

Maximum Multicommodity Flows Over Time without Intermediate Storage^{*}

Martin Groß and Martin Skutella

Fakultät II – Mathematik und Naturwissenschaften,
Institut für Mathematik, Sekr. MA 5-2
Technische Universität Berlin, Straße des 17. Juni 136,
10623 Berlin, Germany
{gross,skutella}@math.tu-berlin.de

Abstract. Flows over time generalize classical “static” network flows by introducing a temporal dimension. They can thus be used to model non-instantaneous travel times for flow and variation of flow values over time, both of which are crucial characteristics in many real-world routing problems. There exist two different models of flows over time with respect to flow conservation: one where flow might be stored temporarily at intermediate nodes and a stricter model where flow entering an intermediate node must instantaneously progress to the next arc. While the first model is in general easier to handle, the second model is often more realistic since in applications like, e. g., road traffic, storage of flow at intermediate nodes is undesired or even prohibited. The main contribution of this paper is a fully polynomial time approximation scheme (FPTAS) for (min-cost) multi-commodity flows over time without intermediate storage. This improves upon the best previously known $(2+\varepsilon)$ -approximation algorithm presented 10 years ago by Fleischer and Skutella (IPCO 2002).

1 Introduction

Two important characteristics of real-world network routing problems are the facts that flow along arcs varies over time and that flow does not arrive instantaneously at its destination but only after a certain delay. As none of these two characteristics is captured by classical network flows, the more powerful model of flows over time has been shifted into the focus of current research. Various interesting applications and examples can, for instance, be found in the surveys of Aronson [1] and Powell, Jaillet and Odoni [15]. A more recent introduction to the area of flows over time is given in [16], and a recent application can, e.g., be found in [2].

Results from the literature. Network flows over time have first been studied by Ford and Fulkerson [7, 8], who developed a reduction of flow over time problems

^{*} Supported by the DFG Research Center MATHEON “Mathematics for key technologies” in Berlin.

to static flow problems using *time-expanded networks*. This technique requires a discrete time model, but works for virtually all flow-over-time problems at the cost of a pseudo-polynomial blow-up in network size. Note that results for discrete time models often carry over to continuous time models (see, e. g., Fleischer and Tardos [6]). Furthermore, Ford and Fulkerson [7, 8] describe an efficient algorithm for the *maximum flow over time problem*, i. e., the problem of sending the maximum possible amount of flow from a source to a sink within a given time horizon. The algorithm performs a static minimum cost flow computation to find a static flow that is then sent repeatedly through the network.

Related to this problem is the *quickest flow problem*, which asks for the minimum amount of time necessary to send a given amount of flow from the source to the sink. Burkard, Dlaska and Klinz [3] describe a strongly polynomial algorithm for this problem by embedding Ford and Fulkersons' algorithm into Megiddo's parametric search framework [14]. A generalization of this problem, the *quickest transshipment problem*, asks for the minimum amount of time to fulfill given supplies and demands at the nodes. Hoppe and Tardos [11, 12] give a polynomial algorithm relying on submodular function minimization for this.

The *minimum cost flow over time problem* consists of computing a flow of minimum cost, fulfilling given supplies and demands within a specified time horizon. Klinz and Woeginger [13] show that this problem is weakly NP-hard; it can still be solved in pseudo-polynomial time using time-expanded networks.

The problems discussed above can all be generalized to the case of *multiple commodities*. In this setting, we either have a set of source-sink pairs between which flow is sent, or a set of supply-demand vectors specifying different types of supplies/demands that need to be routed in the network. Hall, Hippler, and Skutella [10] show that the multicommodity flow over time problem is NP-hard, even on series-parallel graphs. For multiple commodities, there are cases where storage of flow at intermediate nodes (i. e., nodes that are neither source nor sink nodes) can be necessary for obtaining an optimal solution. If storage at intermediate nodes and non simple flow paths are forbidden, the multicommodity flow over time problem is strongly NP-hard [10] and has no FPTAS, if $P \neq NP$.

Fleischer and Skutella [4, 5] introduce the concept of *condensed time-expanded networks*, which rely on a rougher discretization of time, resulting in a blow-up in network size that is polynomially bounded. This, however, makes it necessary to increase the time-horizon slightly but still yields an FPTAS for the quickest multicommodity transshipment problem with intermediate storage. The result can be generalized to the quickest multicommodity flow over time problem with bounded cost, if costs on the arcs are non-negative. It is important to emphasize that the underlying technique critically relies on the fact that flow might be stored at intermediate nodes and that there are optimal solutions that send flow only along simple paths through the network. For the case where intermediate storage of flow is prohibited, there is a $(2 + \varepsilon)$ -approximation algorithm based on entirely different ideas.

Contribution of this paper. There are numerous flow over time problems where non-simple flow paths are required for optimal solutions. Examples of such prob-

lems are multicommodity flows over time without intermediate storage, minimum cost flows over time with arbitrary (possibly negative) arc costs, and generalized flows over time. As in the first problem, the necessity of non-simple flow paths is often closely related to upper bounds on the amount of flow that can be stored at intermediate nodes. Such bounds do exist in most real-world applications and in many cases like, e. g., traffic networks intermediate storage of flow is forbidden completely.

Inspired by the work of Fleischer and Skutella, we present a new, more elaborate condensation technique for time-expanded networks whose analysis no longer requires that flow is being sent along simple paths only. Fleischer and Skutella use the number of arcs on a path to bound the rounding error, which does not yield good bounds with non-simple paths due to the potentially high number of arcs in them. To this end, we introduce a new type of LP formulation for flow over time problems that is somewhere in between an arc-based and a path-based formulation. This LP has a dual which can be approximately separated in polynomial time for most flow problems (more precisely, for flow problems that use standard flow conservation). We start by first studying the separation problem from a general point of view in Section 2. While this problem is NP-hard, we describe an FPTAS for it (which is sufficient for our purposes) that uses dynamic programming in conjunction with Meggido’s parametric search framework [14]. In Section 3 we apply this result to the maximum multicommodity flow over time problem without intermediate storage and show that it yields an FPTAS for this NP-hard problem. Furthermore, in Section 4 we extend our techniques to the minimum cost multicommodity flow over time problem with non-negative arc costs.

2 The Restricted Minimum Cost s - t -Sequence Problem

In this section, we define the restricted minimum cost s - t -sequence problem. Informally speaking, this problem consists of finding an arc sequence of minimum cost between two nodes whose length is restricted to lie in a given interval. This length constraint implies that it can be necessary to incorporate cycles in an optimal solution. We present an FPTAS for this NP-hard problem. In Section 3 we make use of this result to solve a dual separation problem.

In order to define our problem formally, we need some notations for arc sequences first. Given a directed graph $G = (V, A)$, a v - w -sequence S is a sequence of arcs $(a_1, a_2, \dots, a_{|S|})$, $a_i = (v_i, v_{i+1}) \in A$ with $v = v_1$, $w = v_{|S|+1}$. A v - w -path is a v - w -sequence with $v_i \neq v_j$ for all $i \neq j$. A v -cycle is a v - v -sequence with $v_i = v_j$ implying $i = j$ or $i, j \in \{1, |S| + 1\}$. Thus, our paths and cycles are simple arc sequences.

If an arc a is contained in a sequence S , we write $a \in S$. We refer to the number of occurrences of an arc a in a sequence S as $\#(a, S)$. By $S[\rightarrow a_i]$ we denote the subsequence $(a_1, a_2, \dots, a_{i-1})$ of S . Furthermore, we extend the concept of incidence from arcs to sequences and write (by overloading notation) $S \in \delta_v^+$ if $a_1 \in \delta_v^+ := \{a = (v, \cdot) \mid a \in A\}$ and $S \in \delta_v^- := \{a = (\cdot, v) \mid a \in A\}$

if $a_{|S|} \in \delta_v^-$. We also extend arc attributes $f_a \in \mathbb{R}, a \in A$ (e.g., costs, lengths, transit times, etc.) to sequences $S = (a_1, \dots, a_{|S|})$ by defining $f_S := \sum_{i=1}^{|S|} f_{a_i}$.

Finally notice that we can decompose an s - t -sequence S into an s - t -path and cycles. With these notations, we can define our problem more formally.

Definition 1. *An instance of the restricted minimum cost s - t -sequence problem is given by a graph $G = (V, A)$ with costs $c_a \geq 0$ and lengths $\ell_a \geq 0$ on the arcs $a \in A$, two designated nodes $s, t \in V$ and a threshold value $\Delta > \ell^*$, with $\ell^* := \max_{a \in A} \ell_a$. The task is to find an s - t -sequence S of minimum cost $c_S := \sum_{a \in S} \#(a, S) \cdot c_a$ under the constraint that $\Delta - \ell^* \leq \ell_S \leq \Delta$.*

This problem is obviously NP-hard (e.g., by reduction of the length-bounded shortest path problem). The FPTAS that we will develop for this problem approximates the length constraint, i.e., it computes a solution at least as good as an optimal solution but with a relaxed length constraint $(1 - \varepsilon)\Delta - \ell^* \leq \ell_S \leq (1 + \varepsilon)\Delta$ for a given $\varepsilon > 0$. We begin by classifying the arcs into two groups.

Definition 2. *Given $\varepsilon > 0$, an arc $a \in A$ is short if $\ell_a \leq \varepsilon\Delta/n$, and long otherwise. A sequence is short, if all of its arcs are short, and long otherwise.*

It follows that a solution to the restricted minimum cost s - t -sequence problem can – due to the length constraint – contain at most n/ε long arcs. Since a long cycle needs to contain at least one long arc, it follows that at most n^2/ε arcs of the solution can be part of long cycles; the remaining arcs are part of an s - t -path or short cycles. Due to the fact that a path does not contain duplicate nodes, we make the following observation.

Observation 1. *Let S be a solution to the restricted minimum cost s - t -sequence problem. Then S contains at most $n + n^2/\varepsilon$ arcs that are not part of short cycles.*

For our FPTAS, we would like to round the arc lengths without introducing too much error. Sequences without short cycles are fine, because they have a nicely bounded number of arcs. However, we cannot simply restrict ourselves to sequences without short cycles. Therefore, we now define a *nice v - w -sequence* S to be a v - w -sequence that contains at most one short cycle, but is allowed to make several passes of this cycle. We will see that we can handle such sequences efficiently while losing not too much accuracy. Notice that we can ignore zero-length cycles, as these cycles do neither help us reach t nor can they improve the objective value.

Lemma 1. *For an s - t -sequence S with a path-cycle-decomposition, let C be a short cycle in the decomposition of S with maximal length-to-cost ratio $\frac{\sum_{a \in C} \ell_a}{\sum_{a \in C} c_a}$ and let ℓ_{short} be the total length of all short cycles in the decomposition of S . (Zero-cost cycles are assumed to have infinite ratio.) We define a corresponding nice sequence S' with cycle C by removing all short cycles and adding $\left\lfloor \frac{\ell_{short}}{c_C} \right\rfloor$ copies of C . Then $\ell_S - \varepsilon\Delta \leq \ell_S - \ell_C \leq \ell_{S'} \leq \ell_S$ and $c_{S'} \leq c_S$.*

Proof. Let S^* be the sequence obtained by removing all short cycles from S . S^* is then a sequence without any short cycles and length $\ell_{S^*} = \ell_S - \ell_{short}$. Now we add $\left\lfloor \frac{\ell_{short}}{\ell_C} \right\rfloor$ copies of C and gain S' . It follows that

$$\begin{aligned} \ell_{S'} &= \ell_{S^*} + \left\lfloor \frac{\ell_{short}}{\ell_C} \right\rfloor \ell_C = \ell_S - \ell_{short} + \left\lfloor \frac{\ell_{short}}{\ell_C} \right\rfloor \ell_C \\ &\leq \ell_S - \ell_{short} + \frac{\ell_{short}}{\ell_C} \ell_C \\ &= \ell_S. \end{aligned}$$

Since C is a short cycle, $\ell_C \leq n \cdot \varepsilon \frac{\Delta}{n} = \varepsilon \Delta$ holds, yielding

$$\begin{aligned} \ell_{S'} &= \ell_S - \ell_{short} + \left\lfloor \frac{\ell_{short}}{\ell_C} \right\rfloor \ell_C \\ &\geq \ell_S - \ell_{short} + \frac{\ell_{short}}{\ell_C} \ell_C - \ell_C \\ &= \ell_S - \ell_C \\ &\geq \ell_S - \varepsilon \Delta. \end{aligned}$$

Now let C_1, \dots, C_k be the removed short cycles. Due to C having maximum length-to-cost-ratio and therefore minimum cost-to-length-ratio, it follows that the cost of S' is bounded by

$$\begin{aligned} c_{S'} &= c_S - \sum_{i=1}^k \ell_{C_i} \frac{c_{C_i}}{\ell_{C_i}} + \left\lfloor \frac{\ell_{short}}{\ell_C} \right\rfloor c_C \\ &\leq c_S - \sum_{i=1}^k \ell_{C_i} \frac{c_C}{\ell_C} + \ell_{short} \frac{c_C}{\ell_C} \\ &= c_S. \end{aligned}$$

□

Recall that we are looking for an s - t -sequence of lower or equal cost than the cost of an optimal solution, but with relaxed length constraints. Based on our observations, each optimal solution to the problem yields a corresponding nice sequence of lower or equal cost in the length interval between $\Delta - \ell^* - \varepsilon \Delta$ and Δ . Thus, it is sufficient to describe an algorithm that computes an optimal solution in the set of nice sequences within this length interval. In the following, we describe an algorithm that does so, but with a slightly larger set of allowed sequences. This can only improve the result, however.

Computing the best-ratio-cycles for each node. For each node $v \in V$, we compute the best-ratio short cycle containing v and refer to it as C_v . We can compute the cycle with the best ratio by using Megiddo's framework [14].

Defining rounded lengths. In order to compute the rest of the sequence, we will now introduce rounded lengths. Define $\delta := \varepsilon^2 \Delta / n^2$ and rounded arc lengths ℓ' by $\ell'_a := \lfloor \ell_a / \delta \rfloor \delta$ for all $a \in A$.

Computing the optimal sequences with regard to rounded lengths for each node. We use the rounded arc lengths in a dynamic program. Let $S(v, w, x, y)$ be a v - w -sequence of minimum cost with length x (with regard to ℓ') that contains y arcs. We denote the cost of $S(v, w, x, y)$ by $c(v, w, x, y)$. Notice that storing these values requires only $O(n \cdot n \cdot n^2 \varepsilon^{-2} \cdot (n + n^2 \varepsilon^{-1}))$ space, as there are only n nodes, $n^2 \varepsilon^{-2}$ different lengths and at most $n + n^2 \varepsilon^{-1}$ arcs in the sequences we are looking for (see Observation 1). We initialize the dynamic program with $c(v, w, x, y) := 0$ for $v = w, x = 0, y = 0$ and $c(v, w, x, y) := \infty$ for either $x = 0$ or $y = 0$. For $x \in \{0, \delta, \dots, \lfloor n^2 / \varepsilon^2 \rfloor \delta\}$ and $y \in \{0, 1, \dots, \lfloor n + n^2 \varepsilon^{-1} \rfloor\}$ we compute

$$c(v, w, x, y) := \min \{ c(v, u, x - \ell'_a, y - 1) + c_a \mid a = (u, w) \in A, x - \ell'_a \geq 0 \} .$$

We now define $C(v, w, x) := \min \{ c(v, w, x, y) \mid y \in \{0, 1, \dots, \lfloor n + n^2 \varepsilon^{-1} \rfloor\} \}$ as the minimum cost of a v - w -sequence with length x (with regard to ℓ') and at most y arcs. The sequences corresponding to the costs can be computed with appropriate bookkeeping. This dynamic program will either compute a sequence that does not use short cycles or at least a sequence that does not use more arcs than a nice sequence potentially would – which is also fine for our purposes.

Bounding the rounding error. The rounding error for an arc a is bounded by δ . Our sequences have at most $n + n^2 / \varepsilon$ arcs. Therefore the rounding error for such a sequence S is at most $(n + n^2 / \varepsilon) \delta = (\varepsilon + \varepsilon^2 / n) \Delta$. Therefore we can bound the length of S with regard to the rounded lengths by $\ell_S - (\varepsilon + \varepsilon^2 / n) \Delta \leq \ell'_S \leq \ell_S$.

Putting the parts together. For this computation, we test all possible start-nodes v for short cycles and all ways to distribute length between the s - v -, v - t -sequences and the short cycles. There are at most $O((n^2 \varepsilon^{-2})^2) = O(n^4 \varepsilon^{-4})$ ways to distribute the length. If the s - v - and v - t -sequence do not have sufficient length to match the length constraint, the remaining length is achieved by using the best-ratio cycle at v . We define $\mathbb{L} := \{0, \delta, \dots, \lfloor n^2 / \varepsilon^2 \rfloor \delta\}$ for brevity. Since we compute s - v - and v - t -sequences with regard to ℓ' , we incur a rounding error of $2(\varepsilon + \varepsilon^2 / n)$ there. Thus, we have to look for sequences with a length of at least $L := (1 - 3\varepsilon - 2\varepsilon^2 / n) \Delta - \ell^*$ with regard to ℓ' in order to consider all sequences with a length of at least $\Delta - \ell^* - \varepsilon \Delta$ with regard to ℓ . We can compute an s - t -sequence S with cost at most the cost of an optimal solution by

$$\min \left\{ C(s, v, \ell'_s) + c_{C_v} \left[\frac{L - \ell'_s - \ell'_t}{\ell_{C_v}} \right] + C(v, t, \ell'_t) \mid v \in V, \ell'_s, \ell'_t \in \mathbb{L}, \ell'_s + \ell'_t \leq L \right\}$$

and using bookkeeping. The minimum can be computed in time $O(n^5 \varepsilon^{-4})$. Thus, we can find a sequence S with cost at most the cost of an optimal solution and $(1 - 3\varepsilon - 2\varepsilon^2 / n) \Delta - \ell^* \leq \ell_S \leq (1 + 2\varepsilon + 2\varepsilon^2 / n) \Delta$.

Theorem 2. *For a given instance I of the restricted minimum cost s - t -sequence problem and $\varepsilon > 0$, we can compute in time polynomial in the input size and ε^{-1} an arc sequence whose cost is at most the cost of an optimal solution to I and whose length satisfies $(1 - \varepsilon)\Delta - \ell^* \leq \ell_S \leq (1 + \varepsilon)\Delta$.*

3 The Maximum Multicommodity Flow over Time Problem

We will now use this problem to approximate maximum multicommodity flows over time without intermediate storage.

3.1 Preliminaries

An instance of the *maximum multicommodity flow over time problem without intermediate storage* consists of a directed graph $G = (V, A)$ with capacities $u_a \geq 0$ and transit times $\tau_a \in \mathbb{N}_0$ on the arcs $a \in A$. Furthermore, we are given a set of commodities $K = \{1, \dots, k\}$ with a source s_i and a sink t_i for all $i \in K$. Arc capacity u_a means that all commodities together can send at most u_a units of flow into arc a at any point in time. A transit time of τ_a means that flow entering arc a at time θ leaves the arc at time $\theta + \tau_a$. Finally, we are given a time horizon $T \in \mathbb{N}_0$ within which all flow needs to be sent. Let $n := |V|$, $m := |A|$, and $k := |K|$. Note that we will assume for every commodity $i \in K$, that there is a single source $s_i \in V$ and a single sink $t_i \in V$. Furthermore we assume that sources and sinks have no incoming and outgoing arcs, respectively (i. e., $\delta_{s_i}^- = \delta_{t_i}^+ = \emptyset$ for all i). This can be achieved by adding k super-sources and super-sinks.

A *multicommodity flow over time without intermediate storage* $f : A \times K \times [0, T) \rightarrow \mathbb{R}_{\geq 0}$ assigns a flow rate $f(a, i, \theta)$ to each arc $a \in A$, each commodity $i \in K$ and all points in time $\theta \in [0, T)$. A flow rate $f(a, i, \theta)$ specifies how much flow of commodity i is sent into arc a at time θ . These flow rates need to fulfill flow conservation: $\sum_{a \in \delta_v^-} f(a, i, \theta - \tau_a) - \sum_{a \in \delta_v^+} f(a, i, \theta) = 0$ for all $i \in K, v \in V \setminus \{s_i, t_i\}, \theta \in [0, T)$. Such a flow is called feasible, if it obeys the capacity constraints as well: $\sum_{i \in K} f(a, i, \theta) \leq u_a$ for all $a \in A, \theta \in [0, T)$.

The objective is to compute a feasible multicommodity flow over time f without intermediate storage that maximizes the amount of flow $|f|$ sent by all commodities within the time horizon, i. e., $|f| := \sum_{i \in K} \sum_{a \in \delta_{t_i}^-} \int_{\theta=0}^{T-\tau_a} f(a, i, \theta)$. For convenience, we define the flow rate of a flow over time to be zero on all arcs at all time points $\theta \notin [0, T)$.

For our setting of an integer time horizon and integral transit times, we can switch between the continuous time model of flows over time given above, where points in time are from a continuous interval $[0, T)$, and the discrete time model, where a discrete set of time points $\mathbb{T} := \{0, \dots, T - 1\}$ is used. Such a flow over time $x : A \times K \times \mathbb{T} \rightarrow \mathbb{R}_{\geq 0}$ assigns flow rates to the arcs and commodities at the specified discrete time points only. It is assumed that flow is sent at this rate

until the next time step. The viability of the switching is due to the following lemma (see, e. g., [5]).

Lemma 2. *Let f be a feasible multicommodity flow over time without intermediate storage in the continuous time model. Then there exists a feasible multicommodity flow over time x without intermediate storage in the discrete time model of the same value, and vice versa.*

Using the discrete time model, we can employ *time-expansion*, a standard technique from the literature first employed by Ford and Fulkerson [7, 8]. This technique is based on copying the network for each time step with arcs linking the copies based on their transit times. Furthermore, super-sources s'_i and -sinks t'_i are introduced for each commodity $i \in K$ and connected to the copies of the commodity's source and sink, respectively. A more thorough introduction can, for example, be found in [16]. For the sake of completeness, we give a formal definition of time-expanded networks here.

Definition 3. *The time expanded network G^T with its node set V^T and its arc set A^T is constructed by “copying the network for each time step”:*

$$\begin{aligned} V^T &:= \{v_\theta \mid v \in V(G), \theta \in \{0, 1, \dots, T-1\}\} \cup \{s'_i, t'_i \mid i \in K\}, \\ E^T &:= \{a_\theta = (v_\theta, w_{\theta+\tau_a}) \mid a = (v, w) \in A, \theta \in \{0, \dots, T-\tau_a-1\}\}, \\ H^T &:= \{(s'_i, (s_i)_\theta), ((t_i)_\theta, t'_i) \mid i \in K, \theta \in \{0, \dots, T-1\}\}, \\ A^T &:= E^T \cup H^T. \end{aligned}$$

Capacities are extended from A to A^T by $u_{a'}^T := u_a$ for $a' = a_\theta \in E^T$ and $u_{a'}^T := \infty$ for $a' \in H^T$.

3.2 LP-Formulations

Using the discrete time model, we can formulate the problem as an LP in arc variables. A variable $x_{a,i,\theta}$ describes the flow rate of commodity i into arc a at time step θ which is equivalent to the flow on the copy of a starting in time layer θ in the time-expanded network. The only special cases for this time-expanded interpretation are the arcs of the form $(s'_i, (s_i)_\theta)$ and $((t_i)_\theta, t'_i)$, as these arcs are outside the copied original network. However, we can just consider the arcs $(s'_i, (s_i)_\theta)$ being copies of each other belonging to time layer θ and the same for arcs $((t_i)_\theta, t'_i)$.

$$\begin{aligned} \max \quad & \sum_{i \in K} \sum_{a \in \delta_{t'_i}^-} \sum_{\theta \in \mathbb{T}} x_{a,i,\theta}, \\ \text{s.t.} \quad & \sum_{i \in K} x_{a,i,\theta} \leq u_a \quad \text{for all } a \in A, \theta \in \mathbb{T}, \\ & \sum_{a \in \delta_v^-} x_{a,i,\theta-\tau_a} - \sum_{a \in \delta_v^+} x_{a,i,\theta} = 0 \quad \text{for all } i \in K, v \in V \setminus \{s'_i, t'_i\}, \theta \in \mathbb{T}, \\ & x_{a,i,\theta} \geq 0 \quad \text{for all } a \in A, i \in K, \theta \in \mathbb{T}. \end{aligned}$$

It is not difficult to see that, due to flow conservation, all flow contributing to the objective function is sent from a source to a sink along some sequence of arcs. We define \mathcal{S} to be the set of all s'_i - t'_i -sequences for all $i \in K$. Then there is always an optimal solution to the maximum multicommodity flow over time problem that can be decomposed into sequences $\mathcal{S}^* \subseteq \mathcal{S}$. We want to split the sequences $S \in \mathcal{S}$ into smaller parts of nearly the same length $\Delta > 0$. Let τ^* be the maximum transit time of an arc. Then we can split S into subsequences of lengths between $\Delta - \tau^*$ and Δ . The length of the last subsequence can be anywhere between 0 and Δ , though, because we have to fit the remaining arcs there. This leads to the following definition.

$$\begin{aligned} \mathcal{S}_{\Delta-\tau^*}^\Delta := & \{v-t'_i\text{-sequences } S \mid i \in K, v \in V, \tau_S \leq \Delta\} \\ & \cup \{v-w\text{-sequences } S \mid i \in K, v, w \in V, \Delta - \tau^* \leq \tau_S \leq \Delta\}. \end{aligned}$$

Since optimal solutions can be decomposed into sequences in $\mathcal{S}_{\Delta-\tau^*}^\Delta$, we can also formulate an LP based on variables for all sequences in $\mathcal{S}_{\Delta-\tau^*}^\Delta$. Variable $x_{S,i,\theta}$ describes the flow rate of commodity i into sequence S at time step θ . Therefore, the following LP, which we will refer to as the *split-sequence LP*, can be used to compute an optimal solution to our problem:

$$\begin{aligned} \max \quad & \sum_{i \in K} \sum_{S \in \delta_{t'_i}^-} \sum_{\theta \in \mathbb{T}} x_{S,i,\theta-\tau_S}, \\ \text{s.t.} \quad & \sum_{i \in K} \sum_{S \in \mathcal{S}_{\Delta-\tau^*}^\Delta} \sum_{j=1, \dots, |S|: a=a_j} x_{S,i,\theta-\tau_{S[\rightarrow a_j]}} \leq u_a \quad \text{for all } a \in A, \theta \in \mathbb{T}, \\ & \sum_{S \in \delta_v^-} x_{S,i,\theta-\tau_S} - \sum_{S \in \delta_v^+} x_{S,i,\theta} = 0 \quad \text{for all } i \in K, v \in V \setminus \{s'_i, t'_i\}, \theta \in \mathbb{T}, \\ & x_{S,i,\theta} \geq 0 \quad \text{for all } S \in \mathcal{S}_{\Delta-\tau^*}^\Delta, i \in K, \theta \in \mathbb{T}. \end{aligned}$$

This formulation of the problem has two advantages over the initial arc-based formulation. Firstly, all sequences in $\mathcal{S}_{\Delta-\tau^*}^\Delta$ – with the exception of those ending in a sink – have a guaranteed length of $\Delta - \tau^*$. Since the time horizon T is an upper bound for the length of any sequence in \mathcal{S} , we can conclude that a sequence $S \in \mathcal{S}$ can be split into at most $\lceil T/(\Delta - \tau^*) \rceil + 1$ segments. Secondly, when rounding up the transit times of the sequence-segments, we know that the error introduced by the rounding is bounded by τ^* for all segments save for the last one, which ends in a sink. These two advantages together allow us to obtain a strong bound on the rounding error.

3.3 Solving the Split-Sequence LP

Unfortunately, the split-sequence LP cannot be solved directly in polynomial time due to its enormous size. We will approach this problem in two steps.

1. We round transit times to generate a number of discrete time steps that is polynomially bounded in the input size and ε^{-1} . This introduces a rounding error, which we bound in Section 3.4.

2. The resulting LP has still exponentially many variables; therefore we will consider its dual, which has exponentially many constraints, but only polynomially many variables (in the input size and ε^{-1}). In Section 3.5 we argue that the dual separation problem can be approximated which yields an FPTAS for the original problem.

We begin by introducing a set $\mathcal{S}_L := \{S \text{ is a } v\text{-}w\text{-sequence with } \tau_S \geq L\}$ that we will necessary later due to the fact that we can only approximate the dual separation problem. We define $\overline{\mathcal{S}}_L^\Delta := \mathcal{S}_{\Delta-\tau^*}^\Delta \cup \mathcal{S}_L$ and replace $\mathcal{S}_{\Delta-\tau^*}^\Delta$ with $\overline{\mathcal{S}}_L^\Delta$ in the LP above. It is easy to see that this does not interfere with our ability to use this LP to solve our problem. We now round the transit times of the sequences in $\overline{\mathcal{S}}_L^\Delta$ up to the next multiple of Δ , i. e., $\tau'_S := \lceil \tau_S / \Delta \rceil \Delta$ for all $S \in \overline{\mathcal{S}}_L^\Delta$. Moreover, we define $T' := \lceil (1 + \varepsilon)T / \Delta \rceil \Delta$ and $\mathbb{T}' := \{0, \Delta, \dots, T' - \Delta\}$. Using these transit times yields the following LP, which we will refer to as the *sequence-rounded LP*:

$$\begin{aligned}
& \max \sum_{i \in K} \sum_{S \in \delta_{t'_i}^-} \sum_{\theta \in \mathbb{T}'} x_{S,i,\theta-\tau'_S}, \\
& \text{s.t.} \quad \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \#(a, S) \cdot x_{S,i,\theta} \leq \Delta u_a \quad \text{for all } a \in A, \theta \in \mathbb{T}', \\
& \quad \sum_{S \in \delta_v^-} x_{S,i,\theta-\tau'_S} - \sum_{S \in \delta_v^+} x_{S,i,\theta} = 0 \quad \text{for all } i \in K, v \in V \setminus \{s'_i, t'_i\}, \theta \in \mathbb{T}', \\
& \quad x_{S,i,\theta} \geq 0 \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in \mathbb{T}'.
\end{aligned}$$

3.4 Bounding the Error of the Sequence-Rounded LP

In this section, we will analyze the error introduced by using sequence-rounded transit times. The lemmas and proofs used in this subsection built on and extend techniques introduced by Fleischer and Skutella [5]. We begin by bounding the error introduced by switching from the given transit times to the sequence-rounded transit times.

Lemma 3. *Let $S \in \mathcal{S}$ be a source-sink sequence and let τ be the normal transit times and τ' the sequence-rounded transit times. Then $\tau_S \leq \tau'_S \leq \tau_S + \varepsilon^2 T$, for $\Delta := \frac{\varepsilon^2 T}{4n}$ and $L := (1 - 1/4\varepsilon^2 / (1 + 1/4\varepsilon^2))\Delta$.*

Proof. The rounding error for an sequence $S' \in \overline{\mathcal{S}}_L^\Delta$ is bounded by $\max(\tau^*, \Delta - L)$ for all sequences not ending in the sink of a commodity, for those ending in the sink it is bounded by Δ . W.l.o.g, we will now assume that $\Delta - L \geq \tau^*$, as we can decrease τ^* to $\Delta - L$ by splitting arcs – this requires at most $m \frac{T}{\Delta - L} \in O(\frac{nm}{\varepsilon^4})$ additional arcs for the L that we choose below. The rounding error for a source-sink-sequence decomposed into sequences of $\overline{\mathcal{S}}_L^\Delta$ is then bounded by

$$\left\lceil \frac{T}{L} \right\rceil (\Delta - L) + \Delta \leq \frac{T}{L} \Delta - T + 2\Delta - L.$$

We will now choose $\Delta := \frac{\varepsilon^2 T}{4n}$ and $L := (1 - \ell)\Delta$. Then

$$\frac{T}{L}\Delta - T + 2\Delta - L = \frac{\ell}{1 - \ell}T + (1 + \ell)\frac{\varepsilon^2 T}{4n}.$$

Now we choose $\ell := \frac{\frac{1}{4}\varepsilon^2}{1 + \frac{1}{4}\varepsilon^2}$ and get

$$\frac{\ell}{1 - \ell}T + (1 + \ell)\frac{\varepsilon^2 T}{4n} = \frac{1}{4}\varepsilon^2 T + \frac{1}{4n}\varepsilon^2 T + \frac{\frac{1}{4}\varepsilon^2}{1 + \frac{1}{4}\varepsilon^2}\frac{\varepsilon^2 T}{4n} \leq \varepsilon^2 T.$$

Thus, the rounding error is bounded by $\varepsilon^2 T$, i. e. a sequence is at most $\varepsilon^2 T$ longer when using the sequence-rounded transit times instead of the given ones. \square

This result enables us construct flows feasible with regard to the sequence-rounded transit times out of normal flows.

Lemma 4. *Let f be a multicommodity flow over time, sending $|f|$ flow units within a time horizon of T . Then, for $\varepsilon > 0$, there exists a multicommodity flow over time that is feasible with regard to the sequence-rounded transit times, sending $|f|/(1 + \varepsilon)$ flow units within a time horizon of $(1 + \varepsilon)T$.*

Proof. W.l.o.g., we can assume that f can be decomposed into s'_i - t'_i -sequences. Otherwise, there exists a flow of equal value with this property. All these s'_i - t'_i -sequences can be further decomposed into sequences of \overline{S}_L^Δ . We will now examine the effects of sending f with sequence-rounded transit times. This means that all flow particles of f still travel along the same sequences, as specified by the decomposition, but the time to traverse a sequence is now different than before. This does not affect flow conservation, since we just change the speed at which flow travels. However, we might violate capacities. Therefore, we define a smoothed flow based on the sequence-decomposition in order to lessen the effects of this:

$$f_{sm}(S, i, \theta) := \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta} f(S, i, \xi) d\xi \quad \text{for all } S \in \overline{S}_L^\Delta, i \in K, \theta \in [0, (1 + \varepsilon)T).$$

Due to the smoothing, this flow requires a time horizon that is εT longer than the original one, resulting in a time horizon of $(1 + \varepsilon)T$ for f_{sm} . The flow value of each arc a at a point in time θ is then bounded by:

$$\begin{aligned} f_{sm}(a, \theta) &:= \sum_{i \in K} \sum_{S \in \overline{S}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} f_{sm}(S, i, \theta - \tau'_{S[\rightarrow a_j]}) \\ &= \sum_{i \in K} \sum_{S \in \overline{S}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \tau'_{S[\rightarrow a_j]}}^{\theta - \tau'_{S[\rightarrow a_j]}} f(S, i, \xi) d\xi \end{aligned}$$

In the next step, we use the bound on the error introduced by the sequence-rounding which we have obtained in Lemma 3. This gives us the following estimation:

$$\begin{aligned}
& \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \tau'_{S[\rightarrow a_j]}}^{\theta - \tau'_{S[\rightarrow a_j]}} f(S, i, \xi) d\xi \\
& \leq \frac{1}{\varepsilon T} \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \int_{\theta - \varepsilon T - \varepsilon^2 T - \tau_{S[\rightarrow a_j]}}^{\theta - \tau_{S[\rightarrow a_j]}} f(S, i, \xi) d\xi \\
& = \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \varepsilon^2 T}^{\theta} \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} f(S, i, \xi - \tau_{S[\rightarrow a_j]}) d\xi \\
& = \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \varepsilon^2 T}^{\theta} f(a, \xi) d\xi \\
& \leq \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \varepsilon^2 T}^{\theta} u_a d\xi = \frac{\varepsilon T + \varepsilon^2 T}{\varepsilon T} u_a = (1 + \varepsilon) u_a.
\end{aligned}$$

Therefore, we can scale the smoothed flow by $\frac{1}{1+\varepsilon}$ and obtain a feasible flow, that sends $\frac{1}{1+\varepsilon}|f|$ flow units within a time horizon $(1 + \varepsilon)T$:

$$f_{sc+sm}(S, i, \theta) := \frac{1}{1 + \varepsilon} f_{sm}(S, i, \theta) \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in [0, (1 + \varepsilon)T].$$

This concludes the proof. \square

Furthermore, we can obtain a solution to the sequence-rounded LP out of any flow feasible with regard to the sequence-rounded transit times.

Lemma 5. *Let f be a feasible multicommodity flow over time with regard to the sequence-rounded transit times. Then there exists a solution x to the sequence rounded LP with equal value.*

Proof. Define x by

$$x_{a,i,\theta} := \frac{1}{\Delta} \int_{\theta_\Delta}^{(\theta+1)\Delta} f(a, i, \xi) d\xi$$

Since f is feasible with sequence-rounded transit times, x is feasible in the sequence-rounded LP. Due to f obeying flow-conservation, so does x . It is easy to see that the total flow value sent does not change, proving the claim. \square

Finally, a solution to the sequence-rounded LP can be turned into a normal flow.

Lemma 6. *Let x be a solution to the sequence-rounded LP. Then, for any $\varepsilon > 0$, one can compute a multicommodity flow over time without intermediate storage that sends $|x|/(1 + \varepsilon)$ units of flow within time horizon $(1 + \varepsilon)T$.*

Proof. Consider the flow over time f corresponding to x and a decomposition $(f_S)_{S \in \overline{\mathcal{S}}_L^\Delta}$ of f . We will now analyze what happens if we send the flow of f using normal transit times. This means that all flow particles of f still travel along the same sequences, as specified by the decomposition, but the time to traverse a sequence is now different than before. This does not affect flow conservation, since we just change the speed at which flow particles travel. However, due to the different transit times, we might violate capacities. Therefore, we define a smoothed flow by

$$f_{sm}(S, i, \theta) := \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta} f(S, i, \xi) d\xi \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in [0, (1 + \varepsilon)T].$$

in order to limit the capacity violations. The new flow requires εT more time, resulting in a time horizon of $(1 + \varepsilon)T$. Now we need to examine how much capacities are violated by f_{sm} if send with the normal transit times:

$$\begin{aligned} f_{sm}(a, \theta) &= \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} f_{sm}(S, i, \theta - \tau_{S[\rightarrow a_j]}) \\ &= \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \tau_{S[\rightarrow a_j]}}^{\theta - \tau_{S[\rightarrow a_j]}} f(S, i, \xi) d\xi \end{aligned}$$

At this point, we use the fact that the error introduced by the sequence rounding is at most $\varepsilon^2 T$ (see Lemma 3). We use this for the following estimation:

$$\begin{aligned} &\sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T - \tau_{S[\rightarrow a_j]}}^{\theta - \tau_{S[\rightarrow a_j]}} f(S, i, \xi) d\xi \\ &\leq \frac{1}{\varepsilon T} \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} \int_{\theta - \varepsilon T - \tau'_{S[\rightarrow a_j]}}^{\theta - \tau'_{S[\rightarrow a_j]} + \varepsilon^2 T} f(S, i, \xi) d\xi \\ &= \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta + \varepsilon^2 T} \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{j=1, \dots, |S|: a=a_j} f(S, i, \xi - \tau'_{S[\rightarrow a_j]}) d\xi \\ &= \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta + \varepsilon^2 T} f(a, \xi) d\xi \\ &\leq \frac{1}{\varepsilon T} \int_{\theta - \varepsilon T}^{\theta + \varepsilon^2 T} u_a d\xi = \frac{\varepsilon T + \varepsilon^2 T}{\varepsilon T} u_a = (1 + \varepsilon)u_a. \end{aligned}$$

Therefore, we can scale f_{sm} to get an flow, that sends $\frac{1}{1 + \varepsilon}|x|$ flow units within a time horizon $(1 + \varepsilon)T$ and obeys the capacities:

$$f_{sc+sm}(S, i, \theta) := \frac{1}{1 + \varepsilon} f_{sm}(S, i, \theta) \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in [0, (1 + \varepsilon)T].$$

Since f is feasible, this flow also fulfills flow conservation making it feasible as well. Consider an arbitrary $i \in K, v \in V \setminus \{s_i, t_i\}, \theta \in [0, T)$:

$$\begin{aligned}
& \sum_{S \in \delta_v^-} f_{sc+sm}(S, i, \theta) - \sum_{S \in \delta_v^+} f_{sc+sm}(S, i, \theta) \\
&= \sum_{S \in \delta_v^-} \frac{1}{1+\varepsilon} f_{sm}(S, i, \theta) - \sum_{S \in \delta_v^+} \frac{1}{1+\varepsilon} f_{sm}(S, i, \theta) \\
&= \sum_{S \in \delta_v^-} \frac{1}{1+\varepsilon} \frac{1}{\varepsilon T} \int_{\theta-\varepsilon T}^{\theta} f(S, i, \xi) d\xi - \sum_{S \in \delta_v^+} \frac{1}{1+\varepsilon} \frac{1}{\varepsilon T} \int_{\theta-\varepsilon T}^{\theta} f(S, i, \xi) d\xi \\
&= \frac{1}{1+\varepsilon} \frac{1}{\varepsilon T} \int_{\theta-\varepsilon T}^{\theta} \sum_{S \in \delta_v^-} f(S, i, \xi) - \sum_{S \in \delta_v^+} f(S, i, \xi) d\xi \\
&= \frac{1}{1+\varepsilon} \frac{1}{\varepsilon T} \int_{\theta-\varepsilon T}^{\theta} 0 d\xi = 0
\end{aligned}$$

This concludes the proof. \square

For the sake of completeness, we also discuss the computation of f_{sc+sm} and related functions used in previous lemmas.

Lemma 7. f_{sc+sm} can be computed efficiently.

Proof. Let x be a solution to the sequence-rounded LP that can be decomposed into sequences. There is no waiting in intermediate nodes – thus, each sequence can let flow wait at the source only. Therefore, each sequence S of the decomposition of x induces a sequence-flow over time f_S that sends x_S flow in a time interval of length Δ and zero flow otherwise; i. e. each f_S is a step-function with one non-zero-step. There are at most time-layer many sequences in the time expanded network that correspond to the same sequence in the static network (recall that waiting at intermediate nodes is forbidden). Thus, the result is a piecewise constant function describing the flow for each sequence, that has at most time-layer many steps ($T' \setminus \Delta \in O(n\varepsilon^{-2})$). Thus, each of $f_{sc+sm}(S, i)$ is efficiently computable and the number of such functions is polynomial in the input as well (since x is a flow in a network of polynomial size). \square

3.5 Solving the Sequence-Rounded LP's Dual

It remains to show that we can solve the sequence-rounded LP efficiently.

Observation 3. The dual of the sequence-rounded LP is

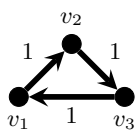
$$\begin{aligned}
& \min \sum_{a \in A} \sum_{\theta \in \mathbb{T}'} \Delta u_a \beta_{a, \theta}, \\
& \text{s.t. } -\alpha_{v, i, \theta} + \alpha_{w, i, \theta + \tau'_S} + \sum_{a \in A} \#(a, S) \beta_{a, \theta} \geq 1 \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in \mathbb{T}', \\
& \beta_{a, \theta} \geq 0 \quad \text{for all } a \in A, \theta \in \mathbb{T}'.
\end{aligned}$$

with $\alpha_{s'_i, i, \theta} = \alpha_{t'_i, i, \theta} = 0$ for all commodities $i \in K$ and $\theta \in \mathbb{T}'$, since these variables do not correspond to constraints in the primal LP and have just been added for notational brevity. The separation problem of this dual is to find a sequence $S \in \overline{\mathcal{S}}_L^\Delta$ with $\sum_{a \in A} \#(a, S) \cdot \beta_{a, \theta} < 1 + \alpha_{v, i, \theta} - \alpha_{w, i, \theta + \tau'_S}$ for some v, w, i, θ .

Notice that the number of different combinations of v, w, i, θ is polynomially bounded such that we can enumerate them efficiently. Thus, the separation problem is to find a v - w -sequence S of minimum cost with regard to β and $S \in \overline{\mathcal{S}}_L^\Delta$. This is the restricted minimum cost s - t -sequence problem introduced in Section 2 with β as the costs and τ as the lengths and Δ as the threshold. As we have seen there, we can only approximately solve this separation problem in the sense that we get a solution with a length between $(1 - \varepsilon)\Delta - \tau^*$ and $(1 + \varepsilon)\Delta$. Recall that we have defined $\overline{\mathcal{S}}_L^\Delta$ as $\mathcal{S}_{\Delta - \tau^*}^\Delta \cup \mathcal{S}_L$ for this purpose. Until this point, choosing $\mathcal{S}_L = \emptyset$ would have been completely fine, we have not needed it so far. However, using the equivalence of separation and optimization [9], we can find a solution to a dual problem where \mathcal{S}_L contains the additional sequences of length at least $(1 - \varepsilon)\Delta - \tau^*$ found by the ellipsoid method. This leads to $L := (1 - \varepsilon)\Delta - \tau^*$ which is also in line with the value of L we have used in the analysis before. We conclude this section by stating our main result.

Theorem 4. *Let I be an instance of the maximum multicommodity flow over time problem without intermediate storage and let OPT be the value of an optimal solution to it. There exists an FPTAS that, for any given $\varepsilon > 0$, computes a solution with value at least $OPT/(1 + \varepsilon)$ and time horizon at most $(1 + \varepsilon)T$.*

We mention that for the multicommodity flow over time problems with storage at intermediate nodes considered in [5], Fleischer and Skutella obtained a stronger approximation result where only the time horizon is increased by a factor $1 + \varepsilon$ but the original, optimal flow values are sent. In our setting, however, we also need to approximate the flow values and can only send a $1/(1 + \varepsilon)$ fraction. This is due to the fact that for the case where intermediate storage is allowed, increasing the time horizon by a factor δ allows to increase flow values by a factor of δ (Lemma 4.8 in [5]). Such a result does not hold for our setting, though (see Example 1 below). In this context, it is also interesting to mention that hardly any results on approximating the flow value for a fixed time horizon are known and almost all approximation algorithms that guarantee an optimal flow value at the price of slightly increasing the time horizon rely on [5, Lemma 4.8].



A gadget where increasing the time horizon does not increase the amount of flow that can be sent. The arc labels specify transit times, the capacities are one on all arcs. There are three commodities, the first having v_1 as a source and v_3 as a sink, the second with v_2 and v_1 , the third with v_3 and v_2 .

Fig. 1. Counterexample to Lemma 4.8 in [5] in our setting.

Example 1. Consider the network in Figure 1. Within a time horizon of $T = 3$, three flow units can be sent by routing exactly one flow unit of each commodity. There is only a single source-sink-path for each commodity, and all paths have length 2. Therefore it is easy to see that this is indeed a maximal flow for this instance. Now consider time horizon $T = 4$. We can still not send more than three flow units, as we can only send flow into the three paths in the interval $[0, 2)$, but flow sent into one path during $[0, 1)$ blocks flow sent into the next path during $[1, 2)$.

4 Extensions

Our approach above is based on the fact that we can efficiently approximate the separation problem of the dual LP. Therefore, we can extend this approach to related problems whose dual LP is also efficiently approximable.

The *multicommodity flow over time problem without intermediate storage* is similar to the maximum multicommodity flow over time problem without intermediate storage, but instead of a source and a sink for each commodity, we are given supplies and demands $b_{i,v}, i \in K, v \in V$ that specify how much flow of each commodity can be sent from or to a node. The task is to find a flow that satisfies all supplies and demands within a time horizon.

The *minimum cost multicommodity flow over time problem without intermediate storage* receives the same input as an instance of the multicommodity flow over time problem without intermediate storage together with arc costs $c_a \in \mathbb{R}$. The task is to find a feasible multicommodity flow over time of minimum cost fulfilling all supplies and demands within the given time horizon. The formal definitions are as follows:

Definition 4 (Multicommodity Flow over Time without Intermediate Storage). *An instance of the multicommodity flow over time problem without intermediate storage consists of a network $G = (V, A)$ with a set of nodes V and a set of arcs A , as well as capacities $u_a \geq 0$ and transit times $\tau_a \in \mathbb{N}_0$ on all arcs $a \in A$. Furthermore, we are given a set of commodities $K = \{1, \dots, k\}$ with balances $b_{i,v} \in \mathbb{R}$ for all $i \in K, v \in V$ with $\sum_{v \in V} b_{i,v} = 0$ for all $i \in K$. Finally we are given a time horizon $T \in \mathbb{N}_0$. The objective is to find a feasible multicommodity flow over time that fulfills all supplies and demands (see below).*

In this setting, a *multicommodity flow over time without intermediate storage* has to fulfill a slightly different type of flow conservation constraint than we have seen before.

Definition 5. *A multicommodity flow over time without intermediate storage $x : A \times K \times \mathbb{T} \rightarrow \mathbb{R}_{\geq 0}$ assigns a flow value $x_{a,i,\theta}$ to each arc $a \in A$, each commodity $i \in K$ and all points in time $\theta \in \mathbb{T}$ such that supplies and demands are fulfilled:*

$$\sum_{a \in \delta_v^+} \sum_{\theta=0}^T x_{a,i,\theta} - \sum_{a \in \delta_v^-} \sum_{\theta=0}^{T-\tau_a} x_{a,i,\theta} = b_{i,v} \text{ for all } i \in K, v \in V.$$

Furthermore, flow conservation has the following requirements:

$$0 \leq \sum_{a \in \delta_v^+} \sum_{\xi=0}^{\theta} x_{a,i,\theta} - \sum_{a \in \delta_v^-} \sum_{\xi=0}^{\theta-\tau_a} x_{a,i,\xi} \leq b_{i,v} \text{ for all } i \in K, v \in V : b_{i,v} \geq 0, \theta \in \mathbb{T}$$

and

$$0 \geq \sum_{a \in \delta_v^+} \sum_{\xi=0}^{\theta} x_{a,i,\theta} - \sum_{a \in \delta_v^-} \sum_{\xi=0}^{\theta-\tau_a} x_{a,i,\xi} \geq b_{i,v} \text{ for all } i \in K, v \in V : b_{i,v} < 0, \theta \in \mathbb{T}.$$

Such a flow is called feasible, if it obeys the capacity constraints as well:

$$\sum_{i \in K} x_{a,i,\theta} \leq u_a \text{ for all } a \in A, \theta \in \mathbb{T}.$$

A *minimum cost multicommodity flow over time* adds an objective function to the multicommodity flow over time problem. Formally, it can be defined as follows.

Definition 6 (Minimum Cost Multicommodity Flow over Time). *An instance of the minimum cost multicommodity flow over time problem receives the same input as an instance of the multicommodity flow over time problem without intermediate storage with the addition of costs $c_a \in \mathbb{R}$ on the arcs $a \in A$. The objective is to find a feasible multicommodity flow over time of minimum cost fulfilling all supplies and demands within the time horizon, where the cost of a flow x is defined as $\sum_{a \in A} \sum_{i \in K} \sum_{\theta \in \mathbb{T}} c_a \cdot x_{a,i,\theta}$.*

We focus now on the minimum cost multicommodity flow over time problem without intermediate storage, as this problem is a generalization of the multicommodity flow over time problem without intermediate storage. The case with intermediate storage can be done similarly. We begin by considering an arc based LP formulation that is again based on a time-expanded network but has a slightly different structure than before.

We still copy the network for each time step and introduce a super-source s'_i and -sink t'_i for each commodity $i \in K$, but we connect the super-nodes differently to the network. For this purpose, we define additional nodes $s_{i,v}$ for each $v \in V$ with supply $b_{i,v} > 0$ and $t_{i,v}$ for each node $v \in V$ with demand $b_{i,v} < 0$. Then we define new supplies and demands b' by $b'_{i,s'_i} := \sum_{v \in V: b_{i,v} > 0} b_{i,v}$, $b'_{i,t'_i} := \sum_{v \in V: b_{i,v} < 0} b_{i,v}$ and $b'_{i,v} = 0$ for the rest of the nodes. Then we connect the super-nodes by arcs $a_{s_{i,v}} = (s'_i, s_{i,v})$ and $a_{t_{i,v}} = (t_{i,v}, t'_i)$ with the supply/demand-nodes. The capacity of the new arcs is determined by the supply and demand of the nodes they are connected to, i. e. $u_{a_{s_{i,v}}} = |b_{i,v}|$, $u_{a_{t_{i,v}}} = |b_{i,v}|$. These arcs have zero transit time and zero cost. The supply nodes $s_{i,v}$ and demand nodes $t_{i,v}$ are then connected with all copies of their node v in the time-expanded network by arcs with infinite capacity, zero transit time and zero cost. This gives us a network where only two nodes per commodity have non-zero

supply and demand, which makes the structure of the LP much nicer. In order to make the LP more similar to the last problem, we transform it into a maximization problem. For brevity, we define $\mathbb{V} := \{s'_i, t'_i \mid i \in K\}$. We get the following LP based on arc variables. Note that the arcs incident to nodes of \mathbb{V} exist only once, i. e. they are outside the copied structure.

$$\begin{aligned}
& \max - \sum_{i \in K} \sum_{a \in A} \sum_{\theta \in \mathbb{T}} c_a \cdot x_{a,i,\theta}, \\
& \text{s.t.} \quad \sum_{i \in K} x_{a,i,\theta} \leq u_a \quad \text{for all } a \in A, \theta \in \mathbb{T}, \\
& \quad \sum_{a \in \delta_v^+} x_{a,i,\theta} - \sum_{a \in \delta_v^-} x_{a,i,\theta-\tau_a} = 0 \quad \text{for all } i \in K, v \in V \setminus \mathbb{V}, \theta \in \mathbb{T}, \\
& \quad \sum_{a \in \delta_v^+} x_{a,i} - \sum_{a \in \delta_v^-} x_{a,i} = b'_{i,v} \quad \text{for all } i \in K, v \in \mathbb{V}, \theta \in \mathbb{T}, \\
& \quad x_{a,i,\theta} \geq 0 \quad \text{for all } a \in A, i \in K, \theta \in \mathbb{T}.
\end{aligned}$$

Note that the changes compared to the maximum multicommodity flow over time problem without intermediate storage are mostly in the objective function and the right side of the flow conservation constraints. Again we can consider a sequence-rounded LP.

$$\begin{aligned}
& \max - \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \sum_{\theta \in \mathbb{T}'} c_S \cdot x_{S,i,\theta}, \\
& \text{s.t.} \quad \sum_{i \in K} \sum_{S \in \overline{\mathcal{S}}_L^\Delta} \#(a, S) \cdot x_{S,i,\theta} \leq \Delta u_a \quad \text{for all } a \in A, \theta \in \mathbb{T}', \\
& \quad \sum_{S \in \delta_v^+} x_{S,i,\theta} - \sum_{a \in \delta_v^-} x_{S,i,\theta-\tau'_S} = 0 \quad \text{for all } i \in K, v \in V \setminus \mathbb{V}, \theta \in \mathbb{T}', \\
& \quad \sum_{S \in \delta_v^+} x_{S,i} - \sum_{S \in \delta_v^-} x_{S,i} = b'_{i,v} \quad \text{for all } i \in K, v \in \mathbb{V}, \theta \in \mathbb{T}', \\
& \quad x_{S,i,\theta}, x_{S,i} \geq 0 \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in \mathbb{T}'.
\end{aligned}$$

The dual of this sequence-rounded LP is:

$$\begin{aligned}
& \min \sum_{a \in A, \theta \in \mathbb{T}'} \Delta u_a \beta_{a,\theta} + \sum_{i \in K, v \in \mathbb{V}, \theta \in \mathbb{T}'} b'_{i,v} \alpha_{v,i,\theta}, \\
& \text{s.t.} \quad \alpha_{v,i,\theta} - \alpha_{w,i,\theta+\tau'_S} + \sum_{a \in A} \#(a, S) \beta_{a,\theta} \geq -c_S \quad \text{for all } S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in \mathbb{T}', \\
& \quad \beta_{a,\theta} \geq 0 \quad \text{for all } a \in A, \theta \in \mathbb{T}'.
\end{aligned}$$

with $\alpha_{s'_i,i,\theta} = \alpha_{t'_i,i,\theta} = 0$ for all commodities $i \in K$ and $\theta \in \mathbb{T}'$, since these variables do not correspond to constraints in the primal LP and have just been added for notational brevity. Note that $c_S = \sum_{a \in A} \#(a, S) c_a$. Therefore, we can write the constraint for all $S \in \overline{\mathcal{S}}_L^\Delta, i \in K, \theta \in \mathbb{T}'$ as

$$\sum_{a \in A} \#(a, S) \cdot (\beta_{a,\theta} + c_a) \geq -\alpha_{v,i,\theta} + \alpha_{w,i,\theta+\tau'_S}.$$

For non-negative costs, this leads once again to a restricted minimum cost s - t -sequence problem as the separation problem.

Theorem 5. *Let I be an instance of the minimum cost multicommodity flow over time problem with non-negative arc costs and let OPT be the value of an optimal solution to it. There exists an FPTAS that, for any given $\varepsilon > 0$, finds a solution of value at most OPT , with time horizon at most $(1 + \varepsilon)T$ that fulfills a $(1 + \varepsilon)^{-1}$ fraction of the given supplies and demands.*

Conclusion. We conclude by stating an open problem that might stimulate further research in this direction: Multicommodity flow over time problems without intermediate storage that are restricted to simple flow paths. It is known that these problems are strongly NP-hard [10] and thus, unless $P=NP$, do not have an FPTAS. On the positive side, there is a $(2 + \varepsilon)$ -approximation algorithm known [4, 5].

Acknowledgments. The authors thank Melanie Schmidt and the anonymous referees for their helpful comments regarding the presentation of this paper.

References

- [1] J. E. Aronson. A survey of dynamic network flows. *Annals of Operations Research*, 20:1–66, 1989.
- [2] M. Braun and S. Winter. Ad hoc solution of the multicommodity-flow-over-time problem. *IEEE Transactions on Intelligent Transportation Systems*, 10(4):658–667, 2009.
- [3] R. E. Burkard, K. Dlaska, and B. Klinz. The quickest flow problem. *ZOR – Methods and Models of Operations Research*, 37:31–58, 1993.
- [4] L. Fleischer and M. Skutella. The quickest multicommodity flow problem. In William J. Cook and Andreas S. Schulz, editors, *Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Computer Science*, pages 36–53. Springer, 2002.
- [5] L. Fleischer and M. Skutella. Quickest flows over time. *SIAM Journal on Computing*, 36:1600–1630, 2007.
- [6] L. K. Fleischer and É. Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23:71–80, 1998.
- [7] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [8] L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1987.
- [9] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- [10] A. Hall, S. Hippler, and M. Skutella. Multicommodity flows over time: Efficient algorithms and complexity. *Theoretical Computer Science*, 379:387–404, 2007.
- [11] B. Hoppe. *Efficient dynamic network flow algorithms*. PhD thesis, Cornell University, 1995.
- [12] B. Hoppe and É. Tardos. The quickest transshipment problem. *Mathematics of Operations Research*, 25:36–62, 2000.
- [13] B. Klinz and G. J. Woeginger. Minimum cost dynamic flows: The series parallel case. *Networks*, 43:153–162, 2004.

- [14] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4:414–424, 1979.
- [15] W. B. Powell, P. Jaillet, and A. Odoni. Stochastic and dynamic networks and routing. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, chapter 3, pages 141–295. North–Holland, Amsterdam, The Netherlands, 1995.
- [16] M. Skutella. An introduction to network flows over time. In *Research Trends in Combinatorial Optimization*, pages 451–482. Springer, 2009.