes an upper bound on the ratio of the lengths of the minimum length column at the start and the end of the GK algorithm.

Lemma 2. *Let $a(0)$ and $a(t)$ be the lengths of the minimum length column at the start and the end of the algorithm, respectively. Then, $\frac{a(t)}{a(0)} \leq \frac{1+\varepsilon}{\delta}$.*

Proof. By the initial values of the dual variables, we have $a(0) = \min_j \sum_i \frac{A(i,j)}{c(j)} y_0(i) = \delta \cdot \min_j \sum_i \frac{A(i,j)}{c(j)b(i)}$. Since now the algorithm stops at the first iteration t such

```
QoS-MCF($G, u, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d}, \boldsymbol{wt}, \boldsymbol{v}, \varepsilon$) {
  forall $e \in E$ { $l(e) = \frac{\delta}{u(e)}$ }
  for $i = 1$ to $k$ { $\phi_i = \frac{\delta}{d_i}$ }
  for $i = 1$ to $k$ { forall $e \in E$ { $X_i(e) = 0$ }}
  $D = (m + k)\delta$;
  for $i = 1$ to $k$ { $p_i = \mathrm{NASP}(G, s_i, t_i, l, wt_i, \varepsilon)$ }
  $\overline{a} = \frac{1}{1+\varepsilon} \min_{1 \le i \le k} \left\{ \frac{l(p_i)+\phi_i h_i(p_i)}{v_i(p_i)} \right\}$;
  $i = 1$;
  while $D \le 1$ {
    $(p, i, \overline{a})$ = QoS-MCF-oracle($G, \boldsymbol{s}, \boldsymbol{t}, l, \boldsymbol{wt}, \boldsymbol{v}, \boldsymbol{\phi}, \varepsilon, i, \overline{a}$);
    $\Delta = \min\{\frac{d_i}{h_i(p)}, \min_{e \in p} u(e)\}$;
    $X_i(p) = X_i(p) + \Delta$;
    forall $e \in p$ do $l(e) = l(e)(1 + \varepsilon\frac{\Delta}{u(e)})$;
    $\phi_i = \phi_i(1 + \varepsilon\frac{\Delta h_i(p)}{d_i})$;
    $D = D + \varepsilon\Delta\frac{l(p)+\phi_i h_i(p)}{v_i(p)}$;
  }
  for $i = 1$ to $k$ { forall $e \in E$ { $X_i(e) = X_i(e)/\log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}$ }}
}

QoS-MCF-oracle($G, \boldsymbol{s}, \boldsymbol{t}, l, \boldsymbol{wt}, \boldsymbol{v}, \boldsymbol{\phi}, \varepsilon, j, \overline{a}$) {
  while true {
    for $i = j$ to $k$ {
      $p = \mathrm{NASP}(G, s_i, t_i, l, wt_i, \varepsilon)$;
      if $\frac{l(p)+\phi_i h_i(p)}{v_i(p)} \le \overline{a}(1+\varepsilon)^2$
        return $(p, i, \overline{a})$;
    }
    $\overline{a} = \overline{a}(1+\varepsilon)$; /* update rule for
                         next phase */
  }
}
```

Fig. 1. The approximation algorithm for the QoS-MCF problem.

that $D(t) > 1$ and the dual variables increase by at most $1+\varepsilon$ in each iteration, it holds that $D(t) \leq 1+\varepsilon$. Consequently, $\sum_i b(i)y_t(i) \leq 1+\varepsilon$, which implies that $y_t(i) \leq (1+\varepsilon)\frac{1}{b(i)}$, $\forall i$. Hence, $a(t) = \min_j \sum_i \frac{A(i,j)}{c(j)} y_t(i) \leq (1+\varepsilon) \cdot \min_j \sum_i \frac{A(i,j)}{c(j)b(i)} = \frac{1+\varepsilon}{\delta} a(0)$. □

The following lemma establishes the approximation guarantee for the oracle.

Lemma 3. *A call to the oracle returns an $(1+\varepsilon)^2$-approximation of the most violated constraint in the dual.*

Proof. Let $\overline{a}_j$ be the value of $\overline{a}$ during the j-th phase of the algorithm. It suffices to show that for all phases $j \geq 1$, $\overline{a}_j \leq a(l,\phi)$.

Initially ($j=1$), we set $\overline{a}_1 = \frac{1}{1+\varepsilon} \min_{1\leq i\leq k}\{\frac{l(p_i)+\phi_i h_i(p_i)}{v_i(p_i)} | p_i = \text{NASP}(G, s_i, t_i, l, wt_i, \varepsilon)\}$. By the definition of the NASP routine, we get $\overline{a}_1 \leq \frac{1}{1+\varepsilon} \min_{1\leq i\leq k}\{(1+\varepsilon) \min_{p\in P_i} \frac{l(p)+\phi_i h_i(p)}{v_i(p)}\} = a(l,\phi)$.

For any subsequent phase $j > 1$, consider phase $j-1$. The oracle finishes the examination of a commodity i and proceeds with $i+1$ only when a call to NASP$(G, s_i, t_i, l, wt_i, \varepsilon)$ in phase $j-1$ returns a path p_i for which $\frac{l(p_i)+\phi_i h_i(p_i)}{v_i(p_i)} > \overline{a}_{j-1}(1+\varepsilon)^2$. This inequality and the definition of the NASP routine imply that at the end phase $j-1$, we have for each commodity i

$$\overline{a}_{j-1}(1+\varepsilon)^2 < (1+\varepsilon) \min_{p\in P_i} \frac{l(p)+\phi_i h_i(p)}{v_i(p)}.$$

Hence, by the definition of $a(l,\phi)$, and since $l(e)$ can only increase during the algorithm, at the end of the phase we have $\overline{a}_{j-1}(1+\varepsilon)^2 < (1+\varepsilon)a(l,\phi)$. Since $\overline{a}_j = \overline{a}_{j-1}(1+\varepsilon)$, we get $\overline{a}_j < a(l,\phi)$. □

To establish a bound on the time complexity of the algorithm, we need to count the number of NASP computations. Clearly, at most one NASP computation is needed per augmentation of flow. The rest of NASP computations (not leading to an augmentation) are bounded by k times the number of phases. The following lemma establishes a bound on the total number of phases.

Lemma 4. *The number of phases of algorithm QoS-MCF is bounded by $\lceil \frac{1}{\varepsilon}(1 + \log_{1+\varepsilon}(m+k)) \rceil + 2$.*

Proof. Let $a(0)$ and $a(t)$ be the lengths of the most violated constraint at the start and the end of the algorithm, respectively. Let now $\overline{a}_j$ be the value of $\overline{a}$ during the j-th phase of the algorithm, and T be the last phase of the algorithm.

Initially, we set $\overline{a}_1 = \frac{1}{1+\varepsilon} \min_{1\leq i\leq k} \left\{ \frac{l(p_i)+\phi_i h_i(p_i)}{v_i(p_i)} | p_i = \text{NASP}(G, s_i, t_i, l, wt_i, \varepsilon) \right\}$ and by the definition of the NASP routine we get that $a(0) \leq (1+\varepsilon)\overline{a}_1$. From the proof of Lemma 3, we have that $\overline{a}_T \leq a(t)$, and from Lemma 2 we get that $a(t) \leq \frac{1+\varepsilon}{\delta} a(0)$. Combining the last three inequalities we get $\overline{a}_T \leq \frac{(1+\varepsilon)^2}{\delta}\overline{a}_1$. By the update rule for $\overline{a}$ on each phase, we have that $\overline{a}_T = \overline{a}_1(1+\varepsilon)^{T-1}$, and therefore $\overline{a}_1(1+\varepsilon)^{T-1} \leq \frac{(1+\varepsilon)^2}{\delta}\overline{a}_1$, which implies that $T \leq \log_{1+\varepsilon} \frac{(1+\varepsilon)^3}{\delta}$. Hence, the number of phases is bounded by $\lceil \log_{1+\varepsilon} \frac{(1+\varepsilon)^3}{\delta} \rceil = \lceil \frac{1}{\varepsilon}(1+\log_{1+\varepsilon}(m+k)) \rceil + 2$, since $\delta = (1+\varepsilon)((1+\varepsilon)(m+k))^{-\frac{1}{\varepsilon}}$. □

We are now ready for the main result of this section.

Theorem 2. *There is an algorithm that computes an $(1-\varepsilon)^{-2}(1+\varepsilon)^2$-approximation to the QoS-aware MCF problem in time $O((\frac{1}{\varepsilon})^3(m+k)\log(m+k)mn^2(\frac{1}{\varepsilon}\log(m+k)+\log(nU))$, where n is the number of nodes, m is the number of edges, k is the number of commodities, and $U = \frac{\max_{e\in E} u(e)}{\min_{e\in E} u(e)}$.*

Proof. From Theorem 1 (with $M = m+k$) and Lemma 3 we have that the algorithm computes an $(1-\varepsilon)^{-2}(1+\varepsilon)^2$-approximation to the optimal and terminates after at most $(m+k)\lceil \frac{1}{\varepsilon}(1+\log_{1+\varepsilon}(m+k)) \rceil$ augmentations. Since for each phase at most k NASP computations do not lead to an augmentation, we get from Lemma 4 that the oracle performs at most $k\lceil \frac{1}{\varepsilon}(1+\log_{1+\varepsilon}(m+k)) \rceil + 2k$ NASP computations not leading to an augmentation. Therefore, the total number of NASP computations during an execution of the algorithm is $O(\frac{1}{\varepsilon}(m+k)\log_{1+\varepsilon}(m+k)) = O((\frac{1}{\varepsilon})^2(m+k)\log(m+k))$.

A NASP computation is carried out in time $O(\frac{1}{\varepsilon}n^2 m\log(nL))$, where $L = \frac{\max_{e\in E} l(e)}{\min_{e\in E} l(e)}$. From the initialization of $l(e)$, and since they can only increase during the algorithm, it is clear that $\min_{e\in E} l(e) \geq \frac{\delta}{\max_{e\in E} u(e)}$. Since now the algorithm stops at the first iteration such that $\sum_{e\in E} l(e)u(e) + \sum_{i=1}^{k} \phi_i d_i > 1$ and the dual variables increase by at most $1+\varepsilon$ in each iteration, it holds that $\sum_{e\in E} l(e)u(e) + \sum_{i=1}^{k} \phi_i d_i \leq 1+\varepsilon$. Consequently at the end of the algorithm we have $l(e) \leq \frac{(1+\varepsilon)}{u(e)}$, $\forall e \in E$, and thus $\max_{e\in E} l(e) \leq \frac{1+\varepsilon}{\min_{e\in E} u(e)}$. Hence, $L \leq \frac{1+\varepsilon}{\delta}U$. By our choice of $\delta = (1+\varepsilon)((1+\varepsilon)(m+k))^{-\frac{1}{\varepsilon}}$, we have that $L \leq ((1+\varepsilon)(m+k))^{\frac{1}{\varepsilon}}U$, and hence the time required for a NASP computation is $O(\frac{1}{\varepsilon}mn^2(\frac{1}{\varepsilon}\log(m+k)+\log(nU)))$. Thus, we get an algorithm that computes an $(1-\varepsilon)^{-2}(1+\varepsilon)^2$-approximation to the QoS-aware MCF problem in time $O([(\frac{1}{\varepsilon})^2(m+k)\log(m+k)][\frac{1}{\varepsilon}mn^2(\frac{1}{\varepsilon}\log(m+k)+\log(nU)]) = O((\frac{1}{\varepsilon})^3(m+k)\log(m+k)mn^2(\frac{1}{\varepsilon}\log(m+k)+\log(nU))$, which is polynomial to the input and $\frac{1}{\varepsilon}$. □

5 Extensions

Better bounds can be obtained for the *constrained* version of the QoS-aware MCF problem. In that version all profits are constant (non QoS-elastic) — i.e., $v_i(p) = v_i$, $\forall i, p \in P_i$ — and there is an upper bound (constraint) on the QoS per path provided — i.e., the consumption functions are now defined as:

$$h_i(p) = \begin{cases} 1 & \text{if } wt_i(p) \leq b_i \\ +\infty & \text{otherwise} \end{cases}, \quad \forall i, p \in P_i.$$

The objective is to maximize the total profit. In this case, we can achieve a FPTAS by implementing the oracle using a FPTAS for the Restricted Shortest Path

(RSP) problem instead of NASP. The currently best FPTAS for general digraphs is due to Lorenz and Raz [7], and runs in $O(mn(\log\log n+1/\varepsilon))$ time. Arguing as in Theorem 2 and taking into account the time complexity of the FPTAS for RSP [7], we can achieve a running time of $O((\frac{1}{\varepsilon})^2(m+k)\log(m+k)mn(\log\log n+1/\varepsilon))$ for the constrained version of the QoS-aware MCF problem.

For the version of the problem with unbounded demands, which constitutes the QoS max flow problem defined in [2], we can achieve a better time bound of $O((\frac{1}{\varepsilon})^2 nm^2 \log m(\log\log n + 1/\varepsilon))$, since the number of constraints in its corresponding LP is m.

6 Conclusions

We considered the QoS-aware MCF problem, a natural and important generalization of the weighted multicommodity flow problem with elastic demands and values that is fundamental in transportation planning (and beyond). We formulated the problem as a fractional packing LP, and provided a FPTAS for its solution by building upon a Lagrangian relaxation method combined with a recent FPTAS for non-additive shortest paths. Finally, we presented better FPTAS for constrained versions of the QoS-aware MCF problem.

Acknowledgments. We are indebted to Naveen Garg, Jochen Könemann, Spyros Kontogiannis, Frank Geraets (aka Wagner) for various useful discussions, and to Christos Papadimitriou for bringing [2] to our attention.

References

1. H. Ackermann, A. Newman, H. Röglin, and B. Vöcking, “Decision Making Based on Approximate and Smoothed Pareto Curves”, in *Algorithms and Computation* – ISAAC 2005, LNCS Vol. 3827 (Springer 2006), pp. 675-684; full version as Tech. Report AIB-2005-23, RWTH Aachen, December 2005.
2. K. Chaudhuri, C. Papadimitriou, and S. Rao, “Optimum Routing with Quality of Service Constraints”, manuscript, 2004.
3. A. Datta, D. Vandermeer, A. Celik, and V. Kumar, “Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users”, *ACM Transactions on Database Systems*, 24:1 (1999), pp. 1-79.
4. L.K. Fleischer, “Approximating fractional multicommodity flows independent of the number of commodities”, *SIAM Journal on Discrete Mathematics*, 13:4 (2000), pp. 505-520.
5. N. Garg and J. Könemann, “Faster and simpler algorithms for multicommodity flow and other fractional packing problems”, in *Proc. 39th IEEE Symposium on Foundations of Computer Science* – FOCS’98, (IEEE CS Press, 1998), pp.300-309.
6. N. Garg and J. Könemann, personal communication, 2005.
7. D.H. Lorenz and D. Raz, “A simple efficient approximation scheme for the restricted shortest path problem”, *Operations Research Letters*, 28 (2001) pp.213-219.

8. P. Van Mieghem, F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Sole-Pareta, and S. Sanchez-Lopez, "Quality of Service Routing", Chapter 3 in *Quality of Future Internet Services*, LNCS Vol. 2856 (Springer-Verlag, 2003), pp. 80-117.
9. PIN project (Projekt Integrierte Netzoptimierung), Deutsche Bahn AG, 2000.
10. S. Plotkin, D. Shmoys, and E. Tardos, "Fast Approximation Algorithms for Fractional Packing and Covering Problems", *Mathematics of Operations Research* 20 (1995), pp. 257-301.
11. G. Tsaggouris and C. Zaroliagis, "Improved FPTAS for Multiobjective Shortest Paths with Applications", CTI Techn. Report TR-2005/07/03, July 2005.
12. G. Tsaggouris and C. Zaroliagis, "Multiobjective Optimization: Improved FPTAS for Shortest Paths and Non-linear Objectives with Applications", CTI Techn. Report TR-2006/03/01, March 2006. Preliminary version in Proc. ISAAC 2006, LNCS (Springer, 2006).
13. F. Wagner, "Challenging Optimization Problems at Deutsche Bahn", AMORE Workshop (invited talk), 1999.
14. F. Wagner (Deutsche Bahn AG), personal communication, 2004.
15. N. Young, "Sequential and Parallel Algorithms for Mixed Packing and Covering", in *Proc. 42nd IEEE Symp. on Foundations of Computer Science* – FOCS 2001, pp. 538-546.

A APPENDIX

A.1 Proof of Theorem 1

The analysis is straightforward from [5]. We only consider an approximate oracle. In order to prove Theorem 1 we need the next two lemmata. In the first lemma, we establish a bound on the ratio of the optimal dual value to the primal objective value at the end of the algorithm.

Lemma 5. *Let* $\beta = \min_y \frac{D(y)}{a(y)}$ *be the optimal dual value and let* t *be the last iteration of the algorithm. The ratio of the optimal dual value to the primal objective value at the end of the algorithm is bounded by* $\frac{\varepsilon(1+w)}{\ln(1/M\delta)}$.

Proof. For each iteration $k \geq 1$ it is

$$\begin{aligned} D(k) &= \sum_i b(i) y_k(i) \\ &= \sum_i b(i) y_{k-1}(i) + \varepsilon \frac{b(p)}{A(p,q)} \sum_i A(i,q) y_{k-1}(i) \\ &\leq D(k-1) + (1+w)\varepsilon(f_k - f_{k-1}) a(k-1) \end{aligned}$$

which implies that

$$D(k) \leq D(0) + (1+w)\varepsilon \sum_{l=1}^{k} (f_l - f_{l-1}) a(l-1).$$

Since $\beta = \min_y D(y)/a(y)$ it is $\beta \leq D(l-1)/a(l-1), \forall l = 1 \dots k$, and thus

$$D(k) \leq M\delta + \frac{(1+w)\varepsilon}{\beta} \sum_{l=1}^{k} (f_l - f_{l-1}) D(l-1).$$

Observe now that for fixed k, this right hand side is maximized by setting $D(l-1)$ to its maximum possible value for all $1 \leq l-1 < k$, and let us denote this maximum value by $D'(k)$, i.e.,

$$D'(k) = M\delta + \frac{(1+w)\varepsilon}{\beta} \sum_{l=1}^{k} (f_l - f_{l-1}) D'(l-1).$$

Consequently,

$$\begin{aligned}
D(k) &\leq D'(k) \\
&= D'(k-1) + \frac{(1+w)\varepsilon}{\beta}(f_k - f_{k-1}) D'(k-1) \\
&= D'(k-1) \left(1 + \frac{(1+w)\varepsilon}{\beta}(f_k - f_{k-1}) \right) \\
&\leq D'(k-1) e^{\frac{(1+w)\varepsilon}{\beta}(f_k - f_{k-1})} \\
&\leq D'(0) e^{\frac{(1+w)\varepsilon}{\beta} \sum_{l=1}^{k}(f_l - f_{l-1})} \\
&\leq D'(0) e^{\frac{(1+w)\varepsilon}{\beta}(f_k - f_0)}
\end{aligned}$$

Since now $D'(0) = M\delta$ and $f_0 = 0$ it follows that

$$D(k) \leq M\delta e^{(1+w)\varepsilon f_k / \beta}.$$

From our stopping condition it is $1 \leq D(t) \leq M\delta e^{(1+w)\varepsilon f_t / \beta}$ and hence

$$\frac{\beta}{f_t} \leq \frac{\varepsilon(1+w)}{\ln(1/M\delta)}.$$

□

The final primal solution constructed may not be feasible since some of the packing constraints may be violated. The second lemma shows that the final primal assignment can be appropriately scaled as to obtain a feasible solution.

Lemma 6. *Scaling the final primal assignment by* $\log_{1+\varepsilon}\left(\frac{1+\varepsilon}{\delta}\right)$, *we obtain a feasible solution to the fractional packing LP.*

Proof. When we pick a column q and increase the left-hand-side of the i-th constraint by $\frac{A(i,q)b(p)}{A(p,q)b(i)}$. Simultaneously we increase the dual variable $y(i)$ by a multiplicative factor of $1 + \varepsilon \frac{A(i,q)b(p)}{A(p,q)b(i)}$. By the definition of p it follows that $\frac{A(i,q)b(p)}{A(p,q)b(i)} \leq 1$ and thus increasing the left-hand-side of the i-th constraint by one causes an increase in $y(i)$ by a multiplicative factor of $1+\varepsilon$. Since t is the first iteration for which $D(t) > 1$, it is $y_{t-1}(i) < 1/b(i)$ and thus $y_t(i) < (1+\varepsilon)/b(i)$. Since now $y_0(i) < \delta/b(i)$ it follows that the left-hand-side of the i-th constraint is no more than $\log_{1+\varepsilon}\left(\frac{1+\varepsilon}{\delta}\right)$ for any i. Thus scaling the primal solution by $\log_{1+\varepsilon}\left(\frac{1+\varepsilon}{\delta}\right)$ gives a feasible solution. □

We now proceed with the proof of Theorem 1.

Proof. In the k-th iteration we increase the dual variable of the "minimum capacity" row by a factor of $1+\varepsilon$. Since we stop the algorithm at the first iteration t such that $D(t) > 1$ it follows that $D(t) < 1+\varepsilon$ and thus $y_t(i) < \frac{1+\varepsilon}{b(i)}$ for any row. Since now $y_0(i) = \frac{\delta}{b(i)}$ and $y_t(i) < \frac{1+\varepsilon}{b(i)}$ and there are M rows the total number of iterations is at most $M\lceil \log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}\rceil = M\lceil \frac{1}{\varepsilon}\log_{1+\varepsilon} M\rceil$, by choosing $\delta = (1+\varepsilon)((1+\varepsilon)M)^{-1/\varepsilon}$.

The ratio of the optimal dual value to objective value of the scaled final primal assignment is $\gamma = \frac{\beta}{f_t}\log_{1+\varepsilon}\left(\frac{1+\varepsilon}{\delta}\right)$ by substituting the bound on $\frac{\beta}{f_t}$ from Lemma 5 we get

$$\gamma \le \frac{\varepsilon(1+w)}{\ln(1/M\delta)}\log_{1+\varepsilon}\left(\frac{1+\varepsilon}{\delta}\right) = \frac{\varepsilon(1+w)}{\ln(1+\varepsilon)}\frac{\ln\frac{1+\varepsilon}{\delta}}{\ln(1/M\delta)}$$

For $\delta = (1+\varepsilon)((1+\varepsilon)M)^{-1/\varepsilon}$, the ratio $\frac{\ln\frac{1+\varepsilon}{\delta}}{\ln(1/M\delta)}$ equals $(1-\varepsilon)^{-1}$, hence we have

$$\gamma \le \frac{\varepsilon(1+w)}{(1-\varepsilon)\ln(1+\varepsilon)} \le \frac{\varepsilon(1+w)}{(1-\varepsilon)(\varepsilon-\varepsilon^2/2)} \le (1-\varepsilon)^{-2}(1+w)$$

□